APPDYNAMICS

AppDynamics App iQ Platform
AppDynamics Platform

Version 4.5.13

Extend AppDynamics

This section describes how you can extend and customize the AppDynamics Application Performance Management (APM) Platform.

The AppDynamics Community Exchange (https://www.appdynamics.com/community/exchange/) includes many pre-built extension you can use to customize AppDynamics and integrate it with other systems.

Alternatively, you can use the AppDynamics REST APIs to create custom integrations and custom automation tasks.

AppDynamics REST APIs

- AppDynamics APIs
- Using the Controller APIs
- Access Swagger and Accounts API

AppDynamics Extensions

- Integration Modules
- Integrate AppDynamics with Splunk
- Integrate AppDynamics with Scalyr

AppDynamics APIs

On this page:

Related pages:

- Extensions and Custom Metrics
- AppDynamics Exchange

The AppDynamics APIs let you extend and customize various aspects of the AppDynamics Application Performance Monitoring (APM) Platform. This page gives you an overview of the AppDynamics APIs.

Overview of the AppDynamics APIs

The AppDynamics APM Platform exposes various APIs for customizing and extending its features. Generally speaking, they can be categorized as platform-side APIs, which are served by the Controller and Events Service, and agent-side APIs.

The AppDynamics platform server components and agents offer the following APIs:

- Controller APIs: Use to administer the Controller, configure, monitor, query metrics, and more. For documentation see the Platform API Index, below.
- Analytics Events API: Use to send custom analytics events from your own data sources to the Events Service. For documentation see the Analytics Events API section in the Platform API Index, below.
- Standalone Machine Agent APIs: HTTP APIs available at the machine agent for uploading custom metrics. See Standalone Machine Agent HTTP Listener.
- Database Agent APIs: Use to get, create, update, and delete Database Monitoring database Collectors. See Database Visibility API.
- Application Agent Instrumentation APIs: Use to control and customize transaction detection and correlation, along with exit point detection. The agent APIs include:
  - PHP Agent API
  - Python Agent API
  - Node.js Agent API Reference
  - C/C++ Application Agent
  - Java Agent API: Customize agent instrumentation. See the SDK folder in the agent home directory
- Mobile RUM: Instrument mobile applications for real user performance monitoring. See Instrument iOS Applications for more information.

The following section lists the general platform APIs, API's served at the Controller and Events Service, that are documented in this section.

Platform API Index

Here are all the methods in the AppDynamics Controller and Events Service APIs:

- **Accounts API**
  - This API is documented using Swagger-based REST browser. For more information, see Using the Controller APIs.
  - Retrieve Controller Audit History
  - Configure Metric Retention by Account
  - Configure Metric Retention by Application
- **Application Model API**
  - Retrieve All Business Applications
  - Retrieve All Business Transactions in a Business Application
  - Retrieve All Tiers in a Business Application

    - Get User by ID
    - Get User by Name
    - Get All Users
    - Update User
    - Delete User
    - Create Group
    - Get Group by ID
    - Get Group by Name
    - Get All Groups
    - Update Group
    - Delete Group
    - Add User to Group
    - Remove User from Group
    - Create Role
    - Add Role to User
    - Remove Role from User
    - Add Role to Group
    - Remove Role from Group
    - Get Role by ID
    - Get Role by Name
    - Get All Roles
    - Update Role
    - Delete Role
- **License Rules API**
    - Creates a New License Rule
    - Returns a Summary of All License Rules for the Current Account
    - Updates a License Rule
    - Deletes a License Rule
    - Retrieve a License Rule via its Id
    - Retrieves a License Rule by Access Key
    - Retrieve a License Rule by Name

API Clients

On this page:

You can create and use the identity type, API Clients, to provide secure access to the Controller through AppDynamics Controller REST API calls. These calls use Open Authorization (OAuth) token-based authentication.

About OAuth Mechanisms

OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile, and desktop applications. See https://oauth.net/ for more information.

OAuth acts as the intermediary on your behalf, providing third-party applications with an access token that authorizes specific account information to be shared. Using the OAuth protocol with AppDynamics Controller REST APIs is the best way to securely grant access to your Controller information.

The OAuth authentication process works by first authenticating a request token. This request token is used to obtain an encrypted access token from your Controller. Once the access token is available, you can use it to make authenticated requests to your Controller until the token expires or is revoked.

The tokens are based on JSON Web Tokens (JWT) authentication format, which is the industry standard RFC 7519 method for representing claims securely between two parties.

Accessing Authentication Provider Settings

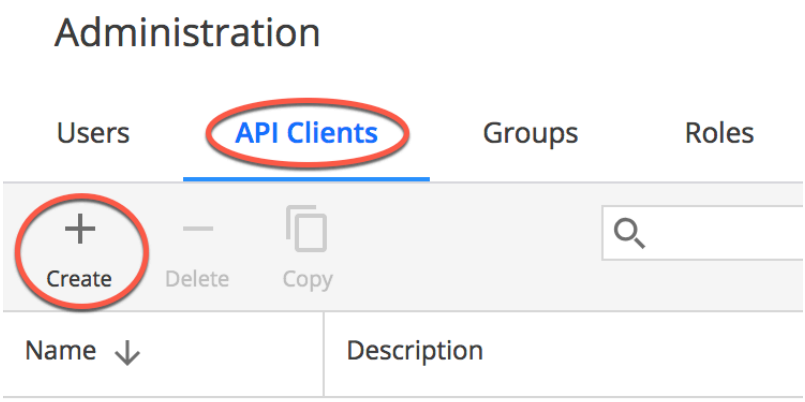
Users with the Account Owner role or the **Administer users, groups, roles ...**permission can view API Clients settings in the **Settings** > **Administration** page.

Creating API Clients

You can create new API Client identity types that can be used to generate OAuth tokens.

To create or edit an API Client

1. While logged in to the Controller UI as an Account Owner (or other role with the **Administer users, groups, roles ...**permissio n), click  (gear icon) > **Administration**.
2. Click the **API Clients** tab to view the list of existing clients.
   From the tab, you can create new clients and modify or delete older ones.
3. Click **+ Create** to create an API Client or select an existing client to edit.



4. Enter the Client Name and Description.
5. Click **Generate Secret** to populate the Client Secret.
   This will generate a UUID as the secret of the API Client.

   > This API Client secret acts as a password. It does not generate the authentication token.

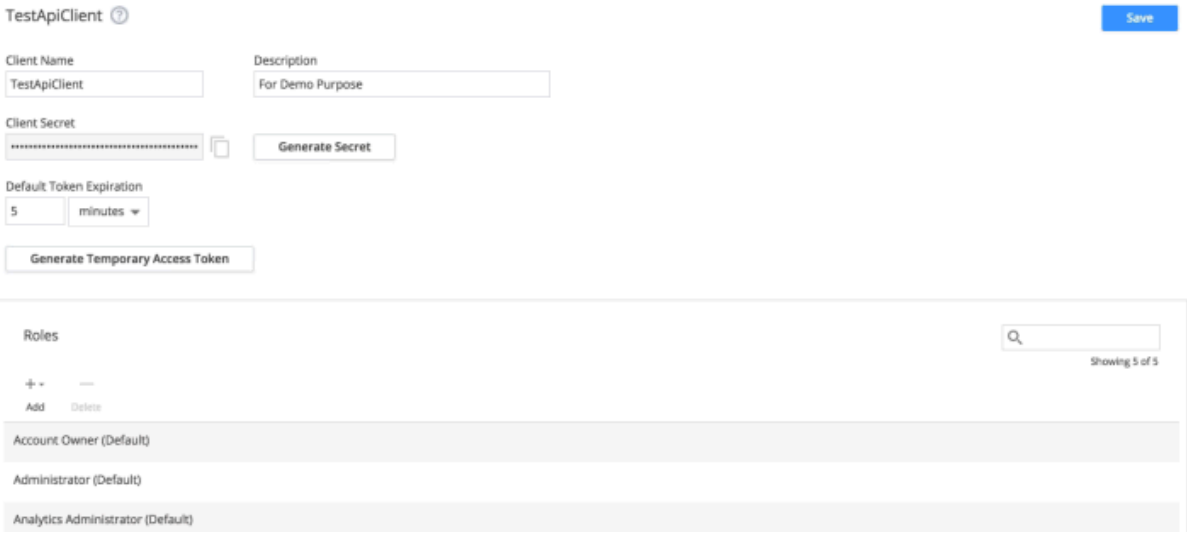
6. Set the Default API-generated Token Expiration. This expiration only applies to authentication tokens generated through the '/controller/api/oauth/access_token' REST API, not to Temporary Access Tokens generated from the UI. See Using the Access Token for more information.

   > Every API-generated access token has an expiration. The default is 5 minutes, but you can set it to any second, minute, or hour limit. The Default API-generated Token has a shorter expiration than the authentication token generated through the Administration UI.

7. Add the Roles you would like to associate with this API Client. You can add or remove roles at any time. See Roles and Permissions for more information.

   > The REST APIs will use the identity which the access token represents to pull up RBAC permissions and check those permissions at the underlying API level.

8. Click **Save** at the top right of the panel to save your API Client.



Using the Access Token

You can now generate and use the access token for each API access call into your Controller by generating the token through one of the following:

- Administration UI: These tokens are usually long-lived and have a longer expiration time. They can be generated by the account administrator and distributed to parties/teams who need to access the Controller but do not want to refresh such tokens too frequently.
- API: These tokens are usually short-lived and have a relatively short expiration time. They are typically generated and refreshed regularly by a program before expiration. These tokens are visible from the UI, and are not individually tracked and managed.

Generate the Token Through the UI

To generate the access token through the Administration UI:

1. Click **Generate Temporary Access Token** when you are ready to use the access token.

   > Remember that temporary token expirations are usually longer than the Default API-generated Token Expiration.

2. Copy and paste the access token from the Temporary Access Token field into the TOKEN portion of your curl command. See the following example:

   ```
   curl -H "Authorization:Bearer <AUTH_TOKEN>"
   "http://master-onprem-controller.e2e.appd-test.com:8090/control
   ler/rest/applications"
   ```

3. Check that your call is successful. It should look like this:

```
                                $ curl -H "Authorization:Bearer eyJraWQiOiIxIiwiYWxnIjoiSFMyNTYifQ.eyJpc3MiOiJBcHBEeW5hbWljcyIsImF1ZCI6IkFwcERfQVBJcyIsImV4c
I6MTUyNDg1MzcyNiwianRpIjoibEZObHRtbFQzbFBTTFk3WEFyaHpPQSIsImlhdCI6MTUyNDc2NzMyNiwibmJmIjoxNTI0NzY3MjA2LCJzdWIiOiJUZXN0QXBpQ2xpZW50IiwidHlwZSI6IkFQSV9DTElFTlQiLC
pZCI6IjRkNDZmYjYzLWNkNjctNGU3NC05Zjg3LWE2ZGFkMjQzNTI2ZSIsImFjY3RJZCI6IjAzMzcyZjlhLThjZmYtNDJlYi05MjA5LTU2OTdmYzM1ZGVmNSIsImFjY3ROYW1lIjoiY3VzdG9tZXIxIn0.pykAwQg
9J9QAQn1soQ_Vi73zGi1GCzBtGVs-gIRkpc" "http://master-onprem-controller.e2e.appd-test.com:8090/controller/rest/applications"
<applications><application>
  <id>8</id>
  <name>ECommerce-Sales-Demo-Fulfillment</name>
</application>
<application>
  <id>11</id>
  <name>AshishTest</name>
</application>
<application>
  <id>20</id>
  <name>IbmBpmApp</name>
</application>
<application>
  <id>5</id>
  <name>ECommerce-Sales-Demo</name>
  <description></description>
</application>
```

Generate the Token Through API

You can use REST API to generate a short-lived access token. This default, on-demand token is not tracked on the UI.

To generate the access token through API:

Use POST /controller/api/oauth/access_token. You will need the API Client name and Client Secret to generate the token:

Example

```
curl -X POST -H "Content-Type:
application/vnd.appd.cntrl+protobuf;v=1" "https://<controller
address>/controller/api/oauth/access_token" -d
'grant_type=client_credentials&client_id=TestApiClient@customer&clie
nt_secret=face10d5-573e-4a75-8396-afa006fd8f19'

{
"access_token":
"eyJraWQiOiIxIiwiYWxnIjoiSFMyNTYifQ.eyJpc3MiOiJBcHBEeW5hbWljcyIsImF1
ZCI6IkFwcERfQVBJcyIsImV4cCI6MTUzNjI3MjI1NiwianRpIjoiT2twZHVDdmduSDdw
MmduMnc4MFRtQSIsImlhdCI6MTUzNjI3MTk1NiwibmJmIjoxNTM2MjcxODM2LCJzdWIi
OiJUZXN0QXBpQ2xpZW50IiwidHlwZSI6IkFQSV9DTElFTlQiLCJpZCI6ImFmZTQxMzg5
LTJlNjMtNDQyYS1hY2U2LTEyYzU5NGFlOGM2OCIsImFjY3ROYW1lIjoiZTJlLWN1c3RvbWVy
In0.95EyDNV5muTN_4zGOXIPQQHOdVDSiEynKSgk08UHlh0",
"expires_in": 300
}
```

It is to be noted that:

- For controllers before 4.5.9, authentication header is required in the request as shown below:

```
curl -X POST -H "Content-Type:
application/vnd.appd.cntrl+protobuf;v=1" -u
'TestApiClient@customer:face10d5-573e-4a75-8396-afa006fd8f19'
"https://<controller address>/controller/api/oauth/access_token" -d
'grant_type=client_credentials&client_id=TestApiClient@customer&cli
ent_secret=face10d5-573e-4a75-8396-afa006fd8f19'

{
"access_token":
"eyJraWQiOiIxIiwiYWxnIjoiSFMyNTYifQ.eyJpc3MiOiJBcHBEeW5hbWljcyIsImF
1ZCI6IkFwcERfQVBJcyIsImV4cCI6MTUzNjI3MjI1NiwianRpIjoiT2twZHVDdmduSD
dwMmduMnc4MFRtQSIsImlhdCI6MTUzNjI3MTk1NiwibmJmIjoxNTM2MjcxODM2LCJzd
WIiOiJUZXN0QXBpQ2xpZW50IiwidHlwZSI6IkFQSV9DTElFTlQiLCJpZCI6ImFmZTQx
Mzg5LTJlNjMtNDQyYS1hY2U2LTEyYzU5NGFlOGM2OCIsImFjY3ROYW1lIjoiZTJlLWN1c3
RvbWVyIn0.95EyDNV5muTN_4zGOXIPQQHOdVDSiEynKSgk08UHlh0",
"expires_in": 300
}
```

- The token generation API requires both an authentication header with username and password as well as the oAuth request body to successfully request a token.

1. Use the generated token to list all applications:

   Example

   ```
   curl -H "Authorization:Bearer <AUTH_TOKEN>"
   https://master-controller.e2e.appd.com/controller/rest/applicat
   ions
   ```

   a. In the case of success: all applications are returned.
   b. In the case of failure: HTTP 401 Unauthorized "Failed to authenticate: invalid access token." is returned.

Managing Access Tokens

Access tokens are based on JWT, so even if you decode them, you will not be able to see any sensitive information.

However, if for any reason you believe that your token has become compromised, then you can revoke it by clicking **Revoke**. Deleting the API Client will also invalidate the token. Calls using revoked access tokens fail to authenticate with a 401 Unauthorized error HTTP status code. You can also click **Regenerate** to refresh a token.

> Regenerated tokens do not disable older tokens. The older tokens will remain active until they expire.

When you regenerate a token, you can set the Temporary Token Expiration.

> The default is 1 day, but you can set it to any hour, day, or year limit.

> There is currently no way to retrieve all previous or currently valid tokens. Therefore, only the current token can be revoked.
>
> Also, API generated tokens, which have Default API-generated Token Expiration, cannot be viewed nor revoked through the UI or REST API.

Using the Controller APIs

On this page:

The Controller APIs, which are served by the Controller instance, rather than by the Events Service or by an agent component, consist of the following:

- Accounts API
- Application Model API
- Metric and Snapshot API
- Alert and Respond API
- Configuration API
- Configuration Import and Export APIs
- Analytics Events API

This page provides general usage information for the APIs.

Controller API Base URI

Except as indicated in the format listing for a particular method, URIs in the Controller API use the following base URI:

```
http://<controller_host>:<controller_port>/controller/rest/<REST_URI>
```

The port that serves the API is the same primary port for the Controller used by Controller UI and agents.

Retrieving Data in JSON Format

The AppDynamics Controller APIs return data in eXtensible Markup Language (XML) or, for certain calls, in JavaScript Object Notation (JSON). The default output format is XML.

Any Controller API with a URI in the /controller/rest/ format shown in Controller API Base URI can return data in JSON format.

To retrieve data in JSON, call the API with the output query parameter set to JSON, as follows:

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications?output=JSON
[
    {
    "description": "",
    "id": 5,
    "name": "ECommerce_E2E"
  },
    {
    "description": "",
    "id": 8,
    "name": "ECommerce_E2E-Fulfillment"
  },
]
```

For any of the Controller APIs, you can similarly specify JSON output format for the response.

When a client uses HTTP 1.1 and accepts gzip content encoding, the Controller returns JSON responses using gzip compression.

Authentication

See API Clients for information on using OAuth identity types for authentication.

To invoke the REST APIs using basic HTTP authentication, you must provide the authentication credentials as well as your account information. These are:

- Account: the AppDynamics tenant account name
- Username: a user in that account
- Password: the password for that account

Pass the credentials in the following form:

```
<your_username>@<your_accountname>:<your_password>
```

For a single-tenant Controller (which is the case for most on-premises Controllers), the account name should be the account name for the primary default account, which is "customer1". For example:

```
<your_username>@customer1:<your_password>
```

If you are using a multi-tenant Controller (most SaaS Controllers), replace customer1 with your own, instance-specific account name. You can find the account name in the License Management page.

Invalid Characters for Usernames and Passwords

Usernames and passwords that contain the following characters are not authenticated for REST API calls:

\ / " [ ] : | < > + = ; , ? * , ' tab space @

If you have already created user credentials that contain any of the disallowed characters, such as "user:customer66", create new credentials without the disallowed character for the purpose of accessing the REST APIs.

For usernames or passwords containing the "@" symbol, URL encode the "@" character as %40.

Copying a Metric URL in the Metric Browser

When you right-click a metric in the Metric Browser you can copy the full REST URL of the metric. You can then paste the REST URL into your code or onto the command line.

For security reasons, AppDynamics only supports making API calls programmatically or at the command line. Do not attempt to paste the REST URL into a browser.

Access Swagger and Accounts API

On this page:

You can view documentation for the Accounts APIs by accessing the Swagger UI. This page provides information on accessing and using Swagger, and a summary of the Accounts APIs.

Controller REST API Browser

Documentation for the Accounts and ACL REST APIs is available via the Controller REST browser, a Swagger-based interactive REST documentation tool. The browser lets you view and try operations in the API on your own Controller. In addition to reviewing the REST API reference below, be sure to refer to the Controller REST browser for additional commands.

To access the REST API Browser, in a Web browser, go to:

```
https://<controller_host>:<primary_port>/api-docs/index.html
```

In the browser, click Show/Hide to see the APIs. If you have an authenticated session for the Controller UI active in the browser, you can view and try the API calls.

The REST Python Client Extension is a tool that you can use with the non-Swagger APIs in the AppDynamics API. Among other functions, it can retrieve the business applications, tiers, and nodes in your AppDynamics model.

> If the user is not already logged into the Controller application, an authentication prompt appears in the browser, where the username and password are to be entered to access the Swagger UI. The username should be of the format <User Name>@<Account Name>.

Endpoint Examples

The API entry point is /api, and the URIs are used as shown below:

```
http[s]://<<hostname>>/controller/api/{URI_from_swagger}
```

Here is an example to get the accountID with name using GET /accounts/myaccount:

```
curl -v --user admin@customer1:root
http://localhost:8090/controller/api/accounts/myaccount
```

Here is an example to get all the applicationIDs with name in a given account using GET /accounts/{acctId}/applications:

```
curl -v --user admin@customer1:root
http://localhost:8090/controller/api/accounts/2/applications
```

Here is an example to get all the Health Rule Names with IDs using GET /accounts/{acctId}/applications/{appId}/healthrules:

```
curl -v --user admin@customer1:root
http://localhost:8090/controller/api/accounts/2/applications/5/healt
hrules
```

Accounts API

The Accounts APIs let you manage and monitor accounts and users, and other aspects of AppDynamics licensing.

For information about licensing, see License Management.

The Accounts APIs are made up of the following modules:

- actionsuppressions
- businesstransactions
- healthrules
- licensemodules
- mdsconfig
- nodes
- policies
- users

Application Model API

On this page:

Related pages:

- Metric and Snapshot API
- AppDynamics Application Performance Management Platform Notes

The applications API lets you retrieve information about the monitored environment as modeled in AppDynamics. This information includes, for example, the names and IDs of the business applications, business transactions, tiers, and nodes in the modeled environment.

Retrieve All Business Applications

The applications API method returns the business applications names and internal numeric identifier. Many of the operations in the Controller APIs occur in the context of a business application, so you can use this method to discover the application names or IDs to use before invoking other methods.

Format

GET /controller/rest/applications

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
timeRange	Time	Time parameter to filter data based on time range options (timeRange, startTime, endTime). If the timeRange option is provided, the query returns alive applications in the provided time range only, otherwise, the query returns all applications. For more information on using time ranges, see Metric and Snapshot API.	No

An alive application is an application with at least one node that submits at least one metric to the Controller in the provided time range.

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications

<applications>
   <application>
     <id>5</id>
     <name>ECommerce_E2E</name>
   </application>
   <application>
     <id>8</id>
     <name>ECommerce_E2E-Fulfillment</name>
   </application>
   <application>
     <id>11</id>
     <name>jimix12110919</name>
     <description></description>
     <accountGuid>429c7884-3f36-4b5a-9412-fdf827e6c86e</accountGuid>
   </application>
</applications>
```

Retrieve All Business Transactions in a Business Application

Format

GET /controller/rest/applications/*application_name*/business-transactions

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The application name or application ID.	Yes
exclude	Query	If false, the query retrieves only the business transactions that are included for monitoring. If true, the query retrieves only the excluded business transactions. Excluded business transactions are those that have been configured to be excluded from monitoring either from the UI or through the REST interface. The default is false.	No
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
timeRange	Time	Time parameter to filter data based on time range options (timeRange, startTime, endTime). If the timeRange option is provided, the query returns alive business transactions in the provided time range only, otherwise, the query returns all business transactions. For more information on using time ranges, see Metric and Snapshot API.	No

An alive business transaction is a transaction that submits at least one metric to the Controller in the provided time range.

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications/5/business-
transactions

<business-transactions>
   <business-transaction>
     <id>92</id>
     <name>/user/.POST</name>
     <entryPointType>WEB_SERVICE</entryPointType>
     <internalName>/user/.POST</internalName>
     <tierId>9</tierId>
     <tierName>ECommerce-Services</tierName>
     <background>false</background>
   </business-transaction>
   ...
   <business-transaction>
     <id>184</id>
     <name>OrderServiceImplService.createOrder</name>
     <entryPointType>WEB_SERVICE</entryPointType>
     <internalName>OrderServiceImplService.createOrder</internalName>
     <tierId>12</tierId>
     <tierName>Inventory-Services</tierName>
     <background>false</background>
   </business-transaction>
</business-transactions>
```

Retrieve All Tiers in a Business Application

Format

GET /controller/rest/applications/*application_name*/tiers

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The application name or application ID.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
timeRange	Time	Time parameter to filter data based on time range options (timeRange, startTime, endTime). If the timeRange option is provided, the query returns alive tiers in the provided time range only, otherwise, the query returns all tiers. For more information on using time ranges, see Metric and Snapshot API.	No

An alive tier is a tier with at least one node in this tier that submits at least one metric to Controller in the provided time range.

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications/5/tiers

<tiers>
<tier>
  <id>8</id>
  <name>Address-Services</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
<tier>
  <id>16</id>
  <name>Customer-Survey-Services</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
<tier>
  <id>9</id>
  <name>ECommerce-Services</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>2</numberOfNodes>
</tier>
<tier>
  <id>12</id>
  <name>Inventory-Services</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
<tier>
  <id>17</id>
  <name>Order-Processing-Services</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
<tier>
  <id>18</id>
  <name>Web-Tier-Services</name>
  <type>Web Server</type>
  <agentType>NATIVE_WEB_SERVER</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
</tiers>
```

Retrieve All Registered Backends in a Business Application With Their Properties

Format

GET /controller/rest/applications/*application_name*/backends

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	Provide either the application name or application id.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications/5/backends

<backends><backend>
  <id>10</id>
  <name>INVENTORY-MySQL DB-DB</name>
  <exitPointType>JDBC</exitPointType>
  <properties>
    <name-value>
      <id>0</id>
      <name>HOST</name>
      <value>DB</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>MAJOR_VERSION</name>
      <value>5.5.44-0ubuntu0.14.04.1</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>PORT</name>
      <value>3306</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>SCHEMA</name>
      <value>INVENTORY</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>URL</name>

<value>jdbc:mysql://db:3306/inventory?useUnicode=true&amp;characterE
ncoding=UTF-8&amp;autoReconnect=true</value>
```

```
    </name-value>
    <name-value>
      <id>0</id>
      <name>VENDOR</name>
      <value>MySQL DB</value>
    </name-value>
  </properties>
  <applicationComponentNodeId>0</applicationComponentNodeId>
  <tierId>0</tierId>
</backend>
 ...
<backend>
  <id>14</id>
  <name>Active MQ-OrderQueue</name>
  <exitPointType>JMS</exitPointType>
  <properties>
    <name-value>
      <id>0</id>
      <name>DESTINATION_NAME</name>
      <value>OrderQueue</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>DESTINATION_TYPE</name>
      <value>QUEUE</value>
    </name-value>
    <name-value>
      <id>0</id>
      <name>VENDOR</name>
      <value>Active MQ</value>
    </name-value>
  </properties>
  <applicationComponentNodeId>0</applicationComponentNodeId>
```

```
  <tierId>0</tierId>
</backend>
</backends>
```

Retrieve Node Information for All Nodes in a Business Application

Format

GET /controller/rest/applications/*application_name*/nodes

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	Provide either the application name or application id.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
timeRange	Time	Time parameter to filter data based on time range options (timeRange, startTime, endTime). If the timeRange option is provided, the query returns alive nodes in the provided time range only, otherwise, the query returns all nodes. For more information on using time ranges, see Metric and Snapshot API.	No

An alive node is a node which submits at least one metric to Controller in the provided time range.

Example

```
curl --user user1@customer1:welcome
http://demo.appdynamics.com:8090/controller/rest/applications/5/nodes


<nodes><node>
  <id>7</id>
  <name>Node_8000</name>
  <type>Tomcat 5.x</type>
  <tierId>12</tierId>
  <tierName>ECommerce Server</tierName>
  <machineId>3</machineId>
  <machineName>TIER1TOMCAT</machineName>
  <machineOSType>Linux</machineOSType>
  <machineAgentPresent>true</machineAgentPresent>
  <machineAgentVersion>Machine Agent v4.2.0.0 GA Build Date
2015-12-18 18:47:15</machineAgentVersion>
  <appAgentPresent>true</appAgentPresent>
  <appAgentVersion>Server Agent v4.2.0.0 GA #10145
r514d60d3122bd992e7152820d2ca5fb5ff4e45c1
8409-master-build</appAgentVersion>
  <agentType>APP_AGENT</agentType>
</node>
...
<node>
  <id>10</id>
  <name>Node_8002</name>
  <type>Tomcat 5.x</type>
  <tierId>14</tierId>
  <tierName>Inventory Server</tierName>
  <machineId>6</machineId>
  <machineName>TIER3TOMCAT</machineName>
  <machineOSType>Linux</machineOSType>
  <machineAgentPresent>true</machineAgentPresent>
  <machineAgentVersion>Machine Agent v4.2.0.0 GA Build Date
2015-12-18 18:47:15</machineAgentVersion>
  <appAgentPresent>true</appAgentPresent>
  <appAgentVersion>Server Agent v4.2.0.0 GA #10145
r514d60d3122bd992e7152820d2ca5fb5ff4e45c1
8409-master-build</appAgentVersion>
  <agentType>APP_AGENT</agentType>
</node>
</nodes>
```

Retrieve Node Information by Node Name

Format

GET /controller/rest/applications/*application_name*/nodes/*node_name*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The application name or application ID.	Yes
node_name	URI	The node name or ID	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user user1@customer1:welcome
http://demo.appdynamics.com:8090/controller/rest/applications/5/node
s/10

<nodes><node>
  <id>10</id>
  <name>Node_8002</name>
  <type>Tomcat 5.x</type>
  <tierId>14</tierId>
  <tierName>Inventory Server</tierName>
  <machineId>6</machineId>
  <machineName>TIER3TOMCAT</machineName>
  <machineOSType>Linux</machineOSType>
  <machineAgentPresent>true</machineAgentPresent>
  <machineAgentVersion>Machine Agent v4.2.0.0 GA Build Date
2015-12-18 18:47:15</machineAgentVersion>
  <appAgentPresent>true</appAgentPresent>
  <appAgentVersion>Server Agent v4.2.0.0 GA #10145
r514d60d3122bd992e7152820d2ca5fb5ff4e45c1
8409-master-build</appAgentVersion>
  <ipAddresses>
    <ipAddress>10.0.32.138</ipAddress>
  </ipAddresses>
  <agentType>APP_AGENT</agentType>
</node>
</nodes>
```

Retrieve Node Information for All Nodes in a Tier

Format

GET /controller/rest/applications/*application_name*/tiers/*tier_name*/nodes

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The application name or application ID.	Yes
tier_name	URI	The tier name or ID.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
timeRange	Time	Time parameter to filter data based on time range options (timeRange, startTime, endTime). If timeRange option is provided, the query returns alive nodes in the provided time range only, otherwise, the query returns all nodes For more information on using time ranges, see Metric and Snapshot API.	No

Example

```
curl --user user1@customer1:welcome
http://demo.appdynamics.com:8090/controller/rest/applications/25/tie
rs/70/nodes

<nodes><node>
  <id>81</id>
  <name>PHP_Node</name>
  <type>Other</type>
  <tierId>70</tierId>
  <tierName>PHP_Tier</tierName>
  <machineId>65</machineId>
  <machineName>232fe50b8f9c</machineName>
  <machineOSType>Linux</machineOSType>
  <machineAgentPresent>false</machineAgentPresent>
  <appAgentPresent>true</appAgentPresent>
  <appAgentVersion>Proxy v4.2.0.0 GA
SHA-1:.c86ec090f4ff77195df065fe56dade4dfc3913aa #9909
8869-master-build</appAgentVersion>
  <ipAddresses>
    <ipAddress>fe80:0:0:0:42:acff:fe11:2%eth0</ipAddress>
    <ipAddress>172.17.0.2</ipAddress>
  </ipAddresses>
  <agentType>PHP_APP_AGENT</agentType>
</node>
</nodes>
```

Retrieve Tier Information by Tier Name

Format

GET /controller/rest/applications/*application_name*/tiers/*tier_name*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The application name or application ID.	Yes
tier_name	URI	Tier name or ID.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user user1@customer1:welcome
http://demo.appdynamics.com:8090/controller/rest/applications/5/tier
s/14

<tiers><tier>
  <id>14</id>
  <name>Inventory Server</name>
  <type>Application Server</type>
  <agentType>APP_AGENT</agentType>
  <numberOfNodes>1</numberOfNodes>
</tier>
</tiers>
```

Metric and Snapshot API

On this page:

The Controller Metrics and Events API allows you to retrieve metric data information and information on various types of activities in your monitored environment, including Controller events. Additionally, you can configure how long you retain metrics.

> The AppDynamics Dexter Data Extraction Enhanced Reporting extension provides an alternative to using a REST client to get metric data. The tool makes AppDynamics data queryable in the manner of a data warehouse. See the extension page on the AppDynamics Community Exchange for more information.

Retrieve Metric Hierarchy

Returns information about the metric tree structure. In the response, if a child element is a container item, its `type` value is `folder`. Otherwise, the `type` tag for the child element is `leaf`.

The API retrieves the first generation of child elements. You can expand only the children of the folder type.

Format

GET /controller/rest/applications/*application_name*/metrics

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	The name or ID of the business or EUM (browser/mobile/IoT) application. Use the call to get the application ID in the Application Model API.	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECommerce_
E2E/metrics"

<metric-items><metric-item>
  <type>folder</type>
  <name>Backends</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Service Endpoints</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>End User Experience</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Errors</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Business Transaction Performance</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Information Points</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Overall Application Performance</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Application Infrastructure Performance</name>
</metric-item>
<metric-item>
  <type>folder</type>
  <name>Mobile</name>
</metric-item></metric-items>
```

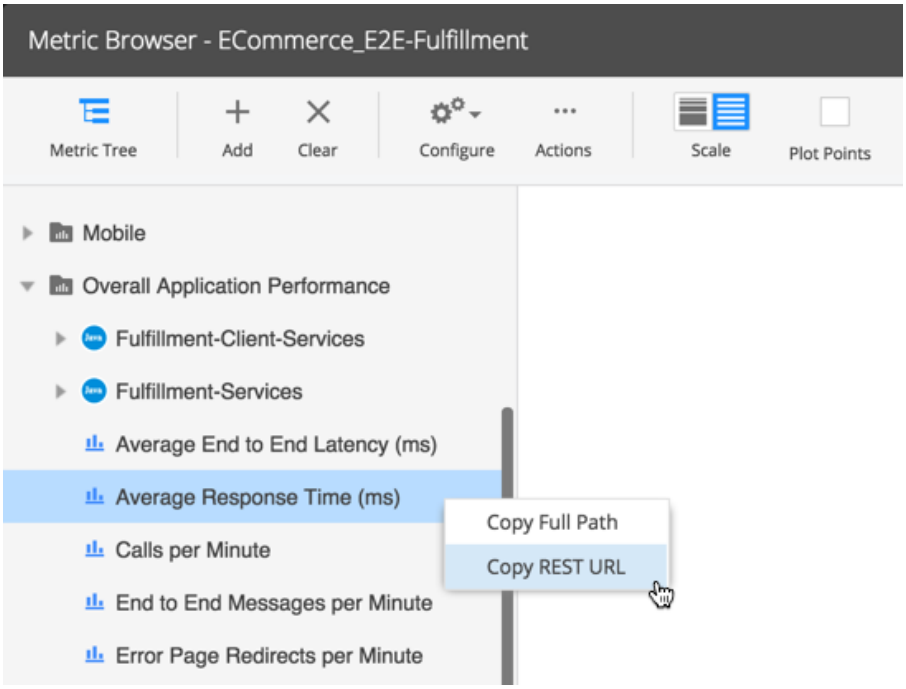
Retrieve Metric Data

The metric data method lets you get values generated for metrics. To use the method, you need to specify these parameters to the API:

- The path of the metric to retrieve.

- The time frame for the data.

The easiest way to learn how to construct the metric path and time range-related parameters is by using the Controller UI. When you right-click on the metric in the Metric Browser, a menu option appears for copying the full REST URL for the metric. The copied URL includes the path to this metric and time range selected in the UI.



Certain clients, such as most web browser, can also accept and properly encode the *full path* value as the metric path parameter. You can see the full path by hovering over the metric in the tree or copy it using the **Copy Full Path** option in the right-click menu. For reading clarity, certain examples below are shown with the full path value rather than the fully encoded URL value. If you test calls with the full path yourself, be sure to avoid having a pipe character appear at the start or end of the path.

The following sections provide additional details and examples for the metric data method:

- Metric Response Values
- Using Wildcards
- Using Time Ranges
- Retrieving All Other Traffic Business Transaction Metrics

Format

GET /controller/rest/applications/*application_name*/metric-data

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
metric-path	Query	The path to the metric in the metric hierarchy.	Yes
rollup	Query	By default, the values of the returned metrics are rolled up into a single data point (rollup=true). To get separate results for all values within the time range, set the rollup parameter to false in the query.	No

Additional mandatory parameters for specifying time ranges are described in Using Time Ranges.

Example

Retrieve metric values for a metric at an absolute path:

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECommerce_
E2E/metric-data?metric-path=Overall%20Application%20Performance%7CAv
erage%20Response%20Time%20%28ms%29&time-range-type=BEFORE_NOW&durati
on-in-mins=15"

<metric-datas><metric-data>
  <metricId>2339</metricId>
  <metricPath>Overall Application Performance|Average Response Time
(ms)</metricPath>
  <metricName>BTM|Application Summary|Average Response Time
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450562160000</startTimeInMillis>
      <value>302</value>
      <min>0</min>
      <max>15212</max>
      <current>15212</current>
      <sum>97800</sum>
      <count>324</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
</metric-datas>
```

Metric Response Values

Metrics results include the following fields:

Name	Definition
current	Value for the current minute. Used only when the time rollup type used by the Controller is current.
count	Number of times the agent collected the metric over the selected time period.
min, max	The minimum and maximum values reported across the selected time period. These are not used for all metric types.
occurrences	Number of data samples taken by the Controller to calculate the standard deviation.
standardDeviation	Intermediate values calculated by the Controller during time rollup used to calculate standard deviation. See Dynamic Baselines for information on how this value is calculated.
startTimeInMillis	The startTimeInMillis is the start time of the time range to which the result metric data applies, in UNIX epoch time.

sum	Total accumulated value for the metric over the selected time period.
useRange	Used internally by the Controller to process the metric.
value	The "value" value is one of the following for all metric values reported across the configured evaluation time length: • arithmetic average, if the metric time rollup type is average • sum, if the metric time rollup type is sum • latest, if the metric time rollup type is current

> `min` and `max` values are not available for any count- or sum-based metric except when the metric is rolled up to hourly or daily data points. Count- and sum-based metrics include errors per minute, calls per minute, and so on.

Using Wildcards

When you copy the REST URL in the Metric Browser, you get the path to a specific metric within a specific application and tier. Alternatively, you can use wildcard characters in one or more the steps in the URL path get metric data for entities, including multiple business transactions, tiers, or nodes.

The following format examples show where to put wildcard characters in various metric paths to achieve particular results. For reading clarity, these format examples use the "full path" for the metric (rather than the REST URL). For a full working example, click the expanding link under each format listing:

- Retrieve the app agent availability time for all tiers in the application using a wildcard for the tier name:

```
/controller/rest/applications/ECommerce_E2E-Fulfillment/metric-
data?metric-path=Application Infrastructure
Performance|*|Agent|App|Availability&time-range-type=BEFORE_NOW
&duration-in-mins=15
```

Click to view a full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E-Fulfillment/metric-data?metric-path=Application%20
Infrastructure%20Performance%7C*%7CAgent%7CApp%7CAvailability
&time-range-type=BEFORE_NOW&duration-in-mins=15"

<metric-datas><metric-data>
  <metricId>2329</metricId>
  <metricPath>Application Infrastructure
Performance|Fulfillment-Services|Agent|App|Availability</metr
icPath>
  <metricName>Agent|App|Availability</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450562460000</startTimeInMillis>
      <value>1</value>
      <min>0</min>
      <max>0</max>
```

```
      <current>1</current>
      <sum>15</sum>
      <count>15</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>2329</metricId>
  <metricPath>Application Infrastructure
Performance|Fulfillment-Client-Services|Agent|App|Availabilit
y</metricPath>
  <metricName>Agent|App|Availability</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450562460000</startTimeInMillis>
      <value>1</value>
      <min>0</min>
      <max>0</max>
      <current>1</current>
      <sum>15</sum>
      <count>15</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
```

```
  </metricValues>
</metric-data>
</metric-datas>
```

- Retrieve the CPU % Busy metric for all the nodes in all tiers using a wildcard for the tier and node names:

```
/controller/rest/applications/ECommerce_E2E-Fulfillment/metric-
data?metric-path=Application Infrastructure
Performance|*|Individual Nodes|*|Hardware
Resources|CPU|%Busy&time-range-type=BEFORE_NOW&duration-in-mins
=15
```

Click to view a full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E-Fulfillment/metric-data?metric-path=Application%20
Infrastructure%20Performance%7C*%7CIndividual%20Nodes%7C*%7CH
ardware%20Resources%7CCPU%7C%25Busy&time-range-type=BEFORE_NO
W&duration-in-mins=15"

<metric-datas><metric-data>
  <metricId>2231</metricId>
  <metricPath>Application Infrastructure
Performance|Fulfillment-Client-Services|Individual
Nodes|FulfillmentClient|Hardware
Resources|CPU|%Busy</metricPath>
  <metricName>Hardware Resources|CPU|%Busy</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563000000</startTimeInMillis>
      <value>10</value>
      <min>2</min>
      <max>82</max>
      <current>6</current>
      <sum>4474</sum>
      <count>450</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
```

```
<metric-data>
  <metricId>2231</metricId>
  <metricPath>Application Infrastructure
Performance|Fulfillment-Services|Individual
Nodes|Fulfillment|Hardware Resources|CPU|%Busy</metricPath>
  <metricName>Hardware Resources|CPU|%Busy</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563000000</startTimeInMillis>
      <value>10</value>
      <min>2</min>
      <max>82</max>
      <current>6</current>
      <sum>4478</sum>
      <count>450</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
```

```
  </metricValues>
</metric-data>
</metric-datas>
```

- Retrieve the Calls per Minute metric for all the business transactions on the ECommerce tier using a wildcard for the business transaction name:

```
/controller/rest/applications/ACME Book Store
Application/metric-data?metric-path=Business Transaction
Performance|Business Transactions|ECommerce Server|*|Calls per
Minute&time-range-type=BEFORE_NOW&duration-in-mins=15
```

Click to view a full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E/metric-data?metric-path=Business%20Transaction%20P
erformance%7CBusiness%20Transactions%7CECommerce-Services%7C*
%7CCalls%20per%20Minute&time-range-type=BEFORE_NOW&duration-i
n-mins=15"
<metric-datas><metric-data>
  <metricId>4042</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/json/cart/all.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:125|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>5</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
```

```
  <metricId>9784</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/appdynamicspilot/WEB-INF|Cal
ls per Minute</metricPath>
  <metricName>BTM|BTs|BT:183|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>5574</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/appdynamicspilot/404.jsp|Cal
ls per Minute</metricPath>
  <metricName>BTM|BTs|BT:140|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4033</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/json/items/all.GET|Calls per
Minute</metricPath>
```

```
  <metricName>BTM|BTs|BT:124|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>5</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4060</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/user/login.POST|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:127|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>5</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>5592</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/cart/{id}.DELETE|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:142|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
```

```
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>5583</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/cart/{id}.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:141|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4024</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/json/cart/co.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:123|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
```

```
      <max>0</max>
      <current>0</current>
      <sum>5</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>2477</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/user/.POST|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:92|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>5</value>
      <min>0</min>
      <max>0</max>
      <current>3</current>
      <sum>71</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>5601</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/cart/co.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:143|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
```

```
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4099</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|UserLogin.memberLogin|Calls
per Minute</metricPath>
  <metricName>BTM|BTs|BT:129|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4138</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/appdynamicspilot/|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:132|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>4</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
```

```
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4108</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewItems.getAllItems|Calls
per Minute</metricPath>
  <metricName>BTM|BTs|BT:130|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>4</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4129</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.sendItems|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:131|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>2</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4051</metricId>
```

```
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/json/cart/{id}.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:126|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>1</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>13</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4156</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>3</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4147</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/json/fault/getfaults.GET|Cal
ls per Minute</metricPath>
  <metricName>BTM|BTs|BT:133|Component:9|Calls per
```

```
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>9</value>
      <min>0</min>
      <max>0</max>
      <current>9</current>
      <sum>130</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>2630</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|/items/all.GET|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:93|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450563420000</startTimeInMillis>
      <value>5</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>76</sum>
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4090</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|UserLogOut.memberLogOut|Calls
per Minute</metricPath>
  <metricName>BTM|BTs|BT:128|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
```

```
    <startTimeInMillis>1450563420000</startTimeInMillis>
    <value>0</value>
    <min>2147483647</min>
    <max>-2147483648</max>
    <current>0</current>
    <sum>0</sum>
    <count>0</count>
    <standardDeviation>0.0</standardDeviation>
    <occurrences>0</occurrences>
    <useRange>false</useRange>
  </metric-value>
```

```
  </metricValues>
</metric-data>
</metric-datas>
```

- Retrieve data for multiple metrics for the ViewCart.addToCart transaction on the ECommerce-Services server:

```
/controller/rest/applications/ECommerce_E2E/metric-data?metric-
path=Business Transaction Performance|Business
Transactions|ECommerce
Server|ViewCart.addToCart|*&time-range-type=BEFORE_NOW&duration
-in-mins=15
```

Click to view a full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E/metric-data?metric-path=Business%20Transaction%20P
erformance%7CBusiness%20Transactions%7CECommerce-Services%7CV
iewCart.addToCart%7C*&time-range-type=BEFORE_NOW&duration-in-
mins=15"

<metric-datas><metric-data>
  <metricId>4155</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Average
Response Time (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Average Response
Time (ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>28</value>
      <min>0</min>
      <max>32</max>
      <current>0</current>
      <sum>84</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
```

```
<metric-data>
  <metricId>4159</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Number of
Very Slow Calls</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Number of Very Slow
Calls</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4157</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Errors per
Minute</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Errors per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4161</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Average
```

```
CPU Used (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Average CPU Used
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>18</value>
      <min>0</min>
      <max>20</max>
      <current>0</current>
      <sum>54</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4160</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Stall
Count</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Stall
Count</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4411</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|95th
Percentile Response Time (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|95th Percentile
Response Time (ms)</metricName>
  <frequency>ONE_MIN</frequency>
```

```
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>28</value>
      <min>0</min>
      <max>32</max>
      <current>0</current>
      <sum>84</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4335</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Normal
Average Response Time (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Normal Average
Response Time (ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>28</value>
      <min>0</min>
      <max>32</max>
      <current>0</current>
      <sum>84</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4162</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Average
Block Time (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Average Block Time
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
```

```
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>0</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4163</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Average
Wait Time (ms)</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Average Wait Time
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>0</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4156</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Calls per
Minute</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Calls per
Minute</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>0</min>
      <max>0</max>
      <current>0</current>
      <sum>3</sum>
```

```
      <count>30</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4331</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Average
Request Size</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Average Request
Size</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>740</value>
      <min>0</min>
      <max>1057</max>
      <current>0</current>
      <sum>2221</sum>
      <count>3</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
<metric-data>
  <metricId>4158</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce-Services|ViewCart.addToCart|Number of
Slow Calls</metricPath>
  <metricName>BTM|BTs|BT:134|Component:9|Number of Slow
Calls</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450566420000</startTimeInMillis>
      <value>0</value>
      <min>2147483647</min>
      <max>-2147483648</max>
      <current>0</current>
      <sum>0</sum>
      <count>0</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>false</useRange>
```

```
    </metric-value>
```

```
  </metricValues>
</metric-data>
</metric-datas>
```

Disabling Data Rollup

By default, metric data is rolled up for the time frame you request. You can set the rollup parameter to false to get all data points within the time frame. For example:

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECommerce_
E2E/metric-data?rollup=false&metric-path=Overall%20Application%20Per
formance%7CAverage%20Response%20Time%20%28ms%29&time-range-type=BEFO
RE_NOW&duration-in-mins=15"

<metric-datas><metric-data>
  <metricId>2339</metricId>
  <metricPath>Overall Application Performance|Average Response Time
(ms)</metricPath>
  <metricName>BTM|Application Summary|Average Response Time
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450503540000</startTimeInMillis>
      <value>334</value>
      <min>0</min>
      <max>3340</max>
      <current>2</current>
      <sum>6678</sum>
      <count>20</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
    <metric-value>
      <startTimeInMillis>1450503600000</startTimeInMillis>
      <value>771</value>
      <min>1</min>
      <max>11235</max>
      <current>4113</current>
      <sum>15424</sum>
      <count>20</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
```

```
    </metric-value>
    <metric-value>
      <startTimeInMillis>1450503660000</startTimeInMillis>
      <value>215</value>
      <min>0</min>
      <max>4249</max>
      <current>3</current>
      <sum>4306</sum>
      <count>20</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  ...
```

```
  </metricValues>
</metric-data>
</metric-datas>
```

Calling metric data with rollups disabled can result in a considerable number of results.

It is important to note that a limit exists on the amount of data that can be returned for the metric data method. The maximum result is 200 metrics.

Using Time Ranges

You can fetch metric data for any time range, including for a range between specific points (for example, from 2:00 to 2:15 pm Monday) or for a relative time range (the last 15 minutes).

Time-based input parameters for the metric data API method let you specify a time range in several ways, as described in the following table.

Time Range Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
time-range-type	Query	Possible values are: • **BEFORE_NOW**: To use this value, you must also specify the "duration-in-mins" parameter. • **BEFORE_TIME**: To use this value, you must also specify the "duration-in-mins" and "end-time" parameters. • **AFTER_TIME**: To use this value, you must also specify the "duration-in-mins" and "start-time" parameters. • **BETWEEN_TIMES**: To use this value, you must also specify the "start-time" and "end-time" parameters. The "BETWEEN_TIMES" range includes the start- time and excludes the end-time.	Yes
duration-in-mins	Query	Duration (in minutes) to return the metric data.	If time-range-type is **BEFORE_NOW**, **BEFORE_TIME**, or **AFTER_TIME**
start-time	Query	Start time (in milliseconds) from which the metric data is returned in UNIX epoch time.	If time-range-type is **AFTER_TIME** or **BETWEEN_TIMES**
end-time	Query	End time (in milliseconds) until which the metric data is returned in UNIX epoch time.	If time-range-type is **BEFORE_TIME** or **BETWEEN_TIMES**

Examples

Most examples on this page use the past 15 minutes as the request time range. The following format examples show other ways you can define the time range for the request.

- Time range of the 15 minutes after December 19, 2015 5:40:00 AM GMT:

```
?time-range-type=AFTER_TIME&start-time=1450532400000&duration-in-mins=15
```

- Time range of the 15 minutes before December 19, 2015 6:00:00 AM GMT.

```
?time-range-type=BEFORE_TIME&end-time=1450533600000&duration-in
-mins=15
```

- Time range between December 19, 2015 6:00:00 AM GMT and December 19, 2015 6:30:00 AM GMT:

```
?time-range-type=BETWEEN_TIMES&start-time=1450533600000&end-tim
e=1450535400000
```

Retrieving All Other Traffic Business Transaction Metrics

The All Other Traffic business transaction is a type of business transaction that aggregates traffic for new transactions once the business transaction registration limits are reached. See Business Transactions for more about the All Other Traffic business transactions.

The All Other Traffic business transaction uses a special identifier in API URI paths, _APPDYNAMICS_DEFAULT_TX_ The following example shows an example of retrieving the average CPU used by the All Other Traffic business transaction:

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com:8090/controller/rest/applications/ACME%
20Book%20Store%20Application/metric-data?metric-path=Business%20Tran
saction%20Performance%7CBusiness%20Transactions%7CECommerce%20Server
%7C_APPDYNAMICS_DEFAULT_TX_%7CAverage%20CPU%20Used%20%28ms%29&time-r
ange-type=BEFORE_NOW&duration-in-mins=15"

<metric-datas><metric-data>
  <metricId>4000</metricId>
  <metricPath>Business Transaction Performance|Business
Transactions|ECommerce Server|_APPDYNAMICS_DEFAULT_TX_|Average CPU
Used (ms)</metricPath>
  <metricName>BTM|BTs|BT:78|Component:12|Average CPU Used
(ms)</metricName>
  <frequency>ONE_MIN</frequency>
  <metricValues>
    <metric-value>
      <startTimeInMillis>1450570800000</startTimeInMillis>
      <value>22</value>
      <min>0</min>
      <max>50</max>
      <current>20</current>
      <sum>3140</sum>
      <count>146</count>
      <standardDeviation>0.0</standardDeviation>
      <occurrences>0</occurrences>
      <useRange>true</useRange>
    </metric-value>
  </metricValues>
</metric-data>
</metric-datas>
```

Retrieve Transaction Snapshots

Snapshots contain details on transactions, by request segment. The time range parameters are the same for snapshots as for retrieving metrics. You can similarly specify a relative time range or a specific range. See Using Time Ranges for more information on using time range parameters.

Format

GET /controller/rest/applications/*application_name*/request-snapshots

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_name	URI	Provide either the application name or application id.	Yes

time-range-type	Query	Possible values are: • **BEFORE_NOW**: To use this value, you must also specify the "duration-in-mins" parameter. • **BEFORE_TIME**: To use this value, you must also specify the "duration-in-mins" and "end-time" parameters. • **AFTER_TIME**: To use this value, you must also specify the "duration-in-mins" and "start-time" parameters. • **BETWEEN_TIMES**: To use this value, you must also specify the "start-time" and "end-time" parameters. The "BETWEEN_TIMES" range includes the start- time and excludes the end-time.	Yes
duration-in-mins	Query	Duration (in minutes) to return the data.	If time-range-type is **BEFORE_NOW**, **BEFORE_TIME**, or **AFTER_TIME**
start-time	Query	Start time (in milliseconds) from which the data is returned.	If time-range-type is **AFTER_TIME** or **BETWEEN_TIMES**
end-time	Query	End time (in milliseconds) until which the data is returned.	If time-range-type is **BEFORE_TIME** or **BETWEEN_TIMES**
guids	Query	Array of comma-separated guids for the transaction snapshots. If not specified, retrieves all snapshots in the specified time range.	No
archived	Query	True to retrieve archived snapshots. Default is false.	No
deep-dive-policy	Query	Array of comma-separated snapshot policy filters to apply. Valid values are: • SLA_FAILURE • TIME_SAMPLING • ERROR_SAMPLING • OCCURRENCE_SAMPLING • ON_DEMAND • APPLICATION_STARTUP • SLOW_DIAGNOSTIC_SESSION • ERROR_DIAGNOSTIC_SESSION • POLICY_FAILURE_DIAGNOSTIC_SESSION • DIAGNOSTIC_SESSION • INFLIGHT_SLOW_SESSION	No
application-component-ids	Query	Array of comma-separated tier IDs to filter. Default is all the tiers in the application.	No
application-component-node-ids	Query	Array of comma-separated node ID filters. Default is all the nodes in the application	No
business-transaction-ids	Query	Array of comma-separated business transaction ID filters. Default is all the business transactions in the application.	No
user-experience	Query	Array of comma-separated user experiences filters. Valid values are: • NORMAL • SLOW • VERY_SLOW • STALL • ERROR	No
first-in-chain	Query	If true, retrieve only the first request from the chain. Default is false.	No

need-props	Query	If true, the values of the following snapshot properties are included in the output. These values correspond to the values of the data-collector-type parameter. If false, the default, these values are empty in the output. • errorDetails • errorIDs • httpParameters • businessData • cookies • httpHeaders • sessionKeys • responseHeaders • logMessages • transactionProperties • transactionEvents • dotnetProperty	No
need-exit-calls	Query	If true, exit calls are included in the result. Default is false.	No
execution-time-in-milis	Query	If set, retrieves only data for requests with execution times greater than this value.	No
session-id	Query	If set, retrieves data only for this session id.	No
user-principal-id	Query	If set, retrieves data only for this user login.	No
error-ids	Query	Array of comma-separated error codes to filter by. Default is to retrieve all error codes.	No
starting-request-id, ending-request-id	Query	If set, retrieves data only for this range of request IDs.	No
error-occurred	Query	If true, retrieves only error requests. Default is false.	No
diagnostic-snapshot	Query	If true, retrieves only diagnostic snapshots. Default is false.	No
bad-request	Query	If true, retrieves only slow and error requests. Default is false.	No
diagnostic-session-guid	Query	Array of comma-separated diagnostic session guids to filter.	No
data-collector-name	Query	Used with data-collector-value to filter snapshot collection based on the value of a data collector.	No
data-collector-value	Query	Used with data-collector-name to filter snapshot collection based on the value of a data collector.	If data-collector-name is set.
data-collector-type	Query	Used with data-collector-name and data-collector-value to filter snapshot collection based on the value of a data collector. Some of the values contain spaces. All are case-sensitive and where indicated the spaces are required. Valid values are: • Error IDs • Stack Traces • Error Detail • Http Parameter • Business Data (This type is a method invocation data collector.) • Cookie • Http Header • Session Key • Response Header • Log Message • Transaction Property • Transaction Event • Dotnet Property • isProtoBuf • EUM Request GUID	
output	Query	HTTP Request parameter included as part of the URL to change the output format Valid values are "XML" (default) or "JSON".	No
maximum-results	Query	A number, if specified, this number of maximum results will be returned. If not specified, default 600 results can be returned at most.	No

Examples

- Retrieve list of transaction snapshots for the ACME Book Store:

```
/controller/rest/applications/ECommerce_E2E-Fulfillment/request
-snapshots?time-range-type=BEFORE_NOW&duration-in-mins=5
```

Click to view full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E-Fulfillment/request-snapshots?time-range-type=BEFO
RE_NOW&duration-in-mins=5"

<request-segment-datas><request-segment-data>
  <id>0</id>
  <archived>false</archived>

<requestGUID>18a9ae17-33a8-4d24-b3fa-558fe42b98b5</requestGUI
D>
  <businessTransactionId>113</businessTransactionId>
  <applicationId>8</applicationId>
  <applicationComponentId>14</applicationComponentId>
  <applicationComponentNodeId>13</applicationComponentNodeId>
  <async>false</async>
  <threadID>58</threadID>
  <threadName>http-nio-8080-exec-8</threadName>
  <localStartTime>1450574075422</localStartTime>
  <serverStartTime>1450574075422</serverStartTime>
  <firstInChain>true</firstInChain>
  <callChain>Component:14</callChain>
  <localID>0</localID>
  <errorOccured>false</errorOccured>
  <hasDeepDiveData>true</hasDeepDiveData>
  <userExperience>NORMAL</userExperience>
  <timeTakenInMilliSecs>3693</timeTakenInMilliSecs>
  <cpuTimeTakenInMilliSecs>19</cpuTimeTakenInMilliSecs>
  <warningThreshold>5318 ms. 3.0x of standard deviation
[453.7 ms] for moving average [3956.8 ms] (minimum baseline:
200 ms) for the last 11617 minutes.</warningThreshold>
  <criticalThreshold>5772 ms. 4.0x of standard deviation
[453.7 ms] for moving average [3956.8 ms] (minimum baseline:
600 ms) for the last 11617 minutes.</criticalThreshold>
  <summary>Scheduled Snapshots: one every 10
minutes.</summary>
  <errorSummary></errorSummary>
  <diagnosticSessionGUID></diagnosticSessionGUID>
```

```
  <deepDivePolicy>TIME_SAMPLING</deepDivePolicy>
  <delayedDeepDive>false</delayedDeepDive>
  <delayedDeepDiveOffSet>0</delayedDeepDiveOffSet>
  <exitCallsDataTruncated>false</exitCallsDataTruncated>
  <URL>/appdynamicspilot/rest/fulfillment</URL>
  <errorIDs/>
  <errorDetails/>
  <httpParameters/>
  <businessData/>
  <cookies/>
  <httpHeaders/>
  <sessionKeys/>
  <responseHeaders/>
  <logMessages/>
  <transactionProperties/>
  <transactionEvents/>

<unresolvedCallInCallChain>false</unresolvedCallInCallChain>
  <dotnetProperty/>
  <endToEndLatency>-1</endToEndLatency>
</request-segment-data>
...
<request-segment-data>
  <id>0</id>
  <archived>false</archived>

<requestGUID>bfce5066-2409-4a4b-a869-6afcc06614d6</requestGUI
D>
  <businessTransactionId>113</businessTransactionId>
  <applicationId>8</applicationId>
  <applicationComponentId>14</applicationComponentId>
  <applicationComponentNodeId>13</applicationComponentNodeId>
  <async>false</async>
  <threadID>60</threadID>
  <threadName>http-nio-8080-exec-10</threadName>
  <localStartTime>1450574082926</localStartTime>
  <serverStartTime>1450574082926</serverStartTime>
  <firstInChain>true</firstInChain>
  <callChain>Component:14</callChain>
  <localID>0</localID>
  <errorOccured>false</errorOccured>
  <hasDeepDiveData>true</hasDeepDiveData>
  <userExperience>NORMAL</userExperience>
  <timeTakenInMilliSecs>3634</timeTakenInMilliSecs>
  <cpuTimeTakenInMilliSecs>16</cpuTimeTakenInMilliSecs>
  <warningThreshold>5318 ms. 3.0x of standard deviation
[453.7 ms] for moving average [3956.8 ms] (minimum baseline:
200 ms) for the last 11617 minutes.</warningThreshold>
  <criticalThreshold>5772 ms. 4.0x of standard deviation
[453.7 ms] for moving average [3956.8 ms] (minimum baseline:
```

```
600 ms) for the last 11617 minutes.</criticalThreshold>
  <summary>[null]</summary>
  <errorSummary></errorSummary>
  <diagnosticSessionGUID></diagnosticSessionGUID>
  <deepDivePolicy>CROSS_APP_POLICY</deepDivePolicy>
  <delayedDeepDive>false</delayedDeepDive>
  <delayedDeepDiveOffSet>0</delayedDeepDiveOffSet>
  <exitCallsDataTruncated>false</exitCallsDataTruncated>
  <URL>/appdynamicspilot/rest/fulfillment</URL>
  <errorIDs/>
  <errorDetails/>
  <httpParameters/>
  <businessData/>
  <cookies/>
  <httpHeaders/>
  <sessionKeys/>
  <responseHeaders/>
  <logMessages/>
  <transactionProperties/>
  <transactionEvents/>

<unresolvedCallInCallChain>false</unresolvedCallInCallChain>
  <dotnetProperty/>
```

```
  <endToEndLatency>-1</endToEndLatency>
</request-segment-data>
</request-segment-datas>
```

- Retrieve list of transaction snapshots including the snapshot fields that are associated with an HTTP parameter data collector:

```
/controller/rest/applications/ECommerce_E2E-Fulfillment/request
-snapshots?time-range-type=BEFORE_NOW&duration-in-mins=5&data-c
ollector-type=Http
Parameter&data-collector-name=param1&data-collector-value=%5B10
0%5D&need-props=true
```

Click to view full example...

```
curl --user user1@customer1:secret
"http://demo.appdynamics.com/controller/rest/applications/ECo
mmerce_E2E-Fulfillment/request-snapshots?time-range-type=BEFO
RE_NOW&duration-in-mins=5&data-collector-type=Http%20Paramete
r&data-collector-name=param1&data-collector-value=%5B100%5D&n
eed-props=true"

<request-segment-datas><request-segment-data>
   <id>0</id>
   <archived>false</archived>

<requestGUID>07532d68-42b8-4a79-877a-dedf2912a2cf</requestGUI
D>
   <businessTransactionId>128</businessTransactionId>
   <applicationId>2</applicationId>
   <applicationComponentId>5</applicationComponentId>
   <applicationComponentNodeId>4</applicationComponentNodeId>
   <async>false</async>
   <threadID>60</threadID>
   <threadName>http-8000-Processor24</threadName>
   <localStartTime>1389164292752</localStartTime>
   <serverStartTime>1389164292752</serverStartTime>
   <firstInChain>true</firstInChain>
   <callChain>Component:5</callChain>
   <localID>0</localID>
   <errorOccured>true</errorOccured>
   <hasDeepDiveData>true</hasDeepDiveData>
   <userExperience>ERROR</userExperience>
   <timeTakenInMilliSecs>105</timeTakenInMilliSecs>
   <cpuTimeTakenInMilliSecs>3839000</cpuTimeTakenInMilliSecs>
```

```
   <summary>[Manual Diagnostic Session] -
org.hibernate.util.JDBCExceptionReporter : Cannot create
PoolableConnectionFactory (Unknown database 'appdy')
</summary>
   <errorSummary/>

<diagnosticSessionGUID>d70a41d9-a96f-46e8-9fbc-31061c6e452f</
diagnosticSessionGUID>
   <deepDivePolicy>ON_DEMAND</deepDivePolicy>
   <delayedDeepDive>false</delayedDeepDive>
   <delayedDeepDiveOffSet>0</delayedDeepDiveOffSet>
   <exitCallsDataTruncated>false</exitCallsDataTruncated>
   <URL>/appdynamicspilot/1.bookslist</URL>

<httpSessionID>088B2A2DD0EF77424DD0EB3346A441F9</httpSessionI
D>
   <errorIDs>
   <long>29</long>
   </errorIDs>
   <errorDetails>
   <name-value>
   <id>0</id>
   <name>1. org.hibernate.util.JDBCExceptionReporter</name>
   <value>org.hibernate.util.JDBCExceptionReporter : Cannot
create PoolableConnectionFactory (Unknown database
'appdy')</value>
   </name-value>
   </errorDetails>
   <httpParameters>
   <name-value>
   <id>0</id>
   <name>param1</name>
   <value>[100]</value>
   </name-value>
   </httpParameters>
   <businessData/>
   <cookies/>
   <httpHeaders/>
   <sessionKeys/>
   <responseHeaders/>
   <logMessages/>
   <transactionProperties>
   <name-value>
   <id>0</id>
   <name>Servlet URI</name>

<value>/appdynamicspilot/WEB-INF/presentation/bookslist.jsp</
value>
   </name-value>
   <name-value>
```

```
<id>0</id>
<name>ProcessID</name>
<value>65331</value>
</name-value>
</transactionProperties>
<transactionEvents/>
```

```
<unresolvedCallInCallChain>false</unresolvedCallInCallChain>
   <dotnetProperty/>
</request-segment-data></request-segment-datas>
```

Retrieve Controller Audit History

The Controller audit history is a record of the configuration and user activities in the Controller configuration. The ControllerAuditHistory API method returns the audit log for the time range specified. The output format can be JSON or CSV. This information is the same as that found in the `audit.log` file. See Platform Log Files for more information.

Format

GET /controller/ ControllerAuditHistory?startTime=<start-time>&endTime=<end-time>&include=<field>:<value>&exclude=<field>:<value>

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
start-time	Query	Start time in the format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Yes
end-time	Query	End time in the format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Yes
time-zone-id	Query	Time zone	No
include	Query	Restricted information in the Controller audit history	No
exclude	Query	Restricted information in the Controller audit history	No

To control the size of the output, the range between the start-time and end-time cannot exceed twenty-four hours. For periods longer than 24 hours, use multiple queries with consecutive time parameters.

Multiple filters of the same type are allowed. The backend API treats include filters with the same <field> with relationship "OR", and filters with different <field> with relationship "AND". There is no direct interaction between include and exclude filters.

Each filter needs to be a parameter, e.g. include=filterName1:filterValue1&include=filterName2:filterValue2. See the below examples.

Examples

```
http://localhost:8080/controller/ControllerAuditHistory?startTime=yyyy-MM-dd%27T%27HH:mm:ss.SSSZ&endTime=yyyy-MM-dd%27T%27HH:mm:ss.SSSZ?include=filterName1:filterValue1&include=filterName1:filterValue1&exclude=filterName1:filterValue1&exclude=filterName1:filterValue1
```

```
http://localhost:8080/controller/ControllerAuditHistory?startTime=yy
yy-MM-dd%27T%27HH:mm:ss.SSSZ&endTime=yyyy-MM-dd%27T%27HH:mm:ss.SSSZ?
include=filterName1:filterValue1&include=filterName1:filterValue1&ex
clude=filterName1:filterValue1&exclude=filterName1:filterValue1
curl --user user1@customer1:welcome
"http://demo.appdynamics.com:8090/controller/ControllerAuditHistory?
startTime=2015-12-19T10:50:03.607-0700&endTime=2015-12-19T17:50:03.6
07-0700&timeZoneId=America%2FSan%20Francisco&include=userName:user1&
include=action:LOGIN&exclude=accountName:system&exclude=action:OBJEC
T_UPDATE"

[{"timeStamp":1450569821811,"auditDateTime":"2015-12-20T00:03:41.811
+0000","accountName":"customer1","securityProviderType":"INTERNAL","
userName":"user1","action":"LOGIN"},{"timeStamp":1450570234518,"audi
tDateTime":"2015-12-20T00:10:34.518+0000","accountName":"customer1",
"securityProviderType":"INTERNAL","userName":"user1","action":"LOGIN
"},{"timeStamp":1450570273841,"auditDateTime":"2015-12-20T00:11:13.8
41+0000","accountName":"customer1","securityProviderType":"INTERNAL"
,"userName":"user1","action":"OBJECT_CREATED","objectType":"AGENT_CO
NFIGURATION"},
...
{"timeStamp":1450570675345,"auditDateTime":"2015-12-20T00:17:55.345+
0000","accountName":"customer1","securityProviderType":"INTERNAL","u
serName":"user1","action":"OBJECT_DELETED","objectType":"BUSINESS_TR
ANSACTION"},{"timeStamp":1450570719240,"auditDateTime":"2015-12-20T0
0:18:39.240+0000","accountName":"customer1","securityProviderType":"
INTERNAL","userName":"user1","action":"APP_CONFIGURATION","objectTyp
e":"APPLICATION","objectName":"ACME Book Store
Application"},{"timeStamp":1450571834835,"auditDateTime":"2015-12-20
T00:37:14.835+0000","accountName":"customer1","securityProviderType"
:"INTERNAL","userName":"user1","action

curl --user user1@customer1:welcome
"http://127.0.0.1:8080/controller/ControllerAuditHistory?startTime=2
019-05-28T08:00:03.607-0700&endTime=2019-05-28T11:32:03.607-0700&tim
eZoneId=America%2FSan%20Francisco&include=applicationName:ACME"
[{"timeStamp":1559066415823,"auditDateTime":"2019-05-28T18:00:15.823
+0000","accountName":"customer1","securityProviderType":"INTERNAL","
userName":"user1","action":"LOGIN","objectId":0,"applicationName":"A
CME"}]
```

Configure Metric Retention by Account

You can configure the Controller to purge stale metrics once a day based on account. Stale metrics are metrics that have not had new data reported based on the number of days configured. This only deletes EUM and SIM metrics that are more than two days old.

To configure this option, you must be the account owner.

Format

POST /controller/api/accounts/<account_id>/metricstaleduration/<number_of_days>

Input Parameters

Parameter Name	Parameter Type	Value
account_id	URI	The account ID.
number_of_days	Integer	The number of days you want to retain stale metrics.

Example

```
curl -X POST -u user1@customer1:secret
"http://demo.appdynamics.com:8090/controller/api/accounts/2/metricst
aleduration/3"
```

Configure Metric Retention by Application

You can configure the Controller to purge stale metrics once a day based on application. Stale metrics are metrics that have not had new data reported based on the number of days configured. This only deletes EUM and SIM metrics that are more than two days old.

To configure this option, you must have administrator permissions or higher.

Format

POST /controller/api/accounts/<account_id>/applications/<application_name>/metricstaleduration/<number_of_days>

Input Parameters

Parameter Name	Parameter Type	Value
account_id	URI	The account ID.
application_id	URI	The application ID.
number_of_days	Integer	The number of days you want to retain stale metrics.

Example

```
curl -X POST -u user1@customer1:secret
"http://demo.appdynamics.com:8090/controller/api/accounts/2/applicat
ions/12/metricstaleduration/3"
```

Alert and Respond API

On this page:

The Alert and Respond APIs let you manage and monitor events, health rules and other aspects of the AppDynamics alert and respond features.

For information about the alert and response features, see Alert and Respond.

Retrieve All Health Rule Violations in a Business Application

Returns all health rule violations that have occurred in an application within a specified time frame.

URI

/controller/rest/applications/application_id/problems/healthrule-violations

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	Provide either the application name or application id.	Yes
time-range-type	Query	Possible values are: **BEFORE_NOW** To use the "BEFORE_NOW" option, you must also specify the "duration-in-mins" parameter. **BEFORE_TIME** To use the "BEFORE_TIME" option, you must also specify the "duration-in-mins" and "end-time" parameters. **AFTER_TIME** To use the "AFTER_TIME" option, you must also specify the "duration-in-mins" and "start-time" parameters. **BETWEEN_TIMES** To use this option, you must also specify the "start-time" and "end-time" parameters. The "BETWEEN_TIMES" range includes the start- time and excludes the end-time.	Yes
duration-in-mins	Query	Duration (in minutes) to return the metric data.	If time-range-type is **BEFORE_NOW**, **BEFORE_TIME**, or **AFTER_TIME**
start-time	Query	Start time (in milliseconds) from which the metric data is returned.	If time-range-type is **AFTER_TIME** or **BETWEEN_TIMES**
end-time	Query	End time (in milliseconds) until which the metric data is returned.	If time-range-type is **BEFORE_TIME** or **BETWEEN_TIMES**
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Click here to view a full example...

```
http://demo.appdynamics.com/controller/rest/applications/7/problems/healthrule-violations?time-range-type=BEFORE_NOW&duration-in-mins=15
```

```
<policy-violations><policy-violation>
  <id>266</id>
  <name>CPU utilization is too high</name>
  <startTimeInMillis>1452630655000</startTimeInMillis>
  <detectedTimeInMillis>0</detectedTimeInMillis>
  <endTimeInMillis>1452630715000</endTimeInMillis>
  <incidentStatus>RESOLVED</incidentStatus>
  <severity>WARNING</severity>
  <triggeredEntityDefinition>
    <entityType>POLICY</entityType>
    <entityId>30</entityId>
    <name>CPU utilization is too high</name>
  </triggeredEntityDefinition>
  <affectedEntityDefinition>
    <entityType>APPLICATION_COMPONENT_NODE</entityType>
    <entityId>16</entityId>
    <name>Fulfillment</name>
  </affectedEntityDefinition>

<deepLinkUrl>http://demo.appdynamics.com/controller/#location=APP_INCIDENT_DETAIL&amp;incident=266</deepLinkUrl>
  <description>AppDynamics has detected a problem.<br><b>errorAbhi</b> is violating.

</description>
</policy-violation>
<policy-violation>
  <id>268</id>
  <name>CPU utilization is too high</name>
  <startTimeInMillis>1452630655000</startTimeInMillis>
  <detectedTimeInMillis>0</detectedTimeInMillis>
  <endTimeInMillis>1452630715000</endTimeInMillis>
  <incidentStatus>RESOLVED</incidentStatus>
  <severity>WARNING</severity>
  <triggeredEntityDefinition>
    <entityType>POLICY</entityType>
    <entityId>30</entityId>
    <name>CPU utilization is too high</name>
  </triggeredEntityDefinition>
  <affectedEntityDefinition>
    <entityType>APPLICATION_COMPONENT_NODE</entityType>
    <entityId>20</entityId>
    <name>FulfillmentClient</name>
  </affectedEntityDefinition>
  <deepLinkUrl>http://demo.appdynamics.com/controller/#location=APP_INCIDENT_DETAIL&amp;incident=268</deepLinkUrl>
  <description>AppDynamics has detected a problem with Node
&lt;b&gt;FulfillmentClient&lt;/b&gt;.&lt;br&gt;&lt;b&gt;CPU utilization is too high&lt;/b&gt;
started violating and is now &lt;b&gt;warning&lt;/b&gt;.&lt;br&gt;All of the following
conditions were found to be violating&lt;br&gt;For Node
&lt;b&gt;FulfillmentClient&lt;/b&gt;:&lt;br&gt;1) Hardware Resources|CPU|%Busy
Condition&lt;br&gt;&lt;b&gt;%Busy's&lt;/b&gt; value &lt;b&gt;76.0&lt;/b&gt; was &lt;b&gt;greater
than&lt;/b&gt; the threshold &lt;b&gt;75.0&lt;/b&gt; for the last &lt;b&gt;30&lt;/b&gt;
minutes&lt;br&gt;</description>
</policy-violation>
</policy-violations>
```

Retrieve Event Data

You can capture data for the event types listed in the event-types parameter.

URI

/controller/rest/applications/*application_id*/events

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	Provides either the application name or application id.	Yes
summary	Query	Provides the summary for the event.	Yes
comment	Query	Provides the comments (if any) for the event.	No
eventtype	Query	APPLICATION_DEVELOPMENT	Yes
time-range-type	Query	Possible values are: **BEFORE_NOW** To use the "BEFORE_NOW" option, you must also specify the "duration-in-mins" parameter. **BEFORE_TIME** To use the "BEFORE_TIME" option, you must also specify the "duration-in-mins" and "end-time" parameters. **AFTER_TIME** To use the "AFTER_TIME" option, you must also specify the "duration-in-mins" and "start-time" parameters. **BETWEEN_TIMES** To use the "BETWEEN_TIMES" option, you must also specify the "start-time" and "end-time" parameters. The "BETWEEN_TIMES" range includes the start- time and excludes the end-time.	Yes
duration-in-mins	Query	Specify the duration (in minutes) to return the metric data.	If time-range-type is **BEFORE_NOW**, **BEFORE_TIME**, or **AFTER_TIME**
start-time	Query	Specify the start time (in milliseconds) from which the metric data is returned.	If time-range-type is **AFTER_TIME** or **BETWEEN_TIMES**
end-time	Query	Specify the end time (in milliseconds) until which the metric data is returned.	If time-range-type is **BEFORE_TIME** or **BETWEEN_TIMES**
event-types	Query	Specify the comma-separated list of event types for which you want to retrieve event information. See the Events Reference for the valid event types.	Yes
severity	Query	Provides the severity level. Specify the comma-separated list of severities for which you want to retrieve event information. Allowed values are: • INFO • WARN • ERROR In the UI these values become "Info", "Warning", and "Critical".	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No
tier	Query	Name of the tier in the application	No

Example

Retrieve the list of events of type "APPLICATION_ERROR" or "DIAGNOSTIC_SESSION" of any severity that occurred in the specified time range:

```
curl --user user1@customer1:secret
http://demo.appdynamics.com//controller/rest/applications/6/events?t
ime-range-type=BEFORE_NOW\&duration-in-mins=30\&event-types=%20APPLI
CATION_ERROR,DIAGNOSTIC_SESSION\&severities=INFO,WARN,ERROR
```

```
<events><event>
  <id>44658</id>
  <type>DIAGNOSTIC_SESSION</type>
  <subType>ERROR_DIAGNOSTIC_SESSION</subType>
  <eventTime>1451343453085</eventTime>
  <severity>WARN</severity>
  <summary>Starting Diagnostic Session after series of errors for a
Business Transaction 18% (2/11) of requests had errors in the last
minute starting 12/28/15 10:57 PM local time</summary>
  <affectedEntities>
    <entity-definition>
      <entityType>APPLICATION</entityType>
      <entityId>6</entityId>
      <name>ECommerce</name>
    </entity-definition>
    <entity-definition>
      <entityType>APPLICATION_COMPONENT</entityType>
      <entityId>11</entityId>
      <name>ECommerce-Services</name>
    </entity-definition>
    <entity-definition>
      <entityType>APPLICATION_COMPONENT_NODE</entityType>
      <entityId>19</entityId>
      <name>ECommerce_WEB2</name>
    </entity-definition>
    <entity-definition>
      <entityType>BUSINESS_TRANSACTION</entityType>
      <entityId>35</entityId>
      <name>/items/all.GET</name>
    </entity-definition>
    <entity-definition>
      <entityType>MACHINE_INSTANCE</entityType>
      <entityId>8</entityId>
      <name>ECommerce-web1</name>
    </entity-definition>
  </affectedEntities>
  <triggeredEntity>
    <entityType>APPLICATION_COMPONENT_NODE</entityType>
    <entityId>19</entityId>
    <name>ECommerce_WEB2</name>
  </triggeredEntity>
  <markedAsRead>false</markedAsRead>
  <markedAsResolved>false</markedAsResolved>
  <archived>false</archived>

<deepLinkUrl>http://demo.appdynamics.com:8090/controller/#location=A
```

```
PP_EVENT_VIEWER_MODAL&amp;eventSummary=44658</deepLinkUrl>
</event>
</events>
```

Create Events

Application deployment events notify AppDynamics when you upgrade your application, push new code, etc. This lets you correlate these application deployment activities with other data inside AppDynamics. This is useful for regression analysis, root cause analysis, and performance studies. A useful practice is to include injection of your application deployment event into AppDynamics as part of the build process for deploying a new version of your application.

The AppDynamics REST API lets you integrate events of type "APPLICATION_DEPLOYMENT" with other systems.

For example, suppose you want to create an event automatically in your AppDynamics monitored system for every new release. To integrate these systems, use the following REST API to create an event of type "APPLICATION_DEPLOYMENT" in your managed environment.

You should receive the event ID after the successful invocation of the request.

Roles and Permissions

Creating events requires the **Create Events** permission. See Application Permissions for more information.

URI

POST /controller/rest/applications/*application_id*/events

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	Provide either application name or application id.	Yes
summary	Query	Provide the summary for the event.	Yes
comment	Query	Provide the comments (if any) for the event.	No
eventtype	Query	APPLICATION_DEPLOYMENT	Yes
severity	Query	Provide a severity level. Allowed values • "INFO" • "WARN" • "ERROR" In the UI, these become "Info", "Warning", and "Critical"	Yes

Create a Custom Event

You can create custom events to be reported in the AppDynamics event viewer and in the events panels on the AppDynamics dashboards. See Monitor Events to learn how to filter on your custom events so that you can find them. Then you can create alerts triggered by these events as you do for AppDynamics standard events.

You should receive the event ID after the successful invocation of the request.

Roles and Permissions

Creating a custom event requires the **Create Events** permission. See Application Permissions for more information.

URI

POST /controller/rest/applications/*application_id*/events

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	Provide either application name or application id.	Yes
summary	Query	Provide a summary describing the event.	Yes
comment	Query	Provide a comment for the event.	No
severity	Query	Provide a severity level. Allowed values • "INFO" • "WARN" • "ERROR" In the UI, these become "Info", "Warning", and "Critical"	Yes
eventtype	Query	CUSTOM	Yes
customeventtype	Query	Provide a name for the "type"; for example, the source, like "nagios"	No
node	Query	Provide the affected node name	No
tier	Query	Provide the affected tier name	Yes, if node and bt are specified
bt	Query	Provide the affected business transaction name	No
propertynames	Query	Provide a property name as a pair, i.e., the "key"	No, but if one element of the pair is defined, the other must be defined also
propertyvalues	Query	Provide the property value as a pair, i.e., the "value"	No, but if one element of the pair is defined, the other must be defined also

Example

```
curl -X POST --user user1@customer1:secret
'http://demo.appdynamics.com/controller/rest/applications/5/events?s
everity=INFO&summary=test1&eventtype=CUSTOM&customeventtype=mycustom
event&propertynames=key1&propertynames=key2&propertyvalues=value1&pr
opertyvalues=value'
```

Note the pattern for custom properties: propertynames and propertyvalues get matched up by order position, so to set N property values, you need N occurrences of propertynames and N occurrences of propertyvalues.

Create Custom URLS for Notifications

Single tenants in a multi-tenant Controller instance should use this API method to specify a custom "vanity" URL for notification purposes. So instead of a URL such as paid8.appdynamics.com being displayed as the host, the custom URL can be displayed as something like "mycompany.appdynamics.com" in the notification.

URI

POST /controller/rest/accounts/*customer_name*/update-controller-url

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
customer_name	URI	The customer account name	Yes

Body Parameter

As Application/JSON content:

```
{
   "controllerURL": "http://<my-custom-hostname:port>"
}
```

If the URL in the alerts is invalid, you can set it using the following curl command:

```
curl -k --basic --user root@system --header "Content-Type:
application/json" --data '{ "controllerURL":
"http://<controller>:<port>" }'
http://<controller>:<port>/controller/rest/accounts/<ACCOUNT-N
AME>/update-controller-url
```

For example:

```
curl -k --basic --user root@system --header "Content-Type:
application/json" --data '{ "controllerURL":
"https://myVIP:443" }'
https://myhost:8181/controller/rest/accounts/customer1/update-
controller-url
```

There is no need to reset the controller. Upgrading the controller will reset the deeplink URL settings.

Create and Delete Action Suppressions

By default any response is in JSON, although XML can be requested by using the following header:
Name : Accept
Value : application/vnd.appd.cntrl+xml;v=1

Retrieve All Existing Action Suppressions

Gets a list of all existing action suppressions.

URI

GET /controller/api/accounts/*account_id*/applications/*application_id*/actionsuppressions

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
account_id	URI	The account ID	Yes
application_id	URI	The application ID	Yes

Example request to get all action suppressions:

```
/controller/api/accounts/2/applications/9/actionsuppressions
```

Example response:

```
Status : 200 ok
Output Data :
{"actionSuppressions": [{"id": "15","name": "App-ASW","timeRange":
{"startTimeMillis": "2014-10-25T04:16:30+0000","endTimeMillis":
"2014-10-25T06:16:30+0000"},"affects": {"type": "APP"}},{"id":
"16","name": "Node-ASW","timeRange": {"startTimeMillis":
"2014-10-25T04:16:57+0000","endTimeMillis":
"2014-10-25T05:16:57+0000"},"healthRuleIds": [60,61],"affects":
{"type": "NODE","nodeAffectedEntities": {"type":
"SPECIFIC","nodeType": "ALL","nodes": [17,18]}}}],"actions":
[{"href":
"http://demo.appdynamics.com:8090/controller/api/accounts/2/applicat
ions/9/actionsuppressions/%7Bactionsuppressions.id%7D/%7Bactions.nam
e%7D","method": ["POST","DELETE"],"name": "enabled"}],"links":
[{"href":
"http://ec2-54-80-163-175.compute-1.amazonaws.com:8090/controller/ap
i/accounts/2/applications/9/actionsuppressions/%7Bactionsuppressions
.id%7D","name": "actionsuppressions"}]}
```

Retrieve a Specific Action Suppression by ID

Get the action suppression by specified ID.

URI

/controller/api/accounts/*account_id*/applications/*application_id*/actionsuppressions/*actionsuppression_id*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
account_id	URI	The account id	Yes
application_id	URI	The application id	Yes
actionsuppressions_id	URI	The action suppression id	Yes

Example request:

```
/controller/api/accounts/2/applications/9/actionsuppressions/15
```

Example response:

```
Status : 200 ok
Output Data :
{"id": "15","name": "App-ASW","timeRange": {"startTimeMillis":
"2014-10-25T04:16:30+0000","endTimeMillis":
"2014-10-25T06:16:30+0000"},"affects": {"type": "APP"}}
```

Create a New Action Suppression

This is a POST request. Should return a 201 - created response.

URI

POST /controller/api/accounts/*account_id*/applications/*application_id*/actionsuppressions

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
account_id	URI	The account id	Yes
application_id	URI	The application id	Yes
name	body key	The name of the action suppression window	Yes
timeRange	body key	The start and end time of the window Includes: startTimeMillis, eg 2014-10-25T04:16:57+0000 endTimeMillis, eg 2014-10-25T05:16:57+0000	Yes
healthRuleIds	body key	The ids of the affected health rules. If not provided, all rules affected	No
affects	body key	Type of entity and corresponding ids (see below)	Yes

Content of "affects"

Scope	Type	Value	Example
Application	APP	Covers the entire application	"affects": {"type": "APP"}

Business Transaction	BT	Covers one or more Business Transactions	
		All Business Transactions	"affects": {"type": "BT","btAffectedEntities": {"type": "ALL"}}
		Business Transactions from specific tiers	"affects": {"type": "BT","btAffectedEntities": {"type": "WITHIN_TIERS","tiers": [11,12]}} where 11,12 are Tier Ids
		Specific Business Transactions by id	"affects": {"type": "BT","btAffectedEntities": {"type": "SPECIFIC","bts": [1,2]}} where 1,2 are BT Ids
		Business Transactions that match criteria	"affects": {"type": "BT","btAffectedEntities": {"type": "CRITERIA","matchesOperator": "CONTAINS","matchesValue": "pojo"}} where "matchesOperator" can be: • CONTAINS • EQUALS • STARTS • ENDS • REGEX_VALUE
Tier	TIER	Covers one or more tiers	
		All tiers	"affects": {"type": "TIER","tierAffectedEntities": {"type": "ALL"}}
		Specific tiers	"affects": {"type": "TIER","tierAffectedEntities": {"type": "SPECIFIC","tiers": [11,12]}} where 11,12 are Tier Ids
Node	NODE	Covers one or more nodes	
		All nodes	"affects": {"type": "NODE","nodeAffectedEntities": {"type": "ALL","nodeType": "ALL"}}
		Nodes belonging to specific tiers	"affects": {"type": "NODE","nodeAffectedEntities": {"type": "WITHIN_TIERS","nodeType": "ALL","tiers": [11,12]}} where 11,12 are Tier Ids
		Specific nodes	"affects": {"type": "NODE","nodeAffectedEntities": {"type": "SPECIFIC","nodeType": "ALL","nodes": [9,10]}} where 9,10 are Node Ids
		Nodes that match criteria	"affects": {"type": "NODE","nodeAffectedEntities": {"type": "NAME_CRITERIA","nodeType": "ALL","nameMatchesOperator": "EQUALS","nameMatchesValue": "Node"}} - **String match** "affects": {"type": "NODE","nodeAffectedEntities": {"type": "PROPERTY_CRITERIA","nodeType": "ALL","metaInfoProperties": [{"name": "ProcessID","value": "12343"}]}} - **Meta Info Properties match** where "matchesOperator" can be: • CONTAINS • EQUALS • STARTS • ENDS • REGEX_VALUE
Machine	MACHINE	Covers one or more machines	"affects": {"type": "MACHINE","machineAffectedEntities": {"type": "SPECITIC","machines": [4,5]}} where 4,5 are Machine Ids note you must spell the value as "SPECITIC" due to a typo in the underlying code.

Example request

```
/controller/api/accounts/2/applications/9/actionsuppressions
```

Header

- Name: Content-Type

- Value: application/vnd.appd.cntrl+json;v=1

Body

```
{"name": "App-ASW_2","timeRange": {"startTimeMillis":
"2014-10-25T04:16:30+0000","endTimeMillis":
"2014-10-25T06:16:30+0000"},"affects": {"type": "APP"}}
```

or

```
{"name": "Node-ASW_1","timeRange": {"startTimeMillis":
"2014-10-25T04:16:57+0000","endTimeMillis":
"2014-10-25T05:16:57+0000"},"healthRuleIds": [60,61],"affects":
{"type": "NODE","nodeAffectedEntities": {"type":
"SPECIFIC","nodeType": "ALL","nodes": [17,18]}}}
```

Delete a Specific Action Suppression by ID

This is a DELETE request. Should return a 204 - No Content.

URI

DELETE /controller/api/accounts/*account_id*/applications/*application_id*/actionsuppressions/*actionsuppression_id*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
account_id	URI	The account id	Yes
application_id	URI	The application id	Yes
actionsuppression_id	URI	The id of the action suppression to be deleted	Yes

Configuration API

On this page:

The configuration API enables you read and modify selected Controller configuration settings programmatically. You can use the API to script or automate tasks that need to be performed frequently or in large batches, such as adding users.

> The Configuration Export and Import API provides the ability to perform select configuration changes using an API as well since you can edit and import Controller configuration definition files.

Create and Modify AppDynamics Users

Creates or modifies user accounts in the Controller.

You pass the user configuration settings as query parameters to the API call. The format of the create and modify user calls are identical except for the user-id parameter, which is not passed for the create operation. The user-id is generated by the create operation.

Format

POST /controller/rest/users

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
user-name	Query	user name	Yes
user-id	Query	user id	No for a create; yes for an update
user-display-name	Query	display name	Yes
user-roles	Query	comma-separated list of roles	No
user-password	Query	user password	Yes for create; optional for update
user-email	Query	user email	Yes

Example

```
curl -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/users?user-name=user2\&u
ser-display-name=User%20Two\&user-password=welcome\&user-email=user2
\@example.com
```

Include or Exclude a Business Transaction from Monitoring

You can exclude or include business transaction for monitoring bypassing the exclude parameter to the business-transactions retrieval API described in Application Model API.

To exclude a business transaction, pass the XML-represented ID of the business transaction to be excluded with the exclude parameter set to true. To turn on monitoring for a currently excluded business transaction, set the exclude parameter to false.

Send the list of business transactions to be excluded or re-included as the XML-formatted POST payload. A sample business-transaction list is:

```
<business-transactions>
    <business-transaction>
        <id>15</id>
    </business-transaction>
    <business-transaction>
        <id>16</id>
    </business-transaction>
</business-transactions>
```

Make sure the Content-Type header is set to "application/xml".

Format

POST /controller/rest/applications/*application_id*/business-transactions

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	Provide either the application name or application id.	Yes
exclude	Post	true|false	Yes

Example

```
curl -X POST -H "Content-Type:text/xml" --user user1@customer1:secret
http://demo.appdynamics.com/controller/rest/applications/6/business-
transactions\?exclude=true -d @businesstransaction.xml
```

Retrieve All Controller Settings

The Controller global configuration values are made up of the Controller settings that are presented in the Administration Console.

Format

GET /controller/rest/configuration

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory

output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user admin@customer1:secret
http://demo.appdynamics.com/controller/rest/configuration

<configuration-items><configuration-item>
  <name>eventsvc.request.segment.data.max.retrieval.size</name>
  <value>5000</value>
  <description>Max number of request segment data that can be
retrieved from event service in a query</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
<configuration-item>
  <name>machine.agent.max.new.actions.per.min</name>
  <value>15</value>
  <description>Maximum number of new actions dispatched per minute
for each machine agent</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
...
<configuration-item>
  <name>tss.retention.period</name>
  <value>336</value>
  <description>Time (in hours) to retain 12 hour tss data values
before they are purged from the system.</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
<configuration-item>
  <name>snapshots.retention.period</name>
  <value>336</value>
  <description>Time (in hours) to retain snapshots before they are
purged from the system.</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
<configuration-item>
  <name>metrics.min.retention.period</name>
  <value>4</value>
  <description>Time (in hours) to retain minute metric data values
before they are purged from the system.</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
<configuration-item>
```

```
  <name>system.notification.event.types</name>

<value>LICENSE,DISK_SPACE,CONTROLLER_AGENT_VERSION_INCOMPATIBILITY,C
ONTROLLER_EVENT_UPLOAD_LIMIT_REACHED,CONTROLLER_RSD_UPLOAD_LIMIT_REA
CHED,CONTROLLER_METRIC_REG_LIMIT_REACHED,CONTROLLER_METRIC_DATA_BUFF
ER_OVERFLOW,CONTROLLER_ERROR_ADD_REG_LIMIT_REACHED,CONTROLLER_ASYNC_
ADD_REG_LIMIT_REACHED,AGENT_ADD_BLACKLIST_REG_LIMIT_REACHED,AGENT_ME
TRIC_BLACKLIST_REG_LIMIT_REACHED,CONTROLLER_STACKTRACE_ADD_REG_LIMIT
_REACHED,CONTROLLER_SEP_ADD_REG_LIMIT_REACHED,CONTROLLER_MEMORY_ADD_
REG_LIMIT_REACHED,CONTROLLER_TRACKED_OBJECT_ADD_REG_LIMIT_REACHED,CO
NTROLLER_COLLECTIONS_ADD_REG_LIMIT_REACHED</value>
  <description>Comma separated list of Event Types (with no spaces
between each) that will shown as System Notifications in the
UI.</description>
  <updateable>true</updateable>
```

```
  <scope>cluster</scope>
</configuration-item>
</configuration-items>
```

Retrieve a Controller Setting by Name

Gets the value of a given Controller configuration setting.

Format

GET /configuration?name=*controller_setting_name*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
name	Query	Name of the Controller setting to retrieve	Yes
output	Query	HTTP Request parameter included as part of the URL to change the output format. Valid values are "XML" (default) or "JSON".	No

Example

```
curl --user admin@customer1:secret
http://demo.appdynamics.com/controller/rest/configuration?name=metri
cs\.min\.retention\.period

<configuration-items><configuration-item>
  <name>metrics.min.retention.period</name>
  <value>4</value>
  <description>Time (in hours) to retain minute metric data values
before they are purged from the system.</description>
  <updateable>true</updateable>
  <scope>cluster</scope>
</configuration-item>
</configuration-items>
```

Configure Global Controller Settings

Set a Controller setting to a specified value.

> You cannot use this REST API to modify Controller settings on SaaS.

Format

POST /controller/rest/configuration

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
name	Query	Name of the Controller setting to get.	Yes
value	Query	Value to set.	Yes

Mark Nodes as Historical

Mark nodes as historical, which directs AppDynamics to stop collecting metrics for the nodes. By default AppDynamics marks as historical (soft deletes) a node that has lost contact with the Controller for the number of hours configured in the node.retention.period controller setting. The default is 500 hours.

Pass one or more identifiers of the node to be marked as historical, up to a maximum of 25 nodes. Multiple IDs should be comma-separated.

Format

POST /controller/rest/mark-nodes-historical?application-component-node-ids=*value*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application-component-node-ids	Query	Comma-separated list of node IDs	Yes

Example

```
curl -X POST --user admin@customer1:secret
http://demo.appdynamics.com/controller/rest/mark-nodes-historical?ap
plication-component-node-ids=44,45

<application-component-node-id>
<44/>
<45/>
</application-component-node-id>
```

Configuration Import and Export API

On this page:

This page describes the AppDynamics API methods you can use to import and export various types of configuration settings in the Controller.

About the Configuration Import/Export APIs

The Configuration Import/Export APIs enables you to migrate configuration settings across Controller accounts, business applications, or Controller instances. You can also use it to add configuration artifacts—such as transaction detection rules, health rules or custom dashboards—to an existing configuration programmatically.

An exported configuration is an XML or JSON representation of the configuration artifact. After exporting the file, you can upload it to another account or application, optionally modifying the configuration.

Export Actions from an Application

Exports all actions in the specified application to a JSON file.

To be able to export actions, the user account you use to make the API call must have permissions to view an action or action template in the application you are exporting from.

Format

GET /controller/actions/*application_id*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/actions/7

[
   {
      actionType: "EmailAction",
      name: "6DA8942B-DF4A-417A-E1NF-59F14231D670",
      priority: 1,
      description: null,
      toAddress: "user1@example.com",
      subject: "",
      timeZone: null
   },
   {
      actionType: "DiagnosticSessionAction",
      name: "MyDiagnotic",
      priority: 0,
      description: null,
      businessTransactionTemplates: [ ],
      numberOfSnapshotsPerMinute: 5,
      durationInMinutes: 10,
      adjudicate: false,
      adjudicatorEmail: null
   }
]
```

Import Actions into an Application

After you have exported actions, you can import them to a different application passing the JSON file created by the export operation as the payload to a POST request.

The user account you use to make the API call must have permissions to create an action or action template in the account.

Actions in the import file that have conflicting names with actions in the existing configuration are not imported. The import for those actions fails, while new actions are imported successfully.

This call takes data as multipart/form-data content. Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

Format

POST /controller/actions/*application_id*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes

Example

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/actions/38 -F
file=@ExportActions.json

{"success":true,"errors":[],"warnings":[]}
```

If there are actions in the file with the same name as ones in the configuration, those actions are not imported, and the response indicates the success of the request as false, as follows:

```
{"success":false,"errors":["Not importing Action with name:
DuplicateExportedDiagnosticAction, since it already
exists."],"warnings":["Imported 1 out of 2 actions"]}
```

Export Email Action Templates from an Account

This API exports all the email action templates in the current account in JSON format.

Format

GET /controller/actiontemplate/email

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/actiontemplate/email

[ {
  "actionPlanType" : "email",
  "name" : "MyCustomEmailTemplate",
  "oneEmailPerEvent" : true,
  "eventClampLimit" : 100,
  "defaultCustomProperties" : [ {
    "id" : 0,
    "version" : 0,
    "name" : "env",
    "value" : "%OS"
  } ],
  "allowCustomRecipients" : true,
  "toRecipients" : [ ],
  "ccRecipients" : [ ],
  "bccRecipients" : [ ],
  "headers" : [ ],
  "subject" : "We've got a situation...",
  "includeTextBody" : true,
  "textBody" : "<h1>Summary of events occurring during the
${policy.digestDurationInMins}+ minute(s) prior to
${action.triggerTime}:</h1> <table> #foreach(${eventList} in
${fullEventsByTypeMap.values()}) #foreach(${event} in ${eventList})
<tr> <td> <!-- Event icon --> <img
src="${event.severityImage.mimeContentRef}" alt="${event.severity}"
/> </td> <td> <!-- Event name with event link --> <a
href="${event.deepLink}">${event.displayName}</a> </td> <td> <!--
Event message --> ${event.eventMessage} </td> </tr> #end #end
</table>"",
  "includeHtmlBody" : true,
  "htmlBody" : "<p>Please look into it.</p>",
  "testLogLevel" : "DEBUG",
  "testPropertiesPairs" : [ ],
  "testToRecipients" : [ ],
  "testCcRecipients" : [ ],
  "testBccRecipients" : [ ],
  "eventTypeCountPairs" : [ ]
} ]
```

Import Email Action Templates

Enables you to import email action templates to an account as JSON file. The import fails if you attempt to import a template with the same name as an existing template of the same type in the destination account.

Data for this call should be in the form of multipart/form-data. Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

Format

POST /controller/actiontemplate/email

Example

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/actiontemplate/email -F
file=@emailactiontemplate.json

{"success":true,"errors":[],"warnings":[]}
```

Export HTTP Request Action Templates from an Account

This API exports all the HTTP request action templates in the current account to a JSON file.

Format

GET /controller/actiontemplate/httprequest/

Example:

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/actiontemplate/httprequest

[ {
  "actionPlanType" : "httprequest",
  "name" : "MyCustomHTTPTemplate",
  "oneRequestPerEvent" : false,
  "eventClampLimit" : -1,
  "defaultCustomProperties" : [ ],
  "method" : "GET",
  "scheme" : "HTTP",
  "host" : "http",
  "port" : 0,
  "path" :
"//demo.appdynamics.com//controller/rest/applications/${latestEvent.
application.name}/nodes/${latestEvent.node.name}",
  "query" : "",
  "urlCharset" : "UTF_8",
  "authType" : "BASIC",
  "authUsername" : "user1",
  "authPassword" : "secret",
  "headers" : [ ],
  "payloadTemplate" : {
    "httpRequestActionMediaType" : "text/plain",
    "charset" : "UTF_8",
    "formDataPairs" : [ ],
    "payload" : ""
  },
  "connectTimeoutInMillis" : 5000,
  "socketTimeoutInMillis" : 15000,
  "maxFollowRedirects" : 0,
  "responseMatchCriteriaAnyTemplate" : [ ],
  "responseMatchCriteriaNoneTemplate" : [ ],
  "testLogLevel" : "DEBUG",
  "testPropertiesPairs" : [ ],
  "eventTypeCountPairs" : [ ]
} ]
```

Import HTTP Action Templates into an Account

After you have exported HTTP request action templates, you can import them to a different account by logging into the destination account and passing the JSON file created by the export operation as the payload to the POST request.

You can modify the exported file before you import it. You might want to do this to remove one or more template configurations or to change their names. The import will fail if you attempt to import a template with the same name as an existing template of the same type in the destination account.

Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

Format

GET /controller/actiontemplate/httprequest

Example

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/actiontemplate/httprequest -F
file=@httpactiontemplate.json

{"success":true,"errors":[],"warnings":[]}
```

Export Custom Dashboards and Templates

You can export and import custom dashboards and custom dashboard templates interactively from the Controller UI or using this API call.

See Import and Export Custom Dashboards and Templates Using the UI for information on importing and exporting custom dashboards and dashboard templates from the AppDynamics UI.

To export the dashboard, the user making the API call must have permissions to view the custom dashboard.

In the export call, you need to identify the dashboard to export by ID. You can discover the ID of a dashboard in the Controller UI. When you open the dashboard in the UI, the ID appears as the dashboard parameter at the end of the URL.

For example, in this URL snippet, the custom dashboard ID is 3: location=CDASHBOARD_DETAIL&mode=MODE_DASHBOARD&dashboard=3.

Format

GET /controller/CustomDashboardImportExportServlet?dashboardId=*dashboard_id*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
dashboardId	Query	The numeric ID of the custom dashboard.	Yes

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/CustomDashboardImportExportSe
rvlet?dashboardId=8

{
  "schemaVersion" : null,
  "dashboardFormatVersion" : "3.0",
  "name" : "Analytics-BrowserData",
   ...
  "warRoom" : false,
  "template" : false
}
```

View a complete response example:

Click to view dashboard export sample...


```
{
  "schemaVersion" : null,
  "dashboardFormatVersion" : "3.0",
  "name" : "Analytics-BrowserData",
  "description" : null,
  "properties" : null,
  "templateEntityType" : "APPLICATION_COMPONENT_NODE",
  "associatedEntityTemplates" : null,
  "minutesBeforeAnchorTime" : 15,
  "startDate" : null,
  "endDate" : null,
  "refreshInterval" : 120000,
  "backgroundColor" : 15395562,
  "color" : 15395562,
  "height" : 768,
  "width" : 1024,
  "canvasType" : "CANVAS_TYPE_GRID",
  "layoutType" : "",
  "widgetTemplates" : [ {
    "widgetType" : "AnalyticsWidget",
    "title" : "Browser_data",
    "height" : 4,
    "width" : 4,
    "x" : 0,
    "y" : 0,
    "label" : "",
    "description" : "",
    "drillDownUrl" : "",
    "useMetricBrowserAsDrillDown" : false,
    "backgroundColor" : 16777215,
```

```
    "backgroundColors" : null,
    "backgroundColorsStr" : null,
    "color" : 4210752,
    "fontSize" : 12,
    "useAutomaticFontSize" : false,
    "borderEnabled" : false,
    "borderThickness" : 0,
    "borderColor" : 0,
    "backgroundAlpha" : 1.0,
    "showValues" : false,
    "compactMode" : false,
    "showTimeRange" : false,
    "renderIn3D" : false,
    "showLegend" : false,
    "legendPosition" : null,
    "legendColumnCount" : null,
    "startTime" : null,
    "endTime" : null,
    "minutesBeforeAnchorTime" : 0,
    "isGlobal" : true,
    "propertiesMap" : null,
    "dataSeriesTemplates" : null,
    "adqlQueries" : [ "SELECT appkey, pageexperience,
distinctcount(pageurl) AS \"URL (Count Distinct)\" FROM
browser_records LIMIT 100,100" ],
    "analyticsWidgetType" : "COLUMN",
    "maxAllowedYAxisFields" : 3,
    "maxAllowedXAxisFields" : 2,
    "min" : null,
    "interval" : 98,
    "max" : null,
    "intervalType" : "By Fixed Number",
    "showMinExtremes" : null,
    "showMaxExtremes" : null,
    "displayPercentileMarkers" : null,
    "percentileValue1" : null,
    "percentileValue2" : null,
    "percentileValue3" : null,
    "percentileValue4" : null,
    "resolution" : "1m",
    "dataFetchSize" : null,
    "percentileLine" : null,
    "timeRangeInterval" : null,
    "pollingInterval" : null,
    "unit" : null
  } ],
```

```
  "warRoom" : false,
  "template" : false
}o
```

Import Custom Dashboards and Templates

You can import custom dashboard and templates based on a previously exported JSON definition, which has optionally been modified. Import the definition as an application/json content type.

Data for this call should be in the form of multipart/form-data. Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

To import a dashboard, the user making the API call must have create dashboard permissions in the Controller.

> Prior to version 4.1, exported custom dashboards were in XML format. You can import custom dashboards previously exported as XML data into the current Controller; however, custom dashboards can only be exported as JSON data.

Format

POST /controller/CustomDashboardImportExportServlet

Example

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/CustomDashboardImportExportSe
rvlet -F file=@customdashboards.json

{"success":true,"errors":[],"warnings":[],"createdDashboardName":"Up
loaded-Analytics-BrowserData"}
```

Export Health Rules from an Application

Returns all health rules in XML format.

The user account you use to make the API call must have permissions to view the health rule.

Format

GET /controller/healthrules/*application_id*?name=*health_rule_name*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes
name	Query	The name of the health rule to export. If not specified, exports all health rules.	No

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/healthrules/38?name=MyCustomH
ealthRule

<health-rules controller-version="004-002-000-000">
    <health-rule>
        <name>MyCustomHealthRule</name>
        <type>BUSINESS_TRANSACTION</type>
        <description/>
        <enabled>true</enabled>
        <is-default>false</is-default>
        <always-enabled>true</always-enabled>
        <duration-min>30</duration-min>
        <wait-time-min>30</wait-time-min>
        <affected-entities-match-criteria>
            <affected-bt-match-criteria>
                <type>ALL</type>
            </affected-bt-match-criteria>
        </affected-entities-match-criteria>
        <warning-execution-criteria>
            <entity-aggregation-scope>
                <type>ANY</type>
                <value>0</value>
            </entity-aggregation-scope>
            <policy-condition>
                <type>leaf</type>
                <display-name>CPU</display-name>

<condition-value-type>BASELINE_STANDARD_DEVIATION</condition-value-t
ype>
                <condition-value>2.0</condition-value>
                <operator>GREATER_THAN</operator>
                <condition-expression/>
                <use-active-baseline>true</use-active-baseline>
                <metric-expression>
                    <type>leaf</type>
                    <function-type>VALUE</function-type>
                    <value>0</value>

<is-literal-expression>false</is-literal-expression>
                    <display-name>null</display-name>
                    <metric-definition>
                        <type>LOGICAL_METRIC</type>
                        <logical-metric-name>Average CPU Used
(ms)</logical-metric-name>
                    </metric-definition>
                </metric-expression>
            </policy-condition>
```

```
        </warning-execution-criteria>
    </health-rule>
</health-rules>
```

Import Health Rules into an Application

You can import health rules defined in an XML file into a business application.

Data for this call should be in the form of multipart/form-data. In the POST request, use UTF-8 URL encoding for the URI; for example, do not replace a space (" ") with "%20" in the URI.

By default, a health rule in the posted data with an identical name to one in the existing configuration does not overwrite the existing health rule. If you want to overwrite an existing health rule of the same name, use the overwrite parameter.

The syntax is the same for importing one health rule configuration or several. All the health rule configurations in the posted XML file are imported.

Format

POST /controller/healthrules/*application_id*?overwrite=*true_or_false*

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes
overwrite	Query	Set to true to have health rules in the posted data overwrite existing health rules with the same name. The default is false.	No

Example

```
curl -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/healthrules/38 -F
file=@uploadhealthrule.xml

Imported 1 health rules successfully.
```

If the health rule exists and you have not enabled the overwrite parameters, you will get the following response:

```
Not importing the health rule: healthrulename since it already
exists.
```

Export Transaction Detection Rules

Gets all transaction detection rules in XML format. This call returns different types of detection rule configurations when MDS is

enabled.

You can get transaction detection rules from a number of different configurations, including the configuration for a specific scope by name.

The URI used by clients for this call should be UTF-8 encoded

Format

GET /controller/transactiondetection/*application_id*/[*scope_name*]/*rule_type*/[*entry_point_type*]/[*rule_name*] >> xml_name.xml

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID. It can be found in the URL in the address bar if you click the specified application. It is an Integer value. It will return an error if you do not specify.	Yes
scope_name	URI	The name of the scope from which you are exporting the entry point configuration. The scope name cannot be "custom" and "auto". If the scope name contains a space, type "%20" instead of space. Example: curl -X GET --user user1@customer1:welcome http://localhost:8080/controller/transactiondetection/10/default%20scope/custom >> result.xml It will export all the rules under all the scopes if you do not specify.	No
rule_type	URI	The type of rule to export, from these options: • auto: Automatic detection rules • custom: Custom detection rules in the configuration • exclude: Custom exclude rules for transaction detection It will return an error if you do not specify.	Yes
entry_point_type	URI	The POJO, Servlet, EJB, Spring Bean, etc. It will export the rules which belong to all the entry point types if you do not specify.	No
rule_name	URI	The name of the rule which you are exporting. It will export all the rules under a scope or all the scopes if you do not specify.	No

- The order of the parameters cannot be change. The order of the parameters should be: *application_id*/[*scope_name*]/*rule_type*/[*entry_point_type*]/[*rule_name*].
- It will return an xml file with error content if you use the wrong name(s) in all the parameters.
- Tier information is provided in the scope list.
- If you do not specify the scope when exporting, then you should also not specify it when importing.
- Please check server.log if you do not get the expected result after exporting.

Scenarios:

- Export the rules from all the scopes.

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/applicati
onID/{rule_type} >> {xml_name}.xml
```

Example:

```
curl -X GET --user
user1@customer1:welcome http://localhost:8080/controller/transa
ctiondetection/10/custom >> result.xml
```

If you do not include scope_name then the output is divided into three parts under <mds-config-data>: scope-list, rule-list, and scope-rule-mapping-list.

Click to view complete response...

```
<mds-data>
    <mds-config-data>
        <scope-list>
            <scope scope-description="" scope-name="scope0"
                scope-type="ALL_TIERS_IN_APP"
scope-version="0"/>
            <scope scope-description="" scope-name="scope1"
                scope-type="SELECTED_TIERS"
scope-version="0"/>
        </scope-list>
        <rule-list>
            <rule agent-type="APPLICATION_SERVER"
enabled="true"
                priority="1"
                rule-description="ruleInScope1_SERVLET"
                rule-name="ruleInScope1_SERVLET"
rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txautodiscoveryrule":{"autod
iscoveryconfigs":[]},"txcustomrule":{"type":"INCLUDE","txentr
ypointtype":"SERVLET","matchconditions":[{"type":"HTTP","http
match":{"uri":{"type":"IS_NOT_EMPTY","matchstrings":[""]},"pa
rameters":[],"headers":[],"cookies":[]}}],"actions":[],"prope
rties":[]},"agenttype":"APPLICATION_SERVER"}</tx-match-rule>
            </rule>
            <rule agent-type="DOT_NET_APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="ASP.NET MVC5 Resource
Handler"
                rule-name="ASP.NET MVC5 Resource Handler"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"POJO","matchconditions":[{"type":"HTT
P","httpmatch":{"parameters":[],"headers":[],"classmatch":{"t
ype":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS","m
atchstrings":["System.Web.Optimization.BundleHandler"],"isnot
```

```
":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT","ht
tpsplit":{}}],"properties":[]},"agenttype":"DOT_NET_APPLICATI
ON_SERVER"}</tx-match-rule>
            </rule>
            <rule agent-type="APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="testPOJO"
                rule-name="testPOJO"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"POJO","matchconditions":[{"type":"HTT
P","httpmatch":{"parameters":[],"headers":[],"classmatch":{"t
ype":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS","m
atchstrings":["System.Web.Optimization.BundleHandler"],"isnot
":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT","ht
tpsplit":{}}],"properties":[]},"agenttype":"APPLICATION_SERVE
R"}</tx-match-rule>
            </rule>
            <rule agent-type="APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="whatever_SERVLET"
                rule-name="whatever_SERVLET"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"SERVLET","matchconditions":[{"type":"
HTTP","httpmatch":{"parameters":[],"headers":[],"classmatch":
{"type":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS"
,"matchstrings":["System.Web.Optimization.BundleHandler"],"is
not":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT",
"httpsplit":{}}],"properties":[]},"agenttype":"APPLICATION_SE
RVER"}</tx-match-rule>
            </rule>
        </rule-list>
        <scope-rule-mapping-list>
            <scope-rule-mapping scope-name="scope1">
                <rule rule-description="ruleInScope1_SERVLET"
rule-name="ruleInScope1_SERVLET"/>
            </scope-rule-mapping>
            <scope-rule-mapping scope-name="scope0">
                <rule rule-description="ASP.NET MVC5 Resource
Handler" rule-name="ASP.NET MVC5 Resource Handler"/>
                <rule rule-description="whatever_SERVLET"
rule-name="whatever_SERVLET"/>
                <rule rule-description="testPOJO"
rule-name="testPOJO"/>
            </scope-rule-mapping>
```

```
        </scope-rule-mapping-list>
    </mds-config-data>
</mds-data>
```

- Export the rules under the specified scope:

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/applicati
onID/scope_name/{rule_type} >> {xml_name}.xml
```

Example:

```
curl -X GET --user
user1@customer1:welcome http://localhost:8080/controller/transa
ctiondetection/10/scope0/custom >> result.xml
```

If you include scope_name then only rule-list is included in the output.

Click to view complete response...

```
<mds-data>
    <mds-config-data>
        <rule-list>
            <rule agent-type="DOT_NET_APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="ASP.NET MVC5 Resource
Handler"
                rule-name="ASP.NET MVC5 Resource Handler"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"ASP_DOTNET","matchconditions":[{"type
":"HTTP","httpmatch":{"parameters":[],"headers":[],"classmatc
h":{"type":"MATCHES_CLASS","classnamecondition":{"type":"EQUA
LS","matchstrings":["System.Web.Optimization.BundleHandler"],
"isnot":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLI
T","httpsplit":{}}],"properties":[]},"agenttype":"DOT_NET_APP
LICATION_SERVER"}</tx-match-rule>
            </rule>
            <rule agent-type="APPLICATION_SERVER"
```

```
                enabled="true" priority="0"
                rule-description="whatever_SERVLET"
                rule-name="whatever_SERVLET"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"SERVLET","matchconditions":[{"type":"
HTTP","httpmatch":{"parameters":[],"headers":[],"classmatch":
{"type":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS"
,"matchstrings":["System.Web.Optimization.BundleHandler"],"is
not":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT",
"httpsplit":{}}],"properties":[]},"agenttype":"APPLICATION_SE
RVER"}</tx-match-rule>
            </rule>
            <rule agent-type="APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="testPOJO"
                rule-name="testPOJO"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"POJO","matchconditions":[{"type":"HTT
P","httpmatch":{"parameters":[],"headers":[],"classmatch":{"t
ype":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS","m
atchstrings":["System.Web.Optimization.BundleHandler"],"isnot
":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT","ht
tpsplit":{}}],"properties":[]},"agenttype":"APPLICATION_SERVE
R"}</tx-match-rule>
            </rule>
```

```
        </rule-list>
    </mds-config-data>
</mds-data>
```

- Export the rules belonging to the specified entry point under all the scopes.

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/applicati
onID/{rule_type}/{entry_point_type} >> {xml_name}.xml
```

Example:

```
curl -X GET --user
user1@customer1:welcome http://localhost:8080/controller/transa
ctiondetection/10/custom/servlet >> {xml_name}.xml
```

If you do not include scope_name then the output is divided into three parts under <mds-config-data>: scope-list, rule-list, and scope-rule-mapping-list.

Click to view complete response...

```
<mds-data>
    <mds-config-data>
        <scope-list>
            <scope scope-description="" scope-name="scope0"
                scope-type="ALL_TIERS_IN_APP"
scope-version="0"/>
            <scope scope-description="" scope-name="scope1"
                scope-type="SELECTED_TIERS"
scope-version="0"/>
        </scope-list>
        <rule-list>
            <rule agent-type="APPLICATION_SERVER"
enabled="true"
                priority="1"
                rule-description="ruleInScope1_SERVLET"
                rule-name="ruleInScope1_SERVLET"
rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txautodiscoveryrule":{"autod
iscoveryconfigs":[]},"txcustomrule":{"type":"INCLUDE","txentr
```

```
ypointtype":"SERVLET","matchconditions":[{"type":"HTTP","http
match":{"uri":{"type":"IS_NOT_EMPTY","matchstrings":[""]},"pa
rameters":[],"headers":[],"cookies":[]}}],"actions":[],"prope
rties":[]},"agenttype":"APPLICATION_SERVER"}</tx-match-rule>
            </rule>
            <rule agent-type="APPLICATION_SERVER"
                enabled="true" priority="0"
                rule-description="whatever_SERVLET"
                rule-name="whatever_SERVLET"
                rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txcustomrule":{"type":"EXCLU
DE","txentrypointtype":"SERVLET","matchconditions":[{"type":"
HTTP","httpmatch":{"parameters":[],"headers":[],"classmatch":
{"type":"MATCHES_CLASS","classnamecondition":{"type":"EQUALS"
,"matchstrings":["System.Web.Optimization.BundleHandler"],"is
not":false}},"cookies":[]}}],"actions":[{"type":"HTTP_SPLIT",
"httpsplit":{}}],"properties":[]},"agenttype":"APPLICATION_SE
RVER"}</tx-match-rule>
            </rule>
        </rule-list>
        <scope-rule-mapping-list>
            <scope-rule-mapping scope-name="scope1">
                <rule rule-description="ruleInScope1_SERVLET"
rule-name="ruleInScope1_SERVLET"/>
            </scope-rule-mapping>
            <scope-rule-mapping scope-name="scope0">
                <rule rule-description="whatever_SERVLET"
rule-name="whatever_SERVLET"/>
            </scope-rule-mapping>
```

```
        </scope-rule-mapping-list>
    </mds-config-data>
</mds-data>
```

- Export the rules belonging to the specified entry point under the specified scope.

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/applicati
onID/scope_name/{rule_type}/{entry_point_type} >> {xml_name}.xml
```

Example:

```
curl -X GET --user
user1@customer1:welcome http://localhost:8080/controller/transa
ctiondetection/10/scope0/custom/servlet >> {xml_name}.xml
```

Click to view complete response...

```
<mds-data>
    <mds-config-data>
        <rule-list>
            <rule agent-type="APPLICATION_SERVER"
enabled="true"
                priority="1"
                rule-description="ruleInScope1_SERVLET"
                rule-name="ruleInScope1_SERVLET"
rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txautodiscoveryrule":{"autod
iscoveryconfigs":[]},"txcustomrule":{"type":"INCLUDE","txentr
ypointtype":"SERVLET","matchconditions":[{"type":"HTTP","http
match":{"uri":{"type":"IS_NOT_EMPTY","matchstrings":[""]},"pa
rameters":[],"headers":[],"cookies":[]}}],"actions":[],"prope
rties":[]},"agenttype":"APPLICATION_SERVER"}</tx-match-rule>
            </rule>
        </rule-list>
    </mds-config-data>
</mds-data>
```

- Export the single rule belonging to the specified entry point under the specified scope.

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/applicati
onID/scope_name/{rule_type}/{entry_point_type}/{rule_name} >>
{xml_name}.xml
```

Example:

```
curl -X GET --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/10/scope0
/custom/servlet/rule_name >> {xml_name}.xml
```

Click to view complete response...

```
<mds-data>
    <mds-config-data>
        <rule-list>
            <rule agent-type="APPLICATION_SERVER"
enabled="true"
                priority="1" rule-description=""
                rule-name="ruleInScope1"
rule-type="TX_MATCH_RULE" version="0">

<tx-match-rule>{"type":"CUSTOM","txautodiscoveryrule":{"autod
iscoveryconfigs":[]},"txcustomrule":{"type":"INCLUDE","txentr
ypointtype":"SERVLET","matchconditions":[{"type":"HTTP","http
match":{"uri":{"type":"IS_NOT_EMPTY","matchstrings":[""]},"pa
rameters":[],"headers":[],"cookies":[]}}],"actions":[],"prope
rties":[]},"agenttype":"APPLICATION_SERVER"}</tx-match-rule>
            </rule>
        </rule-list>
    </mds-config-data>
</mds-data>
```

Import Transaction Detection Rules

Imports automatic detection rules in XML format. This call returns different types of detection rule configurations when MDS is enabled.

Importing action will overwrite a rule or add the new rule to the all/specified scopes.

Data for this call should be in the form of multipart/form-data. The URI for this call should be UTF-8 encoded.

Format

POST /controller/transactiondetection/*application_id*/[*scope_name*]/*rule_type*/[*entry_point_type*]/[*rule_name*] -F file=*@exported_file_name*.xml

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID. It can be found in the URL in address bar if you click the specified application. It is an Integer value. It will return an error if you do not specify.	Yes
scope_name	URI	The name of the scope from which you are importing the entry point configuration. It will import the rules to all the scopes if you do not specify.	No
rule_type	URI	The type of rule to import, from these options: • auto: Automatic detection rules • custom: Custom detection rules in the configuration • exclude: Custom exclude rules for transaction detection It will return an error if you do not specify.	Yes
entry_point_type	URI	The POJO, Servlet, EJB, Spring Bean, etc	No
rule_name	URI	The name of the rule which you are importing. It will import all the rules under a scope or all the scopes if you do not specify.	No

- When you import, if the XML file does not have <scope-list>, then you should type the scope name, otherwise, it will fail, and vice versa.
- It will return an error if you use the wrong name(s) in all the parameters.
- If you do not specify the scope when importing, then you should not specify it when exporting.
- Please check server.log if you do not get the expected result after exporting.

Examples

- Import the rule(s) to the application without scopes specified:

```
curl -X POST --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/{applicat
ion_id}/{rule_type} -F file=@{exported_file_name}.xml
```

- Import the rules to the application with scope specified:

```
curl -X POST --user user1@customer1:welcome
http://localhost:8080/controller/transactiondetection/{applicat
ion_id}/{scope_name}/{rule_type}
-F file=@{exported_file_name}.xml
```

- Please refer to Export Transaction Detection Rules for more scenarios to import.

Export Policies

You can export policies to a JSON file. Before you export policies, export any actions or health rules with their respective APIs.

Format

GET /controller/policies/*application_id*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes

Example

```
curl --user user1@customer1:secret
http://demo.appdynamics.com/controller/policies/application_id

[ {
  "applicationName" : "ECommerce-Books",
  "name" : "My Policy",
  "reactorType" : "IMMEDIATE",
  "enabled" : true,
  "batchActionsPerMinute" : true,
  "durationInMin" : 1,
  "eventFilterTemplate" : {
    "applicationName" : "ECommerce-E2E",
    "healthRuleNames" : null,
    "eventTypes" : [ "POLICY_OPEN_WARNING", "POLICY_OPEN_CRITICAL",
"POLICY_CONTINUES_WARNING", "POLICY_CONTINUES_CRITICAL" ],
    "rsdTypes" : null,
    "customEventFilters" : null,
    "specificEntityNamesByType" : null
  },
  "entityFilterTemplates" : [ ],
  "actionWrapperTemplates" : [ {
    "actionTag" : "ops_viewer@acme.com",
    "type" : null,
    "value" : 0,
    "notes" : "Policy: My Policy",
    "entityIdentifierTemplates" : [ ]
  } ]
} ]
```

Import Policies

You can import policies that you exported with the Export Policies API. Before you import policies, import any actions or health rules with their respective APIs.

You can import a policy after modifying the defined parameter(s) and overwrite the existing policy with the updated one.

Format

POST /controller/policies/*application_id* ?overwrite=*true_or_false*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes
overwrite	Query	Set to true to have updates to a policy overwrite the existing policy with the same name. The default is false.	No

Example

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/policies/38 -F
file=@ImportPolicies.json

{"success":true,"errors":[],"warnings":[]}
```

Example to overwrite a policy

```
curl  -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/policies/38\?overwrite\=true
-F file=@ImportPolicies.json

{"success":true,"errors":[],"warnings":[]}
```

Export Application Analytics Dynamic Service Configuration

The Analytics Dynamic Service is an AppDynamics app agent plugin that performs Analytics client functions for the agent. Enabling the Dynamic Service enables AppDynamics Analytics for an app agent type. You can export the Dynamic Service configuration to back up the configuration or for later import into another Controller.

Format

GET /controller/analyticsdynamicservice/*application_id*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The application name or application ID.	Yes
filename	Query	The name of a file to which the configuration will be exported.	No

Example

```
curl -i --user user1@customer1:secret
http://demo.appdynamics.com/controller/analyticsdynamicservice/10

<analytics-dynamic-service-configurations
controller-version="004-003-000-000">
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>APP_AGENT</agent-type>
        <enabled>true</enabled>
    </analytics-dynamic-service-configuration>
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>DOT_NET_APP_AGENT</agent-type>
        <enabled>true</enabled>
    </analytics-dynamic-service-configuration>
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>NODEJS_APP_AGENT</agent-type>
        <enabled>true</enabled>
    </analytics-dynamic-service-configuration>
</analytics-dynamic-service-configurations>
```

Import Application Analytics Dynamic Service Configuration

The Analytics Dynamics Service configuration determines whether AppDynamics Analytics is enabled for an app agent type. You use this API to import a previously exported configuration to another Controller.

Data for this call should be in the form of multipart/form-data.

Format

POST /controller/analyticsdynamicservice/*application_id*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
application_id	URI	The name or ID identifier of the application to which the Analytics Dynamic Service configuration should be applied.	Yes

Example

```
curl -i -X POST --user user1@customer1:secret
http://demo.appdynamics.com/controller/analyticsdynamicservice/10 -F
file=@dynamicservice.xml
```

The following listing shows sample contents of the dynamicservice.xml file. Notice that the agent-type element indicates the type of app server agent to which the enabled state of the dynamics service applies. The APP_AGENT type represents the Java Agent, the DOT_NET_APP_AGENT the .NET Agent, and so on.

```
<analytics-dynamic-service-configurations
controller-version="004-003-000-000">
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>APP_AGENT</agent-type>
        <enabled>false</enabled>
    </analytics-dynamic-service-configuration>
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>DOT_NET_APP_AGENT</agent-type>
        <enabled>false</enabled>
    </analytics-dynamic-service-configuration>
    <analytics-dynamic-service-configuration>
        <override>true</override>
        <agent-type>NODEJS_APP_AGENT</agent-type>
        <enabled>true</enabled>
    </analytics-dynamic-service-configuration>
</analytics-dynamic-service-configurations>
```

Database Visibility API

On this page:

Related pages:

Use the Database Visibility API to get, create, update, and delete Database Visibility Collectors.

Include the following headers for all Database Visibility API requests:

```
Accept: application/json; Content-type: application/json
```

JSON is currently the only supported format.

Get all Collectors

```
GET /controller/rest/databases/collectors
```

Get a Specific Collector

```
GET /controller/rest/databases/collectors/{configurationId}
```

Create a Collector

```
POST /controller/rest/databases/collectors/create
```

The JSON you send must contain the relevant Collector information. The required fields describing the Collector vary based on the type of database. See the table in the "UI Collector versus JSON Collector Configuration Field Names" section that follows for more information.

Below is a sample JSON request payload:

```
{
 "type":"MYSQL",
 "name":"localdocker_dbagent-MySQLCollector",
 "hostname":"mysql",
 "port":"3306",
 "username":"root",
 "password":"appdynamics_redacted_password",
 "enabled":true,
 "excludedSchemas":null,
 "databaseName":null,
 "failoverPartner":null,
 "connectAsSysdba":false,
 "useServiceName":false,
 "sid":null,
 "customConnectionString":null,
 "enterpriseDB":false,
 "useSSL":false,
 "enableOSMonitor":false,
 "hostOS":null,
 "useLocalWMI":false,
 "hostDomain":null,
 "hostUsername":null,
 "hostPassword":"",
 "dbInstanceIdentifier":null,
 "region":null,
 "certificateAuth":false,
 "removeLiterals":true,
 "sshPort":0,
 "agentName":"localdocker_dbagent",
 "dbCyberArkEnabled":false,
 "dbCyberArkApplication":null,
 "dbCyberArkSafe":null,
 "dbCyberArkFolder":null,
 "dbCyberArkObject":null,
 "hwCyberArkEnabled":false,
 "hwCyberArkApplication":null,
 "hwCyberArkSafe":null,
 "hwCyberArkFolder":null,
```

```
"hwCyberArkObject":null,
"orapkiSslEnabled":false,
"orasslClientAuthEnabled":false,
"orasslTruststoreLoc":null,
"orasslTruststoreType":null,
"orasslTruststorePassword":"",
"orasslKeystoreLoc":null,
"orasslKeystoreType":null,
"orasslKeystorePassword":"",
"ldapEnabled":false,
"customMetrics":null,
"subConfigs":[
{
"type":"MYSQL",
"name":"localdocker_dbagent-MySQLCollector sub-collector",
"hostname":"mysql-remote",
"port":"3388",
"username":"root",
"password":"different-password",
"enabled":true,
"excludedSchemas":null,
"databaseName":null,
"failoverPartner":null,
"connectAsSysdba":false,
"useServiceName":false,
"sid":null,
"customConnectionString":null,
"enterpriseDB":false,
"useSSL":false,
"enableOSMonitor":false,
"hostOS":null,
"useLocalWMI":false,
"hostDomain":null,
"hostUsername":null,
"hostPassword":"",
"dbInstanceIdentifier":null,
"region":null,
"certificateAuth":false,
"removeLiterals":true,
"sshPort":0,
"agentName":"localdocker_dbagent",
"dbCyberArkEnabled":false,
"dbCyberArkApplication":null,
"dbCyberArkSafe":null,
"dbCyberArkFolder":null,
"dbCyberArkObject":null,
"hwCyberArkEnabled":false,
"hwCyberArkApplication":null,
"hwCyberArkSafe":null,
"hwCyberArkFolder":null,
```



```
"hwCyberArkObject":null,
"orapkiSslEnabled":false,
"orasslClientAuthEnabled":false,
"orasslTruststoreLoc":null,
"orasslTruststoreType":null,
"orasslTruststorePassword":"",
"orasslKeystoreLoc":null,
"orasslKeystoreType":null,
"orasslKeystorePassword":"",
"ldapEnabled":false,
"customMetrics":null
```

```
 }
 ]
}
```

Update a Collector

1. Make a GET request for the collector that you want to update.
2. Copy the JSON response body that is returned by the GET request to a text editor, and modify the fields that you want to update.
3. Make a POST request for the collector that you want to update, and include the updated JSON.

```
POST /controller/rest/databases/collectors/update
```

Below is a sample JSON request payload:

```
{
 "id":1,
 "type":"MYSQL",
 "name":"localdocker_dbagent-MySQLCollector",
 "hostname":"mysql",
 "port":"3306",
 "username":"root",
 "password":"appdynamics_redacted_password",
 "enabled":true,
 "excludedSchemas":null,
 "databaseName":null,
 "failoverPartner":null,
 "connectAsSysdba":false,
 "useServiceName":false,
 "sid":null,
 "customConnectionString":null,
 "enterpriseDB":false,
 "useSSL":false,
 "enableOSMonitor":false,
 "hostOS":null,
 "useLocalWMI":false,
 "hostDomain":null,
 "hostUsername":null,
 "hostPassword":"",
 "dbInstanceIdentifier":null,
 "region":null,
 "certificateAuth":false,
 "removeLiterals":true,
```

```
"sshPort":0,
"agentName":"localdocker_dbagent",
"dbCyberArkEnabled":false,
"dbCyberArkApplication":null,
"dbCyberArkSafe":null,
"dbCyberArkFolder":null,
"dbCyberArkObject":null,
"hwCyberArkEnabled":false,
"hwCyberArkApplication":null,
"hwCyberArkSafe":null,
"hwCyberArkFolder":null,
"hwCyberArkObject":null,
"orapkiSslEnabled":false,
"orasslClientAuthEnabled":false,
"orasslTruststoreLoc":null,
"orasslTruststoreType":null,
"orasslTruststorePassword":"",
"orasslKeystoreLoc":null,
"orasslKeystoreType":null,
"orasslKeystorePassword":"",
"ldapEnabled":false,
"customMetrics":null,
"subConfigs":[
{
"id":2,
"type":"MYSQL",
"name":"localdocker_dbagent-MySQLCollector sub-collector",
"hostname":"mysql",
"port":"3388",
"username":"root",
"password":"appdynamics-redacted-password",
"enabled":true,
"excludedSchemas":null,
"databaseName":null,
"failoverPartner":null,
"connectAsSysdba":false,
"useServiceName":false,
"sid":null,
"customConnectionString":null,
"enterpriseDB":false,
"useSSL":false,
"enableOSMonitor":false,
"hostOS":null,
"useLocalWMI":false,
"hostDomain":null,
"hostUsername":null,
"hostPassword":"",
"dbInstanceIdentifier":null,
"region":null,
"certificateAuth":false,
```



```
"removeLiterals":true,
"sshPort":0,
"agentName":"localdocker_dbagent",
"dbCyberArkEnabled":false,
"dbCyberArkApplication":null,
"dbCyberArkSafe":null,
"dbCyberArkFolder":null,
"dbCyberArkObject":null,
"hwCyberArkEnabled":false,
"hwCyberArkApplication":null,
"hwCyberArkSafe":null,
"hwCyberArkFolder":null,
"hwCyberArkObject":null,
"orapkiSslEnabled":false,
"orasslClientAuthEnabled":false,
"orasslTruststoreLoc":null,
"orasslTruststoreType":null,
"orasslTruststorePassword":"",
"orasslKeystoreLoc":null,
"orasslKeystoreType":null,
"orasslKeystorePassword":"",
"ldapEnabled":false,
"customMetrics":null
```

```
}
]
}
```

The JSON you send must contain all the details of the existing collector with only the fields that you want to modify changed. To ensure you have all the fields, use the Get a Specific Collector call.

To add a new sub-collector to an existing collector, provide the sub-collector details without the `id` field.

Delete a Specific Collector

```
DELETE /controller/rest/databases/collectors/{configurationId}
```

Below is an example of a delete request.

```
DELETE /controller/rest/databases/collectors/{1}
```

Batch Delete Multiple Collectors

```
POST /controller/rest/databases/collectors/batchDelete
```

Send an array of the configuration Ids of the Collectors.

Below is an example of a batch delete command.

```
curl --user {username}@{account_name}:{password} -H "Accept:
application/json" -H "Content-type: application/json" -X POST -d
'[1,2,3]'
{Controller_URL}/controller/rest/databases/collectors/batchDelete
```

Get All Monitored Database Servers

```
GET /controller/rest/databases/servers
```

Example

```
curl --user {username}@{account_name}:{password}
{Controller_URL}/controller/rest/databases/servers
```

The output is a list of database servers and their details.

Get Database Server Details

```
GET /controller/rest/databases/servers/{dbserver_id}
```

Example

```
curl --user {username}@{account_name}:{password}
{Controller_URL}/controller/rest/databases/servers/{dbserver_id}
```

The output contains a list of the database's details, including name, node ID, and database type.

Get all Database Agent Events

```
GET /controller/rest/applications/_dbmon/events
```

For a list of query string parameters, see Retrieve Event Data.

Example

```
curl --user {username}@{account_name}:{password}
{Controller_URL}/controller/rest/applications/_dbmon/events?time-range-ty
pe=BEFORE_NOW&duration-in-mins=30&event-types=%20AGENT_EVENT,DB_SERVER_PA
RAMTER_CHANGE&severities=INFO,WARN,ERROR
```

The output gives you a list of events. For each event element, you can determine the node that the event is mapped to by looking for the entity-definition element.

Get all Database Monitoring Application Nodes

```
GET /controller/rest/applications/_dbmon/nodes
```

Example

```
curl --user {username}@{account_name}:{password}
{Controller_URL}/controller/rest/applications/_dbmon/nodes
```

UI Collector versus JSON Collector Configuration Field Names

Use the table below to ensure you use the correct field names for your API calls. The Collector configuration field names are described in Configure the Database Agent to Monitor Server Hardware and Add Database Collectors.

Section	UI Collector Configuration Field Name	JSON Collector Configuration Field Name
		id (AppDynamics assigns this ID to the Collector when you configure the Collector. You need this ID when doing a batch delete.)
	Database Type	type
	Database Agent	agentName
	Database	name
Connection Details	Hostname/IP Address	hostname
	EnterpriseDB	enterpriseDB
	Failover Partner	failoverPartner

	Listener Port	port
	Custom JDBC Connection String	customConnectionString
	Use Service Name	useServiceName
	SID or SERVICE_NAME	sid
	Connect as a sysdba	connectAsSysdba
	Username	username
	Password	password
	Logging Enabled	
Hardware Monitoring	Monitor Operating System	enableOSMonitor
	Operating System	hostOS
	Use Local WMI	useLocalWMI
	Domain	hostDomain
	SSH Port	sshPort
	Use certificate	certificateAuth
	Username	hostUsername
	Password	hostPassword

SSL field
In addition to JSON Configuration Fields listed above, there is also the ssl field. SSL is a configurable property for the Database Agent. If the Database Agent has been configured to use SSL, then you must also provide the ssl field and its value in your Database Visibility API calls.

Analytics Events API

On this page:

AppDynamics Application Analytics provides built-in support for collecting analytics data from various types of sources, such as agent-instrumented Java applications, .NET applications, browser applications, and more. The Analytics Events API lets you supplement the built-in analytics data sources with your own custom data sources and event types.

The custom events collected by the Analytics Events API are metered and are in addition to the transaction events published by the app agents. Publishing custom events requires Transaction Analytics licensing. Transaction Analytics license units determine the limit on the volume of custom events that you can publish.

About the Analytics Events API

In Application Analytics, an event encapsulates a unit of analytics data. In APM, for example, each event corresponds to a method or service invocation, whether an entry point service or downstream service.

With the Analytics API, you define the structure of your own custom event in the data store, capture the event records as they occur in your custom source, and send them to the Events Service, the data store for Analytics. Once your data is in the Events Store, users can query your data through the Controller UI or the Analytics Events API.

Analytics Events API uses a shared key to authenticate clients to the Events Service. As a Controller or Analytics Administrator, you can generate API keys from the Controller UI, see Managing API Keys for information.

A transaction analytics license is required for using the Events Service API. The same licensing model that applies to business transactions for custom events (based on the number of events per unit/day) applies to the API.

Addressing the Events Service Data Store

Unlike most AppDynamics REST APIs, which are presented at the Controller, you access the Analytics Events API by addressing the Events Service instance in the AppDynamics platform.

For a SaaS instance, you address the Events Service at one of the following URLs:

Region	URL
North America	`https://analytics.api.appdynamics.com`
Europe	`https://fra-ana-api.saas.appdynamics.com`

For an on-premises Events Service, address the Events Service instance host (or more likely, the virtual IP presented by a load balancer for the Events Service cluster). Use the primary listening port for the Events Service, 9080, by default.

Calls to the Analytics Events API need to specify the global account name for the Controller account being address, and the API key generated by the administrator for this client. The API expects the values as headers. As cURL arguments, for example, the values would be passed as follows:

```
-H"X-Events-API-AccountName:<global_account_name>"
-H"X-Events-API-Key:<api_key>"
```

You can get the global account name to use from the License page in the Controller UI. The API keys are described in Managing API Keys.

The content type, also as a cURL argument, is:

```
-H"Content-type: application/vnd.appd.events+json;v=2"
```

AppDynamics strongly recommends the use of SSL/HTTPS to access the API. Otherwise, the key is sent in plain text.

> For security reasons, the Analytics Events API, by default, does not accept cross-origin HTTP requests, such as from links embedded directly in web pages.

Data Format

The Analytics Events API takes events as JSON-formatted name-value pair data.

Before sending data that conforms to a custom events schema, you need to define the data structure for the custom schema. The Events Service matches incoming events data to the appropriate schema.

Supported Data Types

- string
- integer
- float
- boolean
- date - Supported time formats include:
  - ISO 8601 format: `yyyy-MM-dd'T'HH:mm:ss.SSSZZ`
  - UNIX epoch date format: A 13-digit number representing the number of seconds/milliseconds since UNIX epoch time (Jan 1 1970). For example, (GMT): Mon, 17 Apr 2017 23:46:22 GMT would be 1492472782000.

Naming Restrictions

Custom event names and field names must conform to the following:

- Contain only a-z, A-Z, _ (underscore), 0-9
- Names can not start with a number.

Timestamp Fields

Two implicit timestamp fields are automatically added to custom schemas:

- eventTimestamp
- pickupTimestamp

The eventTimestamp field represents the time an event occurred. An API client can specify a value for the timestamp field when it creates an event. If it does not, the Analytics Events API uses the same value for eventTimestamp as it uses for another implicit field, pickupTimestamp. The pickupTimestamp field, which is always populated by the Event Service, represents the time the event was

received by the Event Service.

You can express the timestamp fields using ISO 8601 or UNIX epoch time (64-bit milliseconds) format.

Example API Call Flow

The following steps take you through a simple workflow for using the Analytics Events API. The steps show cURL examples for creating a schema, publishing an event to that schema, and then querying the event.

1. Define the schema by associating field names with data types. For example, the following defines a Purchase event type:

```
curl -X POST
"<events_service_endpoint>:9080/events/schema/myProducts"
-H"X-Events-API-AccountName:customer1_1234-567a-bccc-123"
-H"X-Events-API-Key:a123b456-c789-1d23-e456-nnn"
-H"Content-type: application/vnd.appd.events+json;v=2" -d
'{"schema" : { "id": "string", "productBrand": "string",
"userRating": "integer", "price": "float", "productName":
"string", "description": "string" } }'
```

2. Publish an event based on the schema you created:

```
curl -X POST
"<events_service_endpoint>:9080/events/publish/myProducts"
-H"X-Events-API-AccountName:customer1_1234-567a-bccc-123"
-H"X-Events-API-Key:a123b456-c789-1d23-e456-nnn"
-H"Content-type: application/vnd.appd.events+json;v=2" -d
'[{"id": "5653b879ab33a","productBrand": "ACME","userRating":
3,"price": 2006.41,"productName": "Watch","description": "new
watch"},{"id": "5653b879700","productBrand":
"Widget","userRating": 1,"price": 3800.13,"productName":
"Watch","description": "2015 watch"}]'
```

3. Query the event data:

```
curl -X POST
"http://<events_service_endpoint>:9080/events/query"
-H"X-Events-API-AccountName:customer1_7xxx-467a-bccc-xxx"
-H"X-Events-API-Key:a123b456-c789-1d23-e456-nnn"
-H"Content-type: application/vnd.appd.events+text;v=2" -d
'SELECT * FROM myProducts'
```

If including fields with ADQL keywords, enclose the keywords in single quotes. These keywords include, for example, `between`, `in`, `se`

lect, and others.

For a single query request, use this content type:

```
-H"Content-type: application/vnd.appd.events+text;v=2"
```

In a multi-query request, the queries are passed as JSON body text. In this case, use the following content type header:

```
-H"Content-type: application/vnd.appd.events+json;v=2"
```

Custom Event Ingestion Limits

Controller ingestion of custom events has the following limits:

- Fields: 255 maximum per event type
- String attributes: 4 kb maximum length
- Batch total count: 1000 events per call
- Batch total size: 5 Mb maximum per call
- Max custom events for an account: 20

Publish Events

The Publish Events API call takes an array of events and stores them in the Event Service storage. The data must comply with an existing schema. A single request cannot publish to multiple event types.

If the event data doesn't match an event schema, the Events Service makes a best-effort attempt to match the data to the schema and returns a 400 bad request if unsuccessful.

Format

```
POST
http://<events_service_endpoint>:9080/events/publish/{schemaName}
```

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Content-Type	The Content-Type of the request body. The default is "application/vnd.appd.events+json;v=2" which also versions the resource representation (v=2).

Publish response

```
HTTP/1.1 202 ACCEPTED
```

Error Codes

Error Code	Description
400	The given request was invalid.
401	The given authentication information provided in the authorization header was invalid.
404	No event type could be found for this account.
406	The "Accept" header was not "application/vnd.appd.events+json;v=2".
413	The request body is larger than the max allowed size.
415	The "Content-Type" header was not "application/vnd.appd.events+json;v=2".
429	Too many requests. Returned when account or event reaches limits.

Create Event Schema

You use this API method to create your own event schema. The schema defines the overall structure of an event type by field and type.

You only need to use this method if the event you are uploading does not match an existing schema for first class event types (such as logs or transactions). Events that conform to an existing schema automatically match that schema. Be sure to review the supported dat a types and naming restrictions described earlier in this topic.

Format

```
POST http://<events_service_endpoint>:9080/events/schema/{schemaName}
```

Path Params

Name	Description
accountId	Account ID
schemaName	Event schema name

Query Params

N/A

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.

X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Accept	The Content-Type of the response body. The supported value is "application/vnd.appd.events+json;v=2".
Content-type	The Content-Type of the request body. The default is "application/vnd.appd.events+json;v=2" which also versions the resource representation (v=2).

Example Create Request

```
POST http://analytics.api.example.com/events/schema/{schemaName}
HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Content-Type: application/vnd.appd.events+json;v=2
Accept: application/vnd.appd.events+json;v=2
{
   "schema" : {
      "account": "integer",
      "amount": "float",
      "product": "string"
   }
}
```

Example response

```
HTTP/1.1 201 CREATED
```

Retrieve Event Schema

Use this API to retrieve an existing event schema.

Format

```
GET http://<events_service_endpoint>:9080/events/schema/{schemaName}
```

Path Params

Name	Description
accountId	Account ID

schemaName	Event schema name

Query Params

N/A

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Accept	The Content-Type of the response body. The supported value is "aapplication/vnd.appd.events+json;v=2".

Example Retrieve Request

```
GET http://analytics.api.example.com/events/schema/{schemaName}
HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Accept: application/vnd.appd.events+json;v=2
```

Example Response

```
HTTP/1.1 200 OK
{
   "schema" : {
      "account": "integer",
      "amount": "float",
      "product": "string"
   }
}
```

Update Event Schema

Use this API to update an existing event schema by field. The request body defines the updates to be applied to the event schema.

As shown in the example below, you specify each field update action as a named section in the request body. The actions are represented by these fields:

- add field
- rename field

For the add field definition, you need to specify the data format for the new field, as you would when creating the event schema.

The response to this call should be the complete event schema as modified.

Format

```
PATCH http://analytics.api.example.com/events/schema/{schemaName}
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
```

Path Parameters

Name	Description
accountId	Account id
schemaName	Event schema name

Query Params

N/A

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Accept	The Content-Type of the response body. The supported value is "application/vnd.appd.events+json;v=2".
Content-type	The Content-Type of the request body. The default is "application/vnd.appd.events+json;v=2" which also versions the resource representation (v=2).

Example Update Request

```
PATCH http://analytics.api.example.com/events/schema/{schemaName}
HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Content-type: application/vnd.appd.events+json;v=2
Accept: application/vnd.appd.events+json;v=2

[
  {
    "add": {
      "newfield": "integer"
    },
    "rename": {
      "oldname": "newname",
      "oldname2": "newname2"
    }
  }
]
```

Example Response

```
HTTP/1.1 200 OK
```

Delete Event Schema

Use this API to delete an existing event schema.

Format

```
DELETE
http://<events_service_endpoint>:9080/events/schema/{schemaName}
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
```

Path Params

Name	Description
accountId	Account id

eventType	Event schema name

Query Params

N/A

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Accept	The Content-Type of the response body. The following is the supported value: `application/vnd.appd.events+json;v=2`

Example Delete Request

```
DELETE http://analytics.api.example.com/events/schema/{schemaName}
HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Accept: application/vnd.appd.events+json;v=2
```

Querying Events

When querying analytics events data, the following apply:

- Every Events Service API has a limit of 200 searches per minute by each account on each event type.
- The Multi-Query Events API is limited to twenty queries per HTTP request.
- The Analytics Query API can return a maximum of 10,000 results.
- Limits work differently for aggregation and non-aggregation queries. Because we use the limits specified in the ADQL query as the bucket count limit, it isn't possible to use it as the overall result count limit as well. Therefore, the URL parametric limit is used for the overall limit. In the case of non-aggregation queries, there is no bucket limit, so the limit specified in the ADQL query is taken as the row count limit and URL parametric limit becomes the second place to look for it if ADQL query doesn't specify a limit.
    - For aggregation queries: the total returned row count is limited by the limit in the URL query parameter, and is not directly related to the limits specified in the ADQL query statement itself. The limits in the ADQL query apply only to bucket counts in aggregations.
    - For non-aggregation queries: If **LIMIT** is not specified in the **SELECT** statement, the value specified in the URL query parameter is used. If the limit query parameter is also absent the default is 100.

Query Events (Single Query)

There are two ways to use the query events APIs, either as a simple text format query or as a JSON-formatted query. The JSON-formatted query can accommodate multiple queries per call and is described in Query Events (Multiple Queries).

An event type might search against multiple event types. Therefore, the event type is not provided in the URL path or as a query parameter, but as part of the ADQL query provided in the request body itself. Your ADQL queries must adhere to the syntax described in the ADQL Reference.

This section describes the single query form for querying events.

Format

```
POST http://<events_service_endpoint>:9080/events/query
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Content-type: application/vnd.appd.events+text;v=2
```

Query Params

Name	Description
`start`	Filter results based on the minimum event time stamp, specified in ISO 8601 time (https://en.wikipedia.org/wiki/ISO_8601) or Unix time (http://en.wikipedia.org/wiki/Unix_time). If not specified, then the default is no minimum time stamp filtering. Note that data returned will always be limited by the data retention. Specify the time in combined UTC date and time format or epoch milliseconds. Start time is inclusive of limiting timestamps.
`end`	Filter results based on the maximum event time stamp, specified in ISO 8601 time (https://en.wikipedia.org/wiki/ISO_8601) or Unix time (http://en.wikipedia.org/wiki/Unix_time). If not specified, then the default is no maximum time stamp filtering. Note that data returned will always be limited by the data retention. Specify the time in combined UTC date and time format or epoch milliseconds. End time is inclusive of limiting timestamps.
`limit`	Limits the number of results returned. The default value is 100. The upper limit on the results that can be fetched is 10,000.

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Manage API Keys for more information.
Accept	The Content-Type of the response body. The supported value is "application/vnd.appd.events+json;v=2".
Content-type	The Content-Type of the request body. The default is "application/vnd.appd.events+text;v=2" which also versions the resource representation (v=2).

Example Query Request

```
POST
http://analytics.api.example.com/events/query?start=1422823420000&en
d=1423687476000&limit=20000 HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Content-Type: application/vnd.appd.events+text;v=2
Accept: application/vnd.appd.events+json;v=2

SELECT * FROM county WHERE size>=30 AND population>20000
```

Example Response

```
HTTP/1.1 200 OK
{
    "total": 1000,
    "fields": [
      { "label": "eventTimestamp",  "field": "eventTimestamp",
"type": "date",    "aggregation": null },
      { "label": "size",             "field": "size",
"type": "integer", "aggregation": null },
      { "label": "population",          "field": "population",
"type": "integer", "aggregation": null },
      { "label": "pickupTimestamp", "field": "pickupTimestamp",
"type": "date",    "aggregation": null }
    ],
    "results": [
      [ "2015-01-03T23:55:39.801-08:00", 35, 47500,
"2015-02-11T19:52:28.805Z" ],
      ...
    ]
}
```

Query Events (Multiple Queries)

Use this API to execute multiple queries against a particular account and event type in parallel. A query event that specifies multiple queries does so by including multiple ADQL queries in the body of the request. Your ADQL queries must adhere to the syntax described in the ADQL Reference.

The advantage of using queries in this form is that it takes advantage of certain backend optimizations in query performance. Query filter criteria, such as time range and limit, can be overridden by each inner query.

The Multi-Query Events API is limited to twenty queries per HTTP request.

Format

```
POST http://<events_service_endpoint>:9080/events/query
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
```

Path Params

None

Query Params

Name	Description
`start`	Filter results based on the minimum event time stamp, specified in ISO 8601 time (https://en.wikipedia.org/wiki/ISO_8601) or Unix time (http://en.wikipedia.org/wiki/Unix_time). If not specified, then the default is no minimum time stamp filtering. Specify the time in combined UTC date and time format or epoch milliseconds. Start time is inclusive of limiting timestamps.
`end`	Filter results based on the maximum event time stamp, specified in ISO 8601 time (https://en.wikipedia.org/wiki/ISO_8601) or Unix time (http://en.wikipedia.org/wiki/Unix_time). If not specified, then the default is no maximum time stamp filtering. Specify the time in combined UTC date and time format or epoch milliseconds. End time is inclusive of limiting timestamps.
`limit`	Limits the number of results returned. The default value is 100. The upper limit on the results that can be fetched is 10,000.

Headers

Name	Description
X-Events-API-AccountName	The global account name, as shown in the Controller UI License page.
X-Events-API-Key	The Analytics API key. See Managing API Keys for more information.
Accept	The Content-Type of the response body. The supported value is "application/vnd.appd.events+json;v=2".
Content-type	The Content-Type of the request body. The default is "application/vnd.appd.events+json;v=2" which also versions the resource representation (v=2).

Payload

Field	Description
label	(Optional) Friendly name to identify the query.
query	ADQL query to execute.
start	(Optional) Overrides the start parameter value provided as a query parameter.
end	(Optional) Overrides the end parameter value provided as a query parameter.

limit	(Optional) Overrides the limit provided as a query parameter.

Example Multi-Query POST Request

```
POST http://analytics.api.example.com/events/query?limit=100 HTTP/1.1
X-Events-API-AccountName:<global_account_name>
X-Events-API-Key:<api_key>
Content-Type: application/vnd.appd.events+json;v=2
Accept: application/vnd.appd.events+json;v=2
[
    {
      "label": "high_population",
      "query": "SELECT * FROM county WHERE population>50000",
      "limit": 10,
      "start": "2017-02-23T0:0:0Z",
      "end": "2017-03-1T0:0:0Z"
    },
    {
      "label": "small_area",
      "query": "SELECT * FROM county WHERE size<25",
      "start": "2017-02-23T0:0:0Z",
      "end": "2017-03-1T0:0:0Z"
    },
    {
      "label": "high_population_density",
      "query": "SELECT * FROM county WHERE population>50000 AND
size<25",
      "limit": 100,
      "start": "2017-02-23T0:0:0Z",
      "end": "2017-03-1T0:0:0Z"
    }
]
```

Example Response

```
HTTP/1.1 200 OK
[
  {
    "label": "high_population",
    "total": 30,
    "fields": [ ... ],
    "results": [ ... ]
  },
  {
    "label": "small_area",
    "total": 50,
    "fields": [ ... ],
    "results": [ ... ]
  },
  {
    "label": "high_population_density",
    "total": 10,
    "fields": [ ... ],
    "results": [ ... ]
  }
]
```

RBAC API

On this page:

The Role Based Access Control (RBAC) REST API allows you to manage users, groups, and roles for AppDynamics features. These operations provide more flexibility and automation with RBAC management. Relationship settings such as `addUserToGroup` and `removeUserToGroup` are supported.

To use the RBAC REST API, you must be the account owner or have administer user permission.

Note that SAML and LDAP user creations are not supported. Also, you can only create permissions through the UI.

For information about the users and groups, see Manage Users and Groups.

Create User

You can create users in the current account. The request payload should specify `name`, `security_provider_type`, `displayName`, and `password`. The user ID is generated by the server.

Format

POST /controller/api/rbac/v1/users

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
name	Request payload		Yes
security_provider_type	Request payload	"INTERNAL"	Yes
displayName	Request payload		Yes
password	Request payload		Yes

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X POST
-d '{"name": "user10", "security_provider_type": "INTERNAL",
"displayName": "user10", "password": "welcome"}' -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users

Response status code 200 :
{
  "id": 10,
  "name": "user10",
  "displayName": "user10",
  "security_provider_type": "INTERNAL"
}
```

Get User by ID

You can get full user information, including a summary of affiliated groups and roles, using the `userId` in the current account.

Format

GET /controller/api/rbac/v1/users/*userId*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users/4

Response status code 200 :
{
   "id": 4,
   "name": "user1",
   "email": "user1@customer1.com",
   "displayName": "user1",
   "security_provider_type": "INTERNAL",
   "roles":
      [
        {"id": 17,"name": "Workflow Executor"},
        {"id": 18,"name": "DB Monitoring Administrator"},
        {"id": 19,"name": "DB Monitoring User"},
        {"id": 20,"name": "Analytics Administrator"},
        {"id": 21,"name": "Server Monitoring Administrator"},
        {"id": 22,"name": "Server Monitoring User"},
        {"id": 23,"name": "Universal Agent Administrator"},
        {"id": 24,"name": "Universal Agent User"},
        {"id": 13,"name": "Account Administrator"},
        {"id": 14,"name": "Administrator"},
        {"id": 15,"name": "User"},
        {"id": 16,"name": "Dashboard Viewer"}
      ],
   "groups":
      [
        {"id": 1,"name": "group_01"}
      ]
}
```

Get User by Name

You can get full user information, including a summary of affiliated groups and roles, using the `userName` in the current account.

Format

GET /controller/api/rbac/v1/users/name/*name*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users/name/user1

Response status code 200 :
{
   "id": 4,
   "name": "user1",
   "email": "user1@customer1.com",
   "displayName": "user1",
   "security_provider_type": "INTERNAL",
   "roles":
      [
        {"id": 17,"name": "Workflow Executor"},
        {"id": 18,"name": "DB Monitoring Administrator"},
        {"id": 19,"name": "DB Monitoring User"},
        {"id": 20,"name": "Analytics Administrator"},
        {"id": 21,"name": "Server Monitoring Administrator"},
        {"id": 22,"name": "Server Monitoring User"},
        {"id": 23,"name": "Universal Agent Administrator"},
        {"id": 24,"name": "Universal Agent User"},
        {"id": 13,"name": "Account Administrator"},
        {"id": 14,"name": "Administrator"},
        {"id": 15,"name": "User"},
        {"id": 16,"name": "Dashboard Viewer"}
      ],
   "groups":
      [
        {"id": 1,"name": "group_01"}
      ]
}
```

Get All Users

You can get a list of all users in the current account. The list includes user summaries, which includes `userId` and `userName`.

Format

GET /controller/api/rbac/v1/users

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users

Response status code 200 :
{
   "users":
      [
         {"id": 4,"name": "user1"},
         {"id": 10,"name": "user10"}
      ]
}
```

Update User

You can update a user by `userId` in the current account. Only the user object itself is updated, with the relationship to roles and groups remaining unaffected.

Format

PUT /controller/api/rbac/v1/users/*userId*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
id	Request payload		Yes
name	Request payload		Yes
displayName	Request payload		Yes
security_provider_type	Request payload	"INTERNAL"	Yes

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -d
'{"id": 11,"name": "updated_user9","displayName":
"user9","security_provider_type": "INTERNAL"}' -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users/11

Response status code 200 :
{
   "id": 11,
   "name": "updated_user9",
   "displayName": "user9",
   "security_provider_type": "INTERNAL"
}
```

Delete User

You can delete a user by userId in the current account.

Format

DELETE /controller/api/rbac/v1/users/*userId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/users/11

Response status code 200 :
```

Create Group

You can create a group in the current account. The group ID is generated by the server.

Format

POST /controller/api/rbac/v1/groups

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
name	Request payload		Yes
description	Request payload		No
security_provider_type	Request payload	"INTERNAL"	Yes

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X POST
-d '{"name": "group100","description": "new description",
"security_provider_type": "INTERNAL"}' -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups

Response status code 200 :
{
   "id": 2,
   "name": "group100",
   "security_provider_type": "INTERNAL",
   "description": "new description"
}
```

Get Group by ID

You can get full group information by `groupId` in the current account.

Format

GET /controller/api/rbac/v1/groups/*groupId*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/1

Response status code 200 :
{
   "id": 1,
   "name": "group_03",
   "security_provider_type": "INTERNAL"
   "description": "",
   "roles":
      [
         {"id": 19,"name": "DB Monitoring User"},
         {"id": 20,"name": "Analytics Administrator"},
         {"id": 21,"name": "Server Monitoring Administrator"},
         {"id": 22,"name": "Server Monitoring User"},
         {"id": 23,"name": "Universal Agent Administrator"},
         {"id": 13,"name": "Account Administrator"},
         {"id": 16,"name": "Dashboard Viewer"}
      ]
}
```

Get Group by Name

You can get full group information by `groupName` in the current account.

Format

GET /controller/api/rbac/v1/groups/name/*name*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/name/group_03

Response status code 200 :
{
   "id": 1,
   "name": "group_03",
   "security_provider_type": "INTERNAL"
   "description": "",
   "roles":
      [
         {"id": 19,"name": "DB Monitoring User"},
         {"id": 20,"name": "Analytics Administrator"},
         {"id": 21,"name": "Server Monitoring Administrator"},
         {"id": 22,"name": "Server Monitoring User"},
         {"id": 23,"name": "Universal Agent Administrator"},
         {"id": 13,"name": "Account Administrator"},
         {"id": 16,"name": "Dashboard Viewer"}
      ]
}
```

Get All Groups

You can get all groups in the current account. This only returns group summaries, which includes `groupId` and `groupName`.

Format

GET /controller/api/rbac/v1/groups

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups

Response status code 200 :
{
   "groups":
      [
         {"id": 1,"name": "group_03"},
         {"id": 2,"name": "group100"}
      ]
}
```

Update Group

You can update a group by `groupId` in the current account. Only the group itself is updated, while the relationships with users and roles remain unaffected.

Format

PUT /controller/api/rbac/v1/groups/*groupId*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
id	Request payload		Yes
name	Request payload		Yes
description	Request payload		No
security_provider_type	Request payload	"INTERNAL"	Yes

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -d
'{"id": 1, "name": "group2","description": "new description",
"security_provider_type": "INTERNAL"}' -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/1

Response status code 200 :
{
   "id": 1,
   "name": "group2",
   "security_provider_type": "INTERNAL",
   "description": "new description",
   "roles":
      [
         {"id": 19,"name": "DB Monitoring User"},
         {"id": 20,"name": "Analytics Administrator"},
         {"id": 21,"name": "Server Monitoring Administrator"},
         {"id": 22,"name": "Server Monitoring User"},
         {"id": 23,"name": "Universal Agent Administrator"},
         {"id": 13,"name": "Account Administrator"},
         {"id": 16,"name": "Dashboard Viewer"}
      ]
}
```

Delete Group

You can delete a group by `groupId` in the current account.

Format

DELETE /controller/api/rbac/v1/groups/*groupId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/1


Response status code 200 :
```

Add User to Group

You can add a user to a group by `userId` and `groupId`.

Format

PUT /controller/api/rbac/v1/groups/*groupId*/users/*userId*

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -u
user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/2/users/10


Response status code 200 :
```

Remove User from Group

You can remove a user from a group by `userId` and `groupId`.

Format

DELETE /controller/api/rbac/v1/groups/*groupId*/users/*userId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/groups/2/users/10


Response status code 200 :
```

Create Role

You can create a role in the current account. The ID is generated by the server.

Format

POST /controller/api/rbac/v1/roles

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
name	Request payload		Yes
`description`	Request payload		No

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X POST
-d '{"name": "role2","description": "new description"}' -u
user1@customer1 http://localhost:8080/controller/api/rbac/v1/roles

Response status code 200 :
{
   "id": 49,
   "name": "role2",
   "description": "new description"
}
```

Add Role to User

You can add a role to a user by `roleId` and `userId`.

Format

PUT /controller/api/rbac/v1/roles/*roleId*/users/*userId*

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -u
user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/50/users/10


Response status code 200 :
```

Remove Role from User

You can remove a role from a user by `roleId` and `userId`.

Format

DELETE /controller/api/rbac/v1/roles/*roleId*/users/*userId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/50/users/10

Response status code 200 :
```

Add Role to Group

You can add a role to a group by `roleId` and `groupId`.

Format

PUT /controller/api/rbac/v1/roles/*roleId*/groups/*groupId*

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -u
user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/50/groups/2


Response status code 200 :
```

Remove Role from Group

You can remove a role from a group by `roleId` and `groupId`.

Format

DELETE /controller/api/rbac/v1/roles/*roleId*/groups/*groupId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/50/groups/2

Response status code 200 :
```

Get Role by ID

You can get full role information by `roleId` in the current account. This only returns the role object.

Format

GET /controller/api/rbac/v1/roles/*roleId*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/15

Response status code 200 :
{
   "id": 15,
   "name": "User",
   "description": "Can view applications and dashboards but not
modify their configuration"
}
```

Get Role by Name

You can get full role information by `roleName` in the current account.

Format

GET /controller/api/rbac/v1/roles/name/*name*

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/name/User

Response status code 200 :
{
   "id": 15,"name":
   "User",
   "description": "Can view applications and dashboards but not
modify their configuration"
}
```

Get All Roles

You can get all roles in the current account. This only returns role summaries, which includes `roleId` and `roleName`.

Format

GET /controller/api/rbac/v1/roles

Example

```
curl -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles

Response status code 200 :
{
   "roles":
      [
         {"id": 13,"name": "Account Administrator"},
         {"id": 14,"name": "Administrator"},
         {"id": 20,"name": "Analytics Administrator"},
         {"id": 16,"name": "Dashboard Viewer"},
         {"id": 18,"name": "DB Monitoring Administrator"},
         {"id": 19,"name": "DB Monitoring User"},
         {"id": 21,"name": "Server Monitoring Administrator"},
         {"id": 22,"name": "Server Monitoring User"},
         {"id": 23,"name": "Universal Agent Administrator"},
         {"id": 24,"name": "Universal Agent User"},
         {"id": 15,"name": "User"},
         {"id": 17,"name": "Workflow Executor"}
      ]
}
```

Update Role

You can update a role by `roleId` in the current account. This only updates the role object itself, while leaving the relationship with users and groups unaffected.

Format

PUT /controller/api/rbac/v1/roles/*roleId*

Input Parameters

Parameter Name	Parameter Type	Value	Mandatory
id	Request payload		Yes
name	Request payload		Yes
description	Request payload		No

Example

```
curl -H "Content-Type: application/vnd.appd.cntrl+json;v=1" -X PUT -d
'{"id": 49, "name": "role1","description": "new description" }' -u
user1@customer1 http://localhost:8080/controller/api/rbac/v1/roles/49

Response status code 200 :
{
   "id": 49,
   "name": "role1",
   "description": "new description"
}
```

Delete Role

You can delete a role in the current account.

Format

DELETE /controller/api/rbac/v1/roles/*roleId*

Example

```
curl -X DELETE -u user1@customer1
http://localhost:8080/controller/api/rbac/v1/roles/49


Response status code 200 :
```

License Rules API

On this page:

Related Links:

- License Management

The license rules API lets you retrieve information about the license rules as modeled in AppDynamics. For example, this information includes creating, updating and deleting license rules, and also retrieving license rules by its id, access key, and name.

Creates a New License Rule

URL

`http://<host>:<port>/mds/v1/license/rules`

Format

POST /mds/v1/license/rules

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
`license_module_type`	String	String representing the type of the license module. Module types are: • NETVIZ • MACHINE_AGENT • SIM_MACHINE_AGENT • APM • APP_AGENT • DOT_NET_AGENT	Yes
`id`	String	String representing the id or name.	Yes
`version`	Integer	Indicates the version number.	Yes
`number_of_licenses`	Integer	Integer indicating the required number of licenses.	Yes
`account_id`	String	Provides the account name or account id.	Yes
`access_key`	Path	Provides the account access key.	Yes

LicenseRule comprises of license_module_type, id, access_key, account_id, version, and number_of_licenses.

Example Request

```
curl --user <user>@<AccountName>:<password> -X POST \
  http://<host>:<port>/controller/mds/v1/license/rules \
  -H 'Content-Type: application/json' \
  -d '{
    "id": "011116a4-124c-4b73-9343-f315836f103b",
    "version": 0,
    "name": "Default_API9",
    "description": null,
    "enabled": true,
    "constraints": [],
    "entitlements": [
        {
            "license_module_type": "JAVA",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NATIVE_WEB_SERVER",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NATIVE_SDK",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "DOT_NET",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NETVIZ",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "MACHINE_AGENT",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NODEJS",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "SIM_MACHINE_AGENT",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PHP",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PYTHON",
```

```
            "number_of_licenses": 100
        }
    ],
    "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
    "access_key": "ap1845ad-1f3e-4c9b-909b-a3cfa510bfb0",
```

```
    "total_licenses": null,
    "peak_usage": null
}
```

Example Response

```
{
   "id": "011116a4-124c-4b73-9343-f315836f103b",
   "version": 0,
   "name": "Default_API9",
   "description": null,
   "enabled": true,
   "constraints": [

   ],
   "entitlements": [
      {
         "license_module_type": "JAVA",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "NATIVE_WEB_SERVER",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "NATIVE_SDK",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "DOT_NET",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "NETVIZ",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "MACHINE_AGENT",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "NODEJS",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "SIM_MACHINE_AGENT",
         "number_of_licenses": 100
```

```
      },
      {
         "license_module_type": "PHP",
         "number_of_licenses": 100
      },
      {
         "license_module_type": "PYTHON",
         "number_of_licenses": 100
      }
   ],
   "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
   "access_key": "ap1845ad-1f3e-4c9b-909b-a3cfa510bfb0",
```

```
   "total_licenses": null,
   "peak_usage": null
}
```

Note: Version must be 0 when creating.

Returns a Summary of All the License Rules for the Current Account

The license rule DTOs returned will contain a summary only, and not contain any constraints, match conditions or entitlements.

URL

`http://<host>:<port>/mds/v1/license/rules`

Format

GET /mds/v1/license/rules

Example

```
[
{
"id": "f36e08e2-5f57-4b8c-9ad2-f18873adcd87",
"version": 2,
"name": "Default",
"description": null,
"enabled": true,
"constraints": [],
"entitlements": [],
"account_id": "0429fc7b-a2b9-42e3-a7fb-f704d436a054",
"access_key": "account-accesskey-123",
"total_licenses": 120,
"peak_usage": null
}
]
```

Updates a License Rule

URL

`http://<host>:<port>/mds/v1/license/rules`

Format

PUT /mds/v1/license/rules

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
`id`	String	String representing the id or name.	Yes
`license_module_type`	String	String representing the type of the license module. Module types are: • NetViz • Machine Agent • SIM Machine Agent • APM	Yes
`version`	Integer	Indicates the version number.	Yes
`number_of_licenses`	Integer	Integer indicating the required number of licenses.	Yes
`account_id`	String	Provides the account name or account id.	Yes
`access_key`	Path	Provides the account access key.	Yes

Example Request

```
curl -v -X PUT http://<host>:<port>/controller/mds/v1/license/rules \
  --user <user>@<AccountName>:<password> \
  -H 'Content-Type: application/json' \
  -d '{
    "id": "321116a4-124c-4b73-9343-f315836f103b",
    "version": 0,
    "name": "Default_API",
    "description": null,
    "enabled": true,
    "constraints": [],
    "entitlements": [
        {
            "license_module_type": "JAVA",
            "number_of_licenses": 1000
        },
        {
            "license_module_type": "NATIVE_WEB_SERVER",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NATIVE_SDK",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "DOT_NET",
            "number_of_licenses": 100
        },
        {
```

```
            "license_module_type": "NETVIZ",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "MACHINE_AGENT",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NODEJS",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "SIM_MACHINE_AGENT",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PHP",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PYTHON",
            "number_of_licenses": 100
        }
    ],
    "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
    "access_key": "api845ad-1f3e-4c9b-909b-a3cfa510bfb0",
```

```
    "total_licenses": null,
    "peak_usage": null
}
```

Note: For updating the license rules -

1. Get the payload using any of the available GET methods.
2. Update accessKeys / entitlements etc.
3. Send the PUT request without changing other key-value pairs.
4. "version" should be kept as it was during GET. This would be incremental.

Deletes a License Rule

URL

`http://<host>:<port>/mds/v1/license/rules/f36e08e2-5f57-4b8c-9ad2-f18873adcd87`

Format

DELETE /mds/v1/license/rules/{id}

Input parameter

Parameter Name	Parameter Type	Value	Mandatory
LicenseRuleId	Path	Deletes license rule by its id.	Yes

Retrieve a License Rule via its Id

You can retrieve a license rule by the license rule id.

URL

`http://<host>:<port>/mds/v1/license/rules/f36e08e2-5f57-4b8c-9ad2-f18873adcd87`

Format

GET /mds/v1/license/rules/{licenseRuleId}

Input parameter

Parameter Name	Parameter Type	Value	Mandatory
licenseRuleId	Path	Retrieves the UUID of the license rule.	Yes

Output Parameters

Parameter Name	Parameter Type	Value	Mandatory
`id`	String	String representing the id or name.	Yes
`version`	Integer	Indicates the version number.	Yes

`total_licenses`	Integer	Integer indicating the total number of licenses.	Yes
`account_id`	String	Provides the account name or account id.	Yes
`access_key`	path	Provides the account access key.	Yes

Example Request

```
curl -X GET --user <user>@<AccountName>:<password>
http://<host>:<port>/controller/mds/v1/license/rules/091116a4-124c-4
b73-9343-f315836f103b
```

Example Response

```
{
    "id": "091116a4-124c-4b73-9343-f315836f103b",
    "version": 0,
    "name": "Default",
    "description": null,
    "enabled": true,
    "constraints": [],
    "entitlements": [
        {
            "license_module_type": "JAVA",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NATIVE_WEB_SERVER",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NATIVE_SDK",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "DOT_NET",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NETVIZ",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "MACHINE_AGENT",
```

```
            "number_of_licenses": 100
        },
        {
            "license_module_type": "NODEJS",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "SIM_MACHINE_AGENT",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PHP",
            "number_of_licenses": 100
        },
        {
            "license_module_type": "PYTHON",
            "number_of_licenses": 100
        }
    ],
    "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
    "access_key": "bd3845ad-1f3e-4c9b-909b-a3cfa510bfb0",
```

```
    "total_licenses": null,
    "peak_usage": null
}
```

Note: Version gets updated after every update from UI or API.

Retrieves a License Rule by Access Key

URL

`http://<host>:<port>/mds/v1/license/rules/accessKey/account-accesskey-123`

Format

GET /mds/v1/license/rules/accessKey/{accessKey}

Input parameter

Parameter Name	Parameter Type	Value	Mandatory
`access_key`	Path	Retrieves the access key assigned to the license rule.	Yes

Output Parameters

Parameter Name	Parameter Type	Value	Mandatory
`id`	String	String representing the id or name.	Yes
`version`	Integer	Indicates the version number.	Yes
`total_licenses`	Integer	Integer indicating the total number of licenses.	Yes
`account_id`	String	Provides the account name or account id.	Yes

Example Request

```
curl -X GET --user <user>@<AccountName>:<password>
http://<host>:<port>/controller/mds/v1/license/rules
```

Example Response

```
[
    {
        "id": "221116a4-124c-4b73-9343-f315836f103b",
        "version": 0,
        "name": "Default",
        "description": null,
        "enabled": true,
        "constraints": [],
        "entitlements": [],
        "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
        "access_key": "bd3845ad-1f3e-4c9b-909b-a3cfa510bfb0",
        "total_licenses": 1000,
        "peak_usage": null
    },
    {
        "id": "321116a4-124c-4b73-9343-f315836f103b",
        "version": 0,
        "name": "Custom1",
        "description": null,
        "enabled": true,
        "constraints": [],
        "entitlements": [],
        "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
        "access_key": "api845ad-1f3e-4c9b-909b-a3cfa510bfb0",
        "total_licenses": 1000,
        "peak_usage": null
    }
]
```


Retrieve a License Rule by Name

URL

`http://<host>:<port>/mds/v1/license/rules/name/Default`

Format

GET /mds/v1/license/rules/name/{name}

Input parameter

Parameter Name	Parameter Type	Value	Mandatory
name	Path	Retrieves the name of the license rule.	Yes

Output Parameters

Parameter Name	Parameter Type	Value	Mandatory
`id`	String	String representing the id or name.	Yes
`version`	Integer	Indicates the version number.	Yes
`total_licenses`	Integer	Integer indicating the total number of licenses.	Yes
`account_id`	String	Provides the account name or account id.	Yes

Example Request

```
curl -X GET --user <user>@<AccountName>:<password>
http://<host>:<port>/controller/mds/v1/license/rules
```

Example Response

```
[
    {
        "id": "221116a4-124c-4b73-9343-f315836f103b",
        "version": 0,
        "name": "Default",
        "description": null,
        "enabled": true,
        "constraints": [],
        "entitlements": [],
        "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
        "access_key": "bd3845ad-1f3e-4c9b-909b-a3cfa510bfb0",
        "total_licenses": 1000,
        "peak_usage": null
    },
    {
        "id": "321116a4-124c-4b73-9343-f315836f103b",
        "version": 0,
        "name": "Custom1",
        "description": null,
        "enabled": true,
        "constraints": [],
        "entitlements": [],
        "account_id": "8b1e92f2-387c-45b6-9bb5-ebd7f67587d4",
        "access_key": "api845ad-1f3e-4c9b-909b-a3cfa510bfb0",
        "total_licenses": 1000,
        "peak_usage": null
    }
]
```


Note for all APIs:

- id - This is GUID (or UUID) which is an acronym for Globally Unique Identifier.
- access_key - It must be unique across all licenseRules.
- name - It must be unique across all licenseRules.
- account_id - Use GET api to know your current account_id

Integration Modules

The AppDynamics Community Exchange includes many extensions that supplement the capabilities of the AppDynamics Application Performance Monitoring (APM) Platform. Documentation on setting up and using the integration module appears with the module.

The following integration modules are built into the AppDynamics platform and described at the pages linked to here.

Integration Module	More information
Apica	Apica & AppDynamics
DB CAM	Integrate AppDynamics with DB CAM
Scalyr	Integrate AppDynamics with Scalyr
Splunk	Integrate AppDynamics with Splunk
Compuware Strobe	Integrate AppDynamics with Compuware Strobe
AppDynamics for Databases	Integrate AppDynamics for Databases with AppDynamics Pro 3.7 and higher
Cisco® Application Centric Infrastructure (Cisco ACI™)	Integrate AppDynamics with Cisco ACI™

Integrate AppDynamics with DB CAM

On this page:

Related pages:

- Access the Administration Console
- Monitor Databases

You can link to DB CAM for any DB CAM-monitored database that is discovered by AppDynamics. This integration provides access to the database performance metrics provided by DB CAM.

Prerequisites for DB CAM Integration

To use this integration you must have a DB CAM license.
DB CAM must be configured to monitor the databases that you want to link to from AppDynamics.

Configuring AppDynamics to Interface with DB CAM

You configure DB CAM integration at the account level and at the app agent level.

Configure DB CAM at the Account Level

Configure the integration at the account level using the Administration Console at

```
<host>:<port>/controller/admin.jsp
```

Create two new properties as name-value pairs in each account for which you want to enable DB CAM integration.

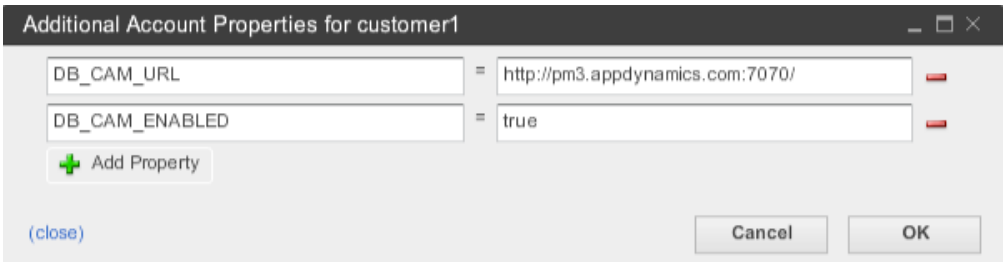
Configure One AppDynamics Account for DB CAM Integration

1. Log in to the Administrator Console with the administrator root password.
2. Select Accounts.
3. In the accounts list, double-click the account for which you want to configure DB CAM integration.
4. In the upper right corner of the account screen, click **Additional Account Properties**.
5. In the Additional Account Properties screen, click **Add Property** to add the DB_CAM_URL property.
6. In the left field enter "DB_CAM_URL".
7. In the right field enter the URL of the AppDynamics Controller that you are configuring using the syntax:

```
http[s]://<host>:<port>
```

8. Click **Add Property** again to add the DB_CAM_ENABLED property.
9. In the left field enter "DB_CAM_ENABLED".
10. In the right field enter "true".

11. Click **OK** to save the properties.
12. Log out of the Administrator Console.



Configure DB CAM at the Agent Level

For each app agent for which you want to enable access to deep diagnostics from DB CAM:

1. Open the AppServerAgent/conf/app-agent-config.xml file for the app agent.
   Locate the TransactionMonitoringService element:

```
<agent-service name="TransactionMonitoringService"
enabled="true">
```

2. Add the jdbc-dbcam-integration-enabled property for the service:

```
<agent-service name="TransactionMonitoringService"
enabled="true">

<service-dependencies>BCIEngine,SnapshotService</service-depend
encies>
         <configuration-properties>
             <property name="jdbc-dbcam-integration-enabled"
value="true"/>
```

3. Save the file.

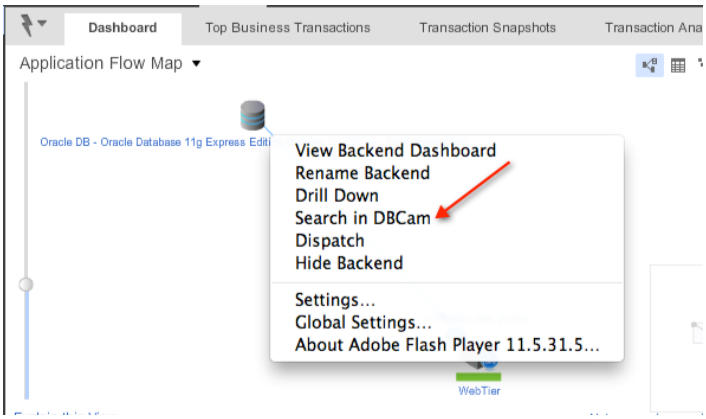
Linking to DB CAM from AppDynamics

You can link to DB CAM from any AppDynamics flow map that displays a discovered DB CAM-monitored database. The flow map could be in a dashboard or a transaction snapshot.

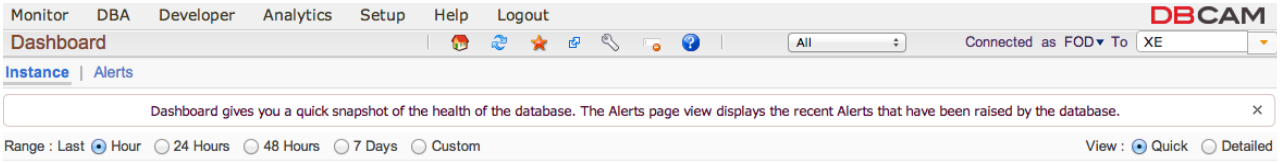
If you link to DB CAM from a dashboard you will land in the DB CAM instance dashboard. If you link to be DB CAM from a transaction snapshot, you will land in the DB CAM Session Drill Down screen.

To Link to DB CAM from a Dashboard

1. In the flow map of a dashboard, right-click on the link below a database icon.
2. Click **Search in DBCam**.

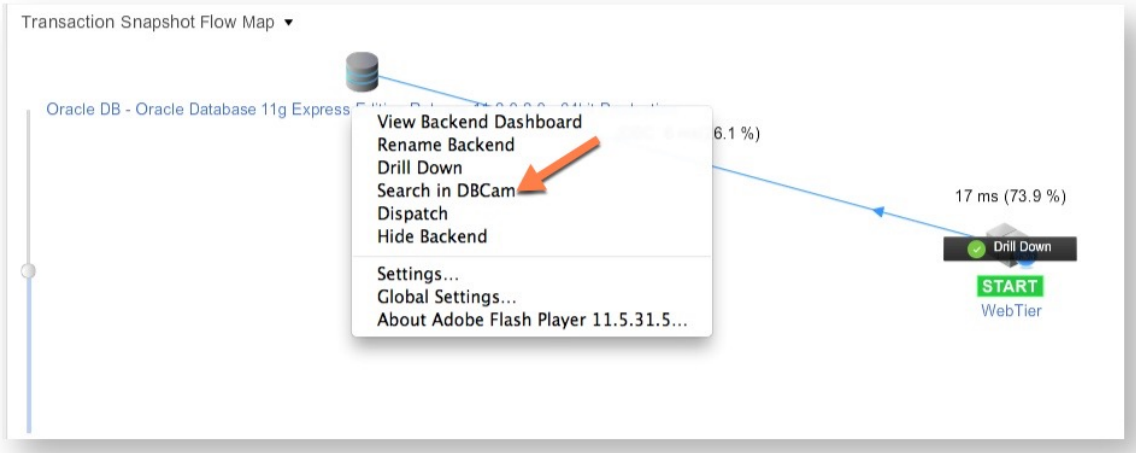


DB CAM launches and displays the instance dashboard for the selected database.

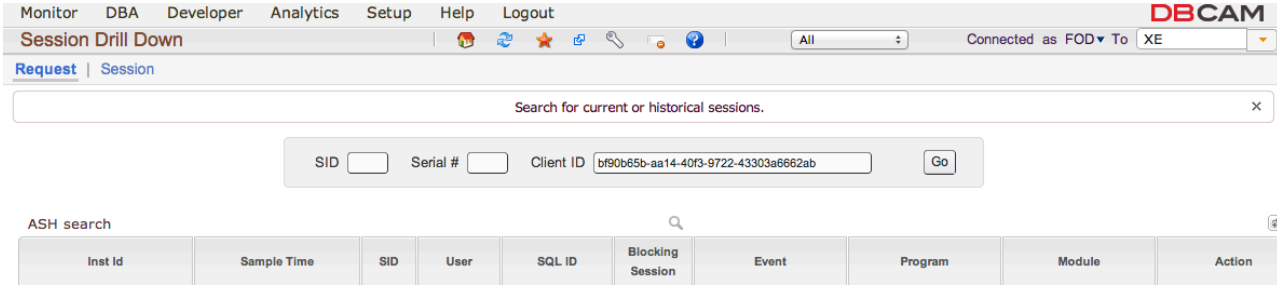


To Link to DB CAM from a Transaction Snapshot

1. In the flow map of a transaction snapshot, right-click on the link below the database icon.
2. Click **Search in DBCam**.



DB CAM launches and displays the session drill down for the selected database.



Integrate AppDynamics with Splunk

On this page:

Related pages:

- Splunkbase Apps
- Splunk documentation

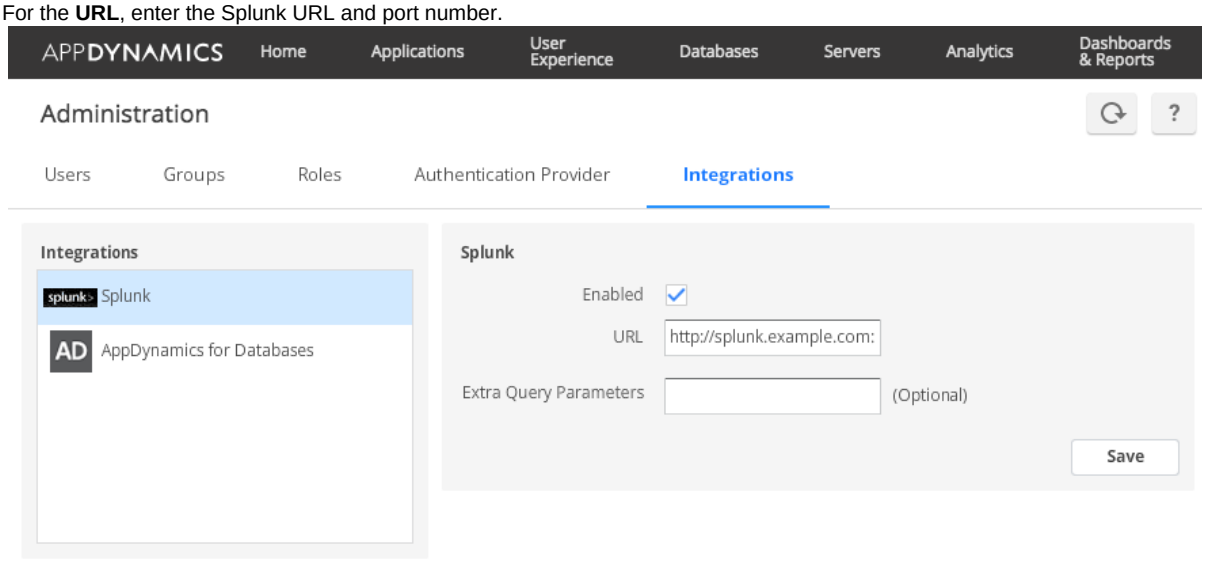
The AppDynamics-Splunk integration gives you a single, cohesive view of data gathered by AppDynamics and Splunk. The AppDynamics-Splunk integration gives you the ability to:

- Launch Splunk searches using auto-populated queries from the AppDynamics Console based on criteria such as time ranges and node IP address.
- Push notifications on policy violations and events from AppDynamics to Splunk.
- Mine performance data from AppDynamics using the Controller REST API and push it into Splunk.

This topic covers the Splunk search capability that is launched from AppDynamics. Pushing AppDynamics notifications to Splunk and mining performance data from AppDynamics to Splunk requires the Splunk extension. See the extension documentation for more information on those capabilities.

Configuring Splunk Integration

1. To enable and configure Splunk integration, log in the Controller UI as an administrator.
2. Click **Settings > Administration**.
3. In the Administration window, click the **Integration** tab and then Splunk from the Integrations list.
4. Check the **Enabled** checkbox to enable the integration.
5. For the **URL**, enter the Splunk URL and port number.



6. (Optional) Enter Extra Query Parameters. These parameters are appended to each Splunk search initiated from AppDynamics.
7. Click **Save.**

Launching a Splunk Search from AppDynamics

You can launch a search of your Splunk logs for a specific time frame associated with a transaction snapshot from several points in

AppDynamics.

To be able to launch a Splunk search, you need:

- Splunk credentials. The first time that you launch a Splunk search, you are prompted to provide your Splunk credentials. After this, your credentials are cached by the browser.
- Splunk Server must be running.
- Your browser is configured to allow pop-ups.

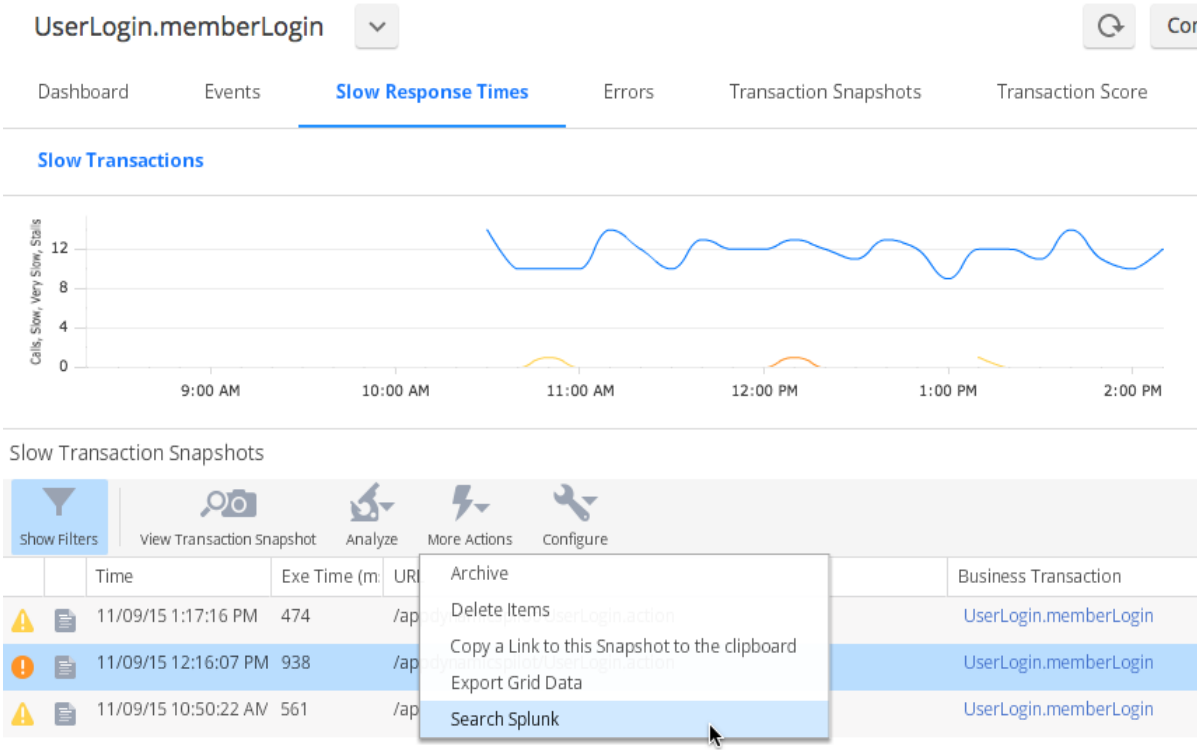
> **Enable Pop-ups**
> The first time that you access the Splunk Server, you are prompted to log in. If nothing happens, it is most likely that either your browser is blocking the Splunk login pop-up or the Splunk Server is not running.

You can access the **Search Splunk** option from the node dashboard as follows:

1. Navigate to a node dashboard.
2. From the **Actions** menu, select the **Search Splunk** item in the Integrations section.

In the list of transaction snapshots:

1. From the business transaction dashboard, select the Transaction Snapshot tab.
2. Select a transaction snapshot and right-click to access a list of options or click **More Actions** to see **Search Splunk** in the list of options.
   More Actions:



Integrate AppDynamics with Scalyr

On this page:

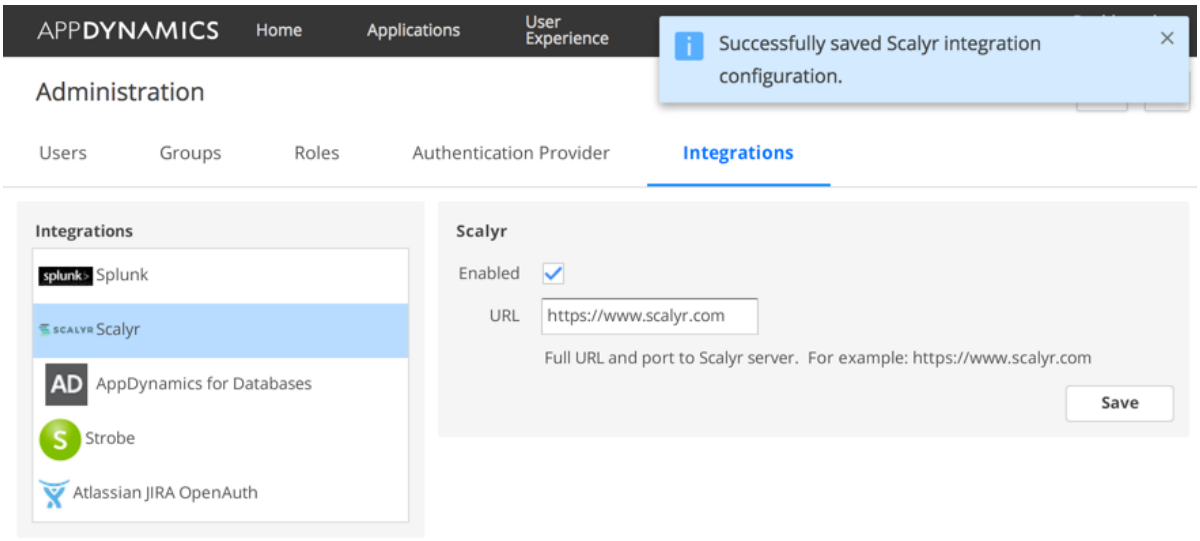
The Scalyr integration module gives you a single entry point for viewing and analyzing data gathered by AppDynamics and Scalyr. With the Scalyr integration module enabled, a user in the Controller can launch Scalyr searches directly from the AppDynamics Console.

The Scalyr search includes context from the AppDynamics UI session, including the time range and IP or hostname of the node being investigated in the AppDynamics UI, as described here.

Enable the Scalyr Integration Module

To enable and configure the Scalyr integration:

1. Log in to the Controller UI as an administrator.
2. Click **Settings > Administration**.
3. In the Administration window, click the **Integration** tab and then **Scalyr** from the Integrations list.
4. Check the **Enabled** check box to enable the integration.
5. For the **URL**, enter the full URL for Scalyr, https://www.scalyr.com.
6. Click **Save.**



Search Scalyr Data from AppDynamics

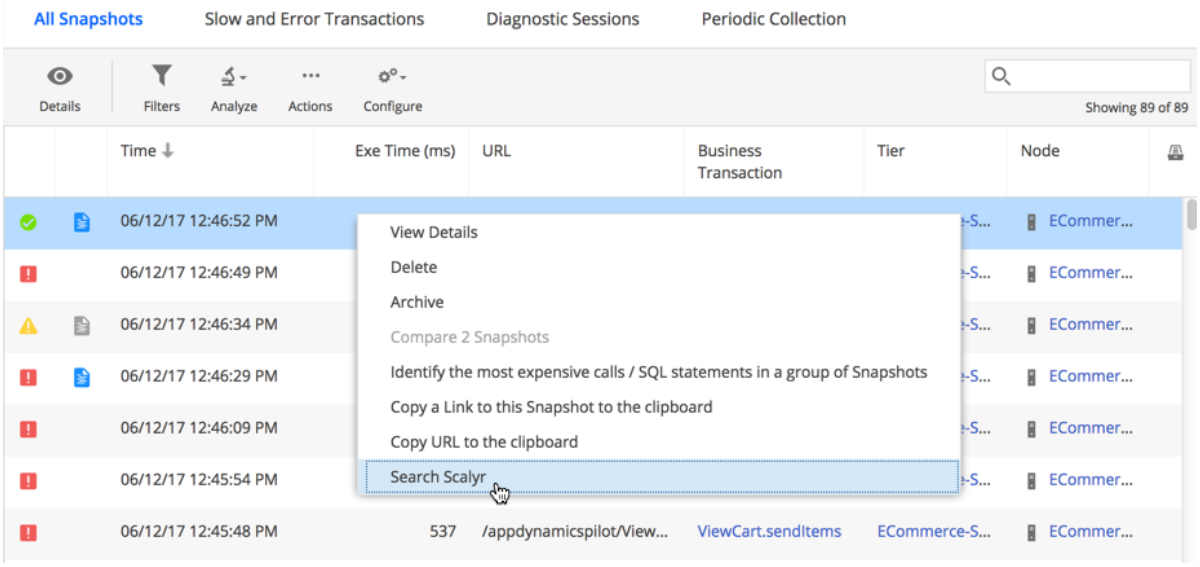
To be able to launch a Scalyr search, Controller UI users need to have an active browser session in the Scalyr UI. Thus, users will need:

- Scalyr credentials.
- A browser with pop-ups permitted.

Controller users can search Scalyr logs while viewing nodes, transaction snapshots, or segments of a transaction snapshot. The time frame passed to Scalyr varies for each context. Those time frames, along with the details for accessing the search option, are as follows:

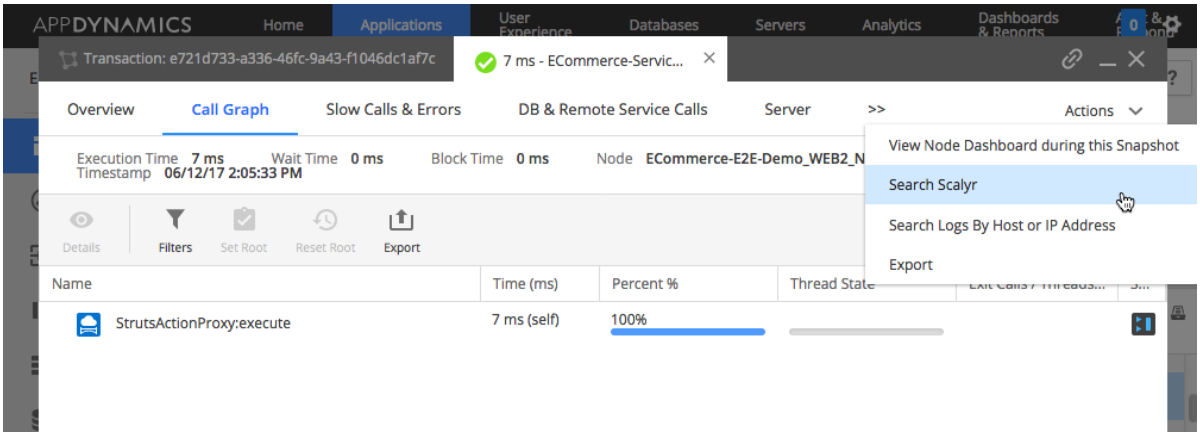
- For a node, choose **Search Scalyr** from the **Actions** menu of the node dashboard. This passes a search for that node with the timeframe currently selected in the Controller UI.
- To search Scalyr from the context of a transaction snapshot, select the transaction snapshot from the list and choose **Search**

Scalyr from the right-click or **Actions** menu. The same option is available from the snapshot view:



This generates a search on the nodes where the transaction occurred within the general time frame of the transaction execution. Specifically, the time frame is from 30 minutes before the transaction start time to 30 minutes after.

- To narrow the context further, you can search Scalyr from the context of a call drill down. This generates a search on the machine on which the snapshot was generated for the time frame of that segment. The **Search Scalyr** option is available from the **Action** menu of the call drill-down view, as shown here:



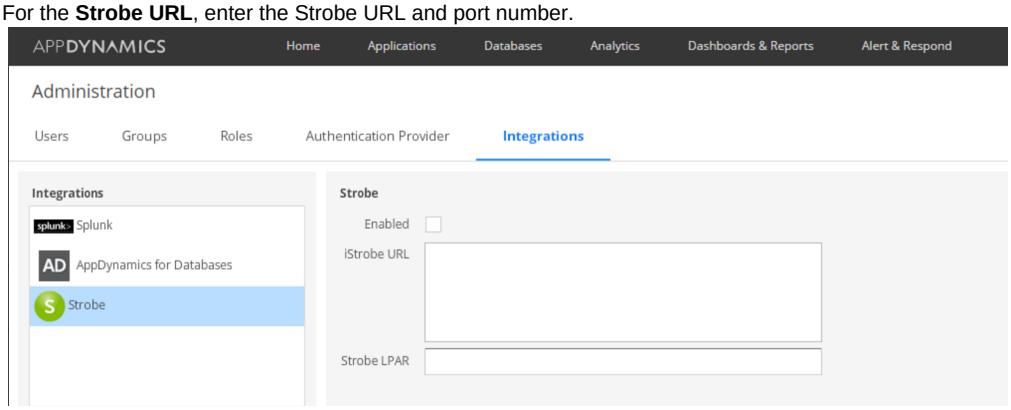
Integrate AppDynamics with Compuware Strobe

On this page:

The AppDynamics-Compuware Strobe integration gives you the ability to drill down from an SQL Call of a transaction snapshot to view and analyze the calls in Strobe.

Configure Compuware Strobe Integration

1. To enable and configure Strobe integration, log in the Controller UI as an administrator.
2. Click **Settings > Administration**.
3. In the Administration window, click the **Integration** tab and then select **Strobe** from the Integrations list.
4. Check the **Enabled** check box to enable the integration.
5. For the **Strobe URL**, enter the Strobe URL and port number.

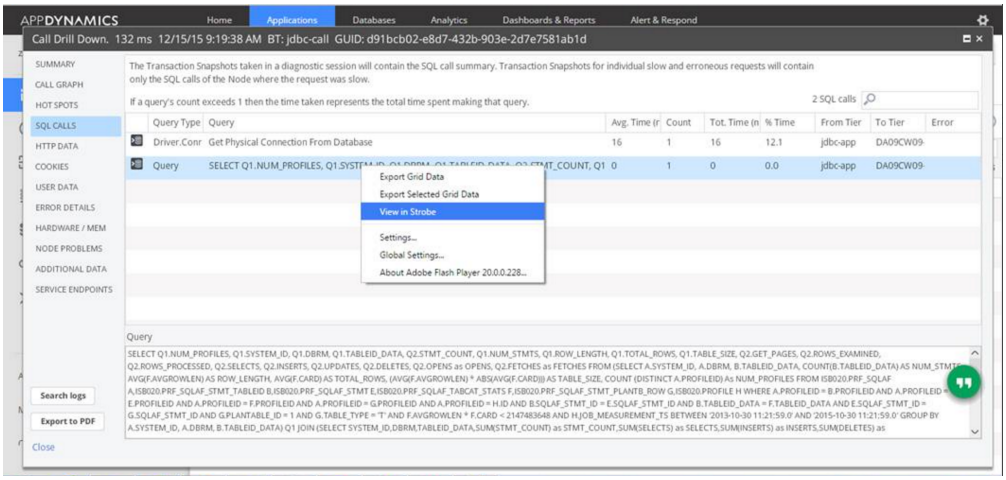


6. For the **Strobe LPAR**, enter the name of the logical partition where Strobe is running.
7. Click **Save.**

Launch Strobe from AppDynamics

You can access the **View in Strobe** option from a transaction snapshot as follows:

1. From the Business Transactions dashboard, select the Transaction Snapshot tab.
2. Select a transaction snapshot containing DB2 z/OS database access.
3. Choose the SQL Calls tab within the snapshot viewer.
4. Right-click an SQL query and then select **View in Strobe** from the list of options that appears.



Integrate AppDynamics with Cisco® Application Centric Infrastructure

Related pages:

Getting Started with AppDynamics- Cisco ACI™ Integration

Troubleshoot Using the Integrated Solution

The integration of AppDynamics with Cisco® Application Centric Infrastructure (Cisco ACI™) solution provides a unified view from the application code to underlying network layers for business-critical applications running in a data center. This enables the network operations admin and the application operations admin to quickly troubleshoot issues both at the application and the network levels.

You can integrate the Cisco ACI™ solution with select AppDynamics Controllers only. If you are interested in using the integrated solution, contact help@appdynamics.com.

Getting Started with AppDynamics- Cisco ACI™ Integration

On this page:

Related pages:

Troubleshoot Using the Integrated Solution

The AppDynamics integration with Cisco ACI™ solution provides end-to-end visibility into the application and network layers within a data center. The following sections provide instructions to integrate AppDynamics with Cisco ACI™ and verify inter-operability between the applications.

System Requirements

Ensure that the following minimum system requirements are met for the integration:

- Version 4.5.2 or above of the Controller with a Network Visibility license per OS instance. For more information, refer Controller System Requirements.
- Hosts with Network Visibility agents (version 4.5.2 or above). For more information, refer to Network Visibility Requirements and Important Notes.
- Hosts with Java App Server agents (version 4.5.2 or above) with an App Agent license. For more information, refer Install Java Agent.
- `sudo` or `root` access permissions on the agent host to install the Network Agent
- Network Visibility is supported on Linux hosts with Java App Server agents only. The following platforms are supported:
  - CentOS 6 and higher (32-bit and 64-bit)
  - Ubuntu 14 and higher (32-bit and 64-bit)
  - Red Hat Enterprise Linux 6 and higher (32-bit and 64-bit)
  - Fedora 24 and higher (32-bit and 64-bit)

> If your applications are running in a Kubernetes environment, enable the Kubernetes-ACI integration, before you proceed with the AppDynamics and Cisco ACI™ integration. For more information, see Cisco ACI and Kubernetes Integration

Enable SSL for Cisco ACI Integration

The Cisco APIC controller supports the import and storing of an SSL certificate and private key into the controller. After import, the certificate and private key can be used to create a secure and trusted environment between Cisco APIC controller and AppDynamics.

1. Access the AppDynamics controller using the SSH.
2. At a command prompt, change directories to the following location:

   ```
   <controller_home>/appserver/glassfish/domains/domain1/config
   ```

3. Import the self-signed certificate from the appropriate directory as follows:

   ```
   keytool -import -keystore cacerts.jks -file /path/to/RootCA.crt
   ```

Enable Cisco ACI Integration

> If you are using AppDynamics controller version 4.5.10 or earlier, enable the Cisco ACI Integration manually as follows. The Cisco ACI Integration is enabled by default for AppDynamics controller version 4.5.11 or later.

Configure AppDynamics Controller for Cisco ACI integration as follows:

1. Log in to the administration console:

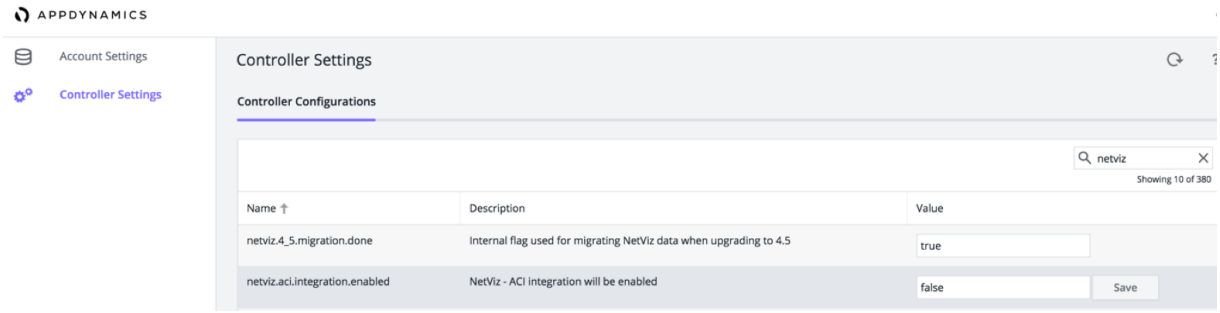
   ```
   http:<controller-hostname>:<controller-port>/controller/admin.jsp
   ```

   For example:

   ```
   https:<controller-hostname>:8800/controller/admin.jsp
   https:<controller-hostname>:443/controller/admin.jsp
   ```

   > Use the password set for the root user of the controller when the controller was installed. For more information, refer to User Management.

2. In **Controller Settings**, search for **netviz.aci.integration.enabled** and set this field to **true**. Click **Save**.



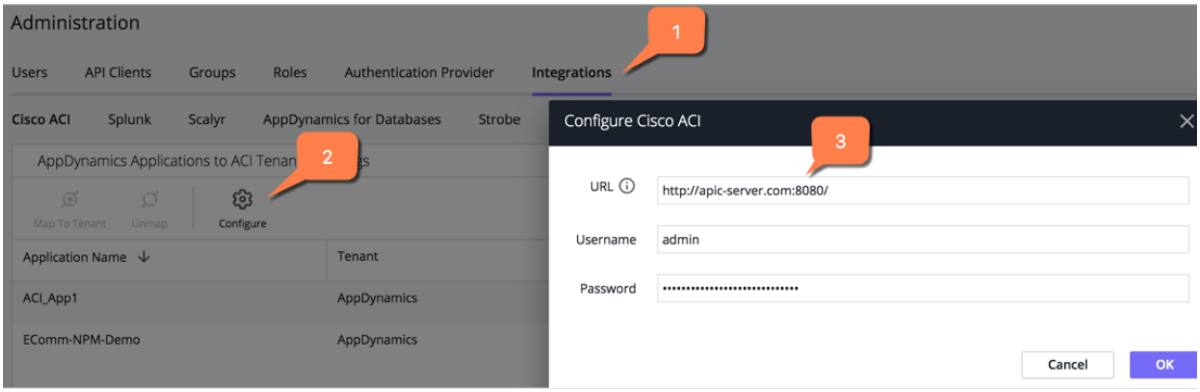
Configure Cisco ACI Credentials

If you are using AppDynamics controller version 4.5.11 or later, the Cisco ACI Integration feature is enabled by default. If you are using AppDynamics controller version 4.5.10 or earlier, enable Cisco ACI Integration manually before you configure your credentials. Configure the Cisco ACI credentials to complete the integration as follows:

1. Log in to the AppDynamics controller as an administrator:

   ```
   http:<controller-hostname>:<controller-port>/controller
   ```

2. Click **Settings > Administration**.
3. In the **Administration** window, click the **Integration** tab.
   Cisco ACI is selected by default in the **Integrations** list.



4. Click **Configure**.
5. In the **Configure Cisco ACI** window, enter the following details:
    a. full URL along with the port details to the Cisco APIC Server. For example,

    ```
    http://apic-server.com:8080/
    ```

    b. user credentials to the Cisco APIC controller and click **OK**.

AppDynamics initiates the background processes required to inter-operate with the Cisco ACI solution.

Correlate AppDynamics and Cisco ACI Components

The AppDynamics-Cisco ACI integration brings together the logical constructs of both the solutions. This enables the network operations admin and the application operations admin to obtain a comprehensive end-to-end view from the application to the network. Correlating or mapping the components of both solutions helps the admins triage and troubleshoot issues quickly.

AppDynamics Components

AppDynamics application model serves as the framework around which AppDynamics organizes and presents application performance information. A typical application environment consists of different components that interact in multiple ways.

Application

A business *application* is the top-level container in the AppDynamics model. A business application contains a set of related services and business transactions.

Node

A *node* in the AppDynamics model corresponds to a monitored server or JVM in the application environment. A node is the smallest unit of the modeled environment. Depending on the agent type, a node may correspond to an individual application server, JVM, CLR, PHP application, and Apache Web server.

Tier

A *tier* is a unit in the AppDynamics application model which is a grouping of one or more nodes. There is no interaction among nodes within a single tier.

AppDynamics components are logically arranged as follows:

Application > Tiers > Nodes

Cisco ACI Components

Cisco ACI is a Software-Defined Networking (SDN) solution. It simplifies, optimizes, and accelerates infrastructure deployment and governance, and expedites the application deployment lifecycle.

Cisco ACI implements Cisco's intent-based networking framework. It captures higher-level business and user intent in the form of a policy and converts this intent into the network constructs necessary to dynamically provision the network, security, and infrastructure services.

Cisco APIC

The infrastructure controller is the main architectural component of the Cisco ACI solution. It is the unified point of automation and management for the Cisco ACI fabric, policy enforcement, and health monitoring. The APIC appliance is a centralized, clustered controller that optimizes performance and unifies the operation of physical and virtual environments. The controller manages and operates a scalable multi-tenant Cisco ACI fabric.

Tenant

A tenant is a logical container for policies that enable an administrator to exercise domain-based access control.

Application Profile

An application profile defines the policies, services, and relationships between endpoint groups (EPGs).

EPG

An EPG is a managed object that is a named logical entity. It is a collection of endpoints (EPs).

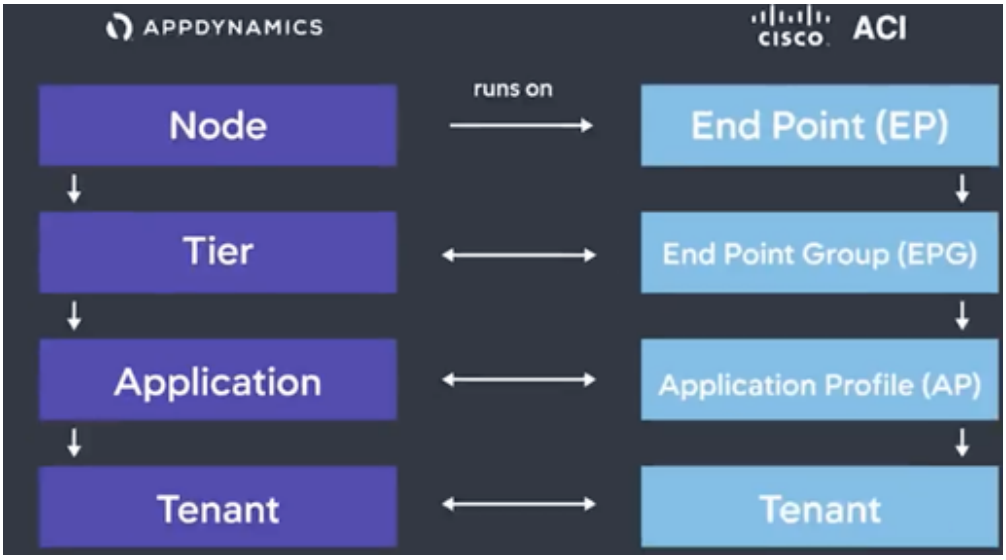
Endpoints

Endpoints are devices that are connected to the network directly or indirectly. They have an address (identity), a location, and attributes (such as version).

The logical topology in the Cisco APIC controller is as follows:

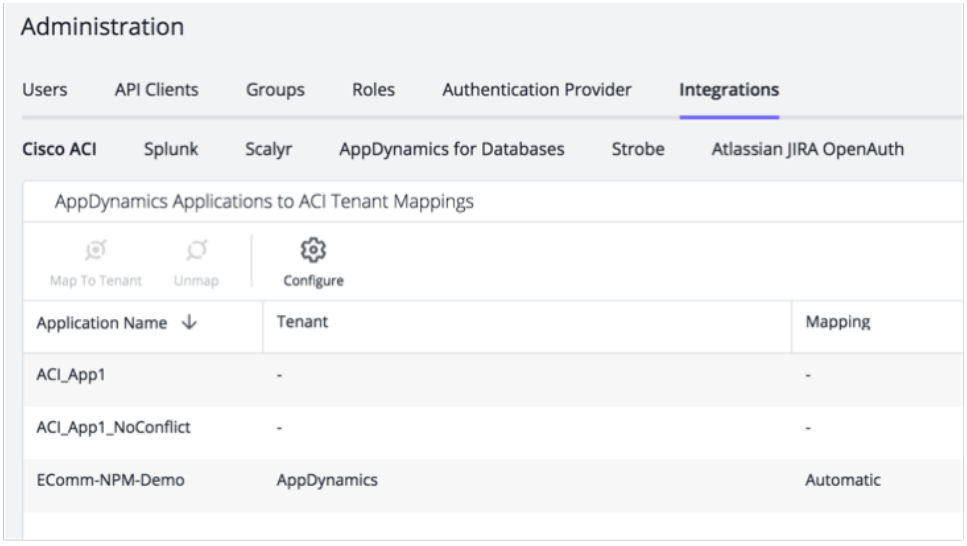
Application profile > EPGs > EPs

The following figure depicts the correlation of AppDynamics and Cisco ACI components.



AppDynamics Applications to ACI Tenant Mapping

After you configure Cisco ACI credentials, all AppDynamics applications are displayed in the **AppDynamics Applications to ACI Tenant Mappings** page.

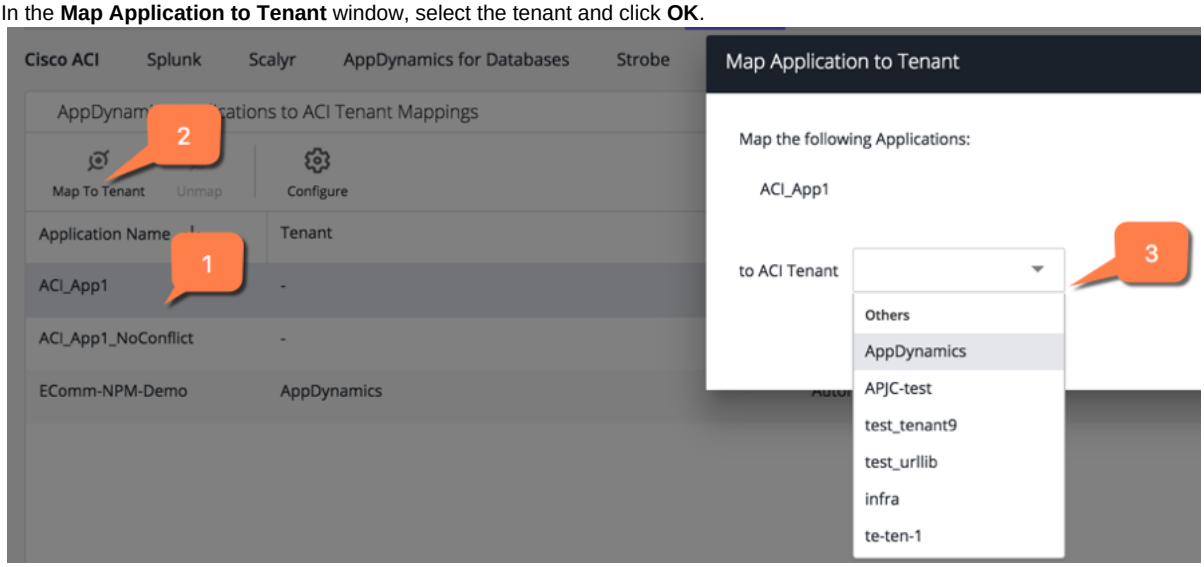


These mappings are based on the mappings between AppDynamics nodes and Cisco ACI endpoints. When the endpoints in different tenants have the same IP address, a node-to-endpoint mapping conflict occurs. AppDynamics resolves these conflicts based on heuristic data. However, if the conflicts persist, you must resolve these conflicts manually, as described in Mapping Applications to Tenants Manually.

Map AppDynamics Application to Cisco ACI Tenant Manually

The application to Cisco ACI tenant map is left blank in *AppDynamics Applications to ACI Tenant Mappings* page when there is a conflict in auto-mapping. You can map applications to Cisco ACI tenants manually:

1. Select the AppDynamics application you want to map manually.
   Multiple applications can be mapped to a tenant at a given time.
2. Click **Map to Tenant**.
3. In the **Map Application to Tenant** window, select the tenant and click **OK**.



Override Auto-mapping

You can override the auto-map and map the application to another Cisco ACI tenant using the **Map Application to Tenant** window. To override the auto-map, **Unmap** the application to tenant map and map it to another Cisco ACI tenant.

To revert to auto-map, **Unmap** the application to tenant map and refresh the page. AppDynamics auto-maps the application to the corresponding Cisco ACI tenant.

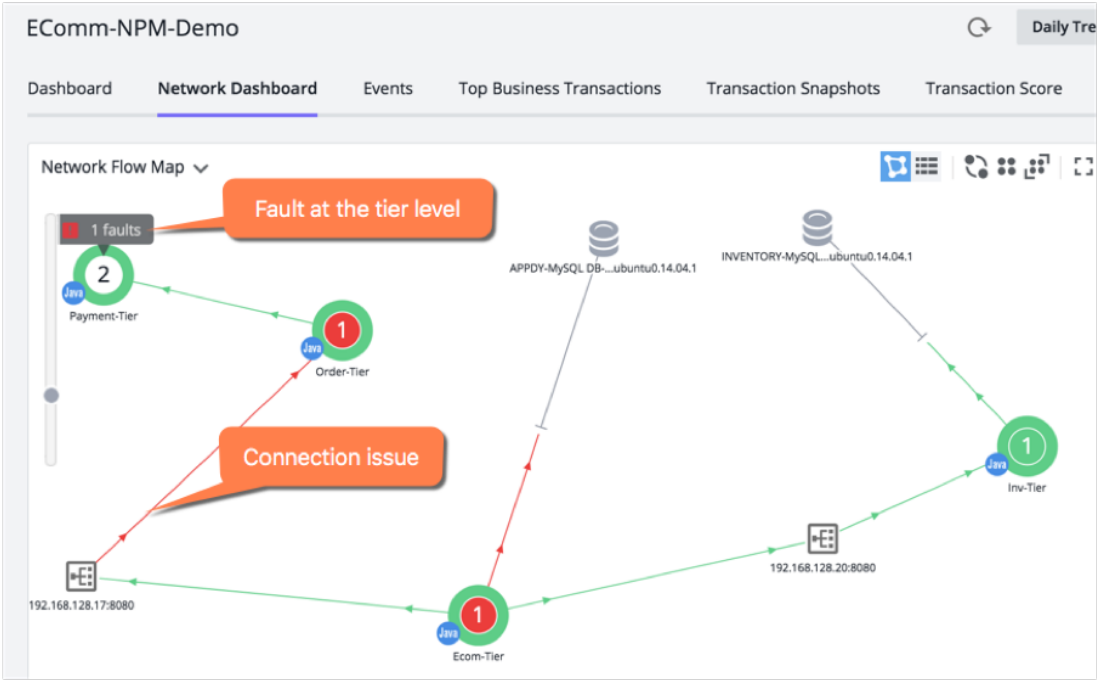
Troubleshoot Using the Integrated Solution

On this page:

Related pages:

Getting Started with AppDynamics- Cisco ACI™ Integration

The AppDynamics Network Dashboard displays the network-layer view of a monitored application. Based on the runtime state of the application, AppDynamics automatically fetches the faults and connection issues and displays them in the network dashboard as shown in the following figure.



Faults are displayed at the tier level. The number of faults displayed indicates an aggregate of faults occurring at all EPGs within the tier. You can view the number of faults at a given point of time in the Network Dashboard.

A connection is the set of all traffic for an application that has the same Source IP, Destination IP, Destination Port, and Protocol. AppDynamics detects if there is an issue in the traffic flow and indicates the connection issue on the network dashboard.

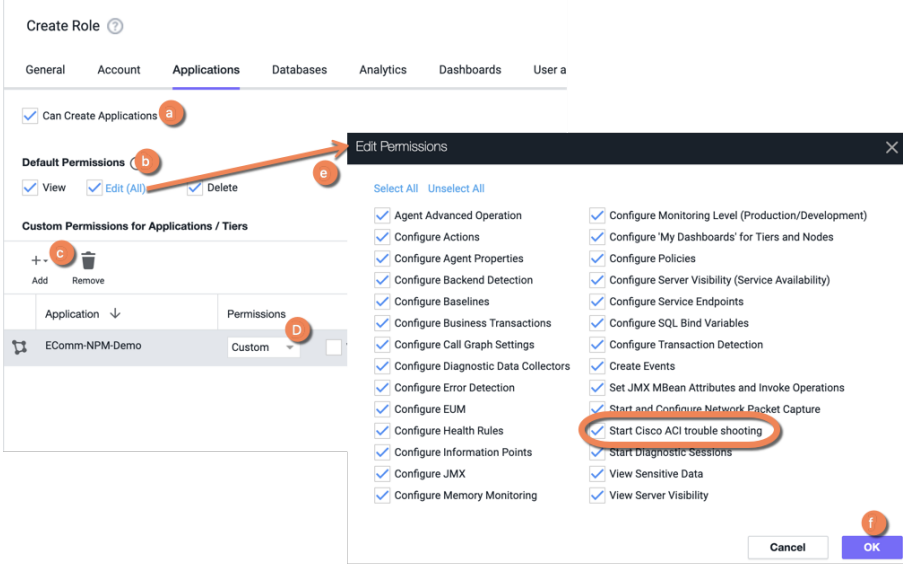
You can initiate a diagnosis to determine the exact location of the connection issue within the network by cross-launching the connection in Cisco APIC controller.

Set Custom Permission to Cross-Launch Cisco APIC Controller

To include permission to cross-launch Cisco APIC controller, follow these steps:

1. While logged in to the Controller UI as an Administrator or Account Owner, click the gear icon ( ) > **Administration**.
2. Click the **Roles** tab to view the list of predefined roles.
3. Click **+ Create** to create a custom role.
4. Specify a **Name** and a **Description** for the newly created role.
5. Customize business application level permission as follows:
    a. On the **Applications** tab, to grant the role permission to create new applications, click **Can Create Applications**.

b. Under Default Permissions, select the default permissions for this role: **View**, **Edit** or **Delete**.
c. Click **+ Add** to add a new custom permission and click **Done**.
The custom permission is added.
d. Choose **Custom** from the Permissions menu for the application (replacing the value of Inherited).
e. Click **Edit(All)**. Ensure that **Start Cisco ACI trouble shooting** is selected.
f. Click **OK.**



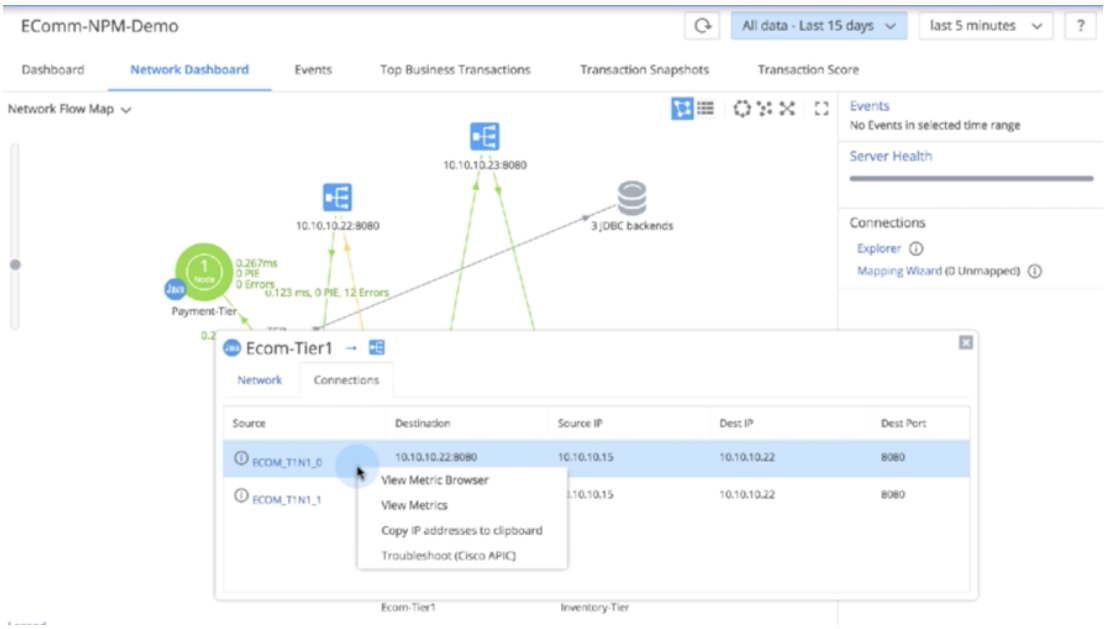
Custom Permission to Cross-Launch a Connection Issue in Cisco APIC Controller

As an AppDynamics administrator, you can now set customise the permissions for various roles to include permission to cross-launch a connection issue in Cisco APIC controller. See Set Custom Permission to Cross-Launch Cisco APIC controller for more information.

Launch Affected Connection in Cisco APIC

After collecting the connection issue information, cross-launch the connection in Cisco APIC controller as follows:

1. On the **Network flow Map**, click on the affected connection.
2. On the popup window, click the **Connections** tab.
3. Select the connection you want to troubleshoot.
4. Right-click on the selection and choose **Troubleshoot (Cisco APIC)**.



5. Enter Cisco APIC login credentials (first time only).
   The connection opens in Cisco APIC controller with the correct contracts (Visibility & Troubleshooting wizard) and the right set of endpoints.

   > If you are using the Cisco ACI Release 4.0, you need not enter the login credentials. The authentication happens automatically. You must enter the login credentials only if you are using a lower version of Cisco ACI.

6. Troubleshoot the issue in Cisco APIC.