



# Configuration API

This page describes the Configuration API methods you can use to read and modify selected Controller configuration settings programmatically. You can use it to script or automate tasks that must 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 as well as you can edit and import Controller configuration definition files.

## Create and Modify AppDynamics Users

Use this to create or modify 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
<code>user-name</code>	Query	user name	Yes
<code>user-id</code>	Query	user id	No for a create; yes for an update
<code>user-display-name</code>	Query	display name	Yes
<code>user-roles</code>	Query	comma-separated list of roles	No
<code>user-password</code>	Query	user password	Yes for a create; optional for an update
<code>user-email</code>	Query	user email	Yes

### Example

```
curl -X POST --user user1@customer1:your_password http://demo.appdynamics.com/controller/rest/users?user-name=user2\&user-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 transactions for monitoring by passing the `exclude` parameter to the business-transactions retrieval API described in the [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>

```

 Ensure that 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:your_password 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:your_password 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,
CONTROLLER_EVENT_UPLOAD_LIMIT_REACHED,CONTROLLER_RSD_UPLOAD_LIMIT_REACHED,
CONTROLLER_METRIC_REG_LIMIT_REACHED,CONTROLLER_METRIC_DATA_BUFFER_OVERFLOW,
CONTROLLER_ERROR_ADD_REG_LIMIT_REACHED,CONTROLLER_ASYNC_ADD_REG_LIMIT_REACHED,
AGENT_ADD_BLACKLIST_REG_LIMIT_REACHED,AGENT_METRIC_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,
CONTROLLER_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

Use this to get 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:your_password http://demo.appdynamics.com/controller/rest/configuration?name=metrics\.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

Use this to 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

Use this to 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
<code>application-component-node-ids</code>	Query	Comma-separated list of node IDs	Yes

## Example

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

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