Collect Transaction Analytics Data

On this page:

Related pages:

- Configuring Application Analytics
- Business Transaction and Log Correlation
- Data Collectors

Works with:

Java .net

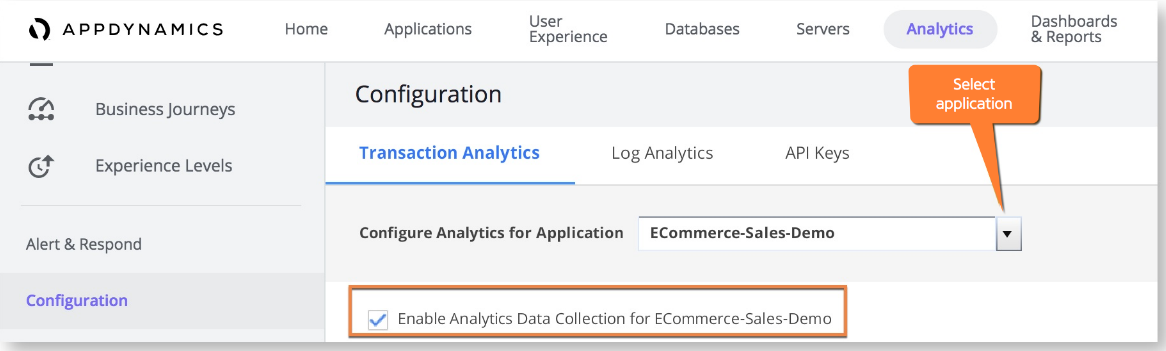
Make sure you have installed and configured the components described in Installing Agent-Side Components and, for on-premises, Custom Install and Events Service Deployment before attempting to configure Transaction Analytics.

Select the Application and the Business Transactions

Configuring Transaction Analytics consists of selecting the application and the specific business transactions that you want to analyze. You can also enable the collection of additional business data using Data Collectors. For Node.js apps, see Configure Analytics for Node.js.

To enable and configure Transaction Analytics for Java and .NET applications:

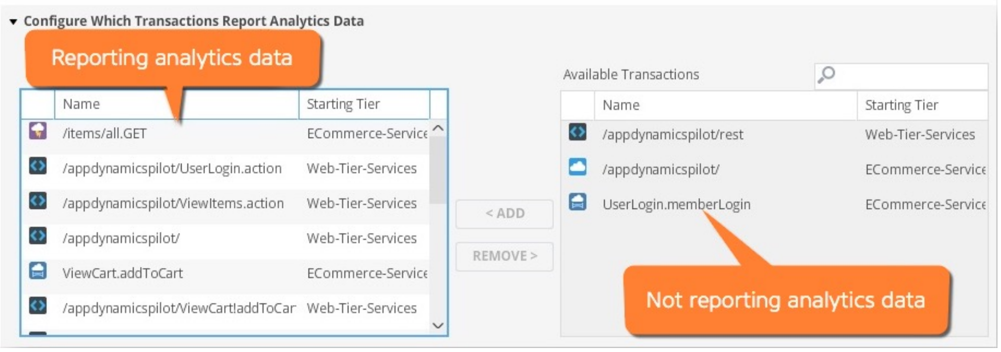
1. In the Controller UI, from the top navigation bar, select **Analytics > Configuration**.
2. From the **Transaction Analytics** tab, select the application.
3. Check the check box labeled **Enable Analytics Data Collection for <*selected_app*>**.



Configure Analytics for Application ECommerce-Sales-Demo

☑ Enable Analytics Data Collection for ECommerce-Sales-Demo

4. Specify which business transactions should report analytics data by adding available transactions from the right-hand list to the left-hand list of transaction reporting analytics data:



Configure Which Transactions Report Analytics Data

Reporting analytics data

	Name	Starting Tier
	/items/all.GET	ECommerce-Service
	/appdynamicspilot/UserLogin.action	Web-Tier-Services
	/appdynamicspilot/ViewItems.action	Web-Tier-Services
	/appdynamicspilot/	Web-Tier-Services
	ViewCart.addToCart	ECommerce-Service
	/appdynamicspilot/ViewCart!addToCar	Web-Tier-Services

Available Transactions

	Name	Starting Tier
	/appdynamicspilot/rest	Web-Tier-Services
	/appdynamicspilot/	ECommerce-Service
	UserLogin.memberLogin	ECommerce-Service

Not reporting analytics data

Even if you have pre-existing Data Collectors defined, you do not get Analytics data unless you confirm that the related business transactions are enabled for analytics.

> The Analytics Dynamic Service loads Transaction Analytics for enabled applications after your JVM instance runs for at least 2 minutes.

Configure Data Collectors

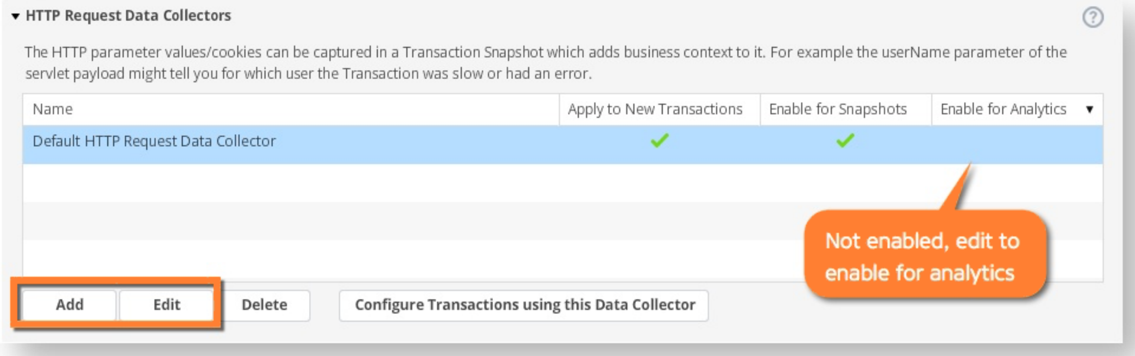
After you select the application and business transactions, you can optionally configure data collectors. The Configure Transaction Analytics page includes sections you can use to configure HTTP and method invocation data collectors, as follows.

HTTP Data Collectors

To collect HTTP request data, you can use the default HTTP Request Data Collector. You need to explicitly enable the collector for Analytics.

To configure HTTP Data Collectors for Analytics, use these steps:

1. In the Controller UI, select **Analytics > Configuration** from the top navigation bar.
2. From the **Transaction Analytics** tab, select the application of interest and scroll to the **HTTP Request Data Collector** section.
3. Select the HTTP Data Collector to enable and click **Edit** (or **Add** to create new collectors).



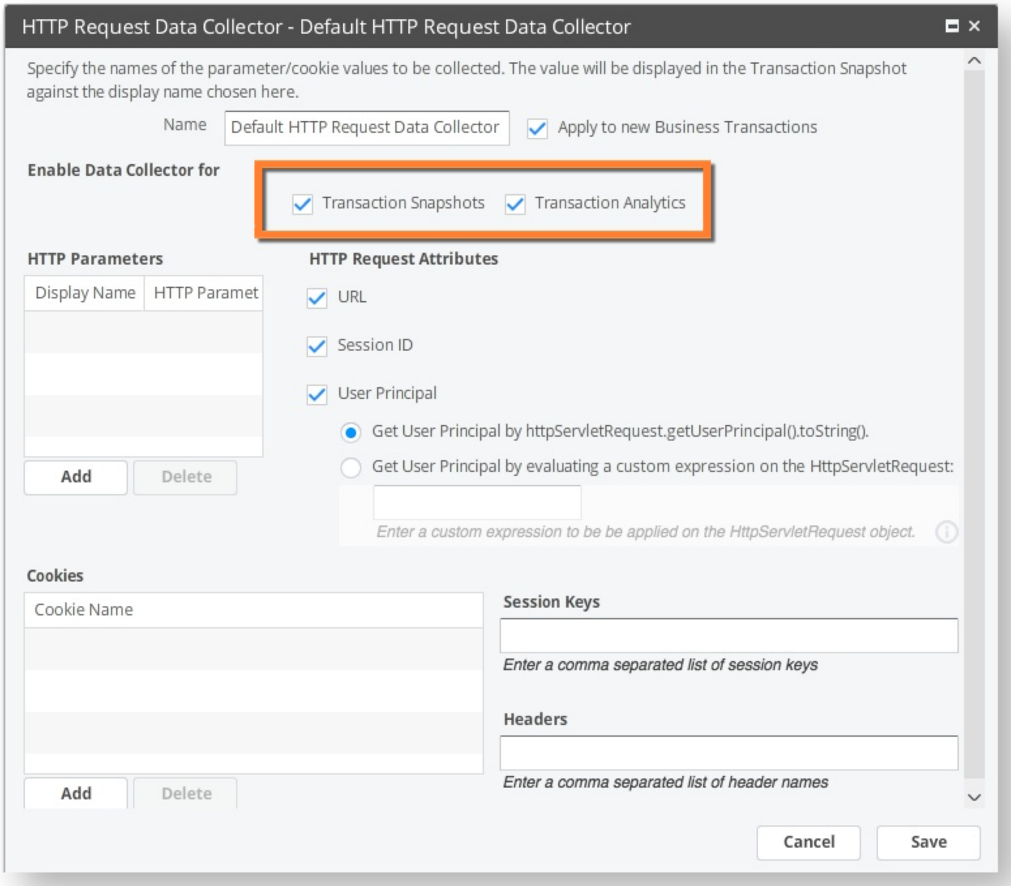
HTTP Request Data Collectors

The HTTP parameter values/cookies can be captured in a Transaction Snapshot which adds business context to it. For example the userName parameter of the servlet payload might tell you for which user the Transaction was slow or had an error.

Name	Apply to New Transactions	Enable for Snapshots	Enable for Analytics
Default HTTP Request Data Collector	✓	✓	

Not enabled, edit to enable for analytics

4. On the **HTTP Request Data Collector** page, confirm the data to collect, confirm **Transaction Analytics** is checked, and click **Save**.



HTTP Request Data Collector - Default HTTP Request Data Collector

Specify the names of the parameter/cookie values to be collected. The value will be displayed in the Transaction Snapshot against the display name chosen here.

Name: Default HTTP Request Data Collector ☑ Apply to new Business Transactions

Enable Data Collector for ☑ Transaction Snapshots ☑ Transaction Analytics

HTTP Parameters

Display Name	HTTP Paramet

HTTP Request Attributes

☑ URL

☑ Session ID

☑ User Principal

● Get User Principal by httpServletRequest.getUserPrincipal().toString().

○ Get User Principal by evaluating a custom expression on the HttpServletRequest:

Enter a custom expression to be be applied on the HttpServletRequest object.

Cookies

Cookie Name

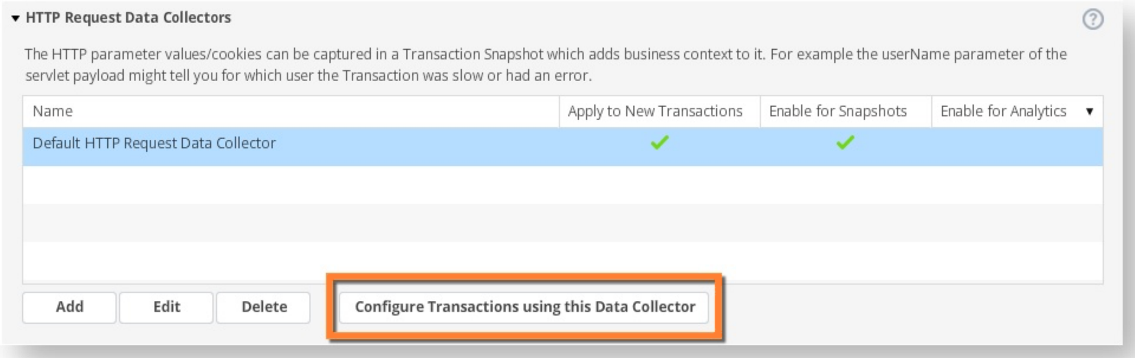
Session Keys

Enter a comma separated list of session keys

Headers

Enter a comma separated list of header names

5. Use **Configure Transactions Using this Data Collector** and confirm that the data collector is enabled on the appropriate business transactions.
   See Data Collectors for more information.



HTTP Request Data Collectors

The HTTP parameter values/cookies can be captured in a Transaction Snapshot which adds business context to it. For example the userName parameter of the servlet payload might tell you for which user the Transaction was slow or had an error.

Name	Apply to New Transactions	Enable for Snapshots	Enable for Analytics
Default HTTP Request Data Collector	✓	✓	

Method Invocation Data Collectors

You can also use the **Analytics > Configuration** page to enable existing or new Method Invocation Data Collectors. Open the **Method Invocation Data Collectors** panel. The process is essentially the same as described in Data Collectors.

Make sure that you:

- Check the **Transaction Analytics** check box to use this collector for Application Analytics
- Check the **Configure Transactions** popup to confirm you have enabled the right Business Transactions.
- Check the name of the MIDC does not exceed the 50 character limit.

> Changing the name of method invocation data collector will create a completely new field in transaction analytics. If you want to change the type of a data collector field, we recommend creating a new field instead to avoid conflicts with the old data.

SQL Data Collectors

Transaction snapshots capture SQL queries. SQL data collectors provide a way to extract business data from parameters used in the SQL statements for use in analytics. See Collect Business Data From SQL Calls.

Enable Transaction Analytics Data Type Conversion for Data Collectors

> Transaction analytics data type conversion requires controller 4.5.6+
>
> For Java deployments, transaction analytics data type conversion is available for Java agents version 4.5.2+

Data collectors determine field type (string, boolean, or number) at runtime. At times the data you collect may not be the type in which you want to use it. For example, you are not able to add metrics if any data collectors are field type string.

While field type cannot be changed, you can specify field type before runtime with *transaction analytics data type conversion*.

To enable transaction analytics data type conversion:

1. Log in to the administration console:
   `http:<controller-hostname>:8080/controller/admin.jsp or`
   `https:<controller-hostname>:45/controller/admin.jsp`
2. On the **Controller Settings** page, change the value of `analytics.type.conversion.enabled` from `false` to true.
3. Press Save.

Data type conversion only appears when enabled in HTTP and Method Invocation Data Collector configuration pages on your controller. If this page was opened while you enabled data type conversion, you will need to refresh your page to see changes.

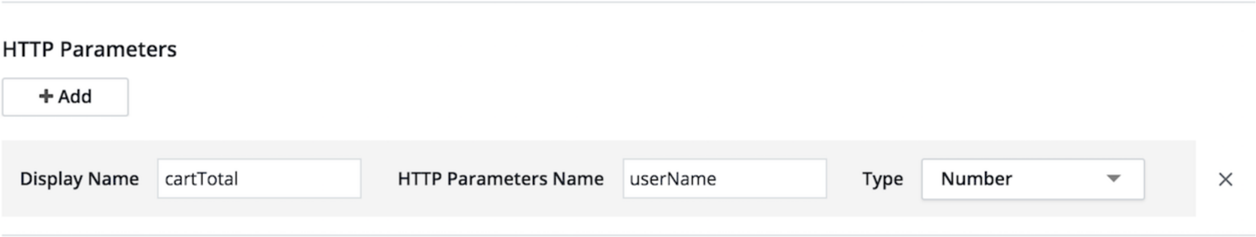
Data type conversion operates by the following rules:

- Available for new data collectors only
- Select type manually for each data collector
- Type cannot be changed once saved

Transaction Analytics Data Type Conversion in HTTP Data Collectors

To use transaction analytics data type conversion, create a new HTTP parameter. You can add to an existing HTTP Request Data Collector or create a new one.

The process is essentially the same as creating an HTTP parameter. However, now that you've enabled transaction analytics data type conversion, you will see "Type" field with a drop down menu, as seen in the screenshot below:



HTTP Parameters

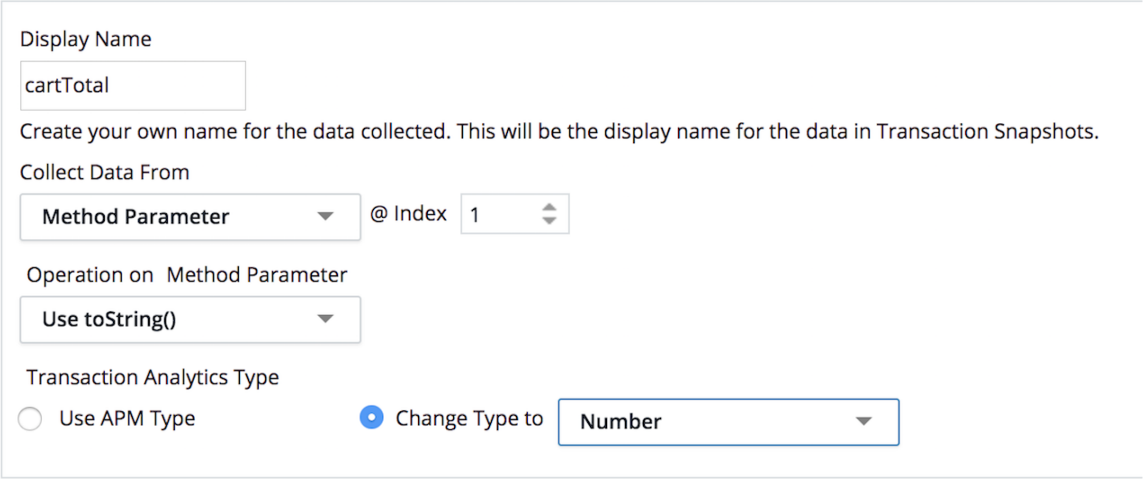
Display Name: cartTotal
HTTP Parameters Name: userName
Type: Number

Select the desired field type for your HTTP parameter. If you do not want to specify a type, leave this field blank. The default type for HTTP data collectors is `string`.

Transaction Analytics Data Type Conversion in Method Invocation Data Collectors

To use transaction analytics data type conversion, create a new Method Invocation Data Collector (MIDC). You can add to an existing MIDC or create a new one.

The process is essentially the same as creating an MIDC. However, now that you've enabled transaction analytics data type conversion, you will see "Change Type to" field with a drop down menu, as seen in the screenshot below:



Display Name: cartTotal

Create your own name for the data collected. This will be the display name for the data in Transaction Snapshots.

Collect Data From: Method Parameter @ Index 1

Operation on Method Parameter: Use toString()

Transaction Analytics Type

○ Use APM Type ● Change Type to: Number

Select the desired field type for your MIDC.

Transaction Analytics Type defaults to "Use APM Type," which determines type at runtime. If you do not want to specify a type, select "Use APM Type."

Naming requirements for transaction analytics data type conversion

Refer to the following naming requirements when using transaction analytics data type conversion:

- Display Names must be unique within each data collector. Names are case sensitive.
- You can repeat Display Names across different data collector configurations, as long the field type is also the same.
- If you delete a field from a data collector and add it again with a different field type, you must rename the data collector.

Changing field types

Field types cannot be changed once they have been saved to the data collector. To change the field type, we recommend to delete and recreate with a new name.

Configure Analytics for Node.js

Collect Default Node.js Transaction Data

To configure the Node.js Agent to send the default transaction data to the analytics agent, modify the require statement in your application. Add the following variables with the proper values to point to your Analytics Agent:

```
analytics: {
    host: <analyticsHostName>,
    port: <analyticsPort>  }
```

The Analytics Agent can be on the same host as the Node.js Agent, or on a different host. For more details on installing the Node.js Agent, see Install the Node.js Agent.

For example, your require statement should look similar to the following where your Analytics Agent is on localhost and listening on port 9090:

```
require("appdynamics").profile({
  controllerHostName: '<controller host name>',
  controllerPort: <controller port number>,
  controllerSslEnabled: false,  // Set to true if controllerPort is
SSL
  accountName: '<AppDynamics_account_name>',
  accountAccessKey: '<AppDynamics_account_key>', //required
  applicationName: 'your_app_name',
  tierName: 'choose_a_tier_name',
  nodeName: 'choose_a_node_name',
  analyticsHostName: 'host_name',
  analyticsPort: 'port_number',
  analyticsSslEnabled: false, // Set to true if controllerPort is SSL
  analytics: {
      host: 'localhost',
      port: 9090 }});
```

Node.js Data Collectors

You can also collect additional data from your Node.js business transactions by using the Node.js API. To configure transaction analytics data collectors for Node.js, see `txn.addAnalyticsData()` in the Node.js Agent API Reference.

Configuring Analytics for PHP Agent (New in 4.5.5)

1. To configure the PHP Agent to send the default transaction data to the analytics agent, modify the .ini file, depending on the operating system under which your PHP is installed. Add the following variables with the proper values to point to your Analytics Agent:

   ```
   agent.analyticsHostName = <analyticsHostName>
   agent.analyticsPort = <analyticsPort>
   ```

   The Analytics Agent can be on the same host as the PHP Agent, or on a different host. For more details on installing the PHP Agent, see Install the PHP Agent.

   For example, your .ini file should contain the following where your Analytics Agent is on localhost and listening on port 9090:

   ```
   agent.analyticsHostName = localhost
   agent.analyticsPort = 9090
   ```

2. HTTP/Method Invocation Data collectors for analytics must be enabled for both Transaction snapshots and Transaction Analytics for data to be reported from the PHP agent.
3. In order to collect method return value with an MIDC, the value must be assigned to a variable.
   `$ret = function()`
   If the return value is not stored in any variable then it will be seen as null in both snapshot and Analytics data.
4. Analytics data cannot currently be collected from php CLI programs.