API Gateway

Version 7.6.2

14 July 2020

Analytics User Guide

This documentation describes the following Axway software:

Axway API Gateway 7.6.2

Contents

Preface

This guide describes how to set up and how to use API Gateway Analytics. This tool enables you to record, monitor, and report on the history of message traffic between API Gateway instances and various services, remote hosts, and clients running in an API Gateway domain.

Who should read this guide

The intended audience for this guide includes system administrators, database administrators, API Gateway administrators, and policy developers.

How to use this guide

This guide should be used in conjunction with the other guides in the API Gateway documentation set. Before you begin, review this guide thoroughly. The following is a brief description of the contents:

- *Introduction to API Gateway Analytics* on page 10 - Gives a brief overview of API Gateway Analytics and outlines the steps required to set up and use this tool.
- *Install and configure a metrics database* on page 12 - Explains how to create and configure a database for monitoring in API Gateway Analytics.
- *Configure API Gateway with the metrics database* on page 23 - Explains how to configure an API Gateway instance and Node Manager to store metrics on historic traffic in the API Gateway Analytics metrics database.
- *Configure API Gateway Analytics* on page 18 - Explains how to update your configuration (for example, port, database connection, and user credentials) before starting API Gateway Analytics.
- *Launch API Gateway Analytics* on page 30 - Explains how to start API Gateway Analytics and how to change the default API Gateway Analytics user credentials and sample certificate.
- *Monitor traffic in API Gateway Analytics* on page 32: - Explains how to monitor your API Gateway domain and generate reports on historic API Gateway message traffic.
- *Configure scheduled report settings in Policy Studio* on page 41 - Explains how to schedule API Gateway Analytics reports to run on a regular basis, and to email the results to users in PDF format.
- *Purge the metrics database for API Gateway Analytics* on page 43 - Explains how to use the `dbpurger` command to purge old data and to archive data.

Related documentation

The AMPLIFY API Management solution enables you to create, publish, promote, and manage Application Programming Interfaces (APIs) in a secure and scalable environment. For more information, see the *AMPLIFY API Management Getting Started Guide*.

The following reference documents are also available on the Axway Documentation portal at https://docs.axway.com:

- *Supported Platforms*

  Lists the different operating systems, databases, browsers, and thick client platforms supported by each Axway product.

- *Interoperability Matrix*

  Provides product version and interoperability information for Axway products.

Support services

The Axway Global Support team provides worldwide 24 x 7 support for customers with active support agreements.

Email support@axway.com or visit Axway Support at https://support.axway.com.

See "Get help with API Gateway" in the *API Gateway Administrator Guide* for the information that you should be prepared to provide when you contact Axway Support.

Training services

Axway offers training across the globe, including on-site instructor-led classes and self-paced online learning. For details, go to: http://www.axway.com/support-services/training

Accessibility

Axway strives to create accessible products and documentation for users.

This documentation provides the following accessibility features:

Screen reader support

- Alternative text is provided for images whenever necessary.
- The PDF documents are tagged to provide a logical reading order.

Support for high contrast and accessible use of colors

- The documentation can be used in high-contrast mode.
- There is sufficient contrast between the text and the background color.
- The graphics have the right level of contrast and take into account the way color-blind people perceive colors.

Updates and revisions

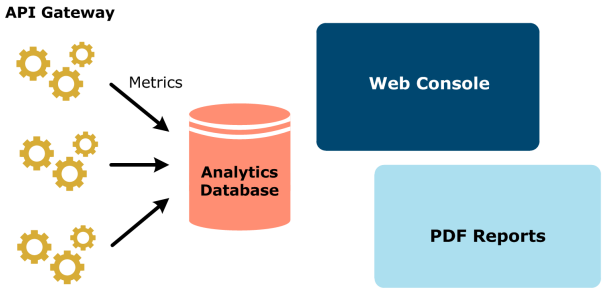
This guide includes the following documentation changes.

Changes in version 7.6.2

- This is a new guide that describes how to set up and use API Gateway Analytics to monitor and report on message traffic between API Gateway instances and services, remote hosts, and clients.

Introduction to API Gateway Analytics 1

API Gateway Analytics is a server runtime and web-based monitoring and reporting console that enables you to generate scheduled reports and analyze API use over time in multiple API Gateways across the domain.



API Gateway Analytics includes the following features:

- Web-based console that monitors and reports on all API Gateways in the domain (multiple API Gateways are shown on the left in the diagram)
- Reporting over an extended time period rather than immediate operational monitoring
- Analysis of what APIs are used, how often APIs are used, when APIs are used, and who is using APIs
- Scheduled reports in PDF format can be emailed to specific users

Install API Gateway Analytics

This guide assumes that you have already installed API Gateway Analytics. For more details, see the *API Gateway Installation Guide*.

For details on upgrading API Gateway Analytics from earlier versions, see the *API Gateway Upgrade Guide*.

Configure your API Gateway Analytics metrics database

Before starting API Gateway Analytics, you must perform the following steps:

1. Create the third-party JDBC database instance used to store metrics. For more details, see *Install and configure a metrics database* on page 12. Alternatively, if you already have an existing database, skip to the next step.
2. Update your API Gateway Analytics configuration with the database details using the `configureserver` script. For more details, see *Configure API Gateway Analytics* on page 18.
3. Configure the database tables using the `dbsetup` script. For more details, see *Install and configure a metrics database* on page 12.
4. Configure your API Gateway instance and Admin Node Manager to store metrics. For more details, see *Configure API Gateway with the metrics database* on page 23.

For more details on managing metrics, see *Purge the metrics database for API Gateway Analytics* on page 43.

Monitoring and reporting with API Gateway Analytics

When you have configured the metrics database and API Gateway Analytics, you can start monitoring your API traffic and generating reports in API Gateway Analytics.

For details, see the following:

- *Launch API Gateway Analytics* on page 30
- *Monitor traffic in API Gateway Analytics* on page 32
- *Configure scheduled report settings in Policy Studio* on page 41

Install and configure a metrics database 2

API Gateway stores and maintains monitoring and transaction data in a JDBC-compliant database, which can be read by API Gateway Analytics, API Manager, and third-party monitoring tools.

This topic describes how to create and configure a database for monitoring in API Gateway Analytics

.

Prerequisites

The prerequisites for setting up the database are as follows:

Install a third-party JDBC database

You must install a JDBC-compliant database to store the monitoring and transaction data. Axway provides setup scripts for the following databases:

- MySQL or MariaDB
- Microsoft SQL Server
- Oracle
- IBM DB2

For details on supported database versions, see the "System requirements" in the *API Gateway Installation Guide*. For details on how to install your chosen third-party JDBC database, see your database product documentation.

Note You must ensure that you have the correct credentials to execute the setup scripts and to access the database for operations on the tables created by the scripts.

Install API Gateway Analytics

You must install API Gateway Analytics to use it to view the monitoring data in the metrics database. For more details, see the *API Gateway Installation Guide.*.

Add third-party JDBC driver files

You must add the JDBC driver files for your chosen third-party database to your API Gateway and Policy Studio installations as appropriate.

Add JDBC drivers to API Gateway

To add the third-party JDBC driver files for your database to API Gateway, perform the following steps:

1. Add the binary files for your database driver as follows:
    - Add `.jar` files to `INSTALL_DIR/apigateway/ext/lib`
    - Add `.so` files to the `INSTALL_DIR/apigateway/platform/lib`
2. Restart API Gateway.

Add JDBC drivers to Policy Studio

To add third-party binaries to Policy Studio, perform the following steps:

1. Select **Window > Preferences > Runtime Dependencies** from the Policy Studio main menu.
2. Click **Add** to select a JAR file to add to the list of dependencies.
3. Click **Apply** when finished. A copy of the JAR file is added to the `plugins` directory in your Policy Studio installation.
4. Click **OK**.
5. Restart Policy Studio using the `policystudio -clean` command.

Add JDBC drivers to API Gateway Analytics

To add the third-party JDBC driver files for your database to API Gateway Analytics , perform the following steps:

1. Add the binary files for your database driver as follows:
    - Add `.jar` files to the `INSTALL_DIR/analytics/ext/lib` directory
    - Add `.dll` files to the `INSTALL_DIR\analytics\Win32\lib` directory
    - Add `.so` files to the `INSTALL_DIR/analytics/platform/lib` directory
2. Restart API Gateway Analytics.

Create the third-party database

API Gateway Analytics reads message metrics from a third-party JDBC database and display this information in a visual format to administrators. This is the same database in which API Gateway stores its message metrics and audit trail data. You must first create this database using the third-party database of your choice:

- MySQL or MariaDB
- Microsoft SQL Server
- Oracle
- IBM DB2

For details on how to do this, see the product documentation for your chosen third-party database. The following example shows creating a MySQL or MariaDB database:

```
mysql> CREATE DATABASE reports;
Query OK, 1 row affected (0.00 sec)
```

In this example, the metrics database is named `reports`, but you can use any appropriate name.

Set transaction isolation to READ COMMITTED

For all supported databases, to ensure atomicity and consistency, you must ensure that the transaction isolation level is set to `READ COMMITTED`. This setting is recommended whether you are installing for the first time or upgrading.

Note Read-committed transaction isolation mode is the default mode for Oracle, Microsoft SQL Server and IBM DB2, but not for MySQL or MariaDB. If you are using MySQL or MariaDB, you must change to read-committed transaction isolation mode after installation and before you start the server for the first time.

For more details, see the product documentation for your chosen third-party database.

Configure the database connection

When you have created the metrics database, you must update your API Gateway Analytics configuration with the database details using the `configureserver` script. For more details, see *Configure API Gateway Analytics* on page 18.

Set up the database tables

When you have created the metrics database and configured the database connection, the next step is to set up the database tables. For API Gateway Analytics monitoring run the `dbsetup` command from the following API Gateway directory:

```
INSTALL_DIR/analytics/posix/bin
```

The following example command shows setting up new database tables:

```
dbsetup
New database
Schema successfully upgraded to:002-leaf
```

Note When you specify command-line arguments to `dbsetup`, the script does not run interactively. You should run `dbsetup` without any options to create the database tables.

Specify options to dbsetup

You can specify the following options to the `dbsetup` command:

Option	Description
`-h, --help`	Displays help message and exits.
`-p PASSPHRASE, --passphrase=PASSPHRASE`	Specifies the configuration passphrase (blank for zero length).
`--dbname=DBNAME`	Specifies the database name (mutually exclusive with `--dburl`,`--dbuser`, and `--dbpass`).
`--dburl=DBURL`	Specifies the database URL.
`--dbuser=DBUSER`	Specifies the database user.
`--dbpass=DBPASS`	Specifies the database password. You must enclose passwords that contain special characters in single quotation marks. For example: `./dbsetup --dburl=mysql://127.0.0.1:3306/reports --dbuser=root --dbpass='AcmeCorp!23'`
`--reinstall`	Forces a reinstall of the database, dropping all data.

Option	Description
`--stop=STOP`	Stops the database upgrade after the named upgrade.

dbsetup examples

The following are some examples of using `dbsetup` command options.

Connect to a named database

You can use the `--dbname` option to connect to a named database connection configured under the **External Connections** node in the Policy Studio tree. For example:

```
dbsetup --dbname=Oracle
Current schema version:001-initial
Latest schema version:002-leaf
Schema successfully upgraded to:002-leaf
```

Connect to a database URL

You can use the `--dburl` option to manually connect to a database instance directly using a URL. For example:

```
dbsetup --dburl=jdbc:mysql://localhost/reports --dbuser=root --dbpass=admin
Current schema version:001-initial
Latest schema version:002-leaf
Schema successfully upgraded to:002-leaf
```

Install a database

You can also use the `--dburl` option to set up a newly created database instance where none already exists. For example:

```
dbsetup --dburl=jdbc:mysql://localhost/reports --dbuser=root --dbpass=admin
New database
Schema successfully upgraded to:002-leaf
```

Reinstall a database

You can use the `--reinstall` option to wipe and reinstall a database. For example:

```
dbsetup --dburl=jdbc:mysql://localhost/reports --dbuser=root --dbpass=admin --
reinstall
Re-installing database...
Schema successfully upgraded to:002-leaf
```

SQL database schema scripts

As an alternative to using the `dbsetup` command, API Gateway also provides separate SQL schema scripts to set up the database tables for each of the supported databases. However, these scripts set up new tables only, and do not perform any upgrades of existing tables. These scripts are provided in the `INSTALL_DIR/analytics/system/conf/sql` directory in the following sub-directories:

- `/mysql`
- `/mssql`
- `/oracle`
- `/db2`

Note The scripts in the `/mysql` folder apply to both MySQL and MariaDB.

You can run the SQL commands in the `analytics.sql` file in the appropriate directory for your database. The following example shows creating the tables for a MySQL database:

```
mysql> \. INSTALL_DIR/analytics/system/conf/sql/mysql/analytics.sql
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected, 1 warning (0.00 sec)
...
```

Configure API Gateway Analytics 3

This topic describes how to update an API Gateway Analytics configuration (for example, the API Gateway Analytics port, database connection, and user credentials) before starting API Gateway Analytics. You can use the `configureserver` script (recommended) to guide you through all the required steps, or you can use Policy Studio to configure the API Gateway Analytics configuration file.

Prerequisites

The prerequisites for configuring API Gateway Analytics are as follows:

Install API Gateway

Because API Gateway Analytics reports on transactions processed by API Gateway in real time, you must ensure that API Gateway is installed. For more details, see the *API Gateway Installation Guide*.

To view API Gateway metrics in API Gateway Analytics, you must also enable the recording of metrics. For more details, see *Configure API Gateway with the metrics database* on page 23.

Install API Gateway Analytics

You must ensure that API Gateway Analytics is installed. For more details, see the *API Gateway Installation Guide*.

Configure a database

You must ensure that a JDBC-compliant database is installed to store the API Gateway monitoring and transaction data. For more details, see *Install and configure a metrics database* on page 12.

Update API Gateway Analytics configuration

By default, API Gateway Analytics is configured to read message metrics from a MySQL database stored on the local machine. You can use the `configureserver` command to configure API Gateway Analytics to use an alternative database, change the user credentials on the default database connection, or use a different listening port.

Update configuration on the command line

Perform the following steps to run `configureserver` in interactive mode:

1. Change to the following directory:

```
INSTALL_DIR/analytics/posix/bin
```

2. Run the `configureserver` command.
3. Enter the port on which the API Gateway Analytics server will listen. Defaults to `8040`. If you have another process already using this port on the machine on which API Gateway Analytics is installed, configure API Gateway Analytics to listen on a different port.
4. Enter the database connection URL. Defaults to `jdbc:mysql://127.0.0.1:3306/reports`.

   The following table lists examples of connection URLs for the supported databases, where `reports` is the name of the database and `DB_HOST` is the IP address or host name of the machine on which the database is running:

Database	Example connection URL
Oracle	`jdbc:oracle:thin:@DB_HOST:1521:reports`
Microsoft SQL Server	`jdbc:sqlserver://DB_HOST:1433;DatabaseName=reports;integratedSecurity=false;`
MySQL/Maria DB	`jdbc:mysql://DB_HOST:3306/reports`
IBM DB2	`jdbc:db2://DB_HOST:50000/reports`

Note You can use the `jdbc:mysql://DB_HOST:3306/reports` URL with both MySQL and MariaDB databases.

5. Enter the database user name. Defaults to `root`.
6. Enter the database password.
7. Enter whether API Gateway Analytics generates PDF-based reports. Defaults to `N`, which means that PDF reports are not generated. When set to `Y`, API Gateway Analytics generates PDF reports that include the same metrics displayed in the API Gateway Analytics window (for example, number of client requests, requests per service, and so on). For more details on generated PDF reports, see *Configure scheduled report settings in Policy Studio* on page 41.
8. Enter the user name to connect to the API Gateway Analytics process that generates PDF reports. Defaults to an administrator user.

Note This is not the operating system user. This is the user that connects to the API Gateway Analytics web server process, which generates the PDF reports. You can add new users under the **Environment Configuration > Users and Groups** node in Policy Studio.

9. Enter the password to connect to the API Gateway Analytics process that generates PDF reports.
10. Enter the directory to which generated PDF reports are output (for example, `c:\reports`).
11. Enter whether to send generated PDF reports to email recipients. You will require an SMTP account with which to send the reports. Defaults to `N`.

The following command shows some example output in interactive mode:

```
/opt/Axway-7.6.2/analytics/posix/bin>configureserver
Connecting to configuration at : federated:file:////opt/Axway-
7.6.2/analytics/conf/fed/
configs.xml

Listening port [8040]:
Configuring Database: Default Database Connection
Database URL [jdbc:mysql://127.0.0.1:3306/reports]:
Database user name [root]:
Database password []: *****
Enable report generation (Y, N) [N]: y
Report generation process connects as user name [admin]:
Report generation process connects using password []: ********
Report output directory []: c:\reports
Email reports (Y, N) [N]: y
Default email recipient []: joe@example.com
Email from []: apigateway@axway.com
Choose SMTP connection type:
       0) None
       1) SSL
       2) TLS/SSL
Choice [0]:
SMTP host []: localhost
SMTP port [25]:
SMTP user name []: jbloggs
SMTP password []: *********
Delete report file after emailing (Y, N) [Y]:
Press enter to exit...
```

Update configuration using command-line options

You can also run the `configureserver` command with various options (`--port`, `--dburl`, `--emailfrom`, `--emailto`, `--smtphost`, and so on). For example, the following command configures the database connection without emailing reports:

```
configureserver --dburl=jdbc:mysql://127.0.0.1:3306/reports
```

```
--dbuser=root --dbpass=changeme --no-email
```

The following command specifies to email reports and the associated SMTP settings:

```
configureserver --dburl=jdbc:mysql://127.0.0.1:3306/reports
--dbuser=root --dbpass=changeme
--email --emailto=joe@example.com --emailfrom=apigateway@axway.com
--smtptype=NONE --smtphost=192.168.0.174 --smtpport=25
--smtpuser=jbloggs --smtppass=changeme
--generate --gpass=changeme --gtemp=c:\reports
```

For descriptions of all available options, enter the `configureserver --help` command.

Note The `configureserver` script does not permit the euro character (`€`) when specifying username and password options. However, you can specify the pound (`£`) and dollar (`$`) characters instead.

Update configuration in Policy Studio

The recommended way to configure API Gateway Analytics is using the `configureserver` command, which guides you through the required settings. However, you can also use the Policy Studio to configure specific settings in your API Gateway Analytics configuration file. For example, to configure the `reports` database, perform the following steps:

1. In your Policy Studio installation directory, run the `policystudio` command.

1. When Policy Studio starts up, select **File > New Project**.
2. In the New Project dialog, enter a name for the project and click **Next**.
3. Select **From existing configuration** and click **Next**.
4. Browse to the directory containing the API Gateway Analytics configuration file (`configs.xml`), for example:

```
INSTALL_DIR/analytics/conf/fed/
```

5. Select **Environment Configuration > External Connections** in the Policy Studio tree, and expand the **Database Connections** tree node.
6. Right-click the **Default Database Connection** tree node, and select **Edit**.
7. The Database Connection dialog enables you to configure the database connection details. By default, the connection is configured to read metrics data from the `reports` database. Edit the details for the **Default Database Connection** on this dialog.

   For example, you should enter a non-default database user name and password. To connect to a database other than the default local database, right-click **Database Connections** in the tree, and select **Add a Database Connection**. For more details, see the *API Gateway Policy Developer Guide*.

Tip You can verify that your database connection is configured correctly by clicking the **Test Connection** button on the Configure Database Connection dialog.

Ensure that metrics have been enabled for your API Gateway host

You must use the `managedomain` tool to enable writing of metrics from the API Gateway instances on your host to the metrics database. This enables the Node Manager to process event logs from your API Gateway instances, and to write metrics data to the metrics database.

The following example uses the interactive `managedomain --menu` command:

```
Select option: 2
Select a host:
1) LinuxMint01
2) Enter host name
Enter selection from 1-2 [2]: 1
Hit enter to continue...
Enter a new host name [LinuxMint01]:
Enter a new Node Manager name [Node Manager on LinuxMint01]:
Enter a new Node Manager port [8090]:
There is only one Node Manager in this domain so it must remain as an Admin Node
Manager
Do you want to create an init.d script for this Node Manager [n]:
Do you want to reset the passphrase for the Node Manager on this host ? [n]:
Do you wish to edit metrics configuration (y or n) ? [n]: y
Do you wish to enable metrics (y or n) ? [y]: y
Enter metrics database URL [jdbc:mysql://127.0.0.1:3306/reports]:
Enter metrics database username [root]:
Enter metrics database plaintext password [*******]:
Testing Database connectivity for : jdbc:mysql://127.0.0.1:3306/reports, user : root
Metrics database connectivity succeeded
Metrics generation enabled. All other specified metrics settings updated.
Metrics settings updated successfully. Please reboot Node Manager on completion of
this program.
Completed successfully.
```

For more details, see *Configure API Gateway with the metrics database* on page 23.

Configure API Gateway with the metrics database

4

This topic explains how to configure an API Gateway instance and Node Manager to store metrics on historic traffic in a relational database used to store metrics. For example, you can configure monitoring in API Gateway Analytics or API Manager to view data stored in the metrics database, or write custom SQL queries to retrieve metrics data as required.

Note This topic explains how to configure API Gateway with a metrics database. This topic assumes that you have already created your metrics database using the steps described in *Install and configure a metrics database* on page 12.

This topic explains how to perform the following tasks:

- Use Policy Studio to configure an API Gateway instance to write audit logging events to the metrics database, and to write metrics data to the transaction event log.
- Use the `managedomain` command to configure the Node Manager to process event logs and update the metrics database.

API Gateway metrics data streams

The following data streams are used to populate the metrics database:

- **Transaction and system data**: Transaction data includes clients, services, remote hosts, and protocols. System data includes CPU, memory and disk usage, and SLA breaches. The API Gateway writes this data to a transaction event log, with a new log file automatically created every 5 minutes. The Node Manager parses completed event logs and updates the metrics database.
- **Transaction audit log events**: These are written directly to the metrics database by the API Gateway instance.

Connect to the API Gateway in Policy Studio

To connect to the API Gateway in Policy Studio, perform the following steps:

1. Ensure the Admin Node Manager and API Gateway are running.
2. Create a new project or open an existing project based on a running API Gateway instance. For more details, see the *API Gateway Policy Developer Guide*.

Configure the metrics database connection

To configure the API Gateway connection to the metrics database, perform the following steps:

1. Expand the **Environment Configuration** > **External Connections** > **Database Connections** node in the Policy Studio tree.
2. Right-click the **Default Database Connection** tree node, and select **Edit**.
3. Configure the database connection to point to your metrics database. For details on connection settings, see the *API Gateway Policy Developer Guide*.
4. Verify that your database connection is configured correctly by clicking the **Test Connection** button on the **Configure Database Connection** dialog.

Tip You can troubleshoot your database connection by viewing the contents of your server `.trc` file in the `INSTALL_DIR/apigateway/trace` directory. For more details, see "Configure API Gateway diagnostic trace" in the *API Gateway Administrator Guide*.

Configure transaction audit logging to the metrics database

To configure the API Gateway instance to write transaction audit log data to the metrics database, perform the following steps:

1. In the Policy Studio tree, select the **Server Settings** node, and select **Logging** > **Transaction Audit Log** in the window on the right.
2. Select the **Database** tab, and select **Enable logging to database**.
3. Select the **Default Database Connection** from the drop-down list if appropriate. Alternatively, select a database connection that you have configured. You must ensure that your database connection points to your metrics database.

For more details, see "Configure API Gateway logging and events" in the *API Gateway Administrator Guide*.

Tip To write the content of message transactions to the database, you must also configure the **Log Message Payload** filter in your policies (for example, at the start and end of the policy). For more details, see the *API Gateway Policy Developer Guide*.

Configure the API Gateway to write to the transaction event log

To configure the API Gateway instance to write transaction data to the transaction event log, perform the following steps:

1. In the Policy Studio tree, select the **Environment Configuration** > **Server Settings** node, and select **Logging** > **Transaction Event Log** in the window on the right.
2. Ensure **Writing to Transaction Event Log** is selected.
3. To enable monitoring of protocol and remote host metrics, select the **Monitoring** > **Traffic Monitor** node, and ensure the following settings are selected:
    - **Enable Traffic Monitor**
    - **Record inbound transactions**
    - **Record outbound transactions**

For more details, see "Configure API Gateway logging and events" in the *API Gateway Administrator Guide*.

Deploy the updated configuration to the API Gateway

You must deploy these configuration changes to the API Gateway. Click the **Deploy** button in the toolbar, or press F6.

The API Gateway now sends transaction audit logging to the metrics database, and writes transaction data to the transaction event log. The final step is to configure the Node Manager to read the transaction event logs and write system and transaction metrics to the metrics database.

Configure the Node Manager to process event logs and update the metrics database

If you have not already done so, you must use the `managedomain` tool to enable the Node Manager to process event logs from your API Gateway host, and to write metrics data to the metrics database.

All API Gateway instances running on the host node generate transaction event log files. These files are all written to the same folder, and are collectively processed and aggregated by the Node Manager on the host, and then written to the metrics database. The metrics database provides the data for the graphical charts in the monitoring views in API Gateway Analytics and API Manager.

Note The Node Manager on each host in the domain must be configured to write metrics data to the same database that API Gateway Analytics reads from. The API Gateway can write to the same database for transaction audit logging if required.

Use the managedomain interactive menu

You can enable metrics using the interactive `managedomain --menu` command. The following shows an example:

```
Select option:2
Select a host:
   1) LinuxMint01
   2) Enter host nameEnter selection from 1-2 [2]:1
Enter a new host name [LinuxMint01]:
Enter a new Node Manager name [Node Manager on LinuxMint01]:
Enter a new Node Manager port [8090]:
There is only one Node Manager in this domain so it must remain as an Admin Node
Manager
Do you want to create an init.d script for this Node Manager [n]:
Do you want to reset the passphrase for the Node Manager on this host ? [n]:
Do you wish to edit metrics configuration (y or n) ? [n]:y
Do you wish to enable metrics (y or n) ? [y]:y
Enter metrics database URL [jdbc:mysql://127.0.0.1:3306/reports]:
Enter metrics database username [root]:
Enter metrics database plaintext password [*******]:
Testing Database connectivity for :jdbc:mysql://127.0.0.1:3306/reports, user :root
Metrics database connectivity succeeded
Metrics generation enabled. All other specified metrics settings updated.
Metrics settings updated successfully. Please reboot Node Manager on completion of
this program.
Completed successfully.
Hit enter to continue...
```

Use the managedomain command options

Alternatively, you can use `managedomain` command options to enable metrics when initializing a host, adding a host other than the Admin Node Manager, or editing a host.

The following example shows enabling metrics when initializing a host machine:

```
./managedomain --initialize --metrics_enabled y
  --metrics_dburl jdbc:mysql://127.0.0.1:3306/reports
  --metrics_dbuser root --metrics_dbpass MY_DB_PWD
```

The following example shows enabling metrics when adding a host machine other than the Admin Node Manager:

```
./managedomain --add --anm_host MY_HOSTNAME --nm_entitystore_passphrase MY_CONFIG_PWD
  --metrics_enabled y --metrics_dbuser root --metrics_dbpass MY_DB_PWD
```

```
  --metrics_dburl jdbc:mysql://1.2.3.4:3306/reports --nm_name MY_NODE_MNGR --port
8055
```

The following example shows enabling metrics when editing a host machine in the domain:

```
./managedomain --edit_host --nm_entitystore_passphrase bonjour
   --metrics_enabled y --metrics_dburl jdbc:mysql://127.0.0.1:3306/reports
   --metrics_dbuser root --metrics_dbpass MY_DB_PWD
```

The `managedomain` metrics options are described as follows:

Option	Description
`--nm_entitystore_pass`	Specifies the encryption passphrase used to access the API Gateway instance configuration. If no passphrase has been set, omit this argument. For more details, see *API Gateway Administrator Guide*.
`--metrics_enabled`	Specifies whether writing of metrics data is enabled. Enter `y` or `n`.
`--metrics_dburl`	Specifies the JDBC URL for the metrics database (for example, `jdbc:mysql://127.0.0.1:3306/reports`).
`--metrics_dbuser`	Specifies the metrics database user (for example, `root` or metrics DB user name).
`--metrics_dbpass`	Specifies the password for the metrics database user.

Note When the `managedomain` command has finished, you must restart the Node Manager.

For more details on `managedomain`, see "Configure an API Gateway domain" in the *API Gateway Administrator Guide*.

Configure additional options for event log processing in the Node Manager

The parameters described in this section specify how transaction event logs are processed in the Node Manager. You can configure these optional settings by editing the Node Manager configuration using the `esexplorer` tool.

For example, perform the following steps:

1. Change to the following directory:
   `INSTALL_DIR/apigateway/posix/bin`
2. Enter the `esexplorer` command.
3. Select **Store** > **Connect**.
4. Browse to `INSTALL_DIR/apigateway/conf/fed/configs.xml`.
5. Select **System Components** > **Metrics Generation Configuration** in the tree on the left.
6. Configure the appropriate fields in the window on the right:

Option	Description
`sourceEventLogDir`	Specifies the folder in which the Node Manager looks for event log files. This should match the API Gateway transaction event log directory set in Policy Studio (see "Configure API Gateway logging and events" in the *API Gateway Administrator Guide*). Defaults to `${environment.VDISTDIR}/events`.
`retainProcessedEventLogs`	Specifies whether processed event logs should be deleted or retained in a separate directory. By default, event logs are deleted when their contents are written to the metrics database. Logs can be retained if they are needed for audit purposes or as input to a custom analytics process. Defaults to `false`.
`processedEventLogDir`	When `retainProcessedEventLogs` is `true`, specifies the directory to which event files are moved after being processed by the Node Manager. Defaults to `${environment.VDISTDIR}/events/processed`.
`dirSizeMb`	If `retainProcessedEventLogs` is `true`, specifies the maximum size of the `processedEventLogDir`. When the configured size is reached, the oldest log files in the directory are deleted. Defaults to 1024 MB.
`processCustomMessageAttributes`	Specifies whether message attributes contained in the transaction event log, are written to the database `transaction_data` table. Defaults to `true`. For more details, see "Configure API Gateway logging and events" in the *API Gateway Administrator Guide*.

Option	Description
`processCustomMetrics`	Specifies whether custom metrics generated by the API Gateway Java Metrics API and written to the transaction event log are written to the database. Defaults to `true`. For more details, see the following:

Note When making changes using `esexplorer` , ensure that you open the latest configuration. For example, you could overwrite changes made using `managedomain` if an old version of the configuration was loaded into `esexplorer` and then updated.

7. Stop and restart the Node Manager after editing its configuration using `esexplorer`.

Launch API Gateway Analytics 5

This topic explains how to launch the API Gateway Analytics web console used to monitor your API Gateway domain. It also explains how to change the default API Gateway Analytics user credentials and sample certificate for TLS.

This topic assumes that you have already performed the steps in *Configure API Gateway with the metrics database* on page 23.

Start API Gateway Analytics

To launch API Gateway Analytics, perform the following steps:

1. Start the API Gateway Analytics server using the `analytics` script in the `/bin` directory of your API Gateway Analytics installation.
2. Using the default port, connect to the API Gateway Analytics interface in a browser at the following URL:

```
https://HOST:8040/
```

   `HOST` points to the IP address or hostname of the machine on which API Gateway Analytics is installed.

3. Log in using the username and password that you selected when installing API Gateway Analytics. See also *Change the default API Gateway Analytics credentials* on page 31.

Start API Gateway Analytics as a service

You can also run the API Gateway Analytics server as a service by creating a script. A sample script and *ReadMe* is provided in the following directory:

```
INSTALL_DIR/analytics/posix/samples/etc/init.d/
```

Note If you change to another metrics database that has a different set of remote hosts or clients configured, you must restart both API Gateway and API Gateway Analytics.

Change the default API Gateway Analytics credentials

You can change the default API Gateway Analytics user credentials by editing your API Gateway Analytics configuration in Policy Studio under **Environment Configuration** > **Users and Groups**.

The default `admin` user is local to API Gateway Analytics. Updating these credentials does not affect access to other Axway products.

Replace the default sample certificate for API Gateway Analytics

The default `Samples Test Certificate` certificate used for SSL/TLS is self-signed by Axway. You can replace this default sample certificate by editing the API Gateway Analytics configuration in Policy Studio:

1. Create a new project from an existing configuration, and select the configuration in `INSTALL_DIR/analytics/conf/fed`.
2. Select **Environment Configuration** > **Certificates and Keys**, and add your own user-signed certificate.
3. Select **Environment Configuration** > **Listeners** > **Axway Analytics** > **Management Services** > **Ports**, and edit the **Reporter HTTP Interface**.
4. On the **Network** tab, click **X.509 Certificate** to change the SSL certificate.
5. Save the project, and copy your Policy Studio project files back to your API Gateway Analytics installation. For example, copy the `.xml` files in `users/apiprojects/my-project` to `INSTALL_DIR/analytics/conf/fed`.

Further information

For more details, see *Monitor traffic in API Gateway Analytics* on page 32

Monitor traffic in API Gateway Analytics 6

API Gateway Analytics monitors, records, and reports on the history of message traffic between API Gateway instances and various services, remote hosts, and clients running in an API Gateway domain.

You can use API Gateway Analytics to monitor traffic and perform root cause analysis at the level of the domain, API Gateway instance, service, remote host, and client. You can also filter the display based on any selected time period. For example, this defaults to the last 7 days, but you can specify any date range.

Tip API Gateway Analytics produces reports based on metrics stored by API Gateway when processing messages. To produce a graph showing the number of connections made by API Gateway to a service, you must first configure a policy that routes messages to that service. When this policy is configured, send messages through the policy so they are routed to the target service.

For details on configuring policies, see the *API Gateway Policy Developer Guide*.

Prerequisites

This topic assumes that you have already performed the steps in:

- *Configure API Gateway with the metrics database* on page 23
- *Launch API Gateway Analytics* on page 30

Monitor the API Gateway system

The API Gateway Analytics **System** view includes the following tabs:

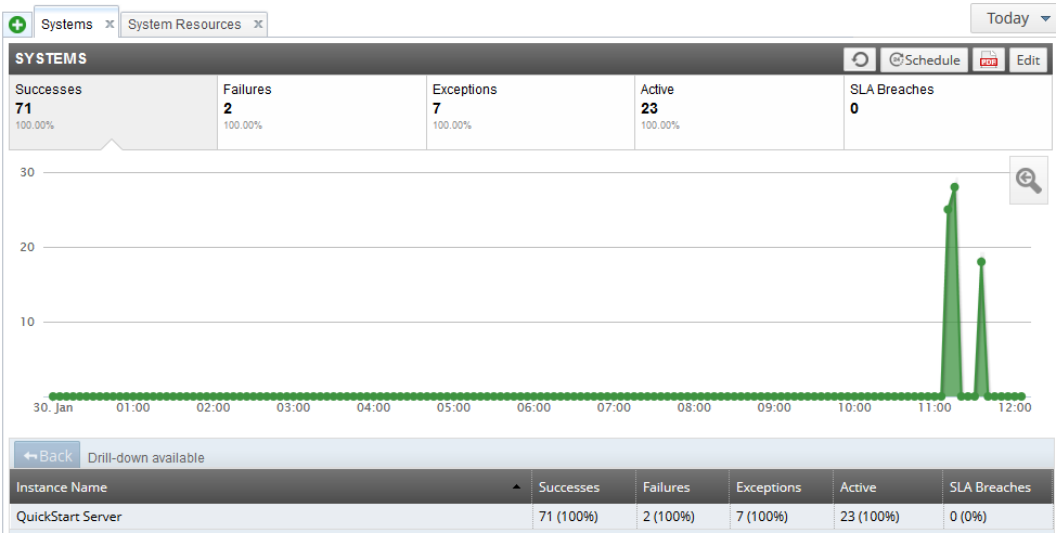
- **Systems**
- **System Resources**

Monitor system-level metrics

On the **Systems** tab, click a panel in the **SYSTEMS** section at the top to display graph for the selected system-level metrics below. For example, the available metrics include the following:

- **Successes**: The number of successful messages processed in the domain (that generated a success in an API Gateway policy).
- **Failures**: The number of blocked messages processed in the domain that generated a failure in an API Gateway policy.
- **Exceptions**: The number of blocked messages in the domain that generated an exception in an API Gateway policy.
- **Active**: The uptime of API Gateway instances.
- **SLA breaches**: The number of Service Level Agreement (SLA) breaches in the domain.

The following example shows messages successfully sent displayed in a simple domain with a single API Gateway instance:



The table at the bottom shows all the API Gateway instances that are sending monitored traffic to protected services, clients, and remote hosts in your domain. You can click an API Gateway instance in the table to drill down and view graphs for the selected instance. Click **Back** on the left to return to the **ALL SYSTEMS** view.

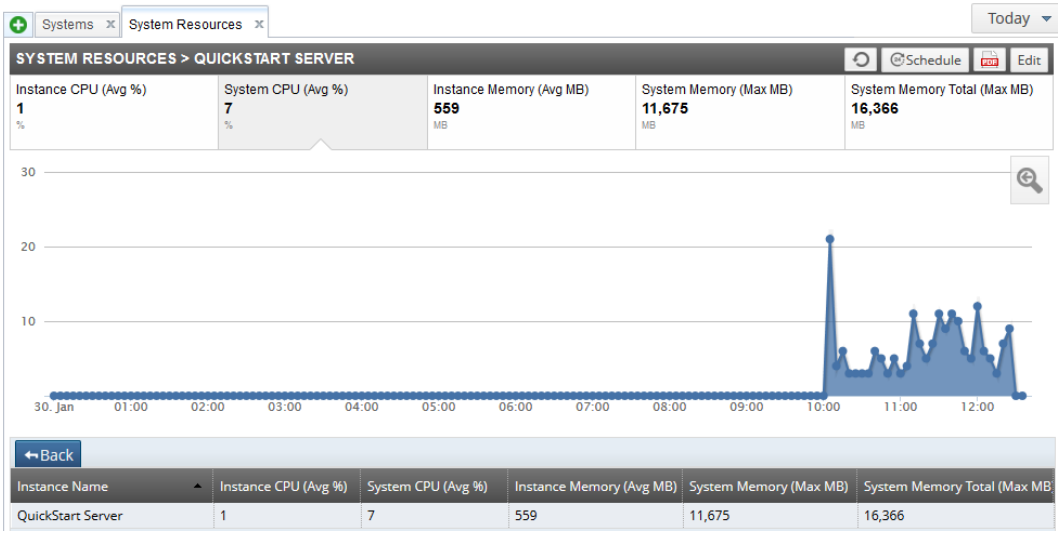
Monitor system resources

On the **System Resources** tab, click a panel in the **SYSTEM RESOURCES** section at the top to display graph for the selected system resource metrics below. For example, the available metrics include the following:

- **Instance CPU (Avg %)**: Average amount of CPU used by the API Gateway instance.
- **System CPU (Avg %)**: Average amount of CPU used on the host machine running the API Gateway.
- **Instance Memory Used (Avg MB)**: Average amount of memory used by the API Gateway instance.

- **System Memory (Max MB)**: Maximum amount of memory used on the machine hosting the API Gateway. This includes memory used by the API Gateway and all other processes running on the machine.
- **System Memory Total (Max MB)**: Total amount of available memory on the machine hosting the API Gateway.

The following example shows the average system CPU displayed in a simple domain with one API Gateway instance:



The table at the bottom shows all API Gateway instances that are sending monitored traffic to protected services, clients, and remote hosts in your domain. You can click an API Gateway instance in the table to drill down and view graphs for the selected instance. Click **Back** on the left to return to the **SYSTEM RESOURCES** view.

Monitor API services, methods, and clients

The **API Services** view shows metrics for services that are virtualized by API Gateway instances in your domain. For more details on virtualizing services, see the *API Gateway Policy Developer Guide*.

The **API Services** view includes the following tabs:

- **Load Balance**: Metrics for the number of messages and processing times.
- **Clients (Service of)**: Metrics for the number of messages, successes, and failures.
- **Clients**: Metrics for the number of messages, successes, failures, and exceptions.
- **API Services (Methods of)**: Metrics for the number of messages, successes, failures, and exceptions.
- **API Services (Clients of)**: Metrics for the number of messages, successes, failures, and exceptions.

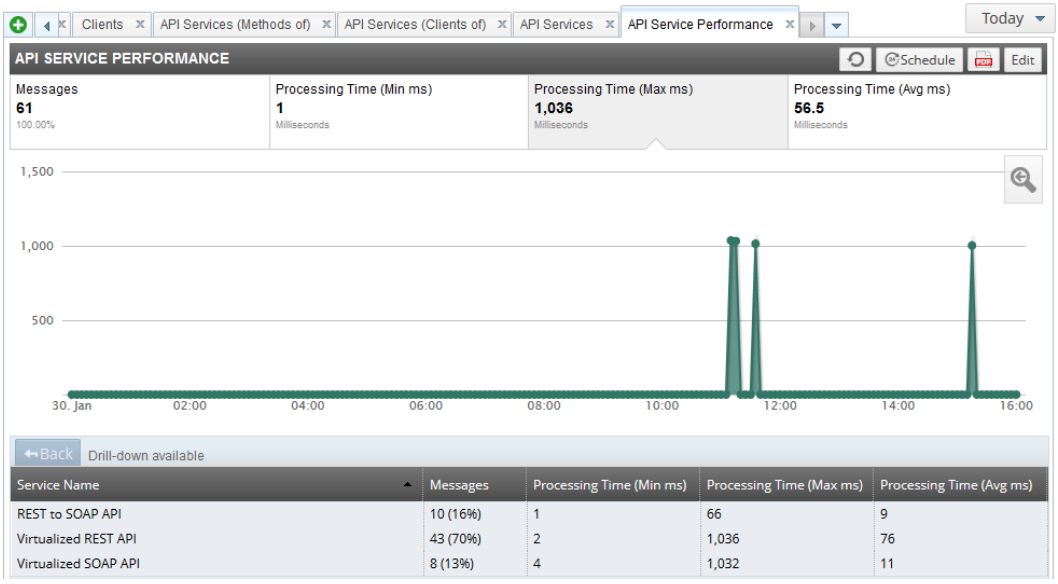
- **API Services**: Metrics for the number of messages, successes, failures, and exceptions.
- **API Service Performance**: Metrics for number of messages and processing times.

Example: API service performance

For example, In the **API Services** > **API Service Performance** tab, click a panel in the **API SERVICE PERFORMANCE** section at the top to display a graph for the selected service-level metric below. For example, the available metrics include the following:

- **Messages**: The number of API service messages processed in the API Gateway domain.
- **Processing Time (Min ms)**: The minimum time taken to process a message, including all calls to remote servers.
- **Processing Time (Max ms)**: The maximum time taken to process a message, including all calls to remote servers.
- **Processing Time (Avg ms)**: The average time taken to process a message, including all calls to remote servers.

The following example shows the maximum processing time in a simple domain with multiple API services:



The table at the bottom shows all services protected by API Gateway instances in your domain. You can click a service in the table to drill down and view graphs for the selected service. Click **Back** on the left to return to the **API SERVICE PERFORMANCE** view.

Note A service must first have been sent a message before it is displayed in the **API Services** view.

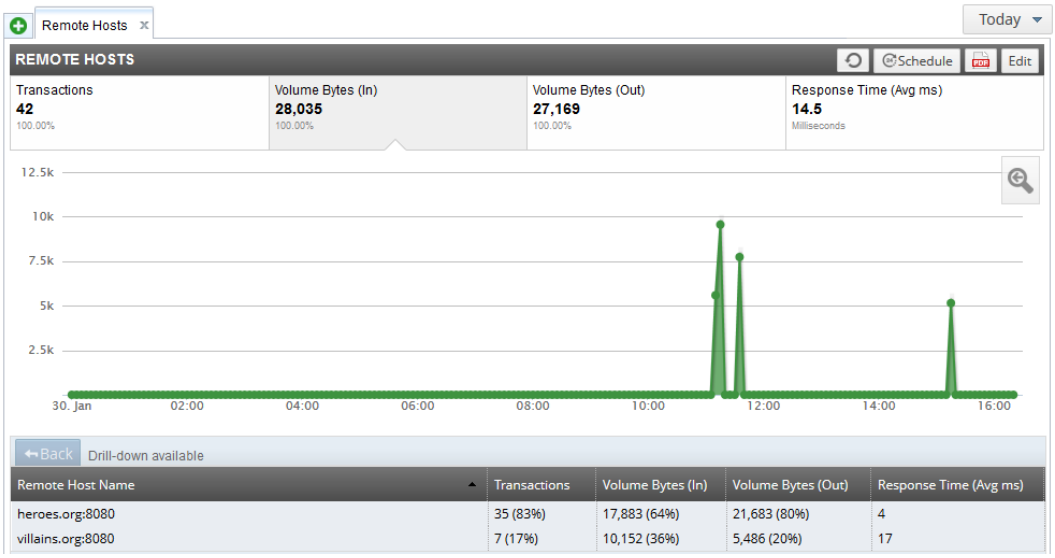
Monitor remote hosts

The **Remote Hosts** view displays metrics for all the remote hosts that have been configured in your domain. It shows details such as the number of message transactions that have been sent to this remote host, together with the total number of bytes sent to and received from this host.

In the **Remote Hosts** view, click a panel in the **REMOTE HOSTS** section at the top to display graph for the selected remote host metric below. For example, the available metrics include the following:

- **Transactions**: The number of message transactions to remote hosts.
- **Volume Bytes (In)**: The total number of bytes sent to remote hosts.
- **Volume Bytes (Out)**: The total number of bytes received from remote hosts.
- **Response Time (Avg ms)**: The average response time to remote hosts.

The following example shows the total number of bytes sent to remote hosts:



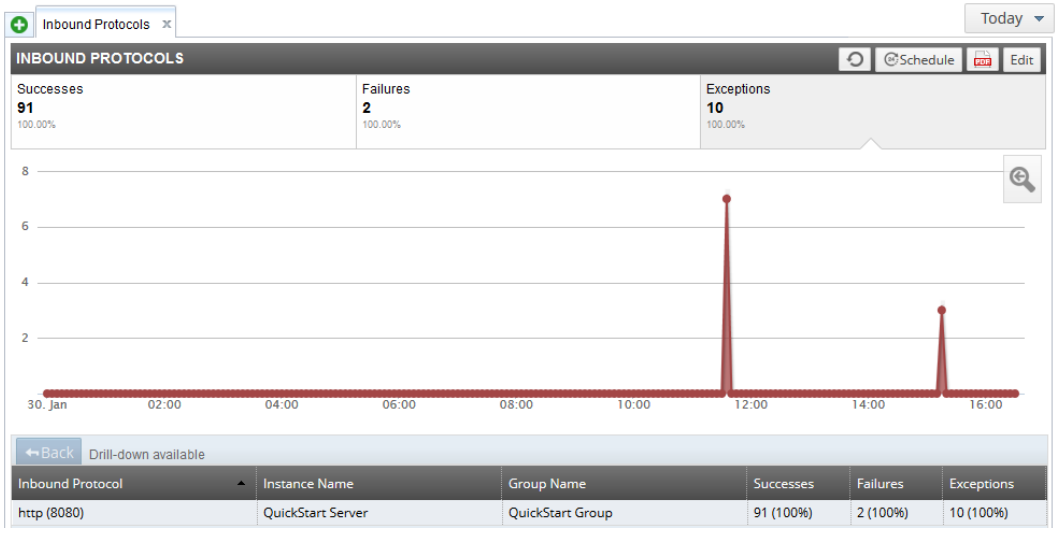
The table at the bottom shows all the remote hosts connected to by API Gateway instances in your domain. You can click a remote host in the table to drill down and view graphs for the selected remote host. Click **Back** on the left to return to the **REMOTE HOSTS** view.

Monitor protocols

The **Protocols** view enables you to monitor the different message protocols in your domain (for example, HTTP, Websocket, JMS, File Transfer, and so on). Click a panel in the **INBOUND PROTOCOLS** section at the top to display graph for the selected protocol metrics below. For example, the available metrics include the following:

- **Successes**: The number of successful messages that are processed in the domain.
- **Failures**: The number of failed messages that are processed in the domain.
- **Exceptions**: The number of messages that generated an exception in an API Gateway policy.

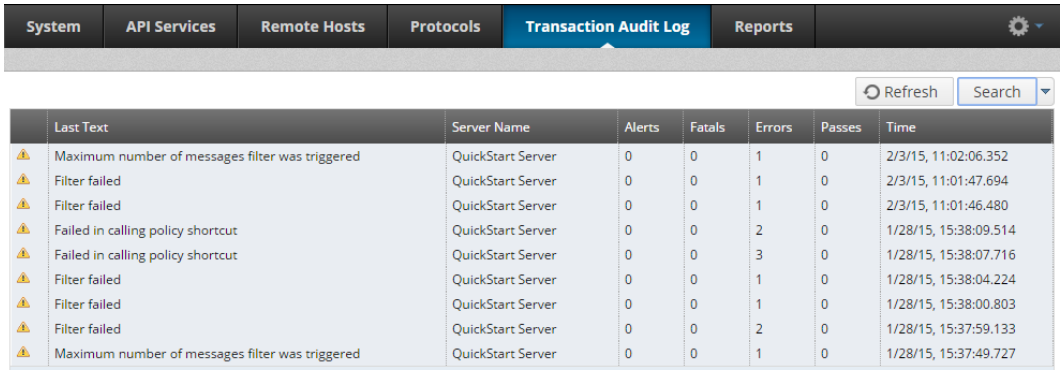
The following example shows the number of HTTP messages that generated an exception in the domain:



The table at the bottom shows all the protocols used by API Gateway instances in your domain. You can click a protocol in the table to drill down and view graphs for the selected protocol (for example, **http 8080**). Click **Back** on the left to return to the **INBOUND PROTOCOLS** view.

Audit transactions

The **Transaction Audit Log** view enables you to filter the transaction audit log messages messages generated by API Gateway instances in your domain. For example:



You can filter the log messages by clicking the **Search** button on the right in the toolbar. The **Query Editor** dialog enables you to create a query to filter log messages by details such as time period, severity level, filter type or name, and message text. When you have added your search criteria, click **Search** at the bottom to run the query. You can also save the query for later use.

When you click **Search**, the log messages that match the search criteria specified in the query are displayed in the table. For example, the details displayed in the table include the log message text, API Gateway name, alerts, and time. You can also double-click an item in the list for more details (for example, transaction ID, filter category, and filter name).

Note To view data in the **Transaction Audit Log** tab, you must configure the API Gateway to write to the metrics database. In the Policy Studio tree, select **Environment Configuration** > **Server Settings** > **Logging** > **Transaction Audit Log** > **Database**. For more details, see the "Configure API Gateway logging and events" in the *API Gateway Administrator Guide*.

Schedule custom reports

API Gateway Analytics uses message metrics stored in a centralized metrics database by the Node Managers running in your domain. The Node Managers store metrics for the virtualized services exposed by the local API Gateway instances, and for the services, clients, and remote host connections that they protect.

API Gateway Analytics can generate usage reports and charts based on the stored metrics data, and enables you to configure custom reports to suit the needs of your environment. This includes viewing available metrics for each target report type, grouping and filtering metrics, and what to display on drill through.

Enable scheduled reporting

You must fensure that reporting is enabled in Policy Studio. For details, see *Configure scheduled report settings in Policy Studio* on page 41.

Create reports in a monitoring view

In API Gateway Analytics, in the **System**, **API Services**, **Remote Hosts**, and **Protocols** views, you can create reports by clicking **Schedule** on the right in the toolbar. For example, to generate reports on API Service clients, perform the following steps:

1. Click the **API Services** > **Clients** view.
2. Click **Edit** on the right in the toolbar to specify the report settings:
   - **Name**: Defaults to the tab name (for example, **Clients**).
   - **Group by**: Select how the report is displayed (for example, by **Client Name**, **Instance Name**, and so on). You can also select the metrics that are displayed (for example, **Messages**, **Successes**, and so on).
   - **Filter by**: Select condition options to filter the display (for example, only show entries with a specific **Client Name**).
   - **Enable drill-down**: Select this to use the value of the **Group by** columns to provide an additional drill-down report. You can choose to group by additional fields. This is enabled by default.

   These filtering and grouping mechanisms enable you to answer questions such as what clients used an API Service, or which API services were used by a client. For example, to show clients that used `Service1` , you can create a custom report that groups by **Client Name** and filters where **Service Name** is `Service1`.

   Note The group-by mechanism only applies to the data table below the report chart. The chart remains the same.
3. Click **Save** in the toolbar.
4. Click **Schedule** in the toolbar.
5. You can also click the **PDF** button in the toolbar to generate a PDF version of the report.

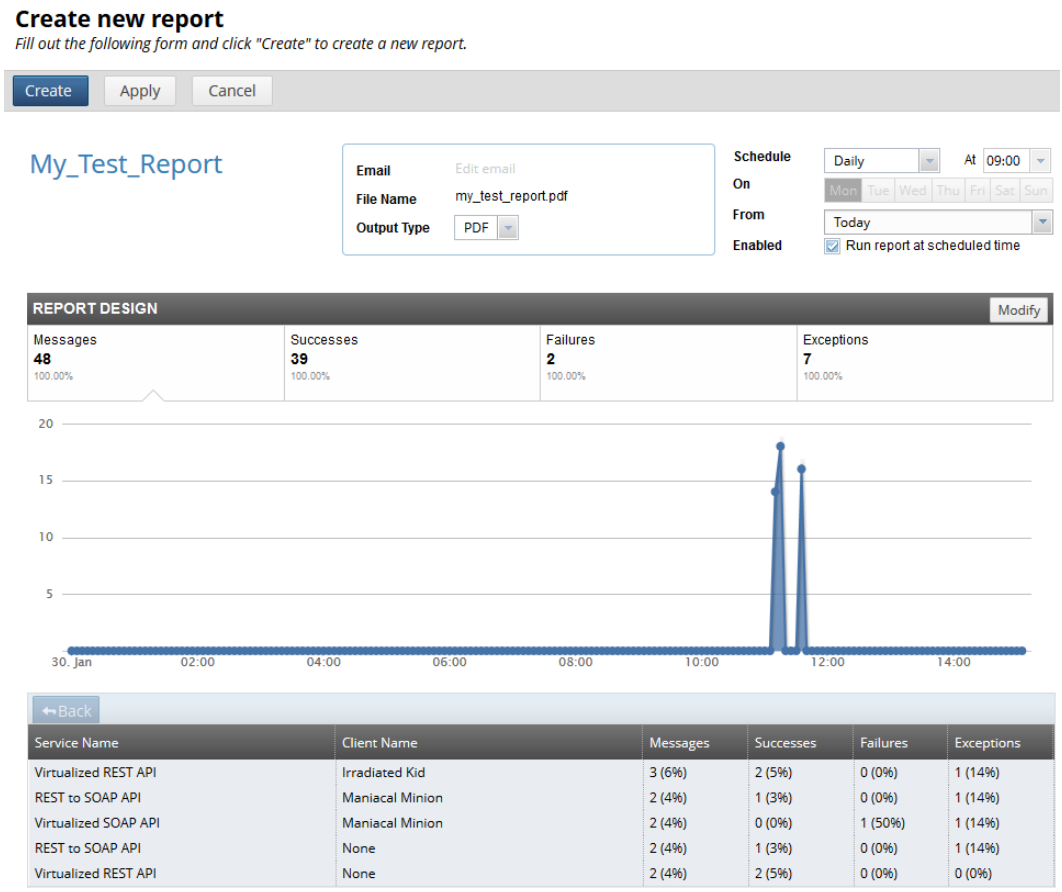
You can follow a similar sequence of steps to generate reports in the **System**, **Remote Hosts**, and **Protocols** views.

Using the reports view

Alternatively, you can create a report in the **Reports** view. Perform the following steps:

1. Click **New report** on the left in the toolbar.
2. Enter the report **Name** in the dialog.
3. Select the report **Type**, and click **OK**.
4. Configure the schedule and output options on the right as appropriate.Defaults to PDF output daily at 9am, starting from today.
5. Click **Apply** to save your settings.
6. Click **Create** to create the report.

The following example shows a custom report grouped by **Service Name**:



When you have created a report, you can select it in list on the **Managing Reports** page, and click **Generate now**. Alternatively, click **Delete** to remove the report from the list.

Further information

For more details, see the following:

- *Configure scheduled report settings in Policy Studio* on page 41
- *Purge the metrics database for API Gateway Analytics* on page 43

Configure scheduled report settings in Policy Studio 7

You can schedule API Gateway Analytics reports to run on a regular basis, and to email the results to users in PDF format. These reports include summary values at the top (for example, the number of requests, SLA breaches, alerts triggered, and unique clients in a specified week) followed by a table of services, and their aggregated usage data (for example, the number of requests on each service).

The report data is for the configured *current week of the report*, which is compared to the week before. You can set the configured *current week of the report* to be the actual current calendar week or any prior week (provided there is corresponding data in the database).

To configure scheduled report settings in Policy Studio, right-click the **Environment Configuration** > **Listeners** > **Axway Analytics** node in the Policy Studio tree, and select **Database Archive**.

Database configuration

Click the browse button the right, and select a pre-configured database connection in the dialog. This setting defaults to the `Default Database Connection`. To add a new database connection, right-click the **Database Connections** node, and select **Add DB connection**.

You can also edit or delete existing nodes by right-clicking and selecting the appropriate option. Alternatively, you can add database connections under the **Environment Configuration** > **External Connections** node in the Policy Studio tree view. For more details on creating database connections, see the *API Gateway Policy Developer Guide*.

Scheduled reports configuration

You can configure the following settings for scheduled reports:

Enable Report Generation:
Select whether to enable scheduled reports in PDF format. When selected, by default, this runs a scheduled weekly report on Monday morning at 0:01. For details on configuring a different time schedule, see the next setting. This setting is not selected by default.

When **Enable Report Generation** is enabled, you can configure the following settings on the **Report Generator Process** tab:

Connect to API Gateway Analytics as User:
Enter the user name and password used to connect to the report generator process. Defaults to the values entered using the `configureserver` script.

Output:
Enter the directory used for the generated report files in the **Output Directory** field, or click **Choose** to browse to the directory. Defaults to the directory entered using the `configureserver` script (for example, `c:\temp\reports`). You can also select to **Do not delete report files after emailing**. This setting is not selected by default.

SMTP configuration

When **Enable Report Generation** is enabled, you can configure the following settings on the **SMTP** tab. These settings default to those entered using the `configureserver` script. For more details, see the *API Gateway Installation Guide*.

Email generated reports:
Select whether to email generated PDF report files. This is not selected by default.

Do not delete report files after emailing:
Select whether to keep generated PDF report files after they are sent. Not selected by default.

Email Recipient (To):
Enter the recipient of the automatically generated email (for example, `user@mycorp.com`). Use a semicolon-separated list of email addresses to send reports to multiple recipients.

Email Sender (From):
The generated report emails appear *from* the sender email address specified here (for example, `no-reply@mycorp.com`).

Note Some mail servers do not allow relaying mail when the sender in the **From** field is not recognized by the server.

SMTP Server Settings:
Specify the following fields:

Outgoing Mail Server (SMTP)	Specify the SMTP server used to relay the report email (for example, `smtp.gmail.com`).
Port	Specify the SMTP server port to connect to. Defaults to port 25.
Connection Security	Select the connection security used to send the report email (`SSL`, `TLS`, or `NONE`). Defaults to `NONE`.

Log on Using:
If you are required to authenticate to the SMTP server, specify the following fields:

User Name	Enter the user name for authentication.
Password	Enter the password for the user name specified.

Purge the metrics database for API Gateway Analytics 8

You can use the `dbpurger` command to connect to your metrics database and to purge old data. This command also enables you to retain a specified amount of data, and to archive all data.

This topic assumes that you have already configured the connection to your metrics database. For more details, see:

- *Configure API Gateway with the metrics database* on page 23

Run the dbpurger command

For API Gateway Analytics metrics, you can run the `dbpurger` command from the following directory:

```
INSTALL_DIR/analytics/posix/bin
```

dbpurger options

You can specify the following options to the `dbpurger` command:

Option	Description
`-h, --help`	Displays help message and exits.
`-p PASSPHRASE, --passphrase=PASSPHRASE`	Specifies the configuration passphrase (leave blank for zero length).
`--dbname=DBNAME`	Specifies the database name (mutually exclusive with `dburl`, `dbuser`, and `dbpass` options).
`--dburl=DBURL`	Specifies the database URL.
`--dbuser=DBUSER`	Specifies the database user.
`--dbpass=DBPASS`	Specifies the database passphrase.

Option	Description
`--archive`	Archive all data.
`--out=OUT`	Archive all data in the specified directory.
`--purge`	Purge data from the database. You must also specify the `--retain` option.
`--retain=RETAIN`	Specifies the amount of data to retain (for example, `30days`, `1month`, or `1year`). You must specify this option with the `--retain` option.

Example dbpurger commands

This section shows examples of running `dbpurger` in default interactive mode and of specifying command-line options.

Run dbpurger in interactive mode

The following example shows the output when running the `dbpurger` command in interactive mode. This example archives all data, retains three months of data, and purges older data from the database:

```
>dbpurger
Choosing:Default Database Connection
Archive database (Y, N) [N]:y
Archive path [./archive]:Purge an amount of data from the database (Y, N) [N]:y
Amount of data to retain (e.g. 1year, 3months, 7days) [3months]:
Wrote archive:./archive/process_groups.xml
Wrote archive:./archive/processes.xml
Wrote archive:./archive/metric_types.xml
Wrote archive:./archive/audit_log_sign.xml
Wrote archive:./archive/time_window_types.xml
Wrote archive:./archive/audit_log_points.xml
Wrote archive:./archive/audit_message_payload.xml
Wrote archive:./archive/transaction_data.xml
Wrote archive:./archive/metric_groups.xml
Wrote archive:./archive/metric_group_types.xml
Wrote archive:./archive/metrics_alerts.xml
Wrote archive:./archive/metrics_data.xml
Purging data older than:Wed Jun 27 15:26:00 BST 2012
Purging table:audit_log_sign... deleted 0 rows
Purging table:transaction_data... deleted 0 rows
Purging table:audit_message_payload... deleted 7 rows
```

```
Purging table:audit_log_points... deleted 16 rows
Purging table:metrics_alerts... deleted 4 rows
Purging table:metrics_data... deleted 703 rows
```

Specify dbpurger command options

The following example shows the output when specifying options the `dbpurger` command. This example retains 30 days of data, and purges older data from the database:

```
dbpurger --dburl=jdbc:mysql://127.0.0.1:3306/reports --dbuser=root --dbpass=fred
   --purge --retain=30days
```

Note You can run `dbpurger` without a password by specifying the name of the database connection. For example:

```
dbpurger --dbname="Default Database Connection" --archive --out=archive.dat
```