



Edge Analytics & Logging

Analytics

Purpose: *Understand API usage, see long-term API usage trends, and segment audiences.*



Dashboards

Check API adoption rates

Developer Metrics

Evaluate performance of APIs used by apps; verify to conformance

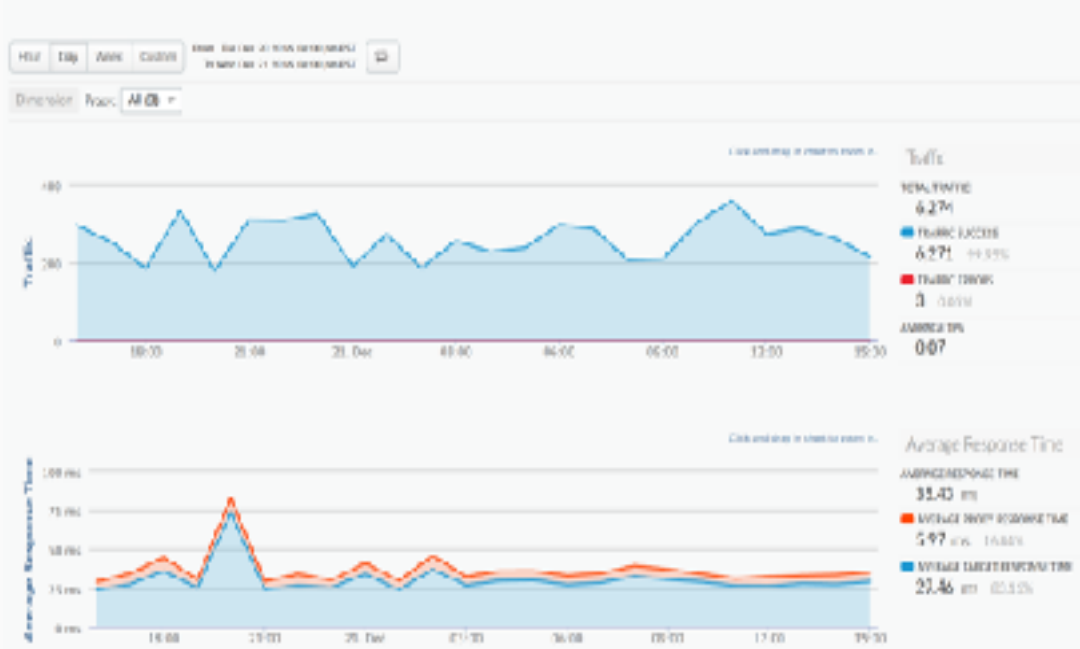
Custom Reports

Evaluate bottom-line impact and ROI

Monitoring

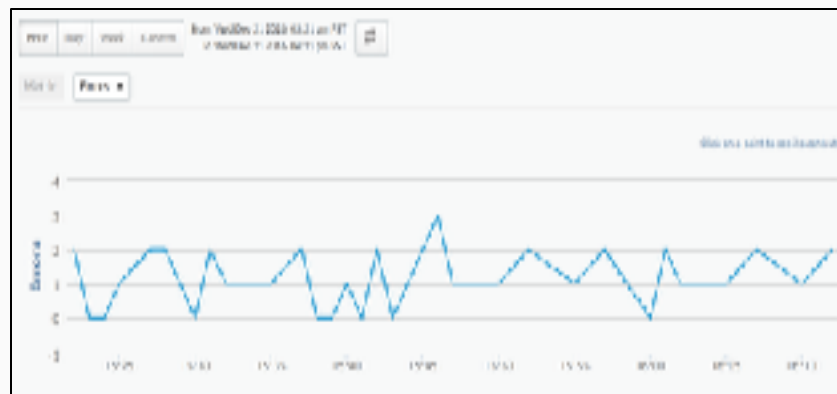
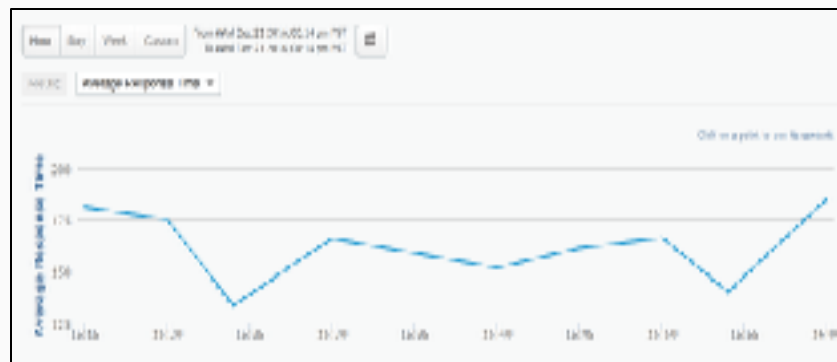
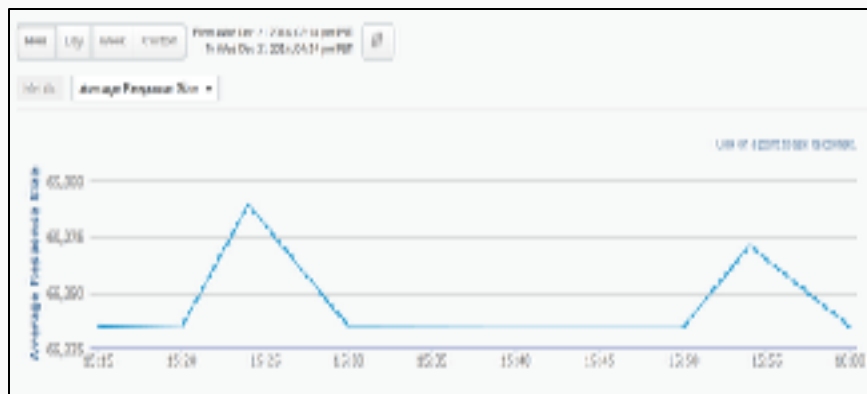
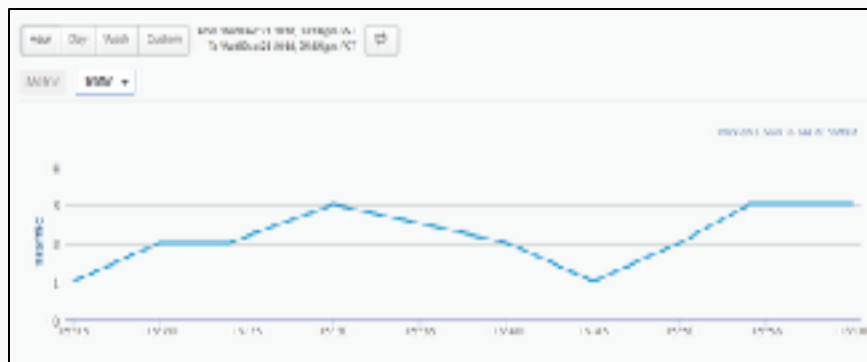
Verify SLA conformance; plan capacity expansion

Edge Analytics Services



- Serves as the repository for analytics data
- Dashboards give information about all APIs or only specific APIs, and support drill-down to allow deeper analysis
- Provides statistical graphs that can be customized to meet specific requirements
- Allows statistics to be extracted using APIs and imported into other systems for further analysis

Edge Analytics Services



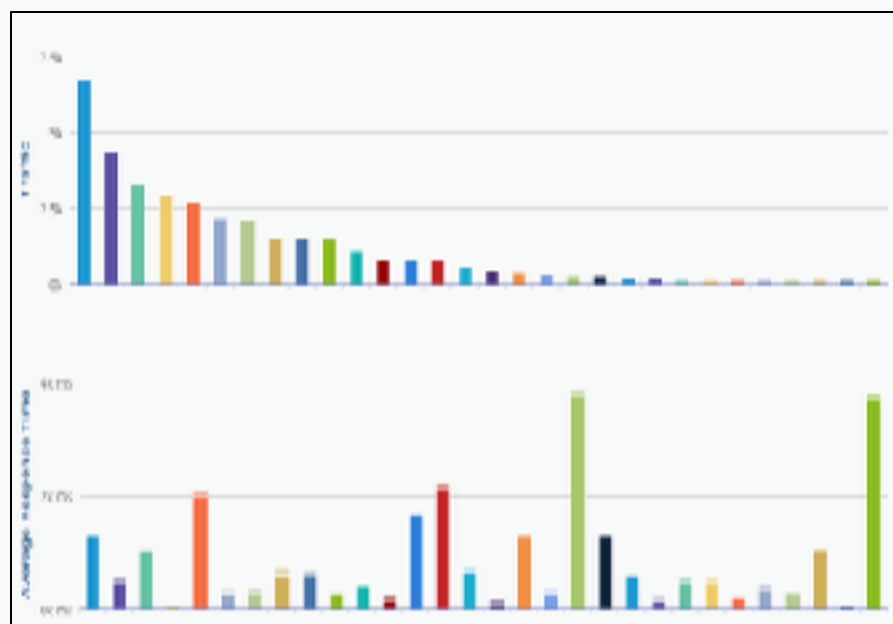
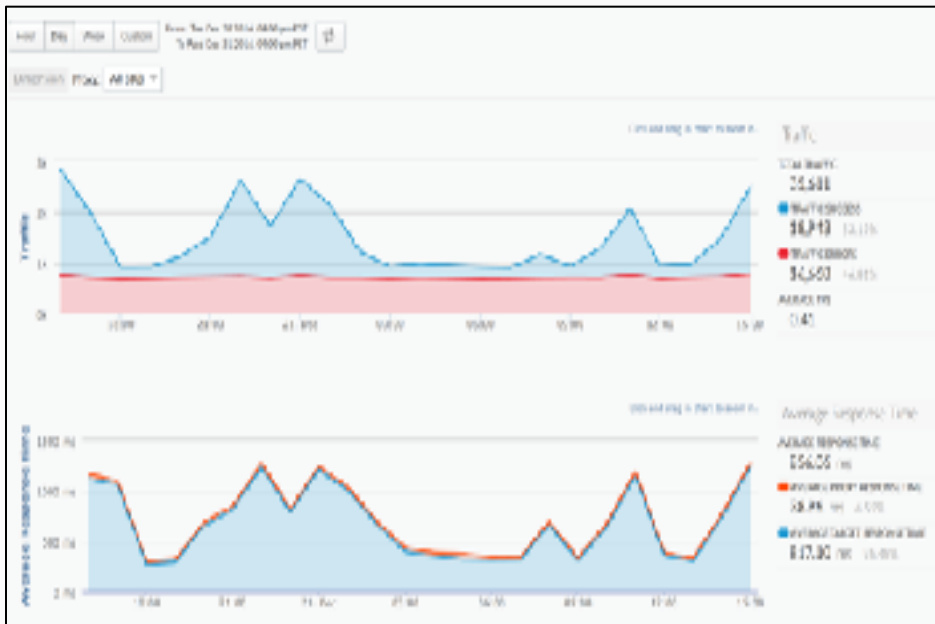
Edge Analytics Services

Analyze trends for your API program, inspect and diagnose problems, and generate reports.

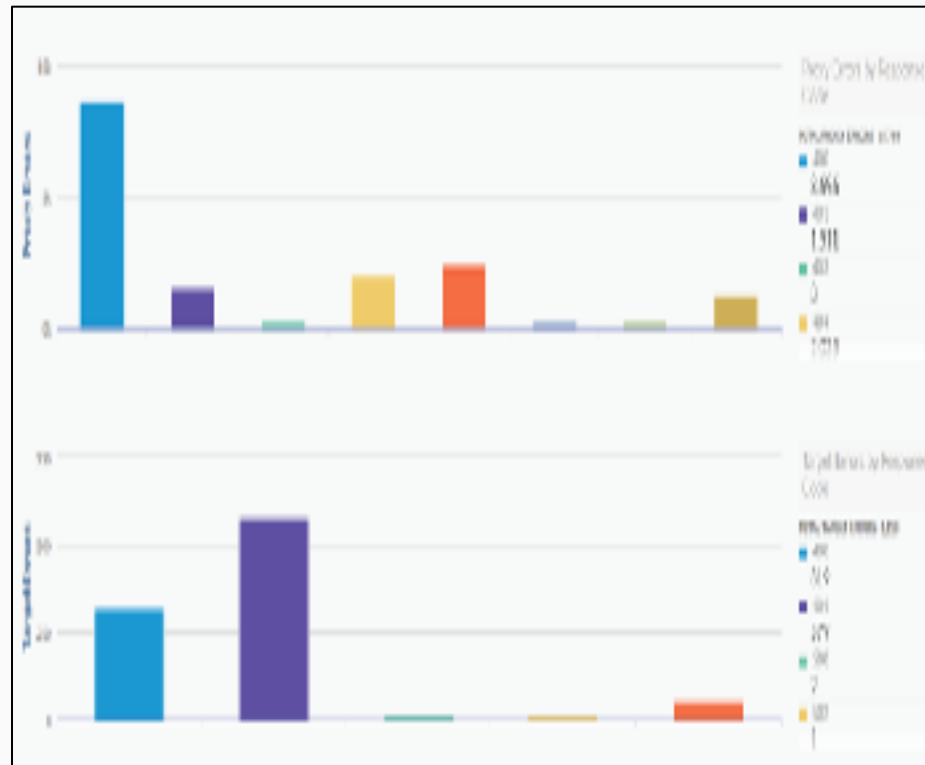
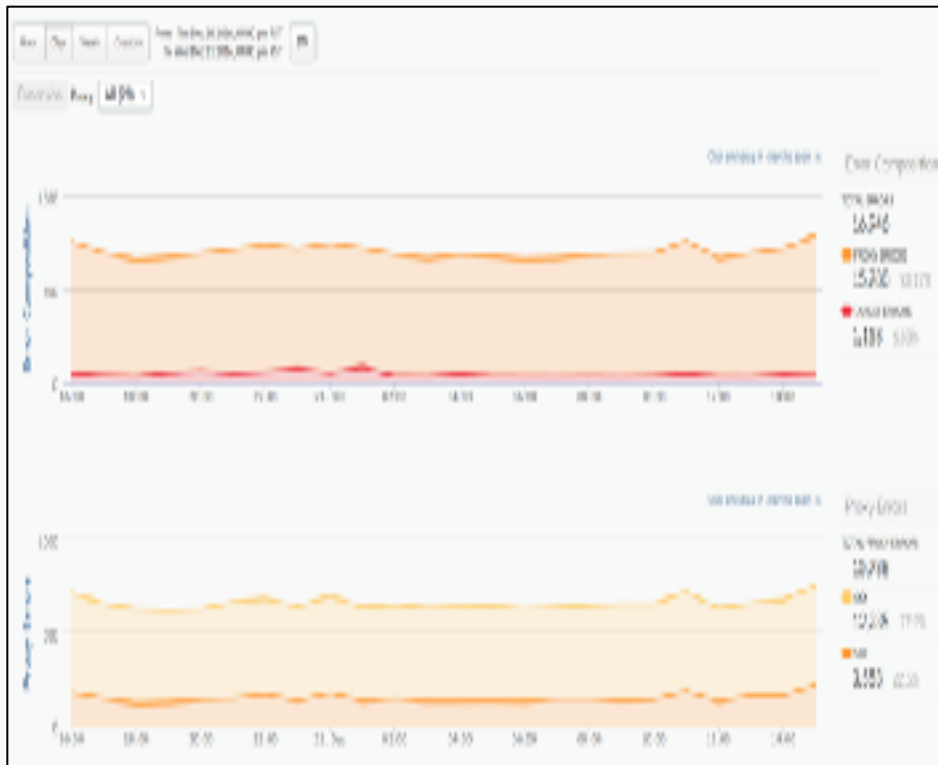
Out of the box reports include:

- **API Proxy Performance** - visualize how much traffic your APIs generate and how long it takes for API calls to be processed
- **Business Transactions** - helps you understand changes in API traffic that might be caused by specific business, marketing, or partner events
- **Cache Performance** - visualize the benefit of the cache in terms of lower latency and reduced load backend servers
- **Developer Engagement** - tells you which of your registered app developers are generating the most API traffic
- **Devices** - tells you about the devices and servers that are being used to access your APIs
- **Error Code Analysis** - tells you about error rates for API proxies and targets.
- **Geomap** - tracks traffic patterns, error patterns, and quality of service across geographical locations
- **Latency Analysis** - can alert you to any latency issues your API proxies may be experiencing
- **Target Performance** - helps you visualize traffic patterns and performance metrics for API proxy backend targets
- **Traffic Composition** - measures the relative contribution of your top APIs, apps, developers, and products to your overall API program
- **Custom Reports** - way to specify precisely what you want to measure across your API program

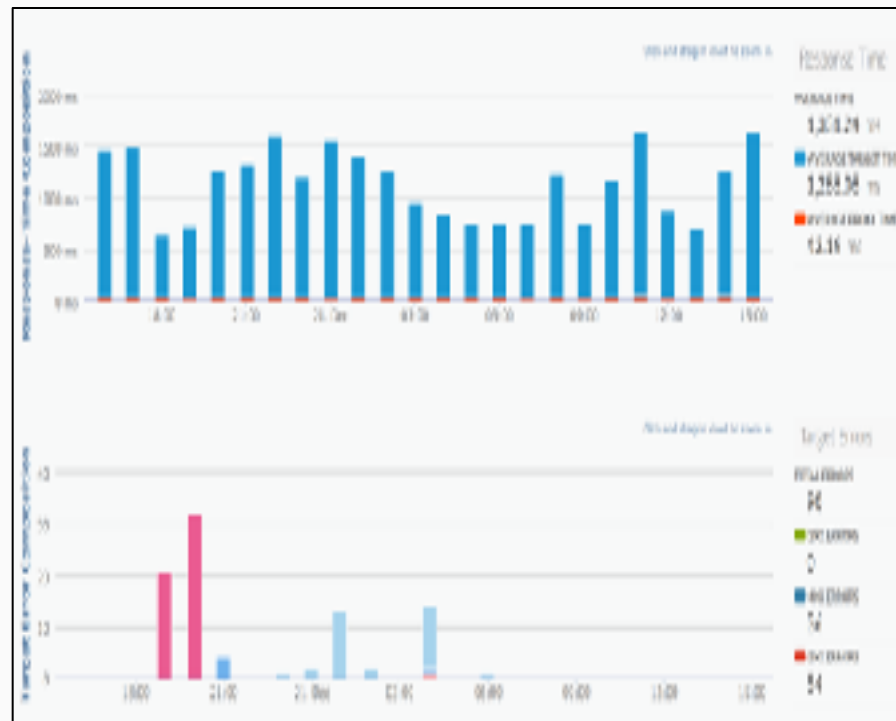
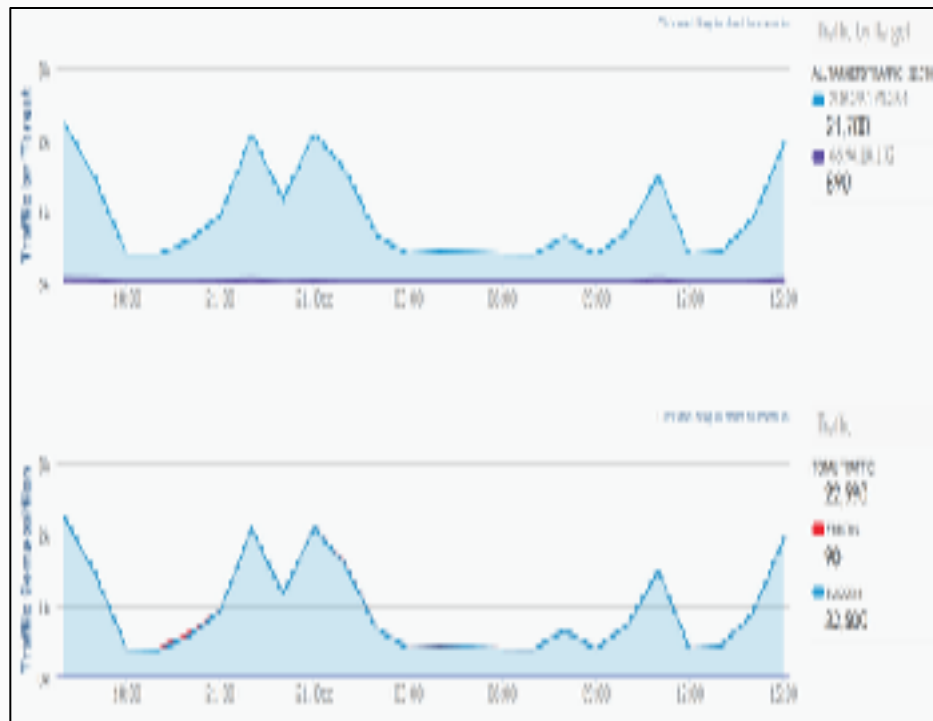
API Proxy Performance



Error Code Analysis



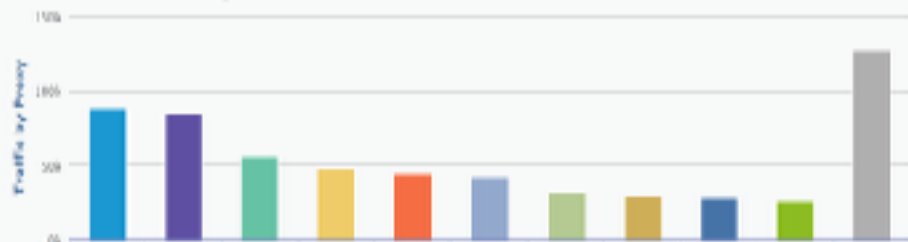
Target Performance



Traffic Composition

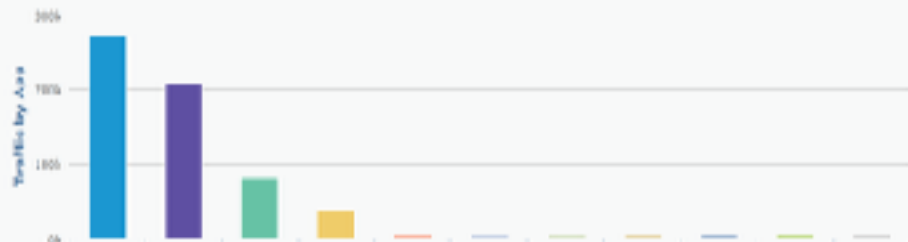
Top 10 Proxies Traffic

See more details for Proxies Traffic Composition



Top 10 Apps Traffic

See more details for Apps Traffic Composition



Top 10 Producers Traffic

See more details for Producers Traffic Composition



Top 10 Developers Traffic

See more details for Developers Traffic Composition



Custom Reports

The screenshot shows the 'Custom Reports' configuration page. It is divided into three main sections: Basics, Metrics, and Filter.

- Basics:** Includes fields for 'Report Name' and 'Report Location'. Below these is a 'Chart Type' selector with radio buttons for 'Column' (selected) and 'Line'. A note states: 'For Column charts, the x-axis represents groups designed by dimensions. For Line charts, the x-axis represents time.'
- Metrics:** The title is 'The y-axis represents metric values'. It contains a table with three columns: 'Metric', 'Aggregate Functions', and 'Actions'.
 - Under 'Metric', a dropdown menu is open, showing a list of metrics: 'Select...', 'Average Transactions per Second', 'Cache Hit', 'L3 Cache Elements Count', 'Policy Errors', 'Proxy Errors', 'Request Processing Latency', 'Request Size', 'Response Cache Eviction', 'Response Processing Latency', 'Response Size', 'Target Errors', 'Target Response Time', 'Total Response Time', and 'Traffic'.
 - Under 'Aggregate Functions', there are radio buttons for 'Sum', 'Average', 'Min', and 'Max'.
 - Under 'Actions', there is a '+ Metrics' button.
- Filter:** The title is 'Filter'. It includes a 'Filter Conditions' section with a table with columns 'Connector', 'Name', 'Operator', 'Value', and 'Actions'. Below this table is a '+ Other Functions...' button. At the bottom right are 'Cancel' and 'Save' buttons.

A custom report is a way to specify precisely what you want to measure across your API program. For instance, you can measure all API traffic generated from a specific client IP address.

Custom reports provide both chart and table views

NOTE: After API calls are made to proxies, it takes about **12 minutes** for the data to appear in dashboards, custom reports, and management API calls.

Custom Reports

Measure	Aggregate Function	Custom
1 Policy Errors	Sum Average Min Max	Delete
2 Target Errors	Sum Average Min Max	Delete
3 Policy Errors	Sum Average Min Max	Delete

→ Measure

- ✓ Select
- Access Token
- API Endpoint
- API Proxy
- API Proxy Revision
- catalog
- catalogId
- catalogId
- Client Host
- Client ID
- Client IP Address
- Developer
- Developer App
- Developer Email
- envId
- Environment
- Flow Resource
- Gateway Flow ID
- Organization
- product
- productId
- productName
- Policy
- Policy Rule
- Policy Rule ID
- Policy Rule ID
- Request Path
- Request URI
- Request Verb
- Response Status Code
- store
- subcategories
- subcategoriesId
- Target
- Target Data Path
- Target Host
- Target IP Address
- Target Resource Code
- Target URL
- User Agent
- version
- versionId
- Virtual Host
- Virtual Host
- Virtual Host

Drilldowns

Drilldowns have two purposes:

- Initially, the drilldown is used to group data, similar to the GROUP BY clause in SQL.
- Once a drilldown is selected, it becomes a filter condition for the WHERE clause in SQL, used for subsequent drilldowns (used for grouping production).

Drilldown	Actions
1 Policy Errors	Delete

→ Drilldown

Filter

Basic Advanced

Filter Conditions	Connector	Name	Operator	Value	Actions
	API Proxy		Contains		or and

Custom Reports

Drilldowns

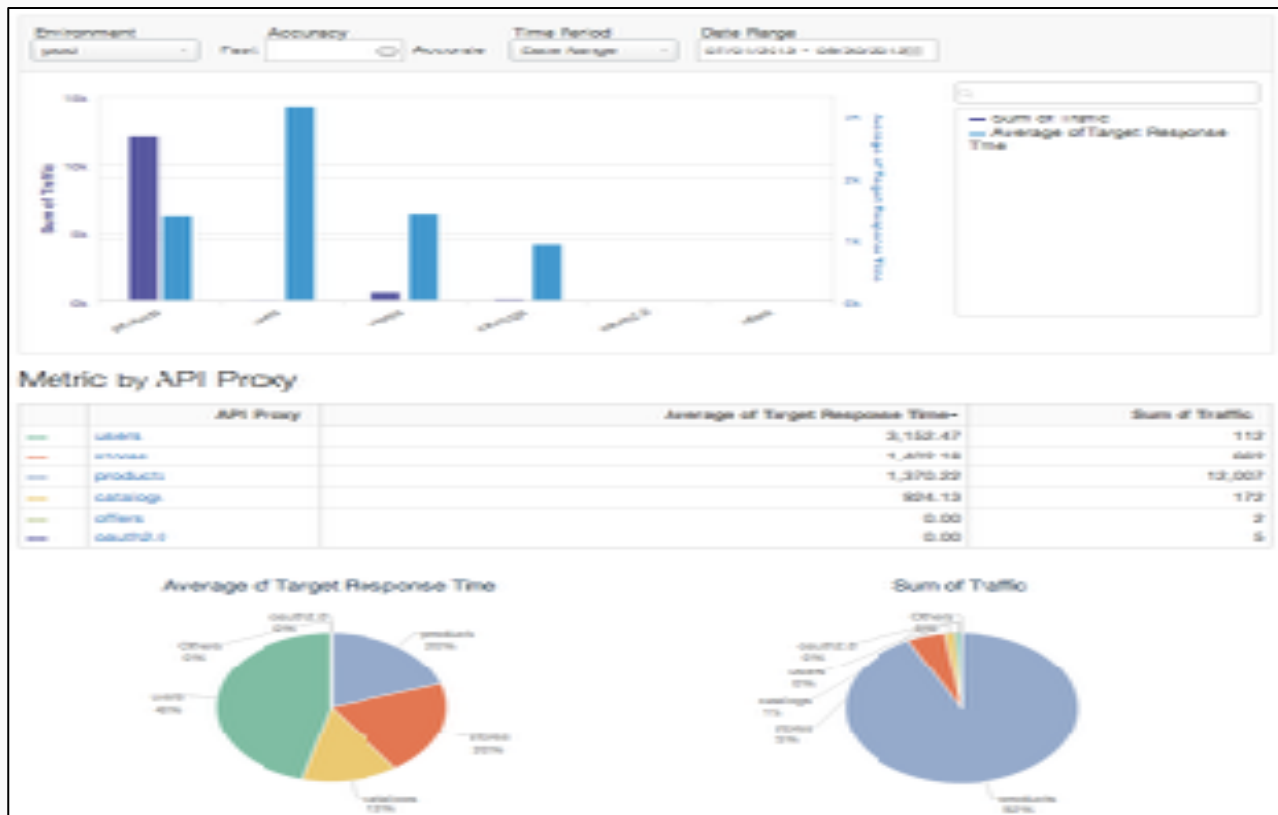
Drilldowns have two purposes:

- Initially, the drilldown is used to group data, similar to the GROUP BY clause in SQL.
- Once a drilldown is selected, it becomes a filter, similar to the WHERE clause in SQL, and the subsequent drilldown becomes the grouping mechanism.

The screenshot shows the 'Drilldowns' section with a table containing one row with a 'Drilldown' field and an 'Actions' column. Below it is a '+ Drilldowns' button. The 'Filter' section shows a table with columns for 'Connector', 'Name', 'Operator', 'Value', and 'Actions'. The 'Name' column contains 'Country ID', and the 'Operator' column contains 'Contains'. An orange arrow points from the 'Drilldowns' section to the 'Filter' section.

Cache Hit
Cache Miss
First Error
Response Processing Latency
REQUEST ID
RESPONSE PROCESSING LATENCY
Response Size
Target Error
Target Response Time
Total Response Time
Traffic
Domain.com
Address: India
API Endpoint
API Name
API Proxy Revision
urlkey
urlweight
category
categoryid
Client Host
Last: 82
Client: IP Address
Developer
Developer App
Developer Email
email
Environment
Flow Resources
Gateway Timeout
Organization
product
productid
productname
PRIORITY
Priority: 8282 Path
Priority: Client IP
Priority: Path Suffix
Request Path
Request URL
Response Value
Response: Status Code
state
subdomainkey
subdomainkeyid
Target
Target: 8282 Path
Target: host
Target: IP ADDRESS
Target: RESPONSE L222
Target URL
User Agent
version
urlweight
Virtual Host
V: 8282828282 8282

Custom Reports



Custom Statistics Collector Policy

1 Analyze HTTP Transaction and Define Collectors

Extract Data from HTTP Transaction

Save in Collector Variable

Location Type	Location Source	Name	Data Type	Action
1 Request JSON Body	JSON Path: \$.requestId	requestId	String	

+ Collector

2 Select Flow

Proxy Endpoint default, Flow Postflow

Solution Results

Segment	Policies
Request	<div><div>Extract Statistics Request</div><div>Collect Statistics Request</div></div>

New collector variables are available in the Report editor as drilldown dimensions after this API proxy has been saved and deployed.

Cancel

Build Solution

Edge allows developers to send any custom data in its out-of-box Analytics engine. This is available through a policy, and even a wizard to set up for you.

Custom Statistics Collector Policy

- Policies are auto-generated by the wizard
- Can be manually customized to extract any part of the HTTP message for used in the StatisticsCollector policy

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <ExtractVariables async="false" continueOnError="true" enabled="true" name="__extract-statistics-request__">
3   <!-- Created by the Custom Analytics Collection tool on: Thu Aug 07 2014 10:17:10 GMT-0700 (PDT) -->
4   <DisplayName>Extract Statistics Request</DisplayName>
5   <JSONPayload>
6     <Variable name="__jsonPayload" type="string">
7       <JSONPath>$.title_id</JSONPath>
8     </Variable>
9   </JSONPayload>
10 </ExtractVariables>
```



```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <StatisticsCollector async="false" continueOnError="false" enabled="true" name="__collect-statistics-request__">
3   <!-- Created by the Custom Analytics Collection tool on: Thu Aug 07 2014 10:17:10 GMT-0700 (PDT) -->
4   <DisplayName>Collect Statistics Request</DisplayName>
5   <Statistics>
6     <Statistic name="titleid" ref="__jsonPayload" type="String"></Statistic>
7   </Statistics>
8 </StatisticsCollector>
```

Custom Analytics Data

The screenshot displays the Google Cloud API Console interface for configuring a custom analytics collector. The main area shows a flow diagram with a 'REQUEST' arrow pointing right and a 'RESPONSE' arrow pointing left. A 'Statistics Collector' icon is visible on the right side of the flow. The 'Property Inspector' on the right shows the 'PreFlow' property with a 'Name' of 'Statistics-Collector-1'. The 'Endpoint default' section shows the XML configuration for the 'Statistics Collector-1' policy.

Property Inspector: PreFlow

Property	Value
name	PreFlow
Request	
Response	
Step	
Name	Statistics-Collector-1

Endpoint default: Policy Statistics Collector-1

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <StatisticsCollector async="false" continueOnError="false" enabled="true" name="
3   <DisplayName>Statistics Collector-1</DisplayName>
4   <Properties>
5   <Statistics>
6     <Statistic name="catalog" ref="myCatalog.id" type="string"/>
7     <Statistic name="catalogName" type="string" ref="myCatalog.name"/>
8   </Statistics>
9 </StatisticsCollector>
```

- Add custom analytics data during API execution, then report on it using custom reports
- Custom analytics data can come from any portion of the inbound request or outbound response

Analytics API

- Edge Analytics Services exposes a RESTful API.
- Can be used to automate certain Analytics functions, such as retrieving metrics periodically using an automation client or script.
- Can also be used to build your own visualizations in the form of custom widgets that you can embed in portals or custom apps.

The base URL that you invoke to retrieve statistics for dimensions is the following:

```
https://api.enterprise.apigee.com/v1/o/{org_name}/e/{env_name}/stats
```

The following dimensions are supported :

`/apis`

`/apiproducts`

For more info on the API, refer the following links:

<https://docs.apigee.com/api/stats>

https://docs.apigee.com/management/apis/get/organizations/%7Borg_name%7D/stats

<https://docs.apigee.com/analytics-services/reference/analytics-command-reference>

Lab

TBD

THANK YOU