



**apigee**

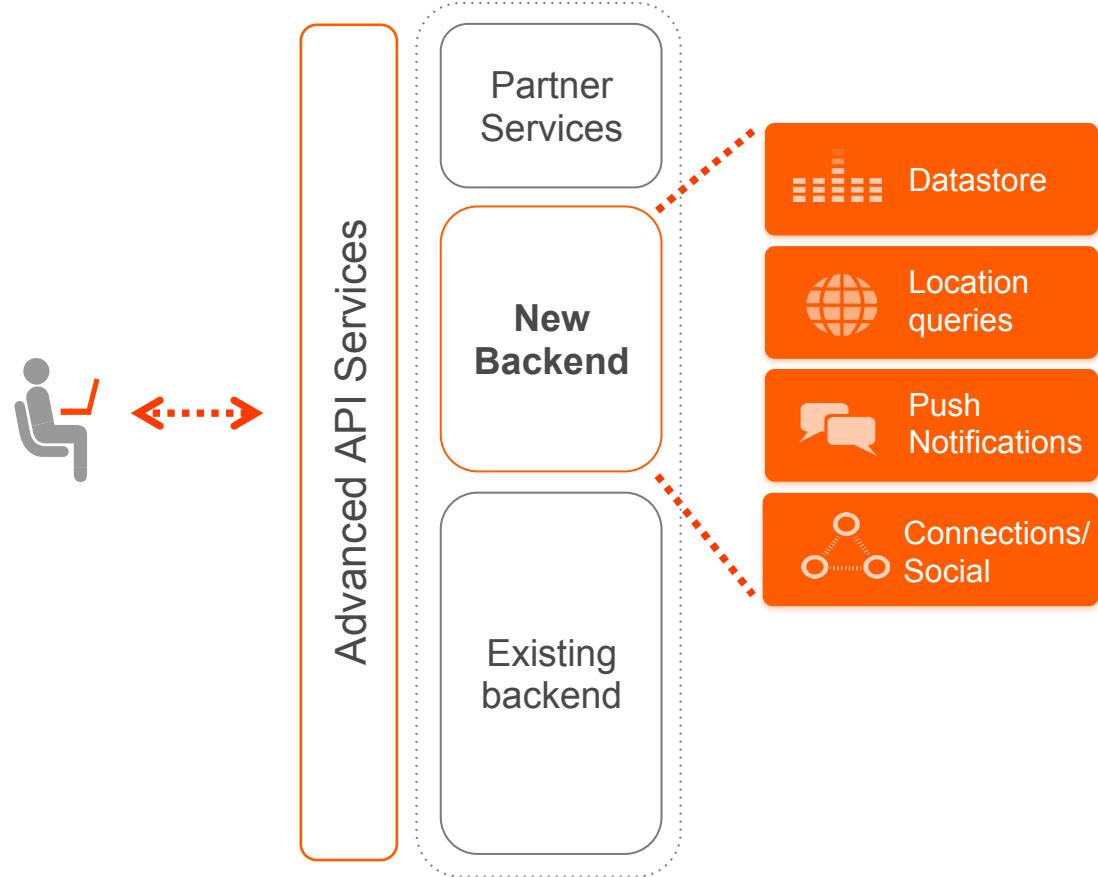
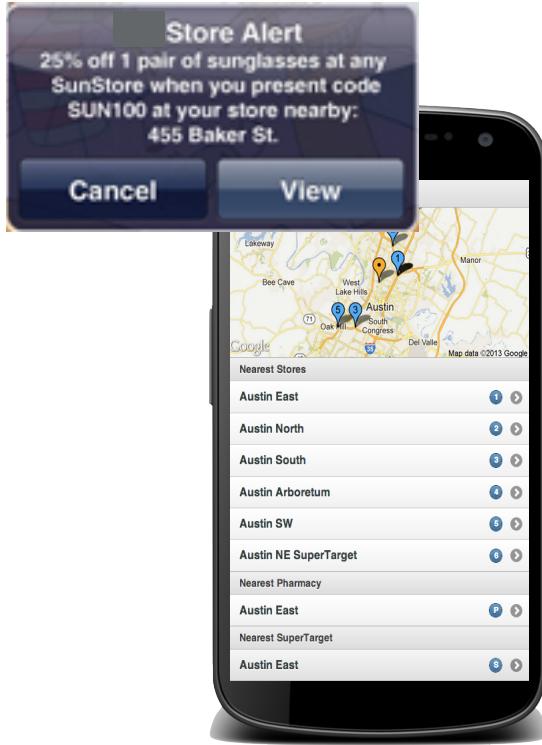
# Foundational Training

Intro to Apigee BaaS

# API BaaS Overview



# The API Backend-as-a-Service (BaaS)



SDKs for:

- iOS
- Android
- JavaScript
- Ruby
- Node.js
- Microsoft .Net

Faster and cost-effective building  
of new back-ends as APIs

# Product Overview

API BaaS is a service that makes it easy to store, retrieve , and query your data. It is also a push notification platform.

Technology-level: leverages open source Usergrid with Cassandra NoSQL database.

- |                      |  |
|----------------------|--|
| Scalable Datastore   | → Accelerate app delivery                      |
| User Data            | → Simplify management of users and preferences |
| Location             | → Use location relevance                       |
| Push Notifications   | → Proactively engage users                     |
| Connections & Social | → Bring in social context to apps              |

# Data Storage

## Cassandra Data Store

- Fast writes to the data store
- Distributed architecture means no single point of failure
- Flexibility in data model design (schema-less)
- Linear scalability

Different types of data can be stored inside of BaaS:

- **Application data** – e.g. listing of houses, a library catalog of books, or even a social graph. This data can be queried using SQL like query language.
- **Asset data** – Image, video, binary, audio files, etc.

## Geolocation

```
{  
  "displayName" : "Santa Clara Library",  
  "name" : "sc-001",  
  "location" : {  
    "latitude" : 37.340512,  
    "longitude" : -121.973504  
  }  
,
```

- Pull data from GPS-enabled devices
- Leverage device entity to capture geolocation data
- Use with push notifications for targeted messaging



## SDK Support

---

The API BaaS product comes with many SDKs that are freely available for use in mobile or server-side clients.

Examples:

- iOS
- Android
- Node.js
- Ruby
- JavaScript
- .NET

# Security and Authentication

- API BaaS provides for Client Credentials and Resource Owner Password Credentials (or password) grant type tokens out of the box.
- Password
  - Application user – based on user entities, based on the roles and permissions assigned to the user
  - Admin user – based on admin users at API BaaS management layer
- Client Credentials
  - Organization client auth – For org-level access to API BaaS. Found at Org Administration screen
  - Application client auth – For individual app access. Found at Getting Started -> Server App Credentials



## How to deploy prod and non-prod

---

- **Without the concept of separate environments, how do we do prod and non-prod deployments?**
  - If possible, a completely separate API BaaS stack provides ultimate data separation between prod and non-prod. Alternatively, you can use organizations to logically separate the data/credentials/users for prod/non-prod purposes.
  - Copy over or recreate data that is similar from prod to non-prod. There are scripts to help do this.
  - Use similarly named apps and entities between prod and non-prod so gateway code can remain similar.

# Data Storage and Querying



# Entities

---

- There are pre-existing default data entity types (user, group, role, application, etc.)
- Default properties : **uuid**, **name**, **type**
  - **type** is the singular form of a collection
  - **uuid** is automatically generated
  - **name** is optional and can be used in the API URI path for accessing the entity (must be unique)
- You can partially update an entity by sending a PUT request with only the attributes' name/value pairs to be updated



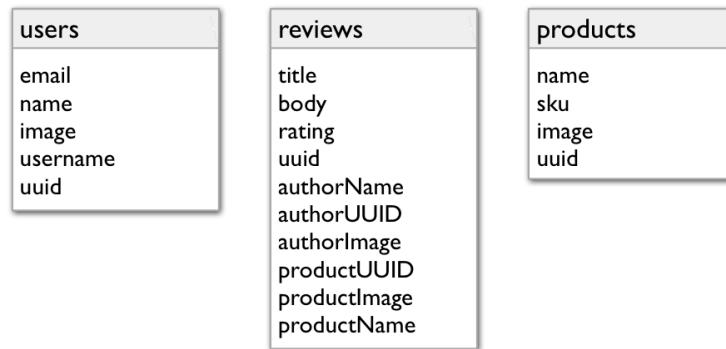
## Collections

---

- There are pre-existing default collections (users, roles, applications, assets, etc.)
- Updating a collection allows you to batch update entities at the same time. Provide multiple entity objects inside of an array
- Collections cannot currently be deleted; you can only empty them
- When accessing a collection, a single page is returned and 10 entities are returned by default

# Data Modeling

- De-normalize and duplicate – avoid costly queries. This takes some additional planning, but well worth it.





## Data Modeling (continued)

---

Leverage **connections** to streamline scoped requests. You can optimize your app's performance by connecting entities that your code retrieves most often. With connections, you can avoid some queries that might add complexity and reduce performance.

- Create a connection between two nouns using a verb that describes the relationship between the two.

### Query Approach

```
GET http://api.usergrid.com/my_org/my_app/reviews?  
ql=select * where authorUUID=<user_uuid>
```

### Connection Approach

```
GET http://api.usergrid.com/my_org/my_app/users/jimmy/  
wrote/reviews
```

# Data Modeling (continued)

## How NOT to model your API BaaS data store:

- More than 3 query parameters within a query on a single column family. *Queries with multiple query parameters can perform very slow.*
- Sticking to rigid entity structure. Can use different properties between entities within an entity array.

- An RDBMS table has a schema-defined set of columns per row.

name	sku	image	description
Bouncy Castle	35450349822	http://path.jpg	
Coffee Maker	60723023589	http://path.jpg	It makes coffee.
Air Mattress	53045985365		

- A Cassandra column family includes rows that can have differing column sets.

name	sku	image	
Bouncy Castle	35450349822	http://path.jpg	
name	sku	image	description
Coffee Maker	60723023589	http://path.jpg	It makes coffee.
name	sku		
Air Mattress	53045985365		

# Query Language Basics

There are 2 parts to the query language used for BaaS API requests:

- **Path to the collection** – <https://api.usergrid.com/<org>/<app>/users>
- **Query language statement** – after `?ql=`
- **Example** - <https://api.usergrid.com/your-org/your-app/users?limit=5&ql=status='active' and age lt 40 order by name asc>
- **Guidelines:**
  - Any values specified in the query must be enclosed in single-quotes
  - No more than 3 parameters in your queries
  - If your query starts with `select * where`, you can omit that part of the statement and just use the condition in the query. Example:

[https://api.usergrid.com/your-org/your-app/users?limit=5&ql=status = 'active'](https://api.usergrid.com/your-org/your-app/users?limit=5&ql=status='active')

# Query Language Examples

Goal	Example	Notes
Find books whose 'title' property contains the full word "tale".	<pre>/books?ql=select * where title contains 'tale'</pre>	contains looks for the occurrence of a full word anywhere in the searched property. Note that contains will not work on the 'name' property, since it is not full-text indexed in the database.
Find books whose 'title' property contains a word that starts with "ta".	<pre>/books?ql=select * where title contains 'ta*'</pre>	contains will look for partial words if you include a wildcard.
Find books whose title property is exactly and only "A Tale of Two Cities".	<pre>/books?ql=select * where title = 'A Tale of Two Cities'</pre>	The = operator is looking for a match in the entire searched property value. Use a * wildcard to look for some set of the first characters only.



## Order by, Limit, and Cursor

---

- **Order by** - You can return query results that are sorted in the order you specify. Use the order by clause to specify the property to sort by, along with the order in which results should be sorted. The syntax for the clause is as follows:

***order by <property\_name> asc | desc***

- **Limit** - When your query might return more results than you want to display to the user at once, you can use the limit parameter with cursors or API methods to manage the display of results. By default, query results are limited to 10 at a time. You can adjust this by setting the limit parameter to a value you prefer. Max number is 1000.
- **Cursor** - An encoded representation of the query position pointing to a set of results. To retrieve the next set of results, pass the cursor with your next call for most results.

## Matrix Queries

**Matrix Queries** allow you to use embedded query strings in the URL. These are Needed if you want to use multiple filters at multiple levels. This becomes very handy for filtering connected entities.

### Example:

```
/users/ed/friends;ql="location eq new york"/achievements?  
ql="level eq mayor"
```

# Geolocation

- Think of where you can use Geolocation data:

Entity type	Usage
user	Save the location of a user's home as part of their profile in the <code>users</code> collection to suggest upcoming special events or activities located nearby, or to display advertisements that are relevant based on the user's proximity to a business.
device	Periodically save the location data returned from a user's device, then query the <code>devices</code> collection to send offers and alerts to user's that are located near your business with a <a href="#">push notification</a> .
activity	Create stronger social connections by associating a user <code>activity</code> with the location where it occurred. The activity can then be displayed to nearby friends and family, or used to enrich the user's activity stream.
asset	Save user photos with location data in the <code>asset</code> collection to allow users to retrieve and sort their memories based on when and where they happened.

- To store location information in an entity -- "**location**" :  
`{"latitude": <latitude_coordinate>, "longitude": <longitude_coordinate>}`. This can be included either in entity creation or on update.

## Geolocation (continued)

- If you've stored location data with your entities, you can query for the proximity of the geographical locations those entities represent.

Goal	Example	Notes
Find stores whose locations are within the specified longitude and latitude.	<pre>/stores?ql=location within 500 of 40.042016, -86.900749</pre>	<p>within will test for values within the value you specify. The within value is expressed as a number of meters.</p> <p>The return results are sorted in order of nearest to furthest. If there are multiple entries at the same location, they're returned in the order they were added to the database.</p> <p>For more on geolocation queries, see <a href="#">Geolocation</a>.</p>

# Assets

- Asset can be any type of file. It can be used to hold data objects like image, text, video or audio.
- An asset can be uploaded to a collection.
- An asset can also be associated with an entity. Only 1 asset can be attached to an entity.

To create a collection “items” with 2 entities milk and bread:

```
curl -X POST "https://api.usergrid.com/<org>/sandbox/item" -d  
'[{"name":"milk", "price":"3.25"}, {"name":"bread", "price":"2.50"}]'
```

To assign an image bread.jpeg to a an entity “bread” :

```
curl -X POST -F name='bread.jpeg' -F file=@bread.jpeg 'https://  
api.usergrid.com/<org>/sandbox/items/bread'
```

## Assets (continued)

Assets can also be organized into folders. For example all item images can be linked to a folder “images”

To create a folder “images”:

```
curl -X POST 'https://api.usergrid.com/<org>/sandbox/folders' -d  
'{"name": "images", "owner": "<valid user uuid>","path": "/images"}'
```

To assign an asset to a folder:

```
curl -X POST 'https://api.usergrid.com/<org>/sandbox/folders<folder  
uuid>/assets/<asset uuid>'
```

# Push Notifications



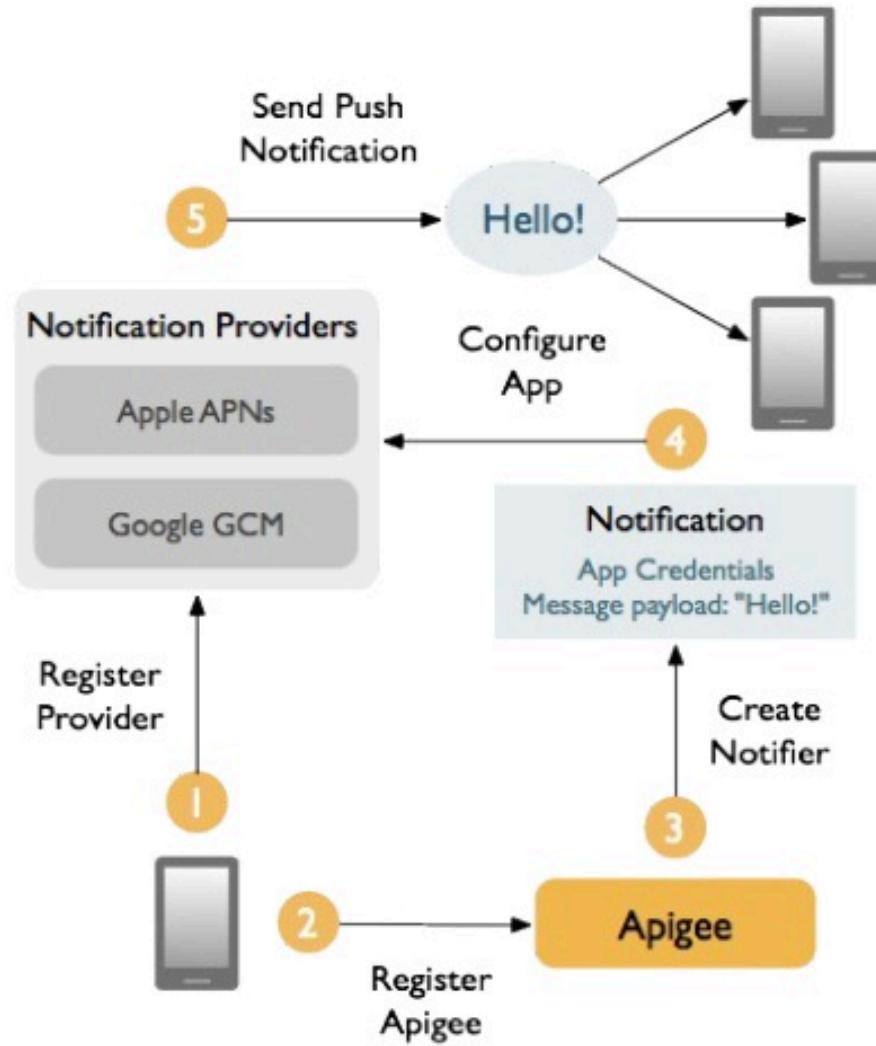


## Push Notifications

---

- Framework for sending push notifications to your apps.
- Messages can be delivered to individual devices, users or group of users.
- Messages can be delivered to users based on their location.
- Notifications can be scheduled.
- BaaS maintains the delivery receipts for notification.

# How Notifications Work





## Push Notifications

---

- To receive notifications application has to register with
  - API BaaS (populates in device collection)
  - Push notification service (Apple APNs or Google GCM)
- Device entity can be tied to users, thereby allowing notifications to be sent to users

# Getting into Baas





Apps Apigee Mail In the Know | Apigee Request Support | Ap Log in to Concur | C OpenAir Other Bookmarks

apigee

Dashboard

srichardson@apigee.com ▾

## API Management

Create & Manage APIs



- Securely expose services
- Publish & monetize APIs
- Analyze API trends

LAUNCH

## API BaaS

Build & Monitor Apps



- Mobile-ready datastore
- Push notifications
- App performance analytics

LAUNCH

## API Consoles

Test & Explore APIs



- Embed anywhere
- Drive adoption
- Customize look-and-feel

LAUNCH

# Another way...

apigee Dashboard APIs Publish Analytics Admin Help richardson+trainingm... API Management

Dashboard / API Proxies / Steves-APIEater

## Steves-APIEater

Deployment to Trace Environment test, Revision 1 ▾ Stop Trace Session Remaining Time: 02:48 Download Trace Session ▾ Node.js Log

Transactions

Status	Method	URI	Elapsed
1 200	GET	/v1/apiateery/trucks?apikey=xZvH0mqNzSNz2oMUA6f7FelbC5aas91n	208 ms

Send Requests

Method URL

GET http://trainingmats-test.apigee.net/v1/apiateery/trucks?apikey=xZvH0mqNzSNz2oMUA6f7FelbC5aas91n

Transaction Map

Phase Details

Request Received from Client  
GET /v1/apiateery/trucks?  
apikey=xZvH0mqNzSNz2oMUA6f7FelbC5aas91n

Response Sent to Client  
200 OK

Request Headers	Response Headers
Accept	/*
Connection	keep-alive
Host	trainingmats-test.apigee.net
User-Agent	NING/1.0
X-Apigee-Trace-Id	f87ce2c9-a44e-40ee-8a37-a1a133d626d4
X-Apigee.application	Steves-APIEater
X-Apigee.environment	test
X-Apigee.organization	trainingmats
X-Apigee.proxy	default
X-Apigee.revision	1
X-Apigee.router.uuid	276741ef-78e0-4b92-ae35-0ab94dd5684b
X-Apigee.vhost	default

View Options

Transaction Map

Show Disabled Policies (none)

Show Skipped Phases (1)

Show All Flowinfos (5)

Phase Details

Automatically Compare Selected Phase

Show Variables

Show Properties

Request Content Response Content

Output from all Transactions

© 2014 Apigee Corp. All rights reserved. Version 140903

# Welcome to Apigee Baas

apigee

Enable Help Take the Tour

srichardson+trainingmats@apigee.com | | | Legacy Portal | API Platform ▾

TRAININGMATS ▾ SANDBOX ADD NEW APP

SDKS AND MODULES

**ORG ADMINISTRATION**

Warning: This application has "sandbox" permissions and is not production ready. Please go to our security documentation to find out more.

**Org Administration** ([need help?](#))

**CURRENT ORGANIZATION**

trainingmats	e06a2c5a-355e-11e4-910d-07094d539f78
--------------	--------------------------------------

**APPLICATIONS** [Add New App](#)

sandbox	e0aab7c0-355e-11e4-8504-072f0663a4c8
---------	--------------------------------------

**ACTIVITIES**

Steven Richardson created a new organization account named trainingmats	Sat, 06 Sep 2014 00:43:53 GMT
---	----------------------------------

**ORGANIZATION API CREDENTIALS**

[Regenerate Org Credentials](#)

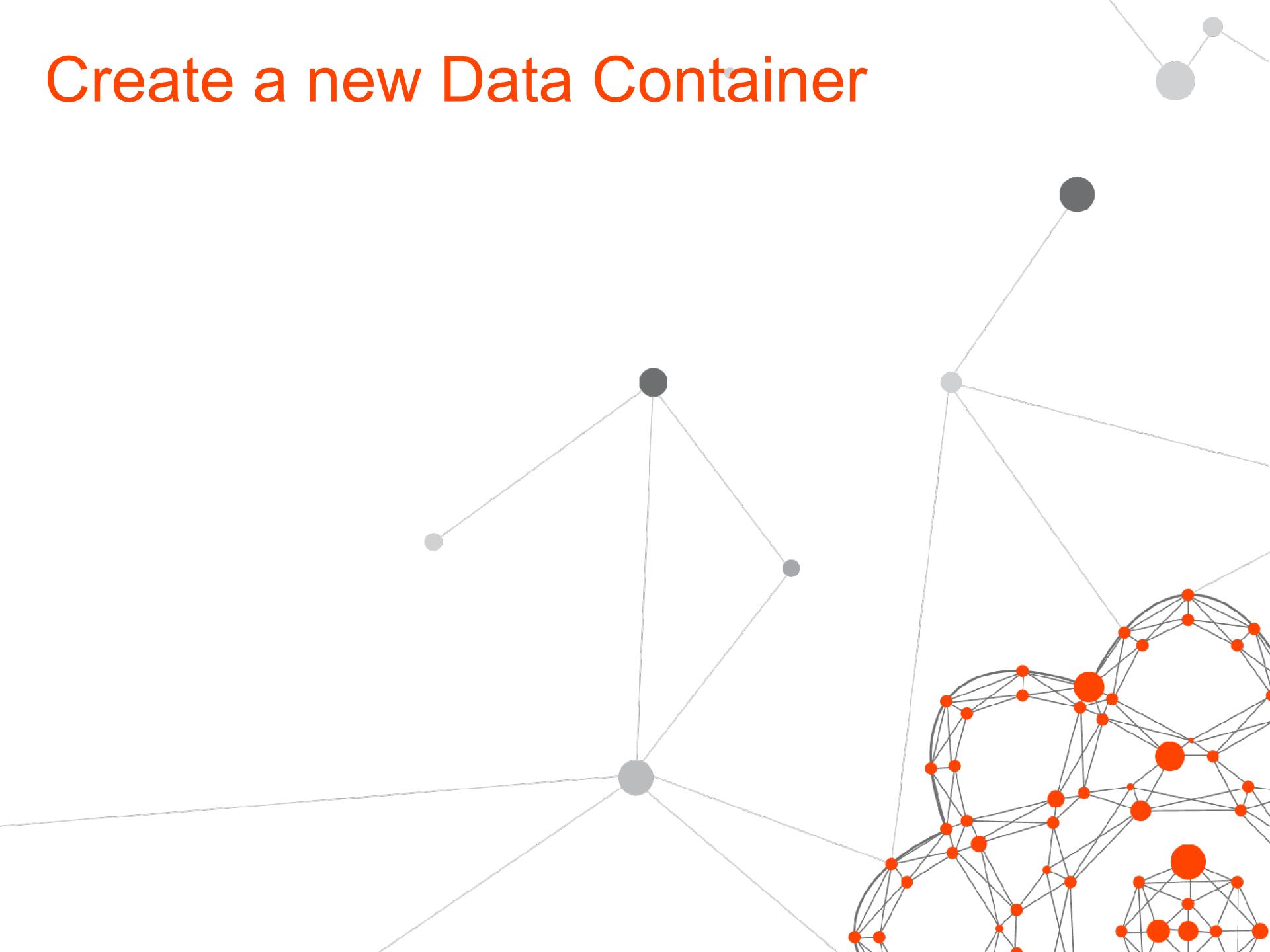
Client ID	b3U64GosWjVeEeSRDQcJTVOfmA
Client Secret	b3U6u33OKIGEHci-um2gVo0AYliLRxA

**ORGANIZATION ADMINISTRATORS**

[Add New Administrator](#)

Steven Richardson	srichardson+trainingmats@apigee.com
-------------------	-------------------------------------

# Create a new Data Container



# BaaS UI



TRAININGMATS ▾  SANDBOX ▾  ADD NEW APP

SDKS AND MODULES     

ORG ADMINISTRATION

GETTING STARTED

APP OVERVIEW

USERS

MONITORING

PUSH

**DATA**

Collections

ACTIVITIES

CONFIGURE

SHELL

**Warning:** This application has "sandbox" permissions and is not production ready. [Please go to our security documentation to learn more.](#)

## Collections (need help?)

 CREATE READ UPDATE DELETE

Path: /activities

Query: ex: select \* where name='fred'

Limit: ex: 10

Run Query

Delete Entity(s)

UUID

## Add a collection

Select the Add Collections Button



Add a ratings collection

A screenshot of a modal dialog box titled "Create new collection". It contains a "Collection Name:" label and a text input field containing "ratings". At the bottom are "Cancel" and "Create" buttons. A large blue arrow points to the "Create" button from the right.

Create new collection	
Collection Name:	<input type="text" value="ratings"/>
<button type="button">Cancel</button>	<button type="button">Create</button>

# Add some data to our collection

Select our new **ratings** collection from menu

## Collections (need help)



/activities

/assets

/devices

/folders

/groups

/ratings

/roles

/users



CREATE    READ    UPDATE    DELETE

Path

/ratings

Query

ex: select \* where name='fred'

JSON Body

```
{  
  "comment": "Great Food, Great Price",  
  "commenter": "Steve Richardson",  
  "score": 5,  
  "truck": "AMS1234"  
}
```

Validate JSON

Run Query

Select the **Create** Radio Button and then add some JSON to represent our data model

## Entity Detail

**comment** : Great Food, Great Price  
**commenter** : Steve Richardson  
**created** : 1410390894238  
**metadata** :  
    **path** : /ratings/45f5bbea-3940-11e4-b587-b5e2b386e216  
**modified** : 1410390894238  
**score** : 5  
**truck** : AMS1234  
**type** : /ratings  
**uuid** : 45f5bbea-3940-11e4-b587-b5e2b386e216

## Edit Entity

```
{  
    "uuid": "45f5bbea-3940-11e4-b587-b5e2b386e216",  
    "type": "/ratings",  
    "created": 1410390894238,  
    "modified": 1410390894238,  
    "comment": "Great Food, Great Price",  
    "commenter": "Steve Richardson",  
    "metadata": {  
        "path": "/ratings/45f5bbea-3940-11e4-b587-b5e2b386e216"  
    },  
    "score": 5,  
    "truck": "AMS1234"
```

Validate JSON

Save

- ORG ADMINISTRATION
- GETTING STARTED
- APP OVERVIEW
- USERS
- MONITORING
- PUSH
- DATA
- ACTIVITIES
- CONFIGURE
- SHELL**

⚠ Warning: This application has "sandbox" permissions and is not production ready. Please do not use this application in a production environment.

### Shell (need help?)

Command Line

Type "help" to view a list of the available commands.

```
>> /ratings
```

```
{  
  "action": "get",  
  "application": "e0aab7c0-355e-11e4-8504-072f0663a4c8",  
  "params": {  
    "access_token": [  
      "YWMt-6iVtjlAEeSpeInuIgj-ngAAAUif6lFl71ZoSmyR4eZQssA0IYdgOL8Wixo"  
    ]  
  },  
  "path": "/ratings",  
  "uri": "https://api.usergrid.com/trainingmats/sandbox/ratings", ← API URL  
  "entities": [  
    {  
      "uuid": "45f5bbea-3940-11e4-b587-b5e2b386e216",  
      "type": "rating",  
      "created": 1410390894238,  
      "modified": 1410390894238,  
      "comment": "Great Food, Great Price",  
      "commenter": "Steve Richardson",  
      "metadata": {  
        "path": "/ratings/45f5bbea-3940-11e4-b587-b5e2b386e216"  
      },  
    },  
  ],  
  "total": 1  
}
```

HTTP/1.1 200 OK  
Date:  
Tue, 07 Jan 2014 01:58:37 GMT  
Access-Control-Allow-Origin:  
\*  
Content-Length:  
705  
x-apigee-serverprocessingtime:  
57  
Set-Cookie:  
rememberMe=deleteMe; Path=/; Max-Age=0; Expires=Mon, 06-Jan-2014 01:58:38 GMT  
Content-Type:  
application/json  
Server:  
Apache-Coyote/1.1

```
{  
  "action": "get",  
  "application": "d0303550-7731-4c9f-833e-cbb5b5",  
  "params": {},  
  
  "uri": "https://api.usergrid.com/bissell@apigee.com/trainingmats/ratings",  
  "entities": [  
    {  
      "uuid": "0eb0812a-7734-11e3-aec0-e54648b8b6c0",  
      "type": "rating",  
      "created": 1389055121970,  
      "modified": 1389055121970,  
      "comment": "Great Food!",  
      "commenter": "Steven Richardson",  
      "score": 5,  
      "truck": "AMS1234",  
      "metadata": {  
        "path": "/ratings/0eb0812a-7734-11e3-aec0-e54648b8b6c0"  
      }  
    },  
    {"timestamp": 1389059918370,  
     "duration": 28,  
     "organization": "srichardson@apigee.com",  
     "applicationName": "apieatery",  
     "count": 1  
  ]}
```

Domain Name

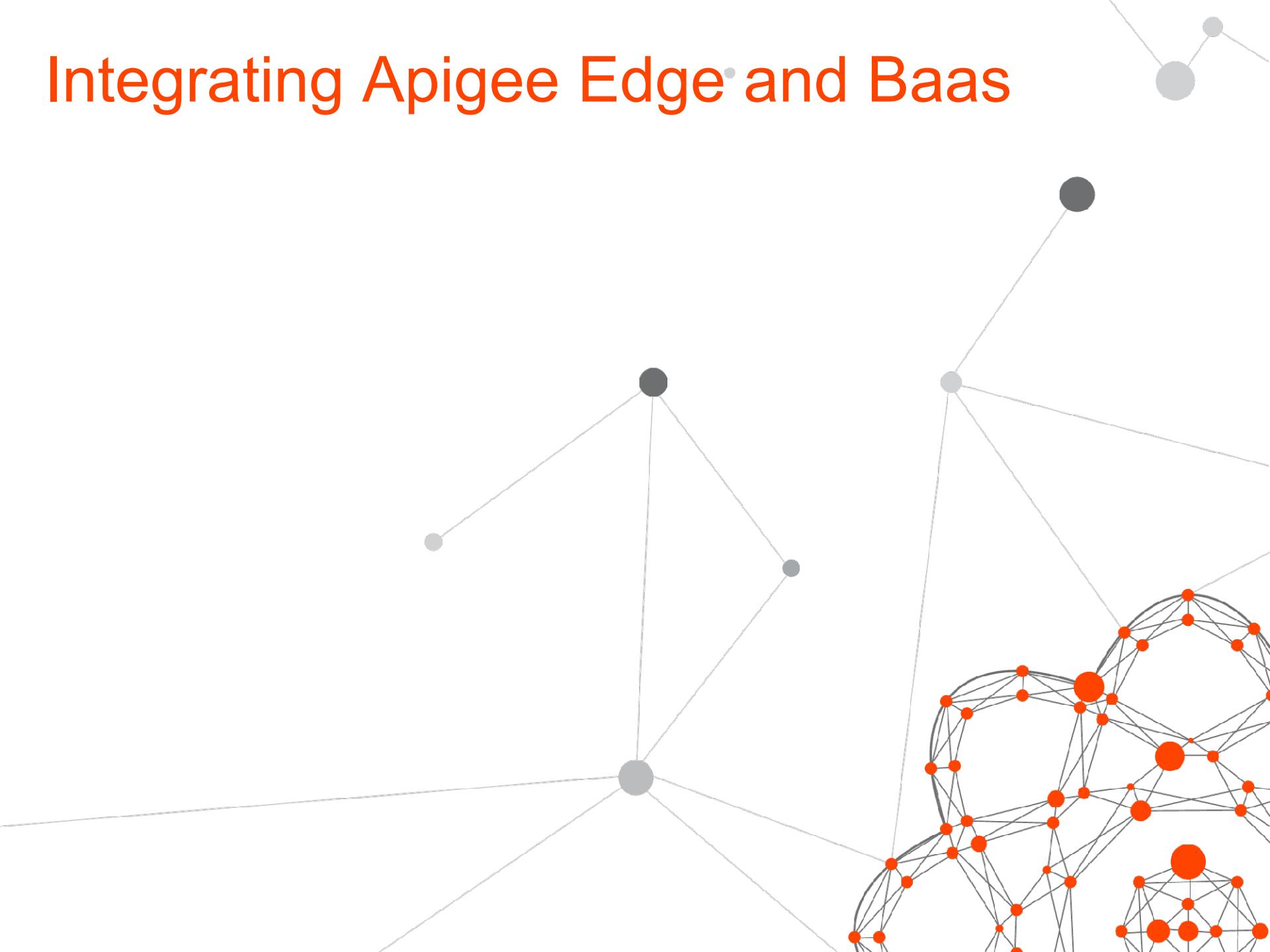
cbb5b5",

Org Name

API Name

Resource/Data Container

# Integrating Apigee Edge and Baas



# Steps to integrate via API

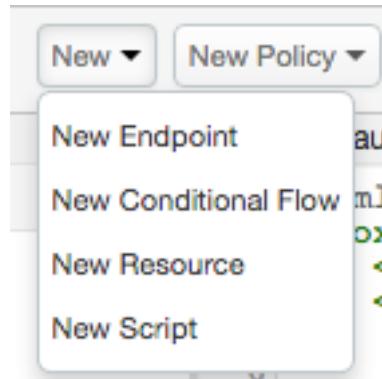
Add GET / POST resources for ratings API

Resources

Name	Proxy Endpoint	Method	Path	URL	Policies	Actions
trucks	default	<button>GET</button>	/trucks	<a href="http://trainingmats-test.apigee.net/v1/apieatery/trucks">http://trainingmats-test.apigee.net/v1/apieatery/trucks</a> [+]	0	<button>Edit</button> <button>Delete</button>
chefs	default	<button>GET</button>	/chefs	<a href="http://trainingmats-test.apigee.net/v1/apieatery/chefs">http://trainingmats-test.apigee.net/v1/apieatery/chefs</a> [+]	0	<button>Edit</button> <button>Delete</button>
cuisines	default	<button>GET</button>	/cuisines	<a href="http://trainingmats-test.apigee.net/v1/apieatery/cuisines">http://trainingmats-test.apigee.net/v1/apieatery/cuisines</a> [+]	0	<button>Edit</button> <button>Delete</button>
routes	default	<button>GET</button>	/routes	<a href="http://trainingmats-test.apigee.net/v1/apieatery/routes">http://trainingmats-test.apigee.net/v1/apieatery/routes</a> [+]	0	<button>Edit</button> <button>Delete</button>
ratings	default	<button>GET</button>	/ratings	<a href="http://trainingmats-test.apigee.net/v1/apieatery/ratings">http://trainingmats-test.apigee.net/v1/apieatery/ratings</a> [+]	0	<button>Edit</button> <button>Delete</button>
post_ratings	default	<button>POST</button>	/ratings			<input checked="" type="checkbox"/> <input type="checkbox"/>

+ Resource

Then add a new Endpoint



Call it baasTarget and press the add button ( make sure target is selected)

New Endpoint

Endpoint Name:

Endpoint Type:  Proxy  Target

## Route Rules

- Route Rules allow you to route requests to different Targets defined within the existing proxy.
- Default Route Rule – will by default route all traffic to the default target

```
<RouteRule name="default">  
    <TargetEndpoint>default</TargetEndpoint>  
</RouteRule>
```

- Lets add another route rule, this time lets make one with a condition

```
<RouteRule name="baasRoute">  
    <TargetEndpoint>baasTarget</TargetEndpoint>  
    <Condition>(proxy.pathsuffix MatchesPath "/ratings")</Condition>  
</RouteRule>
```

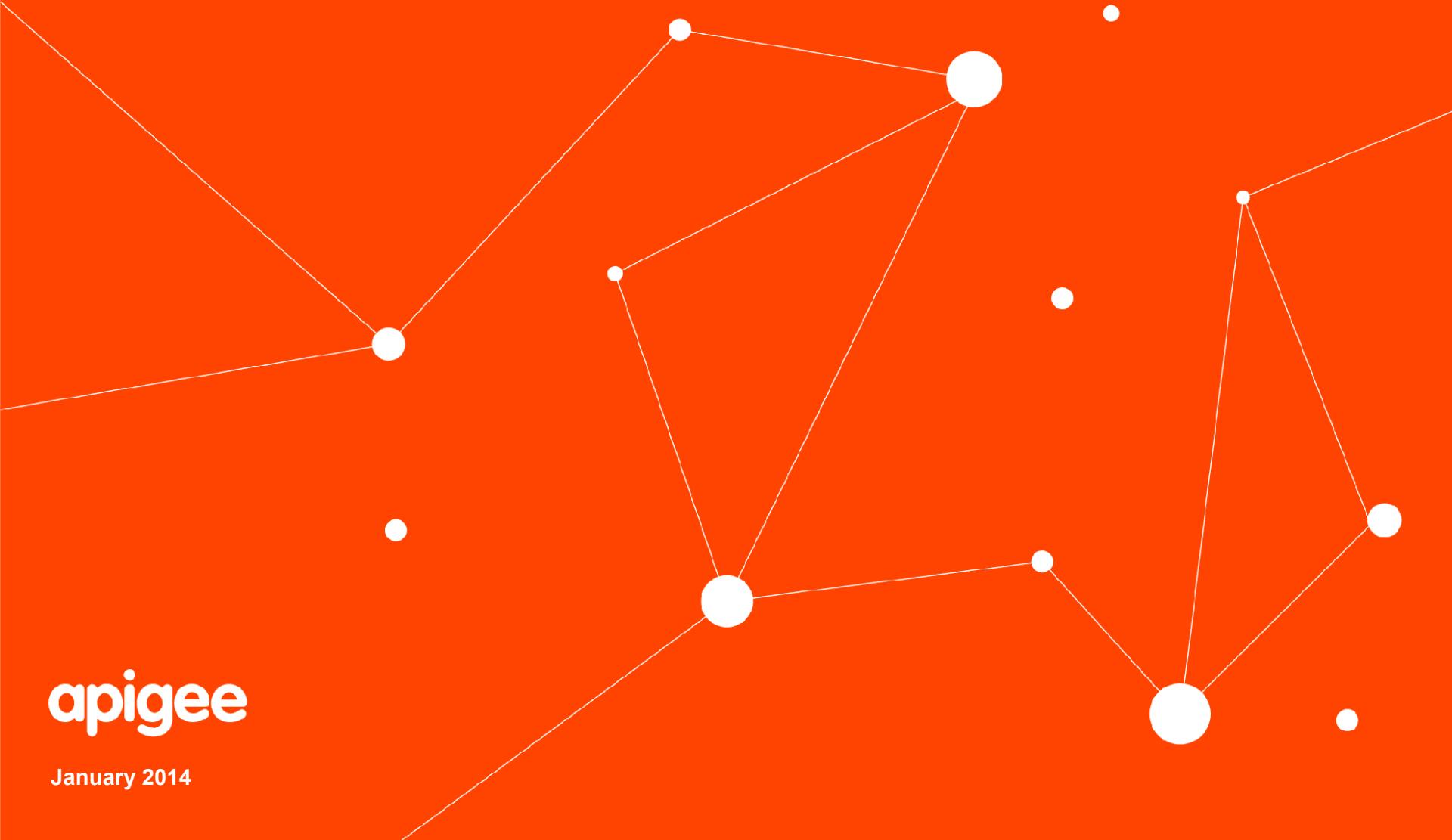
- RouteRules are evaluated top down so make sure your routes are ordered with that in mind

# RouteRules Example

Code: default

```
89     <PostFlow name="PostFlow">
90         <Request/>
91         <Response/>
92     </PostFlow>
93     <HTTPProxyConnection>
94         <BasePath>/v1/apieatery</BasePath>
95         <VirtualHost>default</VirtualHost>
96         <VirtualHost>secure</VirtualHost>
97     </HTTPProxyConnection>
98     <RouteRule name="baasRoute">
99         <TargetEndpoint>baasTarget</TargetEndpoint>
100        <Condition>(proxy.pathsuffix MatchesPath &quot;/ratings&quot;)</Condition>
101    </RouteRule>
102    <RouteRule name="default">
103        <TargetEndpoint>default</TargetEndpoint>
104    </RouteRule>
105 </ProxyEndpoint>
```

# Thank you



apigee

January 2014