Apache Usergrid: An open source (m)BaaS

Lee Grey
lee@apigee.com

Bio

- Solution Architect with Apigee
- Previously at Kony, Intel, NCR, Progress, …
- Ancient past: Avionics
- Georgia Tech EE

- Automated trading system in Groovy

- lee@apigee.com
- @LeeAtApigee (github, twitter, facebook)

Context

Usergrid:

Has been open source since 2011.

Was acquired by Apigee in 2012.

Has been incubating at Apache for about a year.

Is expected to be promoted (hatched?) in a month or two.

Powered by

apache usergrid_

http://usergrid.incubator.apache.org

Features

- A platform, not just a single app
- Users, Groups, Roles, OAuth 2.0 – without code
  - Registration & Login
  - Permissions per object and operation
- JSON data store – Cassandra
  - SQL-like query language
  - Graph queries
- File store – Amazon S3 or other cloud store
  - Text, image, video, etc.
  - Very large file support (TB)

More Features

- SDKs – iOS, Android, HTML5/JS, Node.js, Ruby, Rails, Java, .NET/Windows, PHP, Perl
- Java-based
- Open source
- Trusted
  - Deployed by Korea Telecom, Globo, Apigee and many Fortune 500 companies
  - Hundreds of nodes
  - Millions of users

More Features

- Multi-tenancy at scale
- Deploy to public cloud or traditional servers
- Distributed
- Extensible
- Highly scalable
- Rapid development
- Easy to use from any language

Complete

- Server stack
- Administrative portal website
- SDKs
- Command-line tools

Admin Portal

- HTML5+JS
- Multi-tenant
- Admin
  - Manage developers & features
- Developers
  - Manage apps

Command-line Client

- ugc
- Most maintenance tasks
- Queries, similar to mysql or mongo shell
- sudo gem install ugc
- https://github.com/apache/incubator-usergrid/tree/master/ugc

Server stack

- Java 7
  - Spring
  - Jackson
  - Lucene
  - Shiro
  - Hector -> Astyanax
  - Jersey
  - Mongo-compatible API (binary wire protocol compatible)
  - Netty
- Cassandra
- Maven build

apigee

Usergrid 2.0

- Push
- Elasticsearch (?)

Orgs and Apps

- Multi-tenancy is partitioned into Orgs
  - Company, team, or project
- An Org must have an Admin
  - Full access to any orgs assigned
- Create an App (workspace, database)
- Apps have Users
- Need an access token for both

Why do I need a BaaS?

(m) *obile*

B *ackend*

a *s*

a

S *ervice*

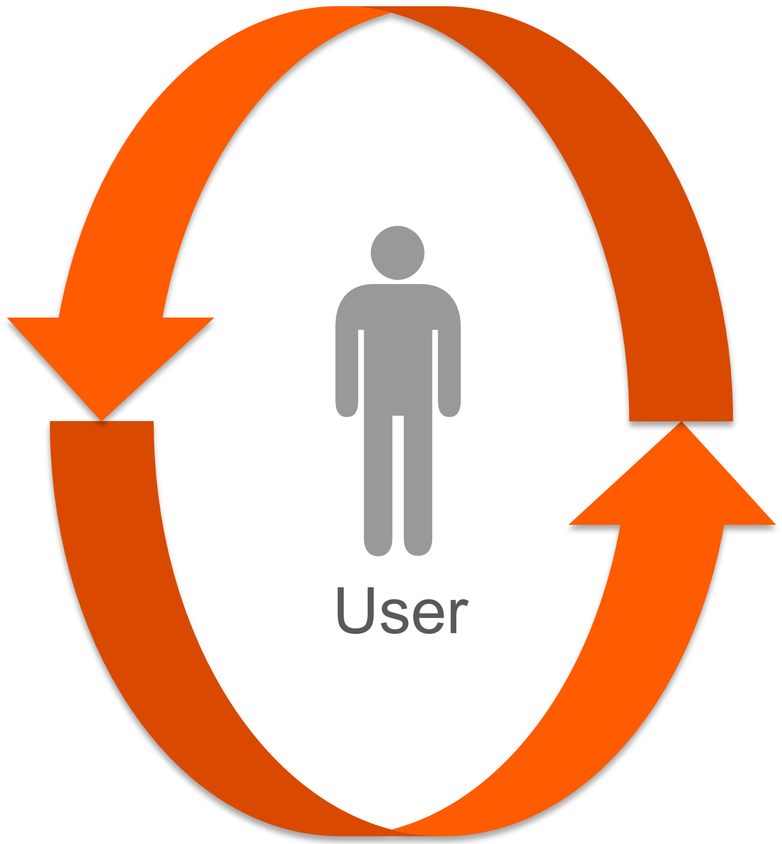
- Datastore
- Connections / Social
- Users
- Location
- Push Notifications
- Performance Mgmt

apigee

Building engaging experiences, powered by APIs

Modern apps are built around the user's context

- ✓ Inherently Mobile
- ✓ Social
- ✓ Location Aware
- ✓ Proactive and Interactive
- ✓ Push notifications (Usergrid 2.0)
- ✓ Agile and Iterative



What can you do with Usergrid?

Datastore

Connections / Social

Users

Location

Push Notifications

Performance Mgmt

Scalable Persistence

Schema-less, multi-datacenter replication.
Automatically exposed as a RESTful API.

apigee

Full-text Indexing

elasticsearch.

Pre-defined Collections

- Users
- Groups
- Folders
- Events
- Assets
- Activities
- Devices
- Notifiers
- Notifications
- Receipts
- Queues
- Any number of user-defined collections
- Schema-less
- Queries are limited in scope to a single collection – no joins

API-Driven Data Access w/ Query

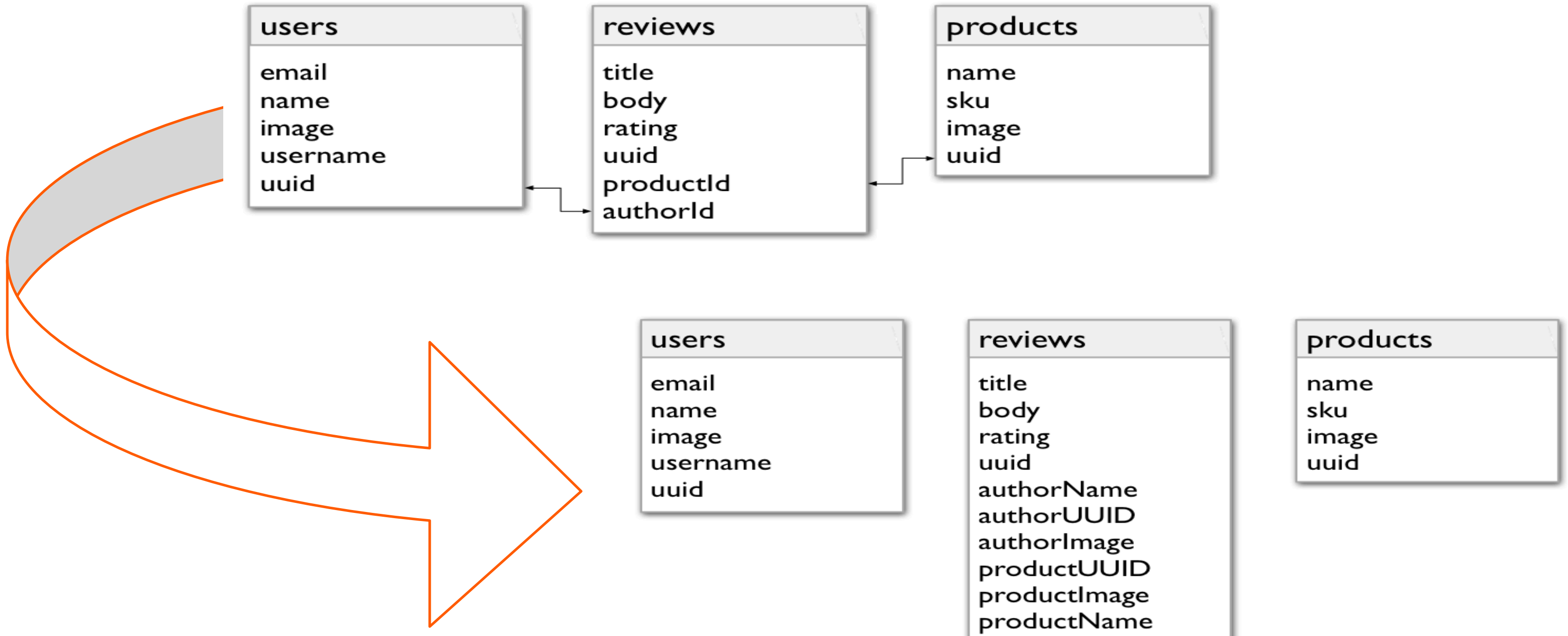
- REST Semantics (CRUD)
  - `GET, PUT, POST, DELETE`
- Filtering:
  - `GET http://api.usergrid.com/MyOrg/MyApp/books?ql=select * where title contains 'ta*'`
  - `GET http://api.usergrid.com/MyOrg/MyApp/books?ql=select * where title contains 'tale`
  - `GET http://api.usergrid.com/MyOrg/MyApp/books?ql=select * where title = 'A Tale of Two Cities'`
- Limit
  - `GET http://api.usergrid.com/MyOrg/MyApp/users?limit=5&ql=status = 'active’`
- Cursors (pagination)
  - `GET http://api.usergrid.com/MyOrg/MyApp/users?ql=select * where name = ’John*'&limit=50&cursor=LTIxNDg0NDUxNDpnR2tBQVFFQWdITUFDWFJ2YlM1emJXbDBhQUNBZFFBUUQyMVZneExxfRWVLR1V3TG9Hc1doZXdDQWRRQVFIYVdjb0JwREV1S1VCd0xvR3NWT0JRQQ`

Querypalooza

- /restaurants?ql=select * where name contains 'diner' order by name asc
- /items?ql=select * where NOT a = 5
- /items?ql=NOT a = 5
- /users?ql=select username,name where name='Gladys Kravitz'
- Date types: string, long, float, boolean, UUID, object
- /mycollection/thing?ql="select * where items.name contains 'rocks'"
- location within <distance in meters> of <latitude>, <longitude>

Data Modeling (duplicate and de-normalize)

- Bringing data together means fewer calls
- Also means less complex queries
- Yes, duplication is ok
- Optimized for reading



TWAE: The Worst Acronym Ever

- HATEOAS
  - Hypermedia As The Engine of Application State
- Responses contain links to related entities
- Traverse an API/data store

Datastore

Connections / Social

Users

Location

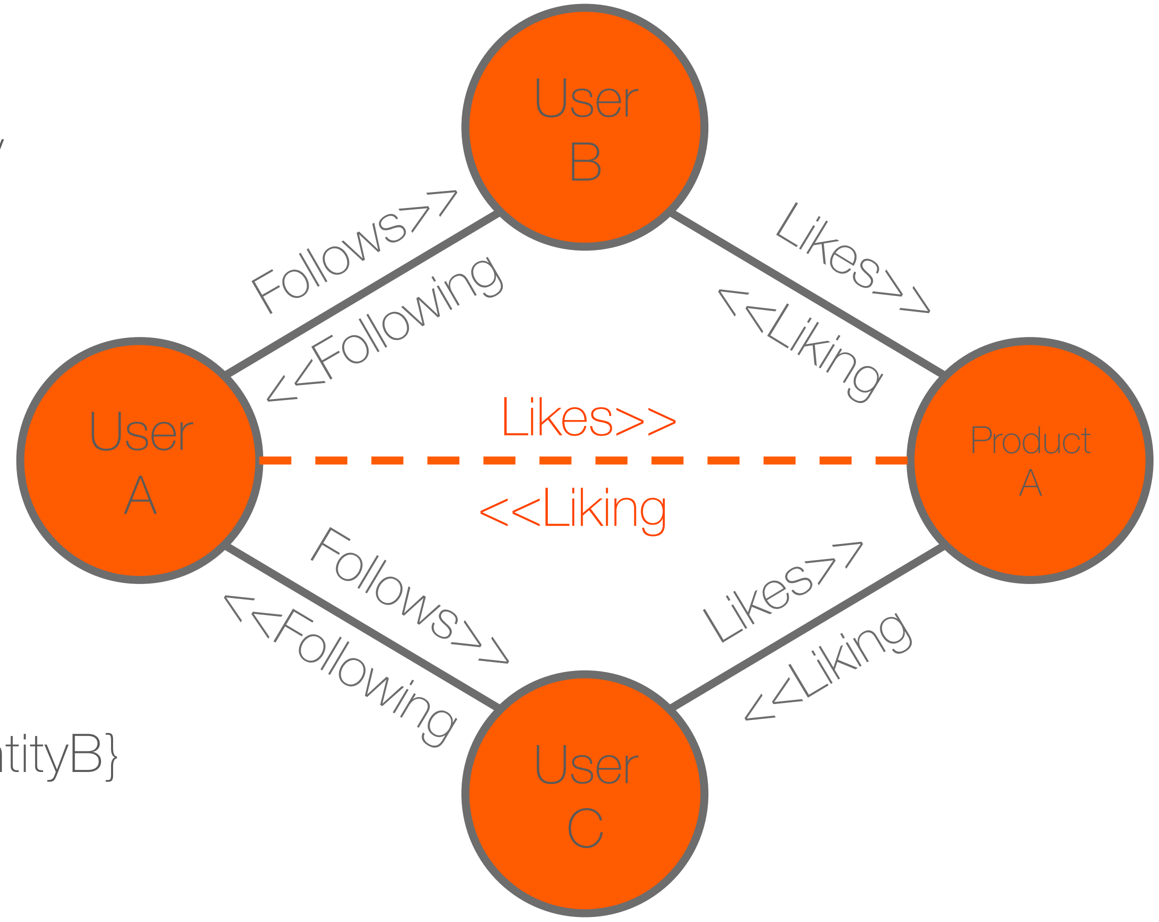
Push Notifications

Performance Mgmt

apigee

Graph Data Persistence: Entities & Connections

- Entities
  - Objects / Documents representing an Entity
- Collections
  - Logical grouping of Entities
- Connections / Entity Relationships
  - Graph Data Structure
  - /{collection}/{entityA}/{verb}/{collection}/{entityB}
  - /Users/me/likes/Products/iPhone6
  - /Recipes/Bread/includes/Ingredients/Flour



Using connections – Following / Followers

Follows/following are a special connection type that forms a reciprocal link – unlike any other connection

```
POST /users/fred/following/users/barney
```

- Fred, who now has a following relationship to Barney will get Barney's posts in his activity stream (just like twitter)
- A reciprocal relationship is created, so this call:

```
GET /users/barney/followers
```

Will return Fred (along with any other followers)

Graph Data Persistence: Entities & Connections

- Creating a connection does not create a reciprocal connection
  - However, you can query either side of the connection
    - http://api.usergrid.com/devnexus/2015/sessions?ql=title contains 'usergrid'
      - Contains a link to the speaker
    - http://api.usergrid.com/devnexus/2015/sessions/9cb14384-c6c4-11e4-bbc8-ab2057fb8d2f/connections
    - http://api.usergrid.com/devnexus/2015/rooms/9fd6031a-c6c4-11e4-bee9-c3c50399c3fe/connecting
    - http://api.usergrid.com/devnexus/2015/timeslots/9fa09b3a-c6c4-11e4-8743-7fd978364ec1/connecting
- Deleting a connection does not delete either of the connected entities

Using Connections

Downside?

- Have to make an extra call or calls during the write
- Can be confusing at first

Upside?

- Optimized for reads so very fast on reads
- Can help structure data in a more normalized way

Connections with Queries

What if we want to query on more fields (e.g. topic = cats)?

Query Approach

```
GET /reviews?ql=select * where author="fred" AND topic="cats"
```

Connection Approach

Just combine queries with connections:

```
GET /users/fred/wrote/reviews?ql=select * where topic="cats"
```

Matrix Queries

NOTE: Matrix queries are not fully supported at this time. Some queries will work, but the feature is not fully complete.

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”
```

Relationships

- Arbitrary connections between entities
- Twitter-like *following* connection
  - Special-case – Usergrid creates mirror *followers* relationship
- Likes, owns, your-favorite-verb
- GET https://api.usergrid.com/my-org/my-app/users/barney/followers
- POST https://api.usergrid.com/my-org/my-app/users/fred/likes/dogs/dino
- GET https://api.usergrid.com/my-org/my-app/users/fred/connections
- GET https://api.usergrid.com/my-org/my-app/users/fred/connecting/likes
- GET https://api.usergrid.com/my-org/my-app/users/fred/likes/dog
- DELETE https://api.usergrid.com/my-org/my-app/users/fred/following/barney

Assets

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>'
```

Datastore

Connections / Social

Users

Location

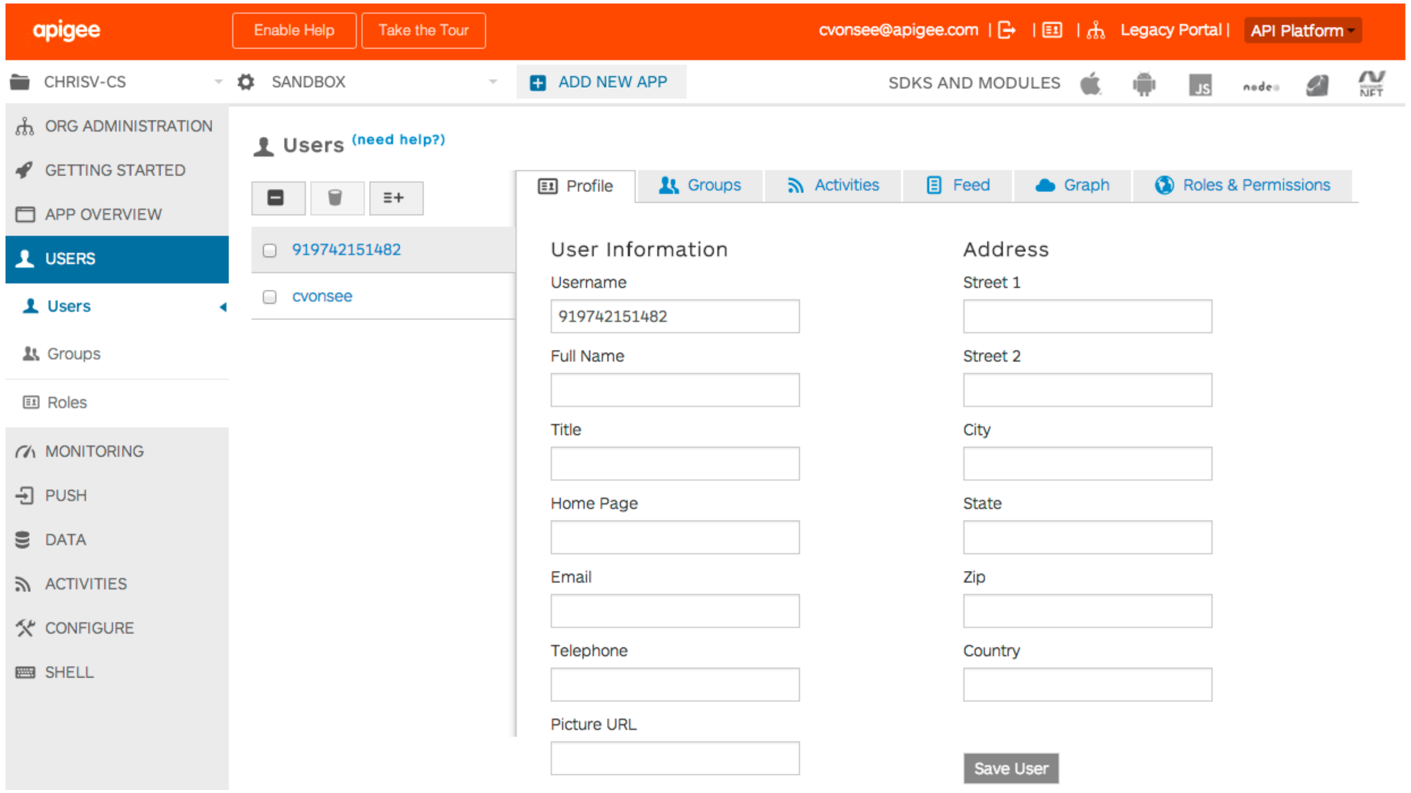
Push Notifications

Performance Mgmt

apigee

Users, Roles & Security

- Organization administrator accounts
- Registration & Login for your app
- End user credentials
- Organization-level client ID and shared secret
- Application-level client ID and shared secret
- Hierarchical Groups
- Fine-Grained User/Group/Role/Permission model
- Facebook OAuth token
- Devices help you track anonymous users & associate Users to Devices



919742151482
cvonsee

User Information

Username: 919742151482
Full Name:
Title:
Home Page:
Email:
Telephone:
Picture URL:

Address

Street 1:
Street 2:
City:
State:
Zip:
Country:

Roles and Permissions

- Role – set of Permissions
- Permissions – HTTP verb + path (optional wildcards)
  - Can be granted to any role, group, or user
  - All permissions are denied, by default
  - A user's permissions are the union of personal, role, and group permissions.
  - Apply only to a single application

Groups

- Organize app users into Groups.
- Groups emulate Facebook Groups.
- Groups have their own activity feeds, their own permissions, and are hierarchical.
- Every member of /groups/proglang/java is a member of /groups/proglang.

```
{
  "path": "comedy",
  ”name": "Comedy Troupes"
}

{
  "path": "comedy/british",
  ”name": ”British Comedy Troupes"
}
```

Activity Streams

- Outbox:
  News items created by user
  /users/{uuid|username}/activities
- Inbox:
  News item feed for the user
  /users/{uuid|username}/feed
- Post activity to a Group's "wall":
  /groups/{uuid|groupname}/activities
- No need to construct publish/subscribe relationships manually
- Post an activity for a user's followers in a group:
  /groups/{uuid|groupname}/users/{uuid|username}/activities

The Messagee sample app is a simple Twitter-style messaging application that leverages the activity stream functionality of Usergrid.

HTML5:
https://github.com/apigee/usergrid-sample-html5-messagee
iOS:
https://github.com/apache/incubator-usergrid-sample-ios-messagee
Android:
https://github.com/apigee/usergrid-sample-android-messagee

Counters

Counters

- Counters can be used to track statistics for an application.
- They can be incremented/decremented by any value or reset to 0.

  To create a counter poll.yes and increment by 1:

  ```
  curl -X POST https://api.usergrid.com/<org>/sandbox/events -d
  '{"timestamp":0, "counters" : { "poll.yes":1}}'
  ```

- Timestamp is specified as UNIX timestamp. A value of 0 assigns the current time
- It may take up to 30 seconds for the counter value to be updated.

  To retrieve the value of the counter:

  ```
  curl -X GET https://api.usergrid.com/<org>/sandbox/counters?
  counter=poll.yes
  ```

Counters (continued)

- Counters can be created and incremented hierarchically using dot notation.
  e.g. poll, poll.yes, poll.no
  Total number of people who took the poll:
  curl -X GET https://api.usergrid.com/<org>/sandbox/counters?counter=poll

Datastore

Connections / Social

Users

Location

Push Notifications

Performance Mgmt

Location Queries

- Add Latitude / Longitude attributes on any Entities. Query any collection with a radius search.
- Radius search:
  - `/stores?ql=location within 500 of 40.042016, -86.900749`
- Combine with Push Notifications for targeted campaigns
  - `/devices;ql="location within 32186.9 of 34.427514,-118.535013"/notifications`

What does Apigee BaaS have that Usergrid doesn’t?

Datastore

Connections / Social

Users

Location

Push Notifications

Performance Mgmt

Flexible Notification Targeting

Push Notification to 20 Miles of a store location

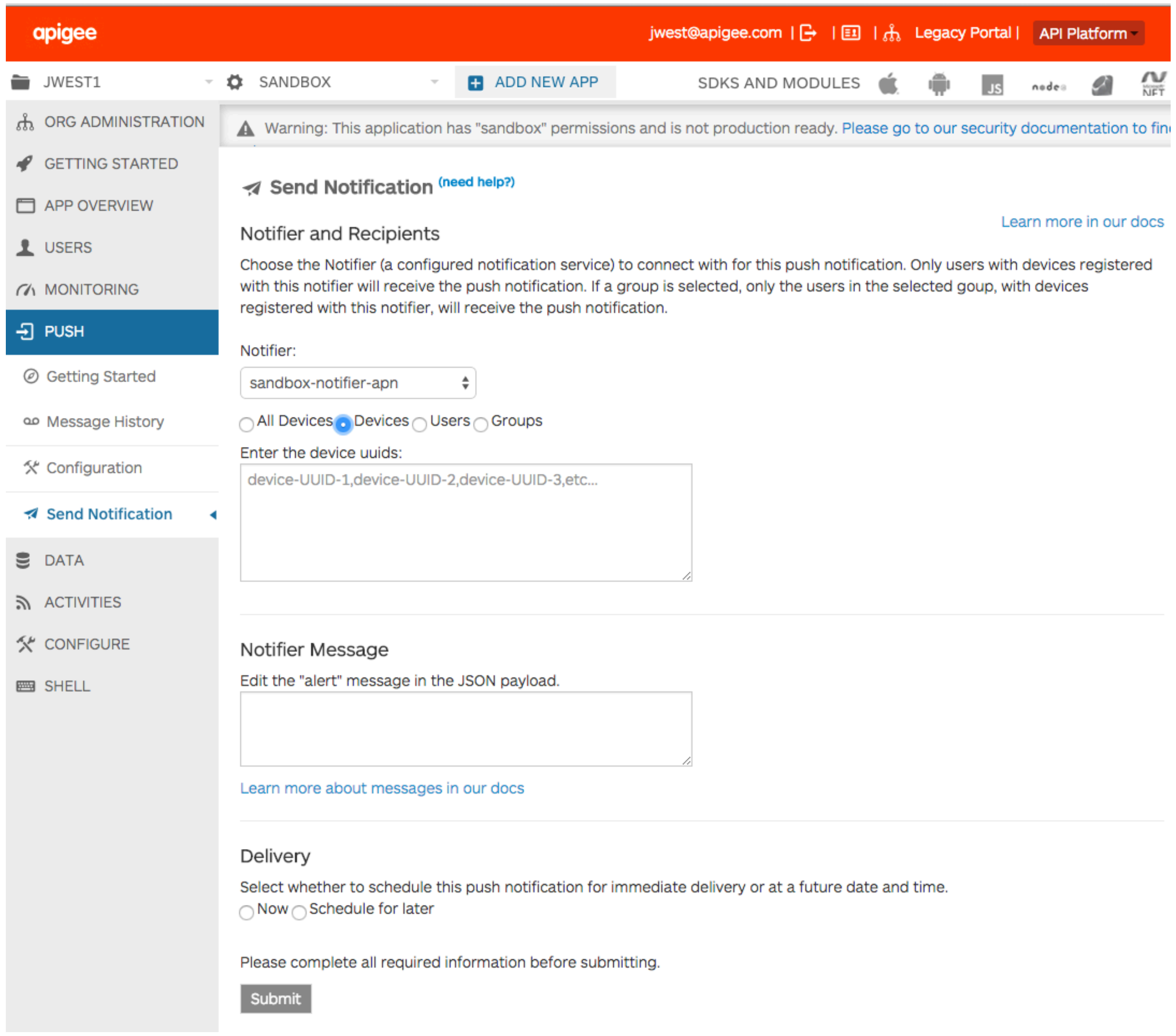
```
POST https://api.usergrid.com/
jeffreyawest/sandbox/
devices;ql="location within 32186.9
of 34.427514,-118.535013"/
notifications
```

Push Notifications to a group of users

```
POST https://api.usergrid.com/
jeffreyawest/sandbox/groups/capos/
notifications
```

Push Notifications to a group of users in a certain location

```
POST https://api.usergrid.com/
jeffreyawest/sandbox/groups/capos/
users;ql=/devices;ql="location within
32186.9 of 034.427514,-118.535013"/
notifications
```



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

Send Notification (need help?)

Learn more in our docs

Notifier and Recipients

Choose the Notifier (a configured notification service) to connect with for this push notification. Only users with devices registered with this notifier will receive the push notification. If a group is selected, only the users in the selected goup, with devices registered with this notifier, will receive the push notification.

Notifier: sandbox-notifier-apn

All Devices [✓ Devices] Users Groups

Enter the device uuids: device-UUID-1,device-UUID-2,device-UUID-3,etc...

Notifier Message

Edit the "alert" message in the JSON payload.

Learn more about messages in our docs

Delivery

Select whether to schedule this push notification for immediate delivery or at a future date and time.

Now Schedule for later

Please complete all required information before submitting.

Submit

Datastore

Connections / Social

Users

Location

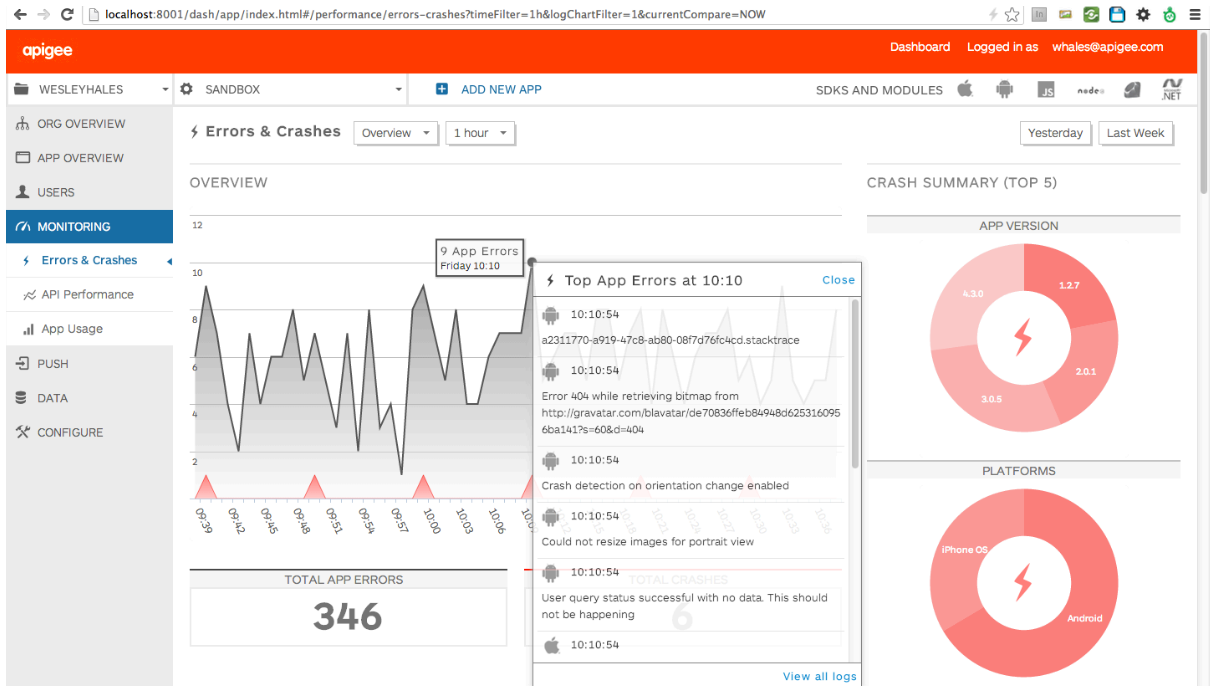
Push Notifications

Performance Mgmt

apigee

Integrated Application Performance Management

- Visualize Errors in Real-Time, slice-n-dice by Version / Device / OS / Carrier
- Detect and diagnose App Crashes
- Research Performance Problems


localhost:8001/dash/app/index.html#/performance/errors-crashes?timeFilter=1h&logChartFilter=1&currentCompare=NOW

Errors & Crashes: Overview, 1 hour

9 App Errors
Friday 10:10

Top App Errors at 10:10

10:10:54
a2311770-a919-47c8-ab80-08f7d76fc4cd.stacktrace

10:10:54
Error 404 while retrieving bitmap from http://gravatar.com/blavatar/de70836ffeb84948d625316095 6ba141?s=60&d=404

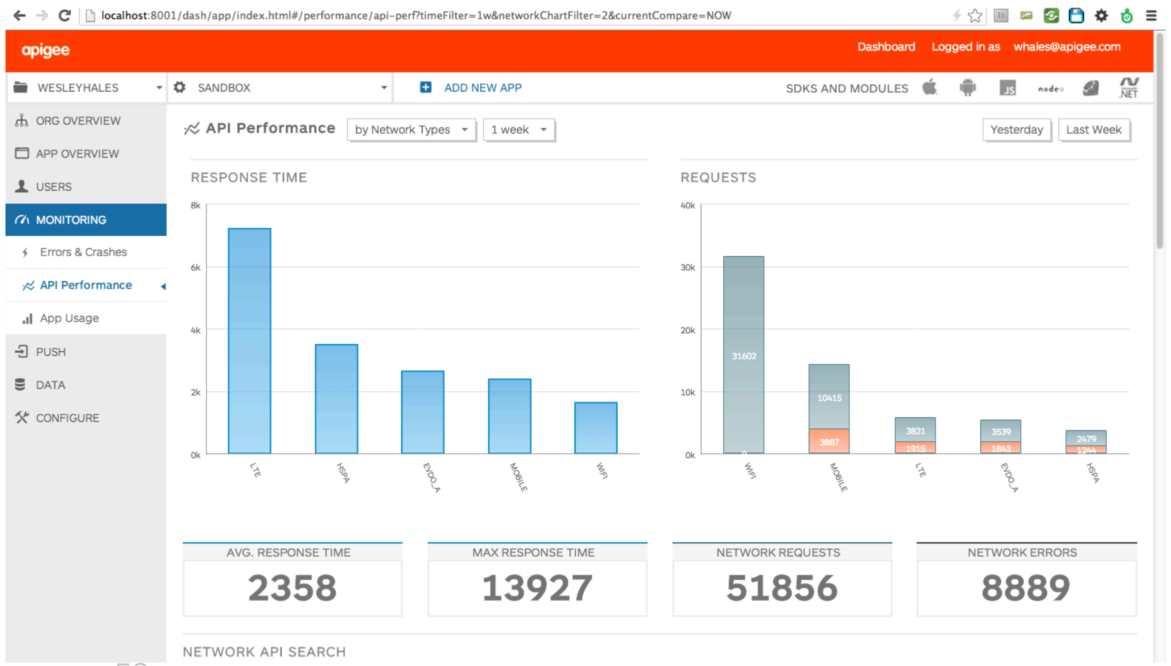
10:10:54
Crash detection on orientation change enabled

10:10:54
Could not resize images for portrait view

10:10:54
User query status successful with no data. This should not be happening

10:10:54

TOTAL APP ERRORS
346


localhost:8001/dash/app/index.html#/performance/api-perf?timeFilter=1w&networkChartFilter=2&currentCompare=NOW

API Performance: by Network Types, 1 week

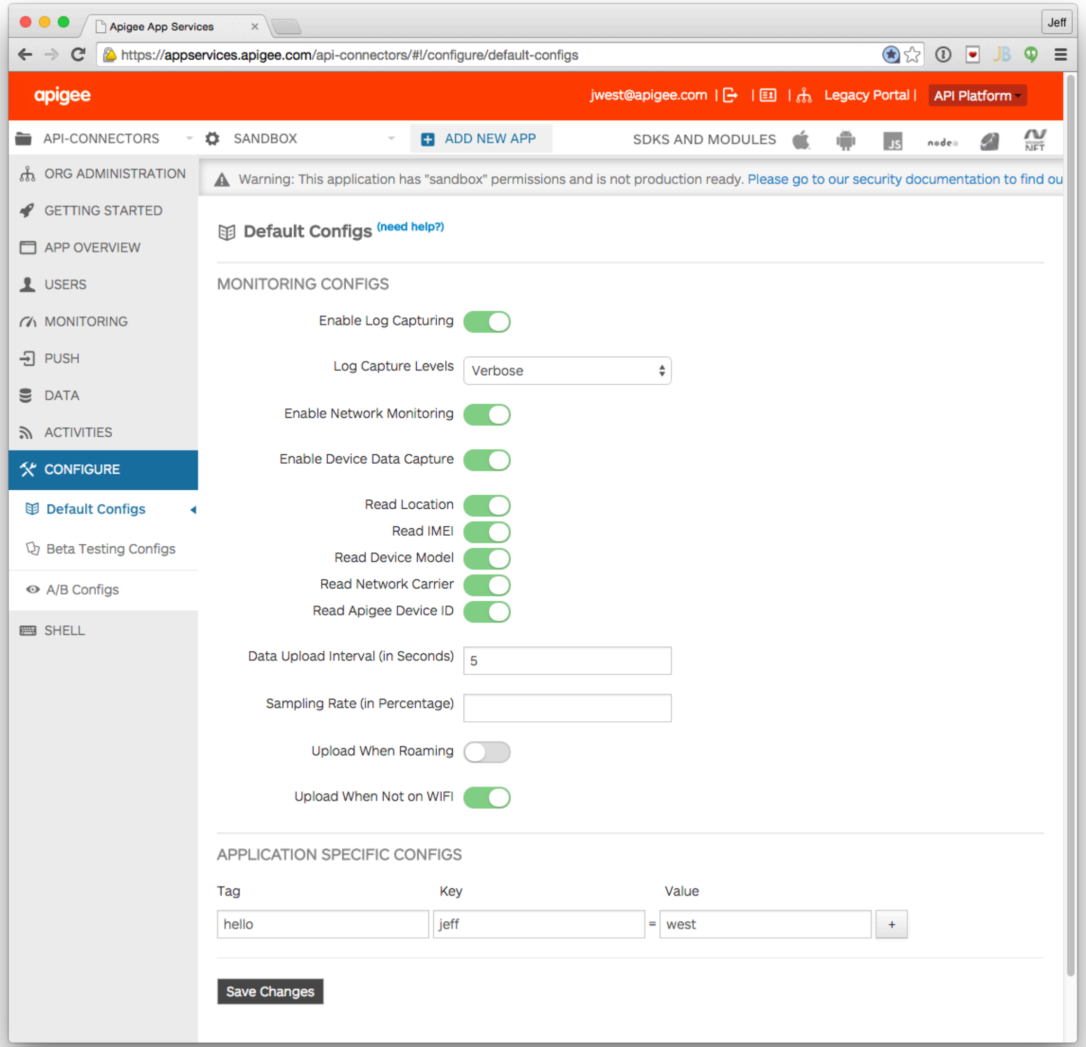
AVG. RESPONSE TIME	MAX RESPONSE TIME	NETWORK REQUESTS	NETWORK ERRORS
2358	13927	51856	8889

- Visualize and track Network Performance by Carrier
- Track API Latency

apigee

Remote Application & SDK Configuration

- Optimize app configuration settings without App Store update
- Push critical configuration fixes in production to fix bugs and improve performance
- Configure SDK Settings for each application
  - Log Capture
  - Log Level
  - Device Data Capture
  - Location Data Capture
  - Configure Sampling



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

Default Configs (need help?)

MONITORING CONFIGS

Enable Log Capturing	On
Log Capture Levels	Verbose
Enable Network Monitoring	On
Enable Device Data Capture	On
Read Location	On
Read IMEI	On
Read Device Model	On
Read Network Carrier	On
Read Apigee Device ID	On
Data Upload Interval (in Seconds)	5
Sampling Rate (in Percentage)	
Upload When Roaming	Off
Upload When Not on WIFI	On

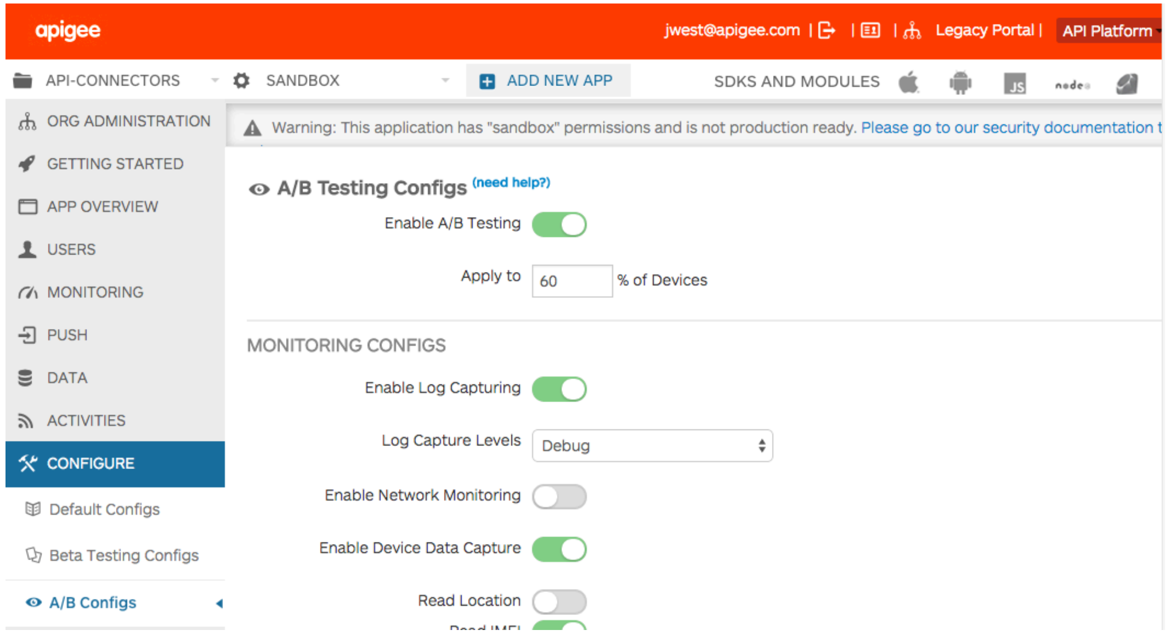
APPLICATION SPECIFIC CONFIGS

Tag	Key	Value
hello	jeff	= west

Save Changes

Beta & A / B Testing Configurations

- Specify distinct device IDs enabled for Beta Testing Configurations
  - Enable configuration profiles for specific devices



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

A/B Testing Configs (need help?)

Enable A/B Testing: on

Apply to 60 % of Devices

MONITORING CONFIGS

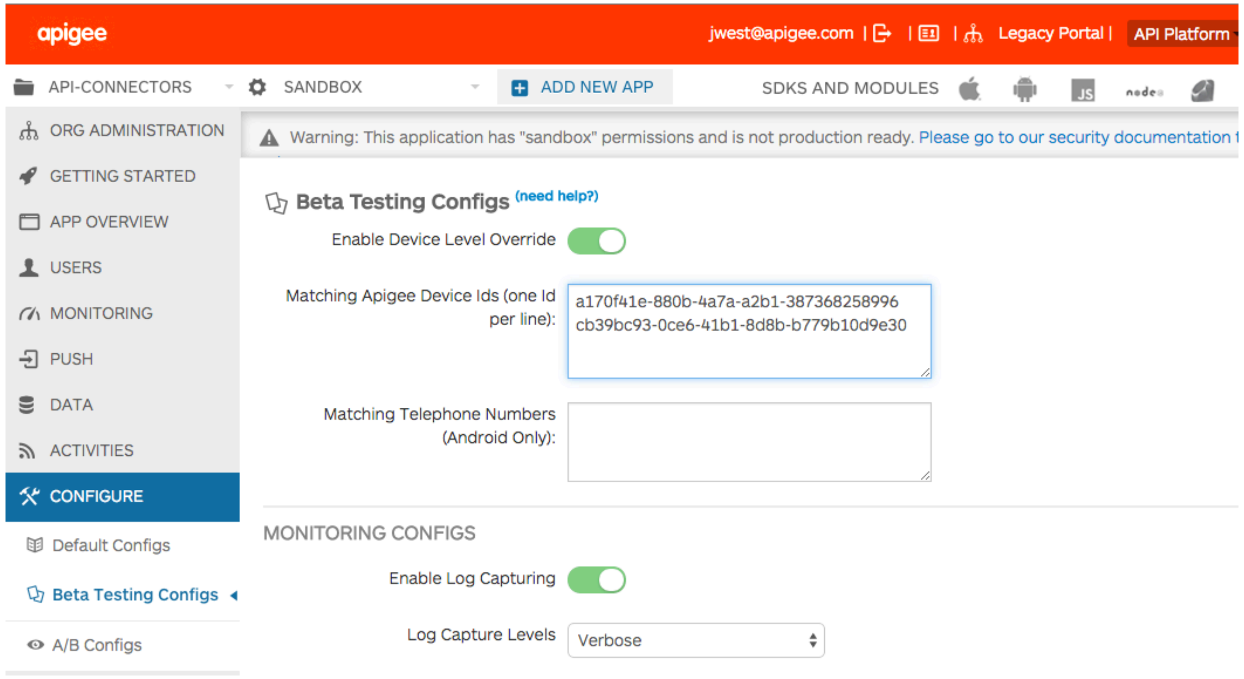
Enable Log Capturing: on

Log Capture Levels: Debug

Enable Network Monitoring: off

Enable Device Data Capture: on

Read Location: off



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

Beta Testing Configs (need help?)

Enable Device Level Override: on

Matching Apigee Device Ids (one Id per line):
a170f41e-880b-4a7a-a2b1-387368258996
cb39bc93-0ce6-41b1-8d8b-b779b10d9e30

Matching Telephone Numbers (Android Only):

MONITORING CONFIGS

Enable Log Capturing: on

Log Capture Levels: Verbose

- Incrementally roll out new configurations with A / B testing
- Manage Testing Configurations applied to a select percentage of Devices/Users

apigee

You Can Do A Lot With Usergrid, But…

- What if you need to add logic?
- You can add it in the client.
- You can add it in a server-side application.
- You can add it in a thin API layer.

Apigee 127

Apigee-127 is toolkit for modeling and building rich, enterprise-class APIs in Node.js on your laptop.

The focal point of Apigee-127 is the Swagger 2.0 specification for defining and describing an API model.

From the Swagger model you can generate clients, servers and interactive documentation for your API.

Apigee 127 in a Nutshell

- https://github.com/apigee-127
- Install: npm install -g apigee-127
- Create app: a127 project create devnexus
- cd devnexus
- Start the API: a127 project start
- New terminal: curl http://127.0.0.1:10010/hello?name=Scott
- In a new tab: a127 project edit
- Install Usergrid: a127 usergrid download
- Start Usergrid: a127 usergrid start
- Open portal: a127 usergrid portal (superuser/superuser, test/test)
- Where from here?
- https://github.com/apigee-127/a127-samples
- https://github.com/apigee-127/example-project

Apigee 127 Capabilities

- Quota
- Spike Arrest
- Caching
- OAuth
- Analytics
- Volos connectors - https://github.com/apigee-127/volos-connectors
  - Databases, SFDC, Amazon, Dropbox, …
- Deploy to Edge
- Integrate with Usergrid
- Mock mode – mock APIs
- Request object has access to all context
  - req.connection.remoteAddress, req.headers, req.url, req.query, req.token, req.swagger.params.client_id.value, req.a127.resource({resource_name})
- Swagger editor
  - *Swagger generator (client and server) – API-driven!*

apigee

Apigee Developer is a free, self-service, non-expiring trial of the Apigee API platform.

Apigee Developer provides the same functionality as Apigee Edge (with some limitations). This includes Usergrid (Apigee BaaS).

http://apigee.com/docs/developer-vs-edge

Resources

http://usergrid.incubator.apache.org/
http://github.com/apache/incubator-usergrid
http://apigee.com/docs/developer-vs-edge

http://youtube.com/apigee
https://apigee.com/about/resources

http://apigee.com/docs/api-baas/content/sdks
https://www.npmjs.com/package/usergrid

https://blog.apigee.com/taglist/apigee_127
http://generator.swagger.io/

More Resources

https://github.com/leeatapigee/devnexus

https://github.com/leeatapigee/devnexus-to-usergrid

https://devnexus-test.apigee.net/devnexus/

Deployment

Install yourself (Tomcat and Cassandra):
http://usergrid.incubator.apache.org/docs/deploy-local/

Docker:
https://github.com/johnament/usergrid-launcher-docker

Brooklyn:
https://github.com/brooklyncentral/baas-usergrid

Apigee 127:
npm install -g apigee-127

Apigee Developer:
https://accounts.apigee.com/accounts/sign_up?callback=https%3A%2F%2Fenterprise.apigee.com

Help

- https://groups.google.com/forum/#!forum/usergrid
- https://twitter.com/usergrid
- http://usergrid.incubator.apache.org/community/#live
  - HipChat
  - IRC
  - Mailing lists for Users and Dev
- http://stackoverflow.com/search?q=usergrid
- Jira
  - https://issues.apache.org/jira/secure/RapidBoard.jspa?rapidView=23&view=planning&selectedIssue=USERGRID-362&epics=visible

Q & A

Thank you!