Google Cloud

API BaaS

Data Model

Entities

There are pre-existing default data entity types (user, group, role, application, etc.)

You can partially update an entity by sending a PUT request with only the attributes' name/value pairs to be updated.

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).

Collections

There are pre-existing default collections (users, groups, roles, applications, etc.)

Updating a collection allows you to batch update entities at the same time. Provide multiple entity objects inside of an array.

Collections cannot be deleted; you can only empty them.

When accessing a collection, a single page is returned and 10 entities is the default limit for the page count.

Data modeling

De-normalize and duplicate to avoid costly queries.

This takes some additional planning depending on the expected queries.

Data modeling

Leverage built-in 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://apibaas-trial.apigee.net/my_org/my_app/reviews?ql=select * where
    authorUUID=<user_uuid>
```

Connection Approach

```
GET http://apibaas-trial.apigee.net/my_org/my_app/users/jimmy/wrote/reviews
```

Data modeling

How NOT to model your API BaaS data store:

Highly normalized data in multiple locations

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. entity array.

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

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

Standard collections

Activity - represents a user activity, and is specifically designed for use in data streams as defined by the JSON Activity Streams 1.0 specification.

Application (hidden in UI) - the base entity for accessing your application data in API Services. Aside from creating the application entity, most apps using API Services will never need to access the application entity directly.

Asset - represents a binary data object stored in API Services infrastructure, such as an image, video or audio file. The asset entity does not contain the binary data, but rather contains information about the data and points to the location where it can be accessed in API Services infrastructure.

Standard collections

Device - represents a unique device that is being used to access your app. Device entities should be associated with a user entity. The API Services push notification feature requires the device entity.

Event (hidden in UI) - used to log application data, primarily for performance and error monitoring. Event entities can be also associated with users and groups. The event mechanism in API Services is optimized to handle large numbers of events, so it is an ideal mechanism for logging in your application.

Folder - used to emulate a file structure for the purpose of organizing assets or custom entities.

Standard collections

Group - allows you to group users based on any criteria. Multiple group entities can be nested to create sub-groups. Users can also belong to multiple groups. Examples of uses for the group entity include grouping users by interest or location.

Notification - represents a push notification, including notification message and details. A notification entity is sent in conjunction with a notifier entity to a notification service, such as Apple Push Notification Service, to initiate a push notification.

Notifier - contains the credentials necessary to securely access push notification service providers, which in turn send your notifications to targeted devices.

Standard collections

Receipt (hidden from UI) - created after a push notification has been sent using API Services. The receipt is a record of an attempted push notification, including if the notification was successful, and when it was sent.

Role - used to define standard permission sets that can be assigned to user and groups entities. For example, you might create an Administrator role to easily grant certain users full access to all app features.

User - represents a registered user of your app, and includes optional properties for common user details, such as real name, email address, and password.

THANK YOU