Hackathon – Postman and General FAQ

Current by GE - Cities

May 2017

Hackathon – General FAQ for Cities

Copyright©2016 General Electric Company. All rights reserved.

GE, the GE monogram, Predix, and Current powered by GE, are either registered trademarks or trademarks of General Electric Company. All other trademarks are the property of their respective owners.

This document may contain Confidential/Proprietary information of GE, GE Software, Current powered by GE, and/or its suppliers or vendors. Distribution or reproduction is prohibited without permission.

THIS DOCUMENT AND IT CONTENTS ARE PROVIDED "AS IS," WITH NO REPRESENTATION OR WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF DESIGN, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. ALL OTHER LIABILITY ARISING FROM RELIANCE UPON ANY INFORMATION CONTAINED HEREIN IS EXPRESSLY DISCLAIMED.

Access to and use of the software described in this document is conditioned on acceptance of a duly executed agreement with GE

Agreement and compliance with its terms.

Table of Contents

Overview

This FAQ is designed for the Current by GE Hackathon users. Make sure you have all of the following at the start of your hackathon efforts, which you should receive by email or from a resources hub:

a. *The **Developer Onboarding Sheet for the Predix Hackathon.** This document has everything you need to get started - links to documentation, Login information, Authentication information, files, and so on.*

b. *Download the **Hackathon VPC Postman Collection json file**. This json file imports Postman collections, which allow you to use Postman to call the APIs and assets for the hackathon.*

This file is named something like: **IC-VPC-APIs-Hackathon.postman_collection.json**.

c. *Download the **Hackathon VPC Postman Environment json file.** This file organizes the variables and URLs utilized by Postman.*

The file is named something like: **IC-VPC-APIs-Hackathon.postman_environment.json.**

If you do not have these artifacts, contact your Hackathon organizer before beginning.

Postman

One of the best ways to utilize the *Current by GE* Intelligent Environments APIs and other RESTful APIs is to use Postman, an open source software tool from Postdot Technologies, Inc. Before developing apps and running APIs using Postman, you should download and install Postman collections and install the *Current by GE* specified Postman environment.

Download Postman

The open source developer App Postman can enable you to develop *Current by GE* APIs. Postman is free to use, and can be downloaded directly onto your development machine.

First, download and install Postman. Start by going here: Install Postman

Download and Import the Postman Collection and Environment files

You need to install collections into Postman for everything to work properly. Collections allow you to organize your calls (GET, POST, etc.) from a single location.

Download the json files (Collections and Environment). Open Postman, click `File / Import` to open the Importer.

You can select the files from where you have stored them, or drag and drop the Collection and Environment files into the Importer. Postman will automatically load the files.

You can also get general information about Postman collections here: Get started with Collections

Environment setup

You will need to setup your environment so it can utilize variables in the API calls.

You can Import an environment in the same way you import a collection. Open Postman, click `File / Import` to open the Importer.

You can select the files from where you have stored them, or drag and drop the Collection and Environment files into the Importer. Postman will automatically load the files.

Here is more information on Setting up an environment with variables.

Important Note: You will be sent the Onboarding Sheet, Postman collections and Environment files in the Hackathon materials by email or on your hackathon resources hub. ***Please see your hackathon organizer / administrator if you do not have these materials.***

Note*:* If you do not import and install Postman collections and setup an environment, Postman will not work properly!

Directions for updating the Postman token

Easy, generate the client token and use it. Easy, right?

First step: Update client token

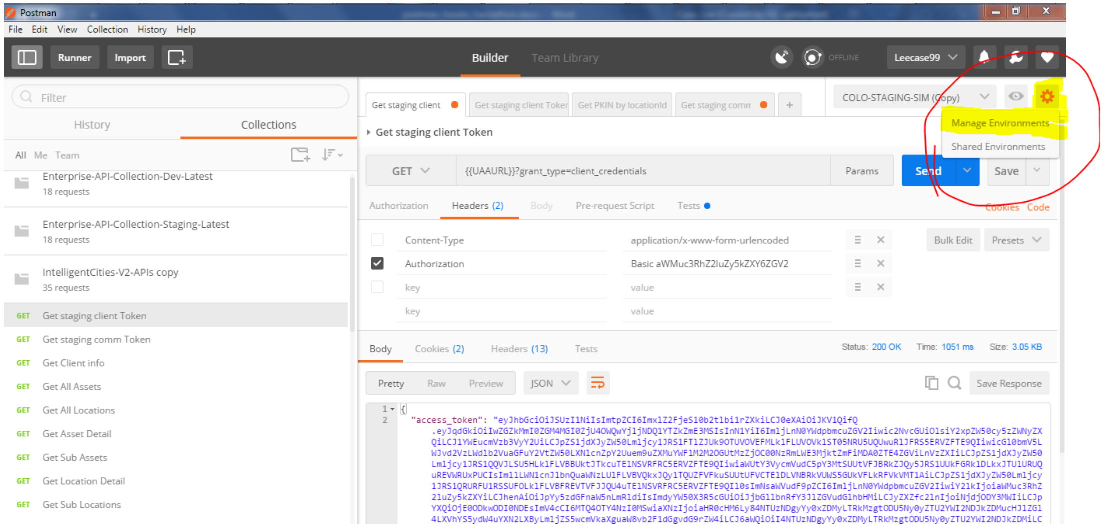
2. *Make sure you have the Developer Onboarding sheet for **your hackathon** which includes:*
    a. *User Account & Authentication (**UAA**) information;*
    b. ***Client ID** & **Client Secret** login information.*
3. *Make sure you have already setup and selected the correct Environment (top right in Postman);*
4. *Authorization tab: set to **Basic Auth**.*
5. *Input the **Client ID** and **Client Secret** from your onboarding sheet into the **Username** and **Password** fields. They will be something like this:*
    a. *hackathon*
    b. *@hackathon*
6. *Click the **Update Request** button to update the authorization Key.*
7. *On the Headers tab, you should see the updated Basic Auth. Key. It will look something like this:*
    a. *< **aGFja2F0aG9uOkBoYWNrYXRob24=** >.*
8. *The Get Call uses this URL for the token:* `{{UAAURL}}?grant_type=client_credentials`

    The UAAURL should already be in the Environment key value variables, and will populate this variable.

9. *If the UAAURL is missing or not updated in the variables, follow this process:*

Navigate to environment variables…

a. *Click on the sprocket icon (far upper right); click **Manage Environments**;*



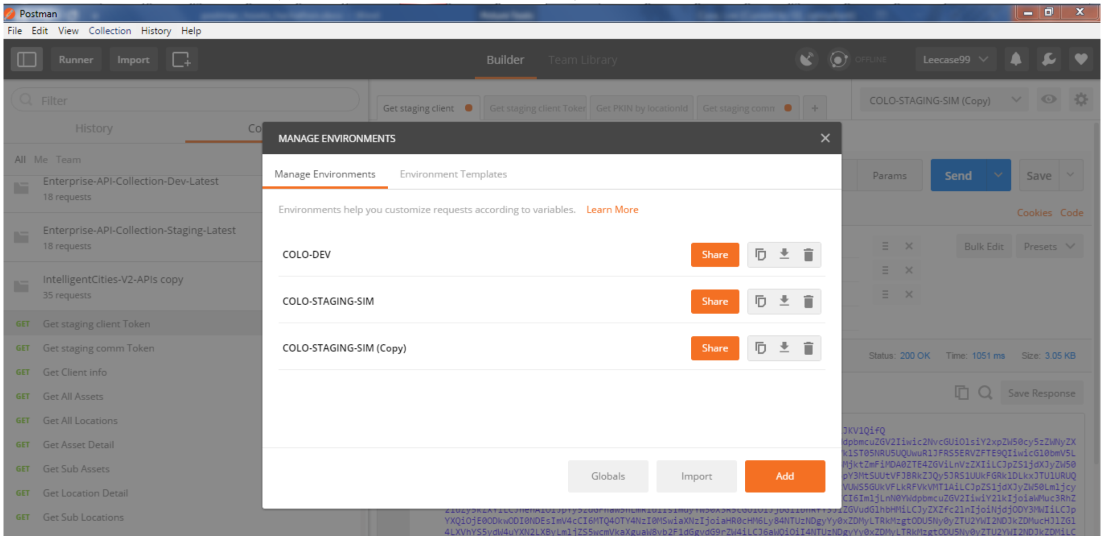
GET {{UAAURL}}?grant_type=client_credentials

□	Content-Type	application/x-www-form-urlencoded
☑	Authorization	Basic aWMuc3RhZ2luZy5kZXY6ZGV2
□	key	value
	key	value

Status: 200 OK Time: 1051 ms Size: 3.05 KB

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2FjeS10b2tlbi1rZXkiLCJ0eXAiOiJKV1QifQ
      .eyJqdGkiOiIwZGZkMmI0ZGM4MGI0ZjU4OWQwYjljNDQ1YTZkZmE3MSIsInN1YiI6ImljLnN0YWdpbmcuZGV2Iiwic2NvcGUiOlsiY2xpZW50cy5zZWNyZX
      QiLCJ1YWEucmVzb3VyY2UiLCJpZS1jdXJyZW50Lmljcy1JRS1FT1ZJUk9OTUVOVEFMLk1FLUVOVk1ST05NRU5UQUwuR1JFRS5ERVZFTE9QIiwicG10bmV5L
      WJvd2VzLWdlb2VuaGFuY2VtZW50LXNlcnZpY2Uuem9uZXMuYWF1M2M2OGUtMzZjOC00NzRmLWE3MjktZmFiMDA0ZTE4ZGViLnVzZXIiLCJpZS1jdXJyZW50
      Lmljcy1JRS1QQVJLSU5HLk1FLVBBUktJTkcuTE1NSVRFRC5ERVZFTE9QIiwiaWUtY3VycmVudC5pY3MtSUUtVFJBRkZJQy5JRS1UUkFGRk1DLkxJTU1URUQ
      uREVWRUxPUCIsImllLWN1cnJlbnQuaWNzLU1FLVBVQkxJQy1TQUZFVFkuSUUtUFVCTE1DLVNBRkVUWS5GUkVFLkRFVkVMT1AiLCJpZS1jdXJyZW50Lmljcy
      1JRS1QRURFU1RSSUFOLk1FLVBFREVTVFJJQU4uTE1NSVRFRC5ERVZFTE9QIl0sImNsaWVudF9pZCI6ImljLnN0YWdpbmcuZGV2IiwiY2lkIjoiaWMuc3RhZ
      2luZy5kZXYiLCJhenAiOiJpYy5zdGFnaW5nLmRldiIsImdyYW50X3R5cGUiOiJjbGllbnRfY3JlZGVudGlhbHMiLCJyZXZfc2lnIjoiNjdjODY3MWIiLCJp
      YXQiOjE0ODkwODI0NDEsImV4cCI6MTQ4OTY4NzI0MSwiaXNzIjoiaHR0cHM6Ly84NTUzNDgyYy0xZDMyLTRkMzgtODU5Ny0yZTU2YWI2NDJkZDMucHJ1ZG1
      4LXVhYS5ydW4uYXN2LXByLmljZS5wcmVkaXguaW8vb2F1dGgvdG9rZW4iLCJ6aWQiOiI4NTUzNDgyYy0xZDMyLTRkMzgtODU5Ny0yZTU2YWI2NDJkZDMiLC
```

b. *Click on the Environment name. Use the Environment for your Hackathon, for example:* ***IC-VPC-APIs-Hackathon.postman_environment.json****}.*



MANAGE ENVIRONMENTS

Manage Environments | Environment Templates

Environments help you customize requests according to variables. Learn More

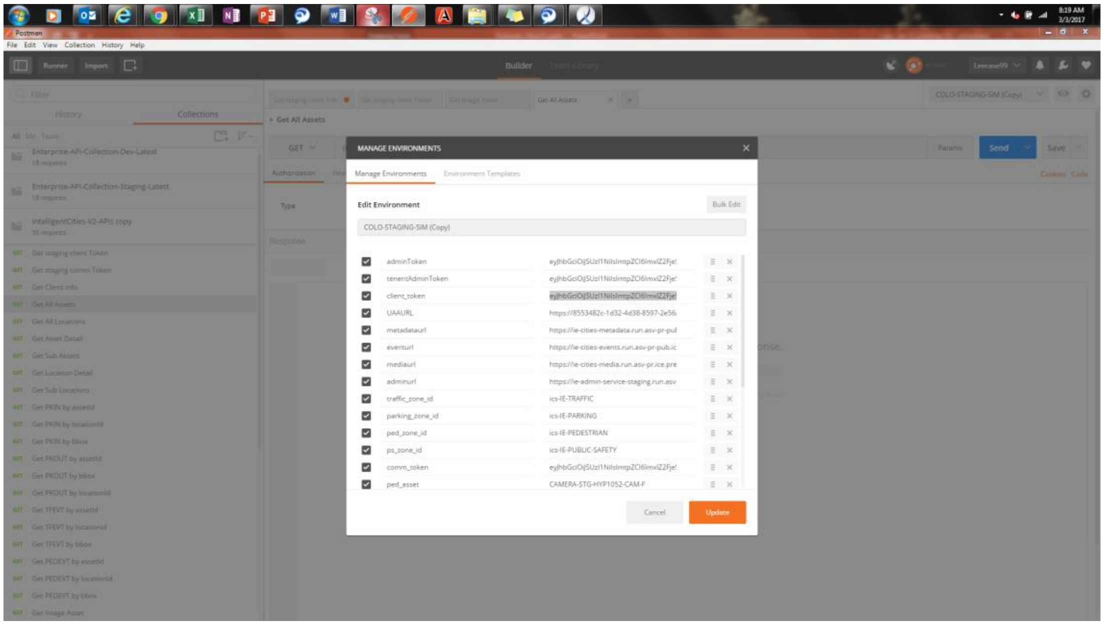
COLO-DEV — Share

COLO-STAGING-SIM — Share

COLO-STAGING-SIM (Copy) — Share

Globals | Import | Add

c. *You should then see the* ***Edit*** *Environment screen. This will bring up a list of key value variables. The 'client token' key value is the value that will get updated with a new client token value.*



MANAGE ENVIRONMENTS

Manage Environments | Environment Templates

Edit Environment — Bulk Edit

COLO-STAGING-SIM (Copy)

☑	adminToken	eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2Fje!
☑	tenentAdminToken	eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2Fje!
☑	client_token	eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2Fje!
☑	UAAURL	https://8553482c-1d32-4d38-8597-2e56
☑	metadataurl	https://ie-cities-metadata.run.asv-pr-pul
☑	eventurl	https://ie-cities-events.run.asv-pr-pub.ic
☑	mediaurl	https://ie-cities-media.run.asv-pr.ice.pre
☑	adminurl	https://ie-admin-service-staging.run.asv
☑	traffic_zone_id	ics-IE-TRAFFIC
☑	parking_zone_id	ics-IE-PARKING
☑	ped_zone_id	ics-IE-PEDESTRIAN
☑	ps_zone_id	ics-IE-PUBLIC-SAFETY
☑	comm_token	eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2Fje!
☑	ped_asset	CAMERA-STG-HYP1052-CAM-F

Cancel | Update

The **UAAURL** field(s) should be:

- *For Cities:*

d. https://890407d7-e617-4d70-985f-01792d693387.predix-uaa.run.aws-usw02-pr.ice.predix.io

e. *If the* ***UAAURL*** *address is different, then copy and paste the address above into the UAAURL variable field, and click* ***Update****.*

f. *Click 'X' to exit.*

g. *In Postman, go to the "****Get client token****" call and click the blue* ***Send*** *button. This will generate the client token and replace the 'client token' value in the Environment.*

Second step: Copy and use client token in Manage Environment

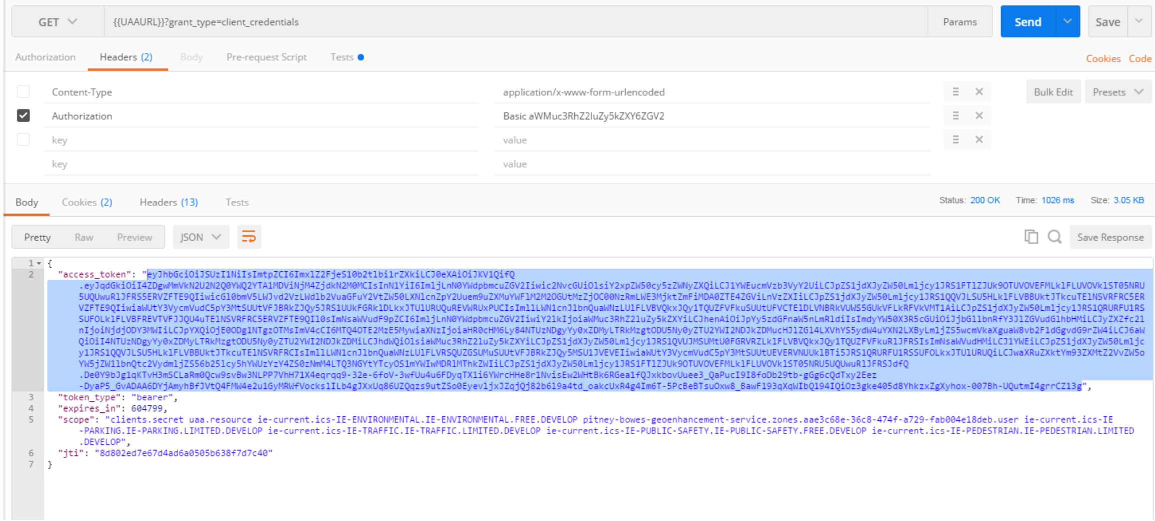
Use this process if the token is not updating for your calls…

The client token should automatically get replaced in the Postman Headers tab/Authorization field:

“ Authorization ……………………………………………………………………………………………… Bearer {{client token }} “

If the client token does **not** automatically update, generate a token and replace the token variable manually.

1. *Generate new token from the* ***Get client Token*** *call; see process above.*
   a. *Copy *only* the token key, which is the content block* ***after*** *“access token” :* ***and*** *between the parentheses.* ***Do not include the parentheses!***



GET {{UAAURL}}?grant_type=client_credentials

□	Content-Type	application/x-www-form-urlencoded
☑	Authorization	Basic aWMuc3RhZ2luZy5kZXY6ZGV2
□	key	value
	key	value

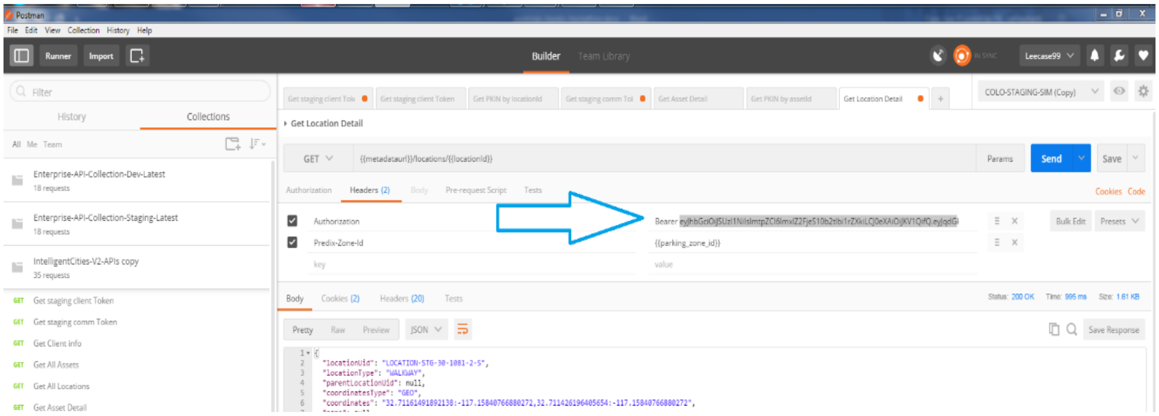
Status: 200 OK Time: 1026 ms Size: 3.05 KB

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImxlZ2FjeS10b2tlbi1rZXkiLCJ0eXAiOiJKV1QifQ
      .eyJqdGkiOiI4ZDgwMmVkN2U2N2Q0YWQ2YTA1MDViNjM4ZjdkN2M0MCIsInN1YiI6ImljLnN0YWdpbmcuZGV2Iiwic2NvcGUiOlsiY2xpZW50cy5zZWNyZXQiLCJ1YWEucmVzb3VyY2UiLCJpZS1jdXJyZW50Lmljcy1JRS1FT1ZJUk9OTUVOVEFMLk1FLUVOVk1ST05NRU
      5UQUwuRlJFRS5ERVZFTE9QIiwicGl0bmV5LWJvd2VzLWdlb2VuaGFuY2VtZW50LXNlcnZpY2Uuem9uZXMuYWFlM2M2OGUtMzZjOC00NzRmLWE3MjktZmFiMDA0ZTE4ZGViLnVzZXIiLCJpZS1jdXJyZW50Lmljcy1JRS1QQVJLSU5HLk1FLVBBUktJTkcuTElNSVRFRC5ER
      VZFTE9QIiwiaWUtY3VycmVudC5pY3MtSUUtVFJBRkZJQy5JRS1UUkFGRklDLkxJTU1URUQuREVWRUxPUCIsImllLWN1cnJlbnQuaWNzLU1FLVBVQkxJQy1TQUZFVFkuSUUtUFVCTE1DLVNBRkVUWS5GUkVFLkRFVkVMT1AiLCJpZS1jdXJyZW50Lmljcy1JRS1QRURFU1RS
      SUFOLk1FLVBFREVTVFJJQU4uTE1NSVRFRC5ERVZFTE9QI10sImNsaWVudF9pZCI6ImljLnN0YWdpbmcuZGV2IiwiY2lkIjoiaWMuc3RhZ2luZy5kZXYiLCJhenAiOiJpYy5zdGFnaW5nLmRldiIsImdyYW50X3R5cGUiOiJjbGllbnRfY3JlZGVudGlhbHMiLCJyZXZfc2l
      nIjoiNjdjODY3MWIiLCJpYXQiOjE0ODg1NTgzOTMsImV4cCI6MTQ4OTE2MzE5MywiaXNzIjoiaHR0cHM6Ly84NTUzNDgyYy0xZDMyLTRkMzgtODU5Ny0yZTU2YWI2NDJkZDMucHJlZGl4LXVhYS5ydW4uYXN2LXByLmljZS5wcmVkaXguaW8vb2F1dGgvdG9rZW4iLCJ6aW
      QiOiI4NTUzNDgyYy0xZDMyLTRkMzgtODU5Ny0yZTU2YWI2NDJkZDMiLCJhdWQiOlsiaWMuc3RhZ2luZy5kZXYiLCJpZS1jdXJyZW50Lmljcy1JRS1QVUJMSUMtU0FGRVRZLk1FLVBVQkxJQy1TQUZFVFkuRlJFRSIsImNsaWVudHMiLCJ1YWEiLCJpZS1jdXJyZW50Lmljc
      y1JRS1QQVJLSU5HLk1FLVBBUktJTkcuTE1NSVRFRCIsImllLWN1cnJlbnQuaWNzLU1FLVRSQUZGSUMuSUUtVFJBRkZJQy5MSU1JVEVEIiwiaWUtY3VycmVudC5pY3MtSUUtUEVERVNUUk1BTi5JRS1QRURFU1RSSUFOLkxJTU1URUQiLCJwaXRuZXktYm93ZXMtZ2VvZW5o
      YW5jZW1lbnQtc2VydmljZS56b25lcy5hYWUzYzY4ZS0zNmM4LTQ3NGYtYTcyOS1mYWIwMDR1MThkZWIiLCJpZS1jdXJyZW50Lmljcy1JRS1FT1ZJUk9OTUVOVEFMLk1FLUVOVk1ST05NRU5UQUwuRlJFRSJdfQ
      .De0Y9bJg1qKTvH3mSCLaRm0Qcw9svBw3NLPP7VhH71X4eqrqq9-32e-6foV-3wfUu4u6FDyqTX1i6YWrcHHe8r1NvisEw2WHtBk6RGea1fQJxkbovUuee3_QaPucI9I8foDb29tb-gGg6cQdTxy2Eez
      -DyaP5_GvADAA6DYjAmyhBfJVtQ4FMW4e2u1GyMRWfVockslILb4gJXxUq86UZQqzs9utZSo0EyevljxJZqjQj82b619a4td_oakcUxR4g4Im6T-5PcBeBTsuOxw8_BawF193qXqWIbQ194IQiOz3gke405d8YhkzxZgXyhox-007Bh-UQutmI4grrCZ13g",
  "token_type": "bearer",
  "expires_in": 604799,
  "scope": "clients.secret uaa.resource ie-current.ics-IE-ENVIRONMENTAL.IE-ENVIRONMENTAL.FREE.DEVELOP pitney-bowes-geoenhancement-service.zones.aae3c68e-36c8-474f-a729-fab004e18deb.user ie-current.ics-IE
      -PARKING.IE-PARKING.LIMITED.DEVELOP ie-current.ics-IE-TRAFFIC.IE-TRAFFIC.LIMITED.DEVELOP ie-current.ics-IE-PUBLIC-SAFETY.IE-PUBLIC-SAFETY.FREE.DEVELOP ie-current.ics-IE-PEDESTRIAN.IE-PEDESTRIAN.LIMITED
      .DEVELOP",
  "jti": "8d802ed7e67d4ad6a0505b638f7d7c40"
}
```

2. *Then, follow the process to replace a variable:*
   a. *Click on the sprocket icon (far upper right); click* ***Manage Environments****; click on the Environment name you use to Edit Environment. This will bring up a list of key value variables.*
   b. *Paste the token into the* `'client_token'` *key value field.*
   c. *Click* ***Update****; then click 'X' to exit. You have updated the client token in the Environment you are using. This token usually works for 24 hours.*
   d. *Now, when you run calls in the Postman Collection, your call will automatically grab and replace the* `{{client token}}` *variable from the environment into your call.*
   e. *To run an API call in Postman, click the blue* ***Send*** *button to run each GET or POST call.*

NOTE: You can also manually replace the newly generated token in the Authorization field in the Headers tab of the call.

Simply copy the new access token, then paste it after ‘Bearer’ in the Authorization key field. It will look like this:



GET {{metadataurl}}/locations/{{locationId}}

☑ Authorization	Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6ImxIZ2FjeS10b2tlbi1rZXkiLCJ0eXAiOiJKV1QifQ.eyJqdGl
☑ Predix-Zone-Id	{{parking_zone_id}}
key	value

Status: 200 OK Time: 995 ms Size: 1.61 KB

```
{
  "locationUid": "LOCATION-STG-30-1081-2-5",
  "locationType": "WALKWAY",
  "parentLocationUid": null,
  "coordinatesType": "GEO",
  "coordinates": "32.71161491892138:-117.15840766880272,32.711426196405654:-117.15840766880272",
```

That said, you will have more consistent results by updating the client token in the environment.

See the “Updating Postman Directions “section in this document.

Special Note: Simulated Data

Current by GE will **not** provide excel spreadsheets containing static metadata or simulated data; these APIs work best with simulated from a reference app or a seed app, or live data from a live implementation.

The use of the postman collections will provide the user with simulated data (metadata such as assets, and so on, vs. events or media) when used with seed data from a data simulator or with live data.

SEED DATA SIMULATOR SITE: https://github.com/CurrentByGE/ie-angular-cities-ref-app

- ***Login*** *(Email): hackathon*
- ***Password****: @hackathon*

This site includes the seed app and documentation describing how to set it up. Login to the simulator, and start using it; data will start accumulating. After about 5 minutes, start running APIs in that environment.

Note: When using the seed app, if you use the same timestamp for start-ts (startTime) and end-ts (endTime), a single result is returned.

Hackathon FAQ

Is there a way to get started using the APIs?

Yes, see the *Getting Started with Intelligent Environments* in the API web-help documentation. This will cover the first things you need to set up to get started. You can also get an overview on how the APIs work in the *General Overview* section of the API web-help.

Start Here:
https://ie-cities-docs.run.aws-usw02-pr.ice.predix.io

How can I run the APIs?

The APIs listed on the Predix.io catalogue essentially contain simulated data, so once you can ping them, you will be able to get simulated data back as a response. Currently, we only have 8-9 assets in the simulator.

For more information, navigate to the ***Intelligent Cities APIs*** section; ***'Appendix A',*** in the API web-help:

https://ie-cities-docs.run.aws-usw02-pr.ice.predix.io

Feel free to expand and make your own dataset for as many assets as you want. Just follow the pattern that we gave you, and you can create your own simulated data.

FAQ -- RECIPES

How long have these cars been parked?

As an example, a Scenario you may come up with could be:

"I would like to see how long the cars in all city parking spaces have been parked during the lunch hour (12-1 PM). Right now, it is 1:15 PM. "

• *Run historical data PKIN API by location*	*Timestamp: 11:45 AM*
• *Run historical data PKIN API by location*	*Timestamp: 12:01 PM*
• *Run PKOUT API by location*	*Timestamp: 1:00 PM*
• *RUN PKIN API by location*	*Timestamp: current time (@ 1:15 PM)*

Then what do I do? First, compare the datasets, extract what is similar along the compared files, and look at the defined EventIds – these will be the identifiers of the specific cars that have remained parked over this total time-period. These cars may be liable for a parking ticket!

Hot Note: Currently, the Predix technology does not "live" monitor the physical parking spaces or Zones, only moving assets (vehicles) that are using the parking lot.

Simulated Parking Data

Do you have simulated parking data for our city?

Right now, our team is developing an app that utilizes available parking spaces, and we want to be able to show local parking areas in the app.

Currently, we do not have parking data for cities other than the simulated data we have provided which has environmental data from the City of San Diego.

Here is a workaround for all hackathon users to use our parking APIs with simulated data:

- *Obtain asset IDs*
- *Call the Car-in & Car-out APIs with the asset IDs; they can obtain the image on demand from the parking events.*
- *Ignore the geo-coordinates. This will give you a basic parking simulation.*

If you choose, you can make a copy of the dataset and replace the geo-coordinates with parking geo-coordinates from your own city. Each `geoCoordinates` field MUST match a unique `locationUid`, and contains 4 distinct geo-coordinate points that denote a parking space.

Sample code from API:

```
"geoCoordinates": "32.7137086:-117.1568737,32.7136872:-
117.1568726,32.7136871:-117.1569479,32.7137092:-117.1569459",
```

As long as you use the same `geoCoordinates` format and make sure each `locationUid` matches the unique set of geoCoordinates for a parking spot, you can create a model showing parking activity for your city or town.

Weather Information

h. *If you also want to include weather information, you can call the Environmental API using the asset IDs.*

current
powered by GE

References

Intelligent Environments – Cities and Enterprises

If you are a developer, or want to build in-house applications, you can use the **Intelligent Environments by Current** to enable a valuable set of location-based services for your organization and your clients.

Start by looking at the CityIQ IoT Platform resources here:
http://developer.currentbyge.com/cityiq

Also, take a look at the *Predix Developer Network* resources:
https://www.predix.io/

On this site, Current reveals the capabilities of the overall Platform and its APIs, Reference Apps, and how to apply for a Business Partnership with Current.

Reference - API documentation

Cities – API online help:
https://ie-cities-docs.run.aws-usw02-pr.ice.predix.io

Enterprises – API online help:
https://ie-enterprises-docs.run.aws-usw02-pr.ice.predix.io