Django GCP IoT Device Manager Documentation

Steve Graham

Jun 24, 2020

Contents:

CHAPTER 1

Django GCP IoT Device Manager

Provides a CRDL (Create, Retrieve, Destroy, List) interface to GCP IoT Core

- Free software: MIT license
- Documentation: https://dj_gcp_iotdevice.readthedocs.io.

1.1 Getting Started

If you are not familiar with GCP IoT Core then please first read the following:

- GCP IoT Core getting started link: https://cloud.google.com/iot/docs/how-tos/getting-started

First create an IoT device registry and then make sure that the service account assigned to your service (VM, Cloud Run, App Engine) has enough GCP permissions to create/read/delete/list from GCP IoT Core.

1.2 Installation

Install `dj_gcp_iotdevice` from pip

```
$ pip install dj_gcp_iotdevice
```

Add to your top level `apps.py`

```
from dj_gcp_iotdevice.apps import GCPIoTDeviceConfig

class MyProjectDeviceConfig(GCPIoTDeviceConfig):
```

(continues on next page)

(continued from previous page)

```
    registry = 'my-iot-registry'
    location = 'us-central1'
    project = 'my-project-id'
```

Add the new app config to your installed apps

```
INSTALLED_APPS = [
    ...
    'apps.MyProjectDeviceConfig',
]
```

Add the provided urls to your list of urls

```
urlpatterns = [
    ...
    path('', include('dj_gcp_iotdevice.urls')),
]
```

Run the migrate command to create the new permissions that you can protect the API with

```
python manage.py migrate
```

1.3 API

The following endpoints will be accessible

```
POST /devices/
GET /devices/{id}
DELETE /devices/{id}
GET /devices/
```

To create a new device you will need to generate a private/public keypair using the following commands

```
openssl genpkey -algorithm RSA -out rsa_private.pem -pkeyopt rsa_keygen_bits:2048
openssl rsa -in rsa_private.pem -pubout -out rsa_public.pem
```

Take the contents of the `rsa_public.pem` and use that for the public_key in the API. Make sure to use \n characters for the line feeds.

The following snippet is the openapi spec for the new devices api

```
/devices/:
    get:
        operationId: devices_list
        summary: Used to list all the devices in the registry.
        description: |-
            :raises PermissionDenied: Likely bad coordinates to registry or not␣
↪enough permissions
                                    to list devices from registry.
        parameters: []
        responses:
            '200':
            description: ''
            schema:
```

(continues on next page)

(continued from previous page)

```
                type: array
                items:
                $ref: '#/definitions/Device'
        tags:
            - devices
    post:
        operationId: devices_create
        summary: Used to add a new IoT device to the registry.
        description: |-
            :raises ParseError: Bad data provided.  Likely a bad public key.
            :raises NotAcceptable: Could not add device.  Probably device Id already␣
↪exists.
            :raises PermissionDenied: Likely wrong GCP coordinates or insufficient␣
↪permissions
                                    on GCP to add devices to the registry.
        parameters:
            - name: data
            in: body
            required: true
            schema:
                $ref: '#/definitions/Device'
        responses:
            '201':
            description: ''
            schema:
                $ref: '#/definitions/Device'
        tags:
            - devices
        parameters: []
/devices/{id}/:
    get:
        operationId: devices_read
        summary: Used to get one device from the registry.
        description: |-
            :raises PermissionDenied: Likely bad coordinates to registry or not enough
                                    permissions to read devices from registry.
            :raises NotFound: Device does not exist.
        parameters: []
        responses:
            '200':
            description: ''
            schema:
                $ref: '#/definitions/Device'
        tags:
            - devices
    delete:
        operationId: devices_delete
        summary: Used to remove a device from the registry.
        description: |-
            :raises PermissionDenied: Likely bad coordinates to registry or not enough
                                    permissions to remove devices from the registry.
            :raises NotFound: Device does not exist.
        parameters: []
        responses:
            '204':
            description: ''
        tags:
```

(continues on next page)

(continued from previous page)

```
        - devices
    parameters:
    - name: id
        in: path
        required: true
        type: string
```

1.4 Permissions

Modifying the IoT device registry is not something you want everyone to be able to do so this app also adds model permissions you can assign to groups or to individual users that can limit what the user is able to do.

In the Admin page under `dj_gcp_iotdevice` there are 4 permissions: can add, can change, can delete, and can view. Use these to control what parts of the CRDL a user or group can access.

CHAPTER 2

dj_gcp_iotdevice

2.1 dj_gcp_iotdevice package

2.1.1 Submodules

2.1.2 dj_gcp_iotdevice.apps module

Application specific configuration. This gets overriden by the user to specify the registry coordinates on GCP.

class dj_gcp_iotdevice.apps.**GCPIoTDeviceConfig**(*app_name*, *app_module*)

Bases: django.apps.config.AppConfig

Users of this app will override this class and specify their registry coordinates.

location = 'GCP project location for device registry - ie us-central1'

name = 'dj_gcp_iotdevice'

project = 'GCP project id for device registry'

registry = 'GCP device registry id'

verbose_name = 'IoT Device Manager'

2.1.3 dj_gcp_iotdevice.serializers module

Simple serializer for the GCP IoT Device structure.

class dj_gcp_iotdevice.serializers.**DeviceSerializer**(*instance=None*, *data=<class 'rest_framework.fields.empty'>*, ***kwargs*)

Bases: rest_framework.serializers.Serializer

Simple serializer that represents the shape of a GCP IoT Device.

create(*validated_data*)

update(*instance*, *validated_data*)

2.1.4 dj_gcp_iotdevice.urls module

2.1.5 dj_gcp_iotdevice.views module

2.1.6 Module contents

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index