FGP Rest-Api javascript interface

Eric Wang

Version 1.0, Feb 12, 2019

Table of Contents

Revision History

Version	Date	Name	Comments
1.0	12/02/2019	Eric Wang	First Draft

Overview

This library will help you work with future-grid/fgp-rest-api, simplify the way of calling most of rest api.

Preparation

All Dependencies

1. fgp-js-kit(19.2.1 or above)
2. AngularJs 1.5.7
3. JQuery 3.0.0
4. fgp-rest-api(19.2.1 or above)

Install fgp-js-kit

install fgp-js-kit via "bower"

```
bower install --save fgp-js-kit:19.2.1
```

import javascript

```
<script src="bower_components/jquery/dist/jquery.min.js"></script>
<script src="bower_components/angular/angular.min.js"></script>
<script src="bower_components/fgp-js-kit/dist/fgp.plugins.bundle.js"></script>
<script src="bower_components/fgp-js-kit/dist/fgp.kit.bundle.js"></script>
```

add module

```
angular.module('app', ['fgp-kit','......'])
```

inject "dataService" into controller, directive....

Reference Api

Use this api for your "searching" page or something stored in database and not need to loaded into

future-grid platform.

1. name:

```
referenceTableRSQL
```

2. parameters:

```
1) api address
2) application name
3) reference type name
4) RSQL: "name=Eric;age=gt=30"
5) page number
6) page size
7) hazelcast proxy? true or false
8) pk
9) timeout default 10000 milliseconds
```

3. example

```
    dataService.referenceTableRSQL(
    "http://10.1.14.45:8082",
    "graphs",
    "ref_test",
    "",
    0,
    10,
    false,
    'id',
    2000).then(function success(resp){
            var result = resp.data;
        }, function error(error){
            console.error(error);
  });
```

```
[
  {"id":"001",
    "birthday":387244800000,
    "firstName":"eric",
    "lastName":"wang",
    "phone":"0478525520",
    "address":"38 incana dr mill park 3082"
    }
  ]
```

Device Api

Get device info by this api, include device dto and extensions.

1. name:

```
getDeviceWithExtensions
```

2. parameters:

```
1) api address
2) application name
3) device name / DeviceKey
4) device type
5) extension types
```

3. example

```
dataService.getDeviceWithExtensions(
"http://10.1.14.45:8082",
"graphs",
"Melbourne",
"city",
["city_ext"]
).then(function success(resp){
        $scope.deviceInfo = resp;
    }, function error(err){
        console.error(err);
    });
```

```
{
  "deviceKey":{"id":"999cebf9-a637-4f35-850c-1f7cce7d445d"},
  "name":"Melbourne",
  "type":"city",
  "description":"City Melbourne",
  "city_ext":{
              "cityUuid":{"id":"999cebf9-a637-4f35-850c-1f7cce7d445d"},
              "name":"Melbourne",
              "latitude":-37.815018,
              "longitude":144.946014
            }
}
```

Relationship Api

Get parent or children device.

1. name:

```
getRelatedDevices
```

2. parameters:

```
1) api address
2) application name
3) device name / DeviceKey
4) device type
5) relation type
6) isParent true/false
```

3. example

```
dataService.getRelatedDevices(
"http://10.1.14.45:8082",
"graphs",
"Melbourne",
"city",
"city_nation",
true
).then(function success(resp){
        $scope.relationData = resp.data;
    }, function error(err){
        console.error(err);
    });
```

```
{
  "deviceKey":{"id":"1329d8b8-4fa6-41de-acb8-aa84e128c8fe"},
  "name":"Australia",
  "type":"nation",
  "description":"Nation Australia"
}
```

```
dataService.getRelatedDevices(
"http://10.1.14.45:8082",
"graphs",
"Australia",
"nation",
"city_nation",
false
).then(function success(resp){
        $scope.relationData = resp.data;
    }, function error(err){
        console.error(err);
    });
```

```
[
  {
    "deviceKey":{"id":"8d1abca6-cfc7-405c-aa82-9609e84d829b"},
    "name":"Perth",
    "type":"city",
    "description":"City Perth"
  },
  {
    "deviceKey":{"id":"999cebf9-a637-4f35-850c-1f7cce7d445d"},
    "name":"Melbourne",
    "type":"city",
    "description":"City Melbourne"
  }
]
```

Store Api

Get device store data.

1. name:

```
getStoreData
```

2. parameters:

```
1) api address
2) application name
3) devices(name)
4) device type
5) store name
6) start timestamp
7) end timestamp
8) fields array (null return all columns)
```

3. example

```
dataService.getStoreData(
"http://10.1.14.45:8082",
"graphs",
["Melbourne"],
"city",
"city_temperature_day",
new Date('2019-01-31').getTime(),
new Date('2019-01-28').getTime(),
["maxTemperature","minTemperature"]
).then(function success(resp){
        $scope.storeData = resp;
    }, function error(err){
        console.error(err);
    });
```

```
[
  {
    "Melbourne":{
      "data":[
        {
          "maxTemperature":24.51,
          "minTemperature":15.37,
          "timestamp":1548892800000
        }
      ],
      "deviceKey":"999cebf9-a637-4f35-850c-1f7cce7d445d"
    }
  }
]
```