

Example of my JSON structure for a single "movie" entity

```
{
  "name": "Castle ",
  "year_released": "1994",
  "actors": [
    {
      "firstName": "Nathan",
      "lastName": "Fillion"
    },
    {
      "firstName": "Stana",
      "lastName": "Katic"
    },
    {
      "firstName": "Susan",
      "lastName": "Sullivan"
    }
  ]
}
```

I added 10 movies to my collection "movies" on <https://apigee.com/appservices> and got their UUIDs

Collections (need help?)

- /activities
- /assets
- /cxcxcs
- /devices
- /folders
- /groups
- /movies**
- /roles
- /users

GET /movies

ex: select * where name="fred" Limit ex: 10 Send

| Name | UUID | Delete Entity(s) |
|-------------------------------------|--------------------------------------|------------------------------|
| Castle | b9a0e75a-0602-11e6-921a-b5ff004e9e28 | View Details |
| The Neon Demon | c487474a-0602-11e6-bfc5-01270831cf5a | View Details |
| Ratchet & Clank | ccd5be4a-0602-11e6-b10e-93e4bd019839 | View Details |
| Compadres | d4d048ea-0602-11e6-8fcf-25fd8717c11e | View Details |
| The Curious Case of Benjamin Button | da39a15a-0602-11e6-a16f-0d410d16fbb6 | View Details |
| Green Lantern | f6ee435a-0602-11e6-b41f-fd9b449f7958 | View Details |
| Drive | fe5e8f5a-0602-11e6-b0fe-8b486d1f2955 | View Details |
| Bad Teacher | 03833daa-0603-11e6-8724-bbea64f78f6f | View Details |
| Jurassic World | 0a6fb3aa-0603-11e6-98da-2ba3bddbd6c7 | View Details |
| Spectre | 127a1cca-0603-11e6-8bc9-05773609bef4 | View Details |

Next >

Snippet of my server app.js

Initiate and connect to the server

```
var usergrid = require('usergrid');
var express = require('express');
var app = express();
var request = require('request');
var client = new usergrid.client({
  orgName: 'hanialghamdi',
  appName: 'sandbox',
});
```

Creating a new entity from the query parameters

```
app.post('/movies', function (req, res){
  if(!req.query.name || !req.query.debut || !req.query.actors) {
    res.send('There was an error in creating the entity.');
```

```
  } else {
    var options = {
      type: 'movies',
      name: req.query.name,
      debut: req.query.debut,
      actors: req.query.actors,
      getOnExist: true
    };
    //Call createEntity to initiate the API call
    dataClient.createEntity(options, function(error, result){
      if(error) {
        var info = JSON.parse(body);
        res.send(info);
      } else {
        res.status(200).send('New movie posted successfully');
      }
    });
  }
});
```

Deleting an entity

```
app.delete('/movies', function (req, res) {
  var properties = {
    type: 'movies',
    name: req.query.name,
    getOnExist: true
  };
  //create the entity object
  var entity = new Usergrid.entity(properties);
  //call destroy() to initiate the API DELETE request
  entity.destroy(function (error, result) {
    if (error) {
      var info = JSON.parse(body);
      res.send(info);
    } else {
      res.status(200).send('Movie deleted successfully');
    }
  });
});
```

