Server RESTful Infrastructure #61

Edit New issue

Closed **ppyordanov** opened this issue on Dec 9, 2014 · 4 comments

ppyordanov commented on Dec 9, 2014

The server's structure needs to be improved to facilitate RESTful communication with the client application (Android).

ppyordanov self-assigned this on Dec 9, 2014

ppyordanov added this to the **Mobile Web Application** milestone on Dec 9, 2014

ppyordanov commented on Dec 9, 2014

The most important communication point is transmitting newly generated routes/ data readings mapped to locations (latitude and longitude) from the Android client application to the server. In order to implement this, multiple approaches were considered and tested:

- considering the fact that data being transmitted is from different domains, there are some conflicts that might arise and cause errors such as CORS standard violation; the CORS (Cross-Origin Resource Sharing) standard is based on the idea that different domains/ web pages can share JSON-formatted resources via their RESTful views to send and consume the data via modifying their headers in a way that authorizes only those specific domains to interact together; this convention has been introduced to address security issues/concerns when designing a RESTful view consuming data.
- CORS: http://en.wikipedia.org/wiki/Cross-origin_resource_sharing
- another option was to use JSONP (JSON with padding) which is flexible and supports different formats. However, there are also some limitations such as GET requests only (again for security reasons). What this means is that all of the information that is transferred is being stored in a parameterized format as a part of the URL; on the contrary, POST stores all this information in the response body so that it can conveniently be deserialized from JSON to models using a serializer (examples: Google's Gson, Jackson); this approach was left out as storing the whole data (considering the amount) as a part of the link is not conventionally acceptable and also requires much more processing/ parsing on the controller side to convert to model objects. Below is how my test request with JSONP looked:

```
$.ajax({
    url:"http://127.0.0.1:8080/addRoute",
    //url:"http://178.62.100.239:8080/addRoute",
    type:"GET",
    //crossDomain: true,
    dataType: "jsonp",
    //contentType: "application/javascript",
    data: JSONdata,
    async: false,
    cache: false,
    processData:false,
    success: function(response)
{
    console.log(response);
    return response;
}});
```

- JSONP: http://en.wikipedia.org/wiki/JSONP

The client is currently communicating with the server using an ajax POST request to

Labels

enhancement

Milestone

Mobile Web Applic...

Assignee

ppyordanov

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

Lock issue

http://178.62.100.239:8080/addRoute and the *dataType* is JSON. There are no concerns using this approach as there are no security issues that could potentially arise at this point. The JSON String is directly being deserialized into a list of *DataReading* models as well as a *Route* model on each transmission. Both entities are then added to the DB with a "routeId" one-to-many association between *Route* and *DataReading*. Here is how the request looks in JavaScript:

```
var dataReading ={json:JSON.stringify(context_data)};
$.ajax({
    type: 'POST',
    //url: "http://127.0.0.1:8080/addRoute",
    url:"http://178.62.100.239:8080/addRoute",
    data: dataReading,
    dataType: "json",
    success: function(response)
    {
        console.log(response);
    }
});
```

An important consideration was whether to reduce the models' atomicity in order to improve query performance as well as reduce code complexity. The change would involve making data readings an integral part of the route model, meaning that the latter would have a list of data readings. Currently each data reading is associated to a route via its "routeId" attribute. This seems to be efficient and satisfactory at this stage of development, but essentially merging the two models is a potential optimization that can be implemented in the future.

PY

ppyordanov commented on Dec 9, 2014

HomeController.java

The main view, consuming JSON, is addRoute():

```
@RequestMapping(value = "/addRoute", method = RequestMethod.POST)
public @ResponseBody
String addRoute( @RequestParam("json") String json)
{
    //save new route
    Route newRoute = new Route();
    routeRepository.save(newRoute);

    //save data reading
    Type listType = new TypeToken<List<DataReading>>() {
    }.getType();
    List<DataReading> data = new Gson().fromJson(json, listType);

    //DataReading dr = new Gson().fromJson(json,DataReading.class);
    LOGGER.info("Route saved: " + newRoute);
    LOGGER.info("ds" + data.size());
    for(DataReading dr: data) {

        dr.setRouteId(newRoute.getId());
        /*
        Map<String, Double> position = dataPopulation.randomPosGen(Constants.MIN_LAT_BOUNDS, Co
        dr.setLatitude(position.get("latitude"));
        dr.setLongitude(position.get("longitude"));
        */

        dataReadingRepository.save(dr);
        LOGGER.info("Data reading saved: " + dr);
    }

    return "success";
}
```

- The function parses the script converting it to a list of model objects, then creates a new route, adds it to the database and associates the data reading objects with it, completing the process by inserting the data readings in the *DataReadings* Mongo DB collection. All operations are logged to the console in order to facilitate easy system monitoring and code debugging.

ppyordanov added the **enhancement** label on Dec 10, 2014

ppyordanov commented on Dec 13, 2014

Update: the web server should be available on the its newly registered domain name:

- **http://ugmap.me/**
- **http://www.ugmap.me/**

ppyordanov referenced this issue from a commit on Feb 4

[LOGO] + [REFACTORING] d01e8a9

ppyordanov referenced this issue from a commit on Feb 4

[LARGE_ITERATION] improved models, better UI, map configuration 32b208b

ppyordanov referenced this issue from a commit on Feb 4

[REFACTORING] menu implementation and optimizations 9d3b885

ppyordanov referenced this issue from a commit on Feb 4

[ITERATION] map configuration c50966c

ppyordanov referenced this issue from a commit on Feb 4

[ITERATION_ROUTES] route data 076432b

ppyordanov referenced this issue from a commit on Feb 4

[IMPROVEMENTS] configuration + user feedback UI 227138a

ppyordanov commented 28 days ago

This task has been completed and can be closed now.

ppyordanov closed this 28 days ago

Write Preview Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Reopen issue Comment