

Assignment 4 - Potholes
Due March 27

Objectives

Navigation, network connections, photos, location

App description

You will build an app that allows people to identify and locate potholes in San Diego streets. When someone sees a pothole in the road they can take a picture of the hole and submit it with a short description. The user can also download the information submitted about potholes and see a list of submissions. When they select one they go to a detailed view (or two) where they see detailed information about the pothole and its image. In an optional version of the app the user can also see all the submissions in a given area on a map so they can visualize where the potholes are located. When the user selects a pothole on the map they go to the detailed view(s) for that pothole.

Information App submits to the Server

When the app submits a report about a pothole it submits the following information

Latitude and longitude of the pothole

Short description: maximum of 500 words describing the issue.

Person or user submitting the report: For this assignment take the last 2 or 3 digits of your red id and add some other characters to create an id for yourself. You can use up to 15 total characters.

Report type: Currently the backend handles "street" and "water". In this assignment we will use "street"

Image type: The image format being sent. For example png, jpg. If not sending an image use "none"

Image: The photo of the pothole.

Interaction with the Server

GET urls

<http://bismarck.sdsu.edu/city/categories>

Returns a json array in the body of the response. The “Content-Type” header in the response will be “application/json”. Each element of the array is string indicating the type of reports the server can handle. Currently the server returns:

```
["water", "street"]
```

<http://bismarck.sdsu.edu/city/count?type=street>

Returns a json object in the body of the response. The “Content-Type” header in the response will be “application/json”. The object contains the current number of records and a timestamp of when the count was created. Given that multiple clients can be adding to the database the count may increase before you can use this information.

```
{"count":1610,"created":"2/18/16, 03:19 AM"}
```

<http://bismarck.sdsu.edu/city/count?type=street&user=rew&end-id=32>

user and end-id are optional.

If the user is given this returns the count of the records added by the given user. You need to replace “rew” with the user id that you use to submit the report.

If end-id is given then returns the number of records added after record id indicated. The above y=url will return the number of records added after record 32 by the user rew. If user is not given it will return the number of records added by all users.

If you are storing the records on the device this url can be used to determine if new records have been added to the server since the app last downloaded records.

<http://bismarck.sdsu.edu/city/batch?type=street&user=rew&size=10&batch-number=0>

[&end-id=15](#)

Returns reports for given user and type. You need to replace “rew” with your id.

Reports are returned in batches. If the database contains 10,000s of reports returning them all in one request could take a long time. So the reports are returned in batches. The number of reports returned is given by the “size”. The above example will return ten reports at a time. The reports start from the most recent and go back in time. The batch-number indicates which group of “size” reports you get. With batch-number set to 0 you get the most recent records. If you set batch-number to 1 you will get in the above the next ten reports. To get all the reports keep increasing the batch-number. When you receive zero reports you know that you have them all.

The user is optional. If user is given only reports from that user are returned. If the user is not given the reports for all users are given. Replace rew with your id. The reason for the user id in this command is to help in development. Multiple people will be using the server. Using the user id in this command will allow you to filter out all the other users.

size, batch-number and end-id are all optional. If size is not give it defaults to 10. If batch-number is not given it defaults to 0. To be useful you need to include batch-number. end-id defaults to 0 if not given.

If you cache the records on the device you can use end-id to reduce the the time it takes to fetch the records. If you include end-id you will only get records that have a higher id number than end-id. That is the records that come after that record number. This is an advanced feature.

Returns an json array of json objects in the body of the response. The “Content-Type” header in the response will be “application/json”. All the possible keys in the json objects are shown in the example below. The keys are all lower case. The “type” is the type of report. “imagetype” is the type of the photo for that report. If there is no image the type will be “none”. “created” is the time the report was sent to the server. The time is Greenwich mean time, so it will be offset by 7 or 8 hours from local time. The format of the time as shown. There is a comma and one space between the date and the time. There is one space between the time and AM or PM. The id is used to obtain the photo.

```
[{"id":92,"latitude":10.34,"longitude":33.45,"type":"water","imagetype":"png","description":"water leak","created":"10/28/15, 10:28 PM"}, {"id":91,"latitude":10.34,"longitude":33.45,"type":"water","imagetype":"png","description":"water leak","created":"10/28/15, 05:13 AM"}]
```

<http://bismarck.sdsu.edu/city/fromLocation?type=water&date=10/6/15&user=rew&start-latitude=12.0&end-latitude=14.1&start-longitude=20&end-longitude=23.5>

Returns all reports for given user, type, date and in the area indicated by start, end latitude and longitude. As in fromDate user and date are optional. You need to replace the type, date, user and latitudes and longitudes.

Returns an json array of json objects in the body of the response with the same data as in fromDate. The “Content-Type” header in the response will be “application/json”.

```
[{"id":92,"latitude":10.34,"longitude":33.45,"type":"water","imagetype":"png","description":"water leak","created":"10/28/15, 10:28 PM"}, {"id":91,"latitude":10.34,"longitude":33.45,"type":"water","imagetype":"png","description":"water leak","created":"10/28/15, 05:13 AM"}]
```

<http://bismarck.sdsu.edu/city/image?id=69>

id is the id that you obtain for a report using either fromDate or fromLocation. This will return the image in binary format in the body of the response. The “Content-Type” header in the response will be “image/XXX”, where XXX is the image type indicated in the report that uploaded the photo.

Post URLs

<http://bismarck.sdsu.edu/city/report>

The body of the post is a json object. The keys are:

type

The value is a string. It is the type of the report. For this assignment it is “street”

latitude
longitude

The values are numbers. The latitude and longitude of the pothole. To be precise it is the latitude and longitude of the device when the report is sent.

user

The value is a string. Your user id

imagetype

The value is a string. It is the type of the photo being sent, for example “png”. If no photo is being sent then it needs to be “none”.

description

The value is a string. This is limited to 300 characters.

image

The value is a string. If imagetype is “none” then this key and value are not needed. The value is a base 64 encoding of the photo being uploaded.

Errors

If your request to the server is missing required data, has keys that are incorrect or your request has some other problems the response you get will status code in the 400's rather than 200. In this case the body of the response will be a string with a brief description of the problem.

Grading

The assignment will be graded as follows:

Points	Item
20	Network connections
10	Table View for Reports sorted by date, most recent on top, display all users reports
10	Detailed view of a report
10	Displaying the image of a report
20	Post a report
10	Post photo take on device with the report
5	GUI quality
10	Navigation between views

Points	Item
5	Proper location of device
5	Displays reports of all users without long delay caused by fetching all data at once (optional)
15	Map view showing location of all potholes in a given region (optional)
5	A map showing the location of a single pothole (optional)

What to Turn in

After removing the build directories in your project zip up your entire project and submit it online in the course portal. We will use the course portal (<http://bismarck.sdsu.edu/CoursePortal>).