CS 646 iPad/iPhone Application Development
Fall Semester, 2020
Assignment 2

San Diego State University -- This page last updated 10/1/20

Due: Oct 18 at 23:59
Points: 100
Objectives

Version 1.1

Networking
Photos

InstaPost

InstaPost allows users to make and view posts. A post contains an optional photo, text and one or more hashtags. A user can create an account by providing an email address, their name and a nick name. Besides making posts a user can see a list of user's nicknames and a list of hashtags. From either list the user can select another user (or hashtag) and see all the posts from that user (or all the posts with that hashtag). One can the rate and/or comment not the post.

We will use a server that is running on bismarck.sdsu.edu. There is a set of POST and GET urls that you use to interact with the server. The server has an interactive web page that describes the urls and allows you to try out each url. The page is at:

https://bismarck.sdsu.edu/instapost/index.html#/

Server Protocol

All server URLs are https not http. The body of all responses are in JSON format. The content type of all POSTs is application/json. The body of all POST URLs is in JSON format.

Test URLs

These URLs are just for testing that you can each the server and get a response.

ping

URL https://bismarck.sdsu.edu/api/ping

returns **pong**

service-calls

URL https://bismarck.sdsu.edu/api/instapost-query/service-calls

Returns JSON object containing the number of calls made to the server

Sample:

```
{
  "service-calls": 0
}
```

Upload URLs

Note that in GET url examples parts in CAPS are values you need to change

newuser

Adds a new user. A user needs a first and last name, a nickname, an email and password. Two users are not allowed to have the same nickname or ones that differ only in case, that is Foo and foo are considered same nicknames. Passwords need to be at least 3 characters long. Passwords are not stored securely so do not use any of your existing passwords. Emails are checked for valid format.

URL GET https://bismarck.sdsu.edu/api/instapost-upload/newuser?
firstname=FIRST_NAME&lastname=LAST_NAME&nickname=NICKNAME&
email=EMAIL&password=PASSWORD

Returns either {"result" : "success", "errors": "none"} or {"result": "fail", "errors" "A STRING INDICATING THE ERRORS"}

URL POST https://bismarck.sdsu.edu/api/instapost-upload/newuser

Post Body The parameters are the same as for the GET version, but are in the body as a JSON object.

```
{
  "firstname": "STRING",
  "lastname": "STRING",
  "nickname": "STRING",
  "email": "STRING",
  "password": "STRING"
}
```

post

Adds a post, which contains email and passwords of a current user, the text of the post and a collection of hashtags. The text is limited to 144 characters. Hashtags must start with # and be at least 2 characters long including the hashtag.

URL POST https://bismarck.sdsu.edu/api/instapost-upload/post

Post Body

```
{"email": "STRING",
  "password": "STRING",
  "text": "STRING",
  "hashtags": ["#STRING1", "#STRING2"] }
```

Response Body

If successful "result" value will be "success", id value is the id of the post, "errors" value will be "none".

If post fails "result" value will be "fail", id value is -1, "errors" value will describe the errors.

General format:

```
{
  "result": "STRING",
  "id": INTEGER,
  "errors": "STRING"
}
```

Sample fail result

```
{
  "result": "fail",
  "id": -1,
  "errors": "Incorrect email or password"
}
```

image

Add an image to an existing post. The image needs to be encoded using base64 encoding before uploading to the server. To upload an image you need the email and password of the person that created the post, the id of the post and the base64 encoding of the image.

URL POST https://bismarck.sdsu.edu/api/instapost-upload/image

Post Body

```
{
  "email": "STRING",
  "password": "STRING",
  "image": "BASE64 STRING",
  "post-id": INTEGER
}
```

Response Body

If the upload was successful then the body will be {"result":"success","errors":"none"}

If unsuccessful "result" will be "fail", "errors" will describe the errors.

```
{
  "result": "STRING",
  "errors": "STRING"
}
```

rating

Add a rating to an existing post. You need the post-id of the post you are rating, the person's email and password that is doing the rating, and a rating which is an integer from 1 to 5 inclusive.

URL POST https://bismarck.sdsu.edu/api/instapost-upload/rating

Post Body

```
{
  "email": "STRING",
  "password": "STRING",
  "rating": INTEGER 1-5,
  "post-id": INTEGER_OF_POST
}
```

Response Body

If successful:

```
{
  "result": "success",
```

```
  "errors": "none"
}
```

If not successful:

```
{
  "result": "fail",
  "errors": "AN ERROR MESSAGE"
}
```

comment

Add a comment to an existing post. You need the post-id of the post you are commenting on, the person's email and password that is doing the commenting, and a string comment.

URL POST https://bismarck.sdsu.edu/api/instapost-upload/comment

Post Body

```
{
  "email": "STRING",
  "password": "STRING",
  "comment": "STRING",
  "post-id": INTEGER_OF_POST
}
```

Response Body

If successful:

```
{
  "result": "success",
  "errors": "none"
}
```

If not successful:

```
{
  "result": "fail",
  "errors": "AN ERROR MESSAGE"
}
```

Query URLS

nicknames

Returns all the existing nicknames.

URL GET https://bismarck.sdsu.edu/api/instapost-query/nicknames

Response Body

```
{"nicknames": [ "STRING1", "STRING2"]}
```

nickname-exists

Indicates if the given nickname is already in use.

URL GET https://bismarck.sdsu.edu/api/instapost-query/nickname-exists?nickname=A_NICKNAME_STRING

Response Body

Either {"result": true} or {"result":false}

email-exists

Indicates if the given email is already in use.

URL GET https://bismarck.sdsu.edu/api/instapost-query/email-exists?email=AN_E-MAIL_STRING

Response Body

Either {"result": true} or {"result":false}

hashtags

Returns all the hashtags in sorted order.

URL GET https://bismarck.sdsu.edu/api/instapost-query/hashtags

Response Body

```
{
  "hashtags": ["HASHTAG1", "HASHTAG2"]
}
```

hashtag-count

Returns the number of hashtags. Useful when using hashtags-batch

URL GET https://bismarck.sdsu.edu/api/instapost-query/hashtag-count

Response Body

```
{
  "hashtag-count": INTEGER_NUMBER_OF_HASHTAGS
}
```

hashtag-batch

Returns hashtags in specified batches. Useful when number of hashtags becomes large. Specify start index and end index. Indexing start at 0 (zero). The batch includes element at start index up to but not including the element at the end index. The list of hashtags is sort-

ed before creating the batch. Start index must be less than end index. If end index is larger that the number of hashtags the server returns the elements from the start index to the end.

URL GET https://bismarck.sdsu.edu/api/instapost-query/hashtags-batch?start-index=START_INDEX&end-index=END_INDEX

Response Body

If successful "result" is "success", "errors" is "none" and "hashtags" is JSON array of strings. Otherwise "result" is "fail", "hashtags" is empty JSON array, "errors" is a description of the error(s).

```
{
  "result": "SUCCESS_OR_FAIL",
  "hashtags": ["HASHTAG1", "HASHTAG2"],
  "errors": "STRING"
}
```

hashtags-post-ids

Returns the ids of the post that have the given hashtags.

URL GET https://bismarck.sdsu.edu/api/instapost-query/hashtags-post-ids?hashtag=STRING

Response Body

If successful "result" is "success", "errors" is "none" and "ids" is JSON array of integer, which are the ids of the posts. Otherwise "result" is "fail", "ids" is empty JSON array, "errors" is a description of the error(s).

```
{
  "result": "STRING",
  "errors": "STRING",
  "ids": [INTEGERS]
}
```

image

Returns the base64 encoded image with the given id

URL GET https://bismarck.sdsu.edu/api/instapost-query/image?id=INTEGER

Response Body

If successful "result" is "success", "errors" is "none" and "image" is JSON string, which is the base64 encoded image. Otherwise "result" is "fail", "image" is "none", "errors" is a description of the error(s).

```
{
  "result": "STRING",
```

```
  "image": "STRING",
  "errors": "STRING"
}
```

nickname-post-ids

Returns the ids of the post posted by the user with the given nickname.

URL GET https://bismarck.sdsu.edu/api/instapost-query/nickname-post-ids?nickname=STRING

Response Body

If successful "result" is "success", "errors" is "none" and "ids" is JSON array of integer, which are the ids of the posts. Otherwise "result" is "fail", "ids" is empty JSON array, "errors" is a description of the error(s).

```
{
  "result": "STRING",
  "errors": "STRING",
  "ids": [INTEGERS]
}
```

post

Returns the post with the given id. Posts with higher ids are the ones posted later. So a post id of 4 was posted before a post with id 9. The post that are returned are JSON objects. Here is a sample post:

```
{
    "comments": [
      "This post is rated too high"
    ],
    "ratings-count": 2,
    "ratings-average": 4.5,
    "id": 7,
    "hashtags": [
      "#first",
      "#demo"
    ],
    "image": -1,
    "text": "This is the first post in Instapost!"
  }
```

If the body has not been rated the "ratings-average" is -1. If the post does not have an image "image" is -1. All the comments for the post are given in the array at "comments".

URL GET https://bismarck.sdsu.edu/api/instapost-query/post?post-id=INTEGER

Response Body

If successful "result" is "success", "errors" is "none" and "post" is JSON object as shown above. Otherwise "result" is "fail", "post" is empty JSON object, "errors" is a description of the error(s).

```
{
  "result": "success",
  "post": {A_JSON_OBJECT_AS_SEEN_ABOVE},
  "errors": "none"
}
```

authenticate

Indicates if email and password are correct for an existing user.

URL GET https://bismarck.sdsu.edu/api/instapost-query/authenticate?email=AN_EMAIL&password=A_PASSWORD

Response Body

```
{
  "result": BOOLEAN
}
```

The boolean indicates whether the email and password are used by an existing user.

Target Device

Your assignment will be grades using Pixel 1 phone/emulator running Android API 29.

What to Turn in

Create a Flutter project for the assignment. Flutter places the project in its own directory. Place the directory (and all its contents) into a zip file. Turn in the zip file. Turn in your assignment at: http://bismarck.sdsu.edu/CoursePortal

Grading

Points	Item
10	**Add User**
10	**List hashtags & nicknames**
10	**Display posts by a nickname selected from list of nicknames**
10	**Display posts with a given hashtag selected from list of hashtags**

Points	Item
10	Can comment on post
10	Can rate a post
15	Can submit a post for the user
5	Can Display image
5	Can add image to post
7	Quality of the Code
8	Quality of the Interface
100	Total
10	Extra Credit - Store hashtags, nicknames and posts on device so can view them when device is off-line
15	Extra Credit - Allow user to create and submit post while device is off-line. Posts are submitted next time app is run when device is on-line

Version History

1.1 Added endpoint of checking if an email exists. Oct 5, 2020