

**COMP3122 Assignment 1**  
**Deadline: 7<sup>th</sup> Oct 2019 (Week 6 Monday) 5pm**

You should submit a softcopy of your python code (bookmarking\_service.py) to blackboard.  
Late submission: within 24 hours (-20%); after 24 hours: no mark

In this assignment, you have to implement a RESTful API for a bookmarking service using python flask framework. You are only allowed to use the following python package(s).

- request, flask, json, sqlite3 (you may install the libraries with pip if needed)

Users

Getting all users

URL	http://localhost/bookmarking/users
HTTP Request method	GET
HTTP Response body	All users (ID, name) in JSON (Sorted by ascending user ID(s))
HTTP Response Code	- 200 OK

Adding one or more new user(s)

URL	http://localhost/bookmarking
HTTP Request method	POST
HTTP Request body	new user/users's id and name in JSON format
HTTP <b>Reponse</b> Body	New user's id and name in JSON format (if operation is successful) Reason for the error (when operation is not successful)
Response Code	<ul style="list-style-type: none"><li>- 201 (Created): Request is Successful</li><li>- 400 (Bad Request): One/more of the user(s) exists already</li><li>- 500 (Internal Server Error): When the input is malformed or request that may raise exceptions</li></ul>

### Deleting a user

URL	http://localhost/bookmarking/[user id]
HTTP Request method	DELETE
Response Code	<ul style="list-style-type: none"><li>- 204 No Content (User is successfully deleted)</li><li>- 404 Not Found (User is not found)</li></ul>

### Bookmarks

#### Getting all bookmarks

URL	http://localhost/bookmarking/bookmarks
HTTP Request method	GET
Query string (optional)	<p>tags=[tag name] Return only those bookmarks with the specified tags</p> <p>count=[number of bookmarks to be returned]</p> <p>offset=[the position of the first bookmark to be returned (counting from 1)]</p>
HTTP Response body	All bookmarks in JSON for all users (Sorted in ascending order of the bookmarks' URL (followed by ascending order of user_id))
HTTP Response Code	<ul style="list-style-type: none"><li>- 200 OK</li></ul>

Getting all bookmarks for a certain user

URL	http://localhost/bookmarking/bookmarks/[user id]
HTTP Request method	GET
Query string (optional)	tags=[tag name] Filter for only those bookmarks with the specified tags (tags are separated by comma if more than 1 tags)  count=[number of bookmarks to be returned] Limit the number of bookmarks to be returned  offset=[the position of the first bookmark to be returned (starting from 1)] Specify the first bookmark to be returned
HTTP Response body	All bookmarks in JSON for all the target user (Sorted in ascending order of the bookmarks' URL)
HTTP Response Code	- 200 OK - 404 Not Found: The user does not exist

Getting a target bookmark for a certain user

URL	http://localhost/bookmarking/bookmarks/[user id] /[encoded bookmark's URL]
HTTP Request method	GET
HTTP Response body	The target bookmark in JSON format
HTTP Response Code	- 200 OK

Adding one or more bookmark(s) for a user

URL	http://localhost/bookmarking/[user id]/bookmarks
HTTP Request method	POST
HTTP Request body	new bookmark to be added (in JSON format)
HTTP Request Body	The new bookmark in JSON format (if operation is successful)  The reason for the error (if operation is not successful)

Response Code	<ul style="list-style-type: none"> <li>- 201 Created: Request is Successful</li> <li>- 400 Bad Request: Bookmark (with a certain URL) already exists</li> <li>- 404 Not Found: The user does not exist</li> <li>- 500 Internal Server Error: when the input is malformed or request that may raise exceptions</li> </ul>
---------------	--

Updating the title/tag(s) for a bookmark for a target user

URL	http://localhost/bookmarking/[user id]/bookmarks/[encoded bookmark URL]
HTTP Request method	PUT
HTTP Request Body	<p>The updated bookmark in JSON format (if operation is successful)</p> <p>The reason for the error (if operation is not successful)</p>
Response Code	<ul style="list-style-type: none"> <li>- 201 (Created): Request is Successful</li> <li>- 404 Not Found: The user does not exists/The bookmark does not exist</li> <li>- 500 (Internal Server Error): When the input is malformed or request that may raise exceptions</li> </ul>

#### Deleting a bookmark for a target user

URL	http://localhost/bookmarking/[user id]/bookmarks/[Encoded bookmark's URL]
HTTP Request method	DELETE
HTTP Response Body	<del>The new bookmark in JSON format (if operation is successful)</del>  The reason for the error (if operation is not successful)
Response Code	<ul style="list-style-type: none"><li>- 204 (No Content): Request is Successful</li><li>- 404 Not Found: The user does not exists/The target bookmark does not exist</li><li>- 500 (Internal Server Error): Response when the input is malformed or request that may raise exceptions</li></ul>

## Sample JSON Format

### Bookmark(s)

```
{
  "count": "2",
  "bookmarks": [
    {
      "url": "https://www.comp.polyu.edu.hk",
      "tags": "computing, polyu, undergraduate, postgraduate",
      "text": "PolyU Department of Computing",
      "user_id": "wchan"
    },
    {
      "url": "https://www.comp.polyu.edu.hk",
      "tags": "polyu",
      "text": "Hong Kong PolyU",
      "user_id": "tlau"
    }
  ]
}
```

### User(s)

```
{
  "count": "2",
  "users": [
    {
      "user_id": "wchan",
      "user_name": "Winney Chan"
    },
    {
      "user_id": "tlau",
      "user_name": "Tom Lau"
    }
  ]
}
```

### JSON format for Error messages

Adding a new user (user ID already exists)

```
{
  "reasons": [
    {
      "message": "User already exists",
    }
  ]
}
```

Adding a new bookmark for an unknown user

```
{
  "reasons": [
    {
      "message": "User not found"
    }
  ]
}
```

Adding an existing bookmark for a certain user

```
{
  "reasons": [
    {
      "message": "Bookmark already exists"
    }
  ]
}
```

Adding/deleting/updating bookmark(s) for a non-existing user

```
{
  "reasons": [
    {
      "message": "User not found"
    }
  ]
}
```

Deleting/updating bookmark(s) for a non-existing user

```
{
  "reasons": [
    {
      "message": "User not found"
    },
  ]
}
```

Deleting/updating bookmark(s) for a non-existing bookmark.

```
{
  "reasons": [
    {
      "message": "Bookmark not found"
    }
  ]
}
```

#### Additional Remarks

- The python flask program should listen at port 5000 during startup.
- The API should retrieve and store the bookmarks in sqlite database file bookmarks.db (stored in the same folder as the python script). The database has two tables: Users and Bookmarks. Please refer to the sample database available at learn@PolyU.
- You should set the correct content type in the HTTP headers when JSON is returned (application/json)
- If bookmark's URL is sent as part of the URL, you should first encode it
  - o e.g. the encoded string of *http://www.comp.polyu.edu.hk* is *http%3A%2F%2Fwww.comp.polyu.edu.hk*
- More than one new bookmarks/users can be added in the same request. You should return errors if one or more bookmark(s)/user(s) is/are invalid.
- Each bookmark has zero/more tags. The bookmark's URL is compulsory field.
- Empty array should be returned if there is no user/no bookmark/bookmarks for a certain users. This request is considered successful (HTTP 200 is returned).
- The input/web API's response format should follow the requirements exactly to be considered correct.



#### *Updates/clarification to the assignment*

- *For adding users, send to the endpoint `http://localhost/bookmarking` instead of `http://localhost/bookmarking/[user id]`*
- *For getting a target bookmark for a certain user, no need to handle the option query strings.*
- *Delete a bookmark for a target user: 204 (No Content) should be returned if request is Successful.*
- *Getting all bookmarks: change the attribute "tag" to plural form "tags"*
- *Deleting/updating bookmark(s) for a non-existing user: Return "message": "User not found".*
- *Deleting/updating bookmark(s) for a non-existing bookmark (user exists). Return "message": "Bookmark not found"*
- *Adding a new bookmark for an unknown user: The error message should be "User not found".*

#### *Hint*

- *You may get an error when matching a pattern in the URL with `"/`. You may use the "path" prefix for getting the bookmark's URL. You may refer to `http://exploreflask.com/en/latest/views.html` for details.*
- *You should follow the exact output format as defined in the assignment sheet to be considered correct. E.g. Instead of responding a JSON with "one or more bookmark is/are not found", you should respond with the message "Bookmark not found" (which is defined in the assignment sheet).*

#### *FAQ (Part 1)*

- *The web API may support adding one/more user(s) in one single request. The list of users to be added are specified in http request body. If successful, the new user(s) in JSON format will be returned in http responses. Otherwise, the web API will return an error response code with reasons for failure (in JSON format).*
- *You should use the same JSON format (refer to the sample JSON assignment) in both the input and output to the web API. E.g. When adding more than one users, the user count is provided by the client and the list of users are specified as array format (even though only one user is added).*
- *One/more bookmarks may be added in one request. For updating/deleting bookmark, the web API only need to handle one bookmark in one request.*
- *For updating bookmark, the bookmark to be updated is specified in URL. The http request body will contain JSON of the bookmark to be updated. The API will return the updated bookmark in the response body if update is successful (with response code 201). Otherwise, the web API will return an error response code with reasons for failure (in JSON format).*
- *For deleting bookmark, the bookmark to be deleted is specified in URL. The web API will return empty response body if update is successful (with response code 204). Otherwise, the web API will return an error response code with reasons for failure (in JSON format).*

## *FAQ (Part 2)*

1. *What should the API respond if the request body is not JSON/JSON with invalid attributes values?*  
*Ans: This is considered to be malformed input. Return response code 500 and provide any reasonable error message.*
2. *Do I need to take care of non-ASCII characters, e.g. Chinese?*  
*Ans: Just consider English is ok.*
3. *For searching bookmarks with tag, is it a partial match/complete match (e.g. When searching for "cat", can we also return bookmark with tags like "catdog")?*  
*Ans: Partial matching is fine for simplifying implementation effort (you may use the SQL 'like' operator for searching the relevant bookmark).*
4. *For bookmarks, is every attribute required?*  
*Ans: You may assume all fields of bookmark is present.*
5. *If a user is deleted, should his/her bookmarks be deleted automatically?*  
*Ans: Also delete the related bookmarks if the user is deleted.*
6. *What should the web API respond when adding a bookmark for an unknown user?*  
*Ans: Return JSON error message "User Not Found" with error code 404.*
7. *What should the web API respond when retrieving/updating/deleting a bookmark for an unknown user?*  
*Ans: In case for unknown user, you can return JSON error message "User Not Found" with error code 404. There is no need to check if the bookmark is there.*
8. *In case more than one user is added, and at least one of them is invalid. How should the web API respond?*  
*Ans: The users are added to DB only if ALL users are valid. Otherwise, it will return error message and error respond codes without adding any of the users.*
9. *What happens if the user ID in the URL and the JSON in HTTP request body does not match (e.g. when adding a bookmark)?*  
*Ans: You may assume they are the same.*
10. *When deleting the bookmark, should I consider the bookmark in URL/the JSON bookmark in HTTP request body? What if they don't match each others?*  
*Ans: You just need to consider the bookmark in URL. You may ignore the JSON bookmark in the HTTP request body (if provided).*
11. *When deleting bookmark/user, what should be returned by the API?*  
*Ans: If the deletion is successful, return an empty string " in the HTTP response body and response code 204 (No Content).*