# **Discussion Forum**

### Web Service API

Your Flask application should expose the following RESTful URL endpoints. All data sent to and from the API should be in JSON format with the Content-Type header field set to application/json.

|  |  |
| --- | --- |
| Method | GET |
| URL | /forums |
| Action | List available discussion forums |
| Request Payload | N/A |
| Successful Response | HTTP 200 OK  [  { "id": 1, name: "redis", creator: "alice" },  { "id": 2, name: "mongodb", creator: "bob" }  ] |
| Error Response | None. Returns an empty array if no forums have been created. |
| Requires Authentication? | No |

|  |  |
| --- | --- |
| Method | POST |
| URL | /forums |
| Action | Create a new discussion forum |
| Request Payload | {  "name": "cassandra"  } |
| Successful Response | HTTP 201 Created  Location header field set to /forums/*<forum\_id>* for new forum. |
| Error Response | HTTP 409 Conflict if forum already exists |
| Requires Authentication? | Yes |
| Notes | Forum’s creator is the current authenticated user. |

|  |  |
| --- | --- |
| Method | GET |
| URL | /forums/*<forum\_id>* |
| Action | List threads in the specified forum |
| Request Payload | N/A |
| Successful Response | HTTP 200 OK  [  {  "id": 1,  "title": "Does anyone know how to start Redis?",  "creator": "bob",  "timestamp": "Wed, 05 Sep 2018 16:22:29 GMT"  },  {  "id": 2,  "title": "Has anyone heard of Edis?",  "creator": "charlie",  "timestamp": "Tue, 04 Sep 2018 13:18:43 GMT"  }  ] |
| Error Response | HTTP 404 Not Found if forum does not exist. |
| Requires Authentication? | No |
| Notes | The timestamp for a thread is the timestamp for the most recent post to that thread.  The creator for a thread is the author of its first post.  Threads are listed in reverse chronological order. |

|  |  |
| --- | --- |
| Method | POST |
| URL | /forums/*<forum\_id>* |
| Action | Create a new thread in the specified forum |
| Request Payload | {  "title": "Does anyone know how to start MongoDB?",  "text": "I'm trying to connect to MongoDB, but it doesn't seem to be running."  } |
| Successful Response | HTTP 201 Created  Location header field set to /forums/*<forum\_id>*/*<thread\_id>* for new thread. |
| Error Response | HTTP 404 Not Found if forum does not exist. |
| Requires Authentication? | Yes |
| Notes | The text field becomes the first post to the thread. The first post’s author is the current authenticated user. |

|  |  |
| --- | --- |
| Method | GET |
| URL | /forums/*<forum\_id>*/*<thread\_id>* |
| Action | List posts to the specified thread |
| Request Payload | N/A |
| Successful Response | HTTP 200 OK  [  {  "author": "bob",  "text": "I'm trying to connect to MongoDB, but it doesn't seem to be running.",  "timestamp": "Tue, 04 Sep 2018 15:42:28 GMT"  },  {  "author": "alice",  "text": "Ummm… maybe 'sudo service start mongodb'?",  "timestamp”: "Tue, 04 Sep 2018 15:45:36 GMT"  }  ] |
| Error Response | HTTP 404 Not Found |
| Requires Authentication? | No |
| Notes | Posts are listed in chronological order. |

|  |  |
| --- | --- |
| Method | POST |
| URL | /forums/*<forum\_id>*/*<thread\_id>* |
| Action | Add a new post to the specified thread |
| Request Payload | {  "text": "@Bob: Derp."  } |
| Successful Response | HTTP 201 Created |
| Error Response | HTTP 404 Not Found |
| Requires Authentication? | Yes |
| Notes | The post’s author is the current authenticated user. |

|  |  |
| --- | --- |
| Method | POST |
| URL | /users |
| Action | Create a new user |
| Request Payload | {  "username": "eve",  "password”: "passw0rd"  } |
| Successful Response | HTTP 201 Created |
| Error Response | HTTP 409 Conflict if username already exists |
| Requires Authentication? | No |

|  |  |
| --- | --- |
| Method | PUT |
| URL | /users/*<username>* |
| Action | Changes a user’s password |
| Request Payload | {  "username": "eve",  "password": "s3cr3t"  } |
| Successful Response | HTTP 200 OK |
| Error Response | HTTP 404 Not Found if username does not exist  HTTP 409 Conflict if username does not match the current authenticated user. |
| Requires Authentication? | Yes |

### Authentication

Use [HTTP Basic authentication](https://en.wikipedia.org/wiki/Basic_access_authentication) to authenticate users. You may implement this yourself, or install the [Flask-BasicAuth](https://flask-basicauth.readthedocs.io/) extension.

To use Flask-BasicAuth, create a [subclass](https://www.digitalocean.com/community/tutorials/understanding-class-inheritance-in-python-3) of [flask.ext.basicauth.BasicAuth](https://flask-basicauth.readthedocs.io/en/latest/" \l "flask.ext.basicauth.BasicAuth) and override its [check\_credentials](https://flask-basicauth.readthedocs.io/en/latest/" \l "flask.ext.basicauth.BasicAuth.check_credentials) method to read from the database instead of comparing to the BASIC\_AUTH\_USERNAME and BASIC\_AUTH\_PASSWORD configuration variables.

### Session State

All requests to the Web Service API must include all information necessary to complete the request; your API may not use the Flask session object to maintain state between requests.

### Flask Installation

The easiest way to install Flask is probably the following shell command:

$ **pip3 install --user flask**

### Database

Use [The Python Standard Library](https://docs.python.org/3.6/library/)’s [sqlite3](https://docs.python.org/3.6/library/sqlite3.html) module as the [database for your Flask application](http://flask.pocoo.org/docs/1.0/patterns/sqlite3/).

#### Initializing the Database

As you test your API, you may find that you need to reset its state by dropping and re-creating the database. When that happens, do not waste time by going back and adding users, forums, threads, and posts by hand. After all, [performing simple, repetitive tasks by hand makes you dumber.](https://dzone.com/articles/automation-people-because)

Create a file init.sql containing SQL CREATE TABLE, CREATE INDEX, and INSERT statements. Add a [custom command](http://flask.pocoo.org/docs/1.0/cli/" \l "custom-commands) to your Flask application so that you can create an [initial schema](http://flask.pocoo.org/docs/1.0/patterns/sqlite3/" \l "initial-schemas) and populate the database with test data by running the following shell command:

$ **flask init\_db**

### Tips

* See [Creating Web APIs with Python and Flask](https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask) to get started with Flask.
* See [Using SQLite 3 with Flask](http://flask.pocoo.org/docs/1.0/patterns/sqlite3/) for several useful tips.
* Your web browser can be used to test GET methods, but to test POST and PUT, consider using a tool like [curl](https://curl.haxx.se/), [httpie](https://httpie.org/), or [Postman](https://www.getpostman.com/) or a Python library like [Requests](http://docs.python-requests.org/en/master/) or the Flask [test\_client](http://flask.pocoo.org/docs/1.0/testing/).
* While you can interact with the database solely through Python code, you may also wish to use a command-line SQLite client like [sqlite3](https://sqlite.org/cli.html) or a GUI client like [DB Browser for SQLite](https://sqlitebrowser.org/) to manage your database.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_