

Team: Nexus

Part 1:

Date met with IA Mentor: 10/20/2021

1. Team Name: Nexus
2. Team Members
 - Vaibhav Goyal (vg2498)
 - Nihar Maheshwari (nm3223)
 - Talya Koschitzky (tk2892)
 - Shantanu Jain (slj2142)
3. Programming Language: Python
4. Mac
5. Github Repo: <https://github.com/niharmaheshwari/nexus>

Part 2:

Overview

We are developing a note-taking system specifically for developers. The idea is to allow developers to save some important routines, functions, classes or CLI commands as part of the service, for easy retrieval at a later date. It is **not** similar to a text-based note taking service. The key difference is that we will be providing language-specific features such as autotagging, linting and search. Users will have an option to upload their snippets, along with adding tags and comments to them.

Thus, this service satisfies the following requirements from the assignment:

- Persistent database of users and their tagged code snippets.
- Supports multiple users with authentication.
- No GUI. Will showcase the service using Postman (tentative).

What will your service do?

Our service will allow users to easily keep track of and search code snippets that they frequently reuse.

What kind of functionality or features will it provide?

1. Uploading code snippets in Python, C++, Java and shell scripts.
2. Tagging and annotating code snippets with descriptions and comments.
3. Autotagging code snippets.
4. Searching code snippets by comments, tags, and descriptions.
5. Linting/static analysis of code snippets.

6. Sharing of code snippets between users.
7. User profiling with authentication

Elaborating important features:

- Sharing of code snippets between users
 - Implement user groups and memberships to those user groups. We plan to use Cognito RBAC for this.
- Uploading code snippets in Python, C++, Java and shell scripts:
 - HTTP: POST
 - The body parameters will include the following:
 - Serialized text of code snippet
 - Session ID of the user
 - Tags (optional) provided by the user
 - Description (optional)
 - Response: 200 OK
- Searching code snippets by comments, tags, and descriptions:
 - HTTP: POST
 - Body:
 - Language Tags: Python, C++ and Java
 - Custom Tags: Manual tags
 - Descriptions: Text based set of keywords
 - Comments: Text based comments (in code snippets) -- Tentative
- Autotagging code snippets
 - In the event where the user does not tag the script during input, we plan to infer the programming language of the snippet uploaded.

Good to have features: (tentative)

1. Compilation engine for the above mentioned languages. To return compiler issues.
2. Image to code synthesis. (Users could upload images of their code snippets, and the note scanner should be able to extract the relevant code.)

Who or what will be its users?

Anybody writing code, specifically developers or CS students.

What might they use the functionality for?

They will use it to store and easily search common code snippets that they otherwise have to look up with google. This is a time consuming and a repetitive process. We wish to eliminate this by allowing the user to have his/her own code snippet repository.

Note that the endpoints / functionalities such as linting of uploaded code, storage and retrieval of code snippets, tagging of code are generic and do not apply only to a note taking service. It has numerous applications potentially e.g. online programming platforms (where users can code on a web browser, code repository browser etc.)

What kind of data will your service create or accumulate?

Unstructured code snippets, user profile information. Every user will have a separate profile, with session identifiers for maintaining their sessions. Hence, *persistent data of the user's session will be maintained across executions*. Thus, *the service does support multiple clients, with different session ids for each client*.

What will the data be used for?

Searching, retrieving, sharing code snippets. Here, by default each user will have access to his/her own set of code-snippets only. But, users can share some code snippets with other users, allowing the access to be shared among them.

Part 3:

How will you test that your service does what it is supposed to do and provides the intended functionality?

Planning to use the following:

- Unit Test cases for checking intended functionality.
- Selenium / Jmeter for stress testing our endpoints on various input cases.
- Postman Collections and Test Events to test individual APIs

How will you check that your service does not behave badly if its clients use it in unintended ways or provide invalid inputs?

- Restrict the users to text based inputs. Users cannot add images / zips etc.
- Validate incoming input to check for errors. For example, the user's input should be one of the languages that our application supports.
- Test with invalid inputs and ensure an appropriate error message is returned

How will you test that your service handles its data the way its supposed to?

- Testing the data along each step of the transformations that are happening throughout the application. Ex. Testing before and after a DB call, testing before and after an API request (which may fetch data from elsewhere) etc. **This will**

ensure that we check that meaningful data is persisted across each step of the execution

- One feature that we ensure is multi-user support. We plan to have client (user) information passed to each and every API call through Cookies. This will help us maintain the user sessions. We potentially plan to test this feature by having scripts that shuffle between valid cookies and make requests to our service. **This will allow us to test API calls simulating multiple users.**
- Apart from call scripts, all our tests will run as collection runs on Postman. This will allow us to test the entire functionality of the service **without having any application UI.**
- Mocking Tests for checking Data Manipulations across the entire application
- For instance : Using postman to POST a code snippet, verifying that it is stored and indexed correctly in the database, and then using postman to GET the same code snippet, and verifying that it is the same.

Part 4

Libraries & Frameworks:

- Flask
- AWS Cognito (User authentication)
- Elasticsearch (For indexing & searching on the description)
- DynamoDB (Code snippet information)
- PyLint (Style checking & static analysis)
- Coverage (Code coverage)
- Postman (for making client side api calls)