**Difficult technical issue:**

1. **token generation and authentication**
2. *Why we need token?*

Token is mainly used for interface protection. We need to make sure the interface will not be visited freely and guarantee the legal user identification for every visit. We also don't want user to input their username and password frequently.

1. *How to use token?*

The source of token is the certificate we give users once they have passed the login authentication. The user will be count as an illegal user when he/she brings a valid certificate to visit our interface. It is unsafe to store token in local client side, so each token will expire after specific time to make sure even anyone will not hold a valid token permanently.

1. *Three essential characteristics of token:*
2. Period of validity.
3. Can identify user’s identity and status.
4. Can be encoded.
5. *Implementation:*

We used jwt packet to encode every useful information inside of user’s token.

图片包含 游戏机, 截图, 鸟

描述已自动生成

Also used jwt to decode a input token. If this token can be decoded and not expire, it will be counted as a valid token and send information inside of token back to certain interface.

图片包含 截图, 游戏机

描述已自动生成

1. *custom decorator:*

图片包含 游戏机, 截图

描述已自动生成We made our own decorator called required\_login, any interface which can be visited after user login will have this decorator before their Api function.

1. **Pagination**

To make users more easily to explore and locate result, pagination is used in book search and review list part. So we need to design an accurate pagination algorithm to show all result in certain page.

1. Get the result and the number of result.
2. By inputting the parameter for number of result on each page, get total number of pages and the number of result which will be shown on last page.
3. Get the index of result which will be shown on the page chose by user.
4. Obtain the result according to the index.

图片包含 游戏机, 文字

描述已自动生成Below is the flowchart for book search pagination.

1. **Recommend**

In this assignment, we have not used machine learning to recommend book. So basically, we gave each book a popular value which decided by number of been collected, number of been read, google rating and our website user rating. Each recommend mode will filter out the books with certain restriction and sort out the books with highest popular value in subset.

At very first, we will create a table to store every book’s information which will be used in the recommendation by SQL query, and create a popular entity for each book. In every recommend mode, we used pandas to filter out the books with requirement and then sort by popular value.