LUNIT EXAM Project Document

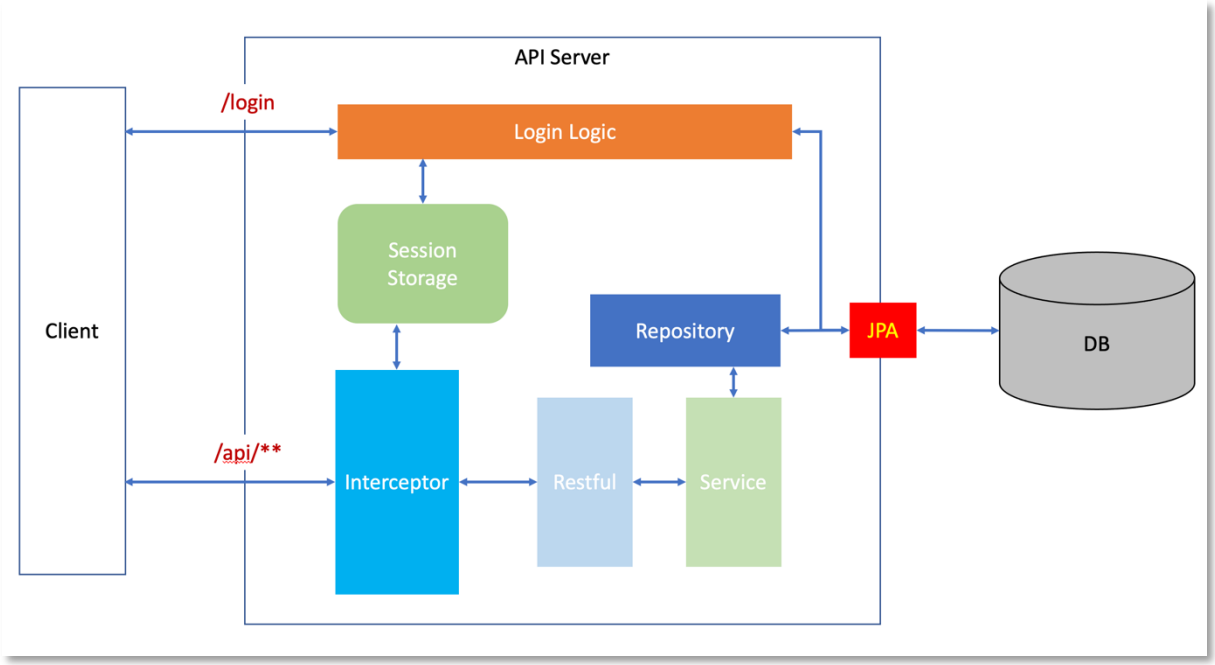
Contents

조 민수 (Minsu Jo)

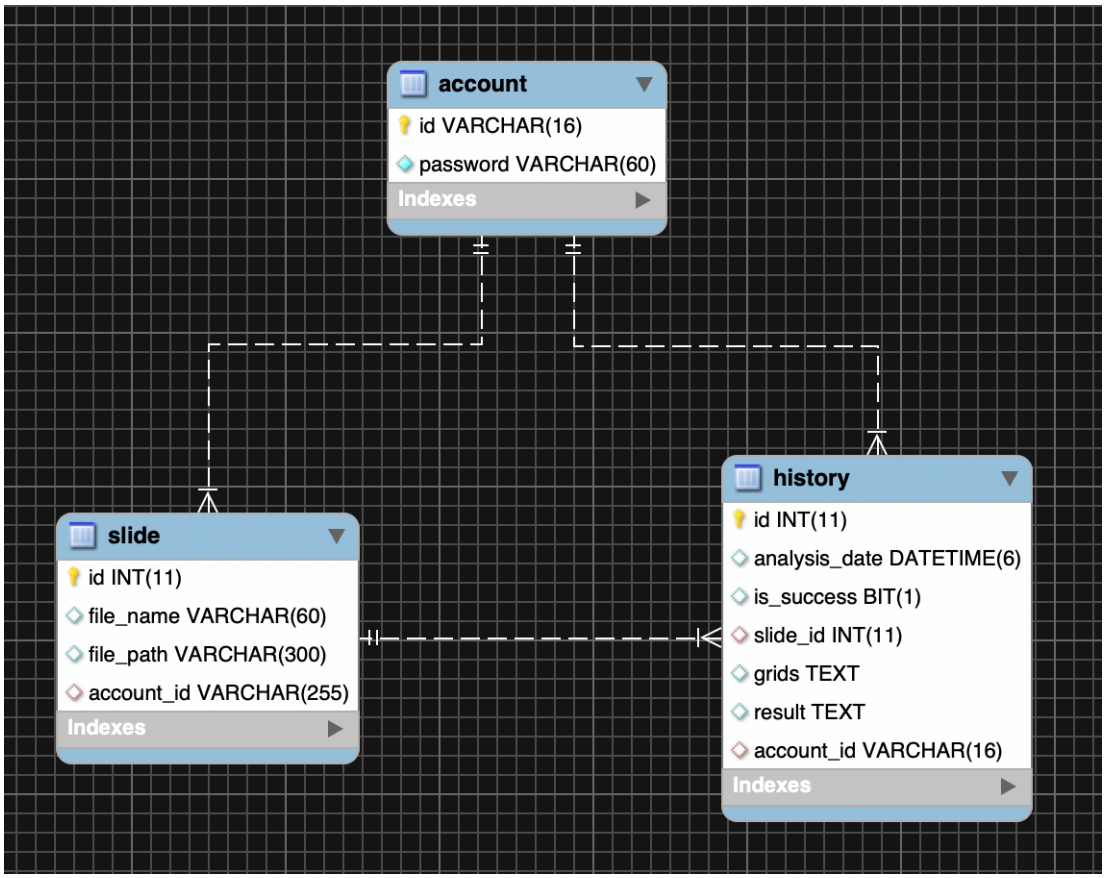
tok0419@naver.com

https://github.com/J0minsu/Lunit_Test

1. Structure



2. DB ERD



3. API Specification

Account		
Method	URI	Description
POST	/login	Login and Session issuance

Slide		
Method	URI	Description
GET	/api/slides	Return downloadable slide list
GET	/api/slides/{search}	Return contain {search} in slide name
GET	/api/slides/download/{slideId}	Download Slide
POST	/api/slides	Upload slide on server storage

History		
Method	URI	Description
GET	/histories	Return findable history list
GET	/histories/{historyId}	Return history info
GET	/histories/analysis/{slideId}	Analysis slide and return result

4. Result

Essential Requirements

1. Login and Authentication (Complete)
2. Slide img file up/download (Complete)
3. Find all slide only you uploaded, Search by file name (Complete)
4. Require slide analysis and immediately receive result (Complete)
5. Whenever you can find your analysis history (Complete)
6. using docker-compose, you can publish development env (Incomplete)

Optional Requirements

1. OAuth2.0 (Incomplete)
2. Divide ‘authentication server’ and ‘API server’ (Incomplete)
3. Consider slide img’s size is bigger then 2GB (Complete)
4. Consider slide analysis take more than 10 minutes (Complete)
5. Good if you have test code (Incomplete)

5. Truble Shooting

1) String form 의 Json 파싱 시 특정 자료형에서 Error 발생

ObjectMapper 를 이용해 String(Json Form) — convert —> dto 시,

Boolean 등의 특정 자료형에서 convert 안되는 현상 발견했습니다.

try 1) ObjectMapper 의 설정을 변경

```
ObjectMapper mapper = new ObjectMapper()
.configure(DeserializationFeature.FAIL_ON_UNKNOWN_PROPERTIES, false);
```

➔ Error 는 발생하지 않았지만 아예 무시해서 NULL 값으로 들어갔습니다.

try 2) dto 의 각 Property 에 @JsonProperty 설정

```
@JsonProperty(value = "Decision")
private Boolean Decision;
```

➔ 정상적 작동 확인

6. Test

1. Login

   localhost:8080/login (POST)

   but, in browser, localhost:8080/test (specially premise for authentication)

   = AccountId = ‘msjo’,

   If, you want to change account to ‘jslee’ -> /test/otherLogin

2. Slide up/download and search

   In test page, you can test

3. Analysis slide, history find and get

   Base on api specification, you can test.

7. takeaway

과제를 진행하면서 많은 것들을 깨닫고, 배울 수 있었으며, 어떤 것이 부족한지 알게되어 한단계 성장할 수 있었습니다. 성장할 수 있는 기회를 주셔서 감사의 말씀 드립니다.

첫째로 API 서버를 개발할 때, Validation에 대해 많은 고뇌와 정교함이 포함된 설계가 필요하다는 것을 알게 되었습니다. 어떤 방식으로 동작하게끔 API를 구현할지, Validation은 어떤 것들, 어느 부분에서 해야하는지, Junit 을 사용해 여러 예외 케이스들을 생성해보면서 개발했습니다. 그렇게 하니 기존의 개발속도보단 더뎠으나, 완성도 높은 API 서버를 구축할 수 있었습니다.

두번째, 대용량 데이터를 사용자에게 전달할 때, 어떤 방식으로 전달해야 하는지 고민을 해보았습니다. 기존에는 조건에 만족하는 모든 데이터들을 전달해주는, 비효율적인 방식을 생각했겠지만, ROW가 수천, 수만의 경우를 생각해봤습니다.
퍼포먼스를 고려했을 때, DB 내에서 Pagination 을 통해 조건에 맞는 일부의 데이터를 제공하는 방식을 알 수 있었습니다.

세번째, 데이터 리스트를 조회할 때, 최소한의 정보만을 제공하고, 리스트에서 원하는 Row 를 상세히 보고자 한다면, 그 때 해당 Row 의 정보를 모두 전달해주는 방식을 고려해봤습니다. Column의 양이 방대하고, row 또한 방대하다면, 리스트 모두를 가져오는 것의 비용이 상당할 것이라고 판단했고, Mapping 클래스를 사용을 해봤습니다.

이외에도 깨달은 점, 배운 점, 스스로를 자아성찰 할 수 있는 최고의 경험이었으며, 터닝포인트가 되는 경험이었습니다. 테스트를 진행하는 3일간 최선을 다했으며, 즐겁게 임했습니다. 다시 한번 감사의 말씀 드립니다. (_ _)

- 조민수