

Todo App

- **Todo App**
- **사용 언어 및 기술**
 - 프레임 워크 : Ruby on Rails, Bootstrap
 - 데이터베이스 : sqlite3
- **Repository**
 - <https://github.com/qkrrudtjr954/todo-app>
- **동기**
 - 바쁜 일정을 소화하는 동안 할일을 구체화하지 못하고 잊어버리는 경우가 생기기 때문에 간단하게 일정을 관리할 수 있는 Todo-app을 개발 하였습니다.
- **설명**
 - ‘멋쟁이 사자처럼’ 동아리를 통해 배운 Ruby on Rails 프레임 워크를 사용하여 개발하였습니다.
 - devise gem 을 사용한 사용자 로그인 로그아웃 기능을 구현하였습니다.
 - scaffold 를 사용한 기본 CRUD를 구현하였습니다.
 - 로그인된 유저는 해야할 일을 등록할 수 있습니다. 그리고 그 아래 세부 사항을 입력할 수 있습니다.
- **결과**
 - 개인 아이디를 사용한 로그인 기능을 통해 일정 관리를 할 수 있습니다.
 - Ruby on Rails의 RESTful 한 개발 방식을 이해할 수 있었습니다.

- PaPaoTalk 메시지 전송 프로그램

- 사용 언어 및 기술

- 언어 : Java
- 프레임 워크 : Swing, JDBC
- 데이터베이스 : Oracle
- 구조 : Spring MVC (DAO와 Service의 분리)

- Repository

- <https://github.com/qkrrudtjr954/papaoTalk>

- 동기

- Java Swing을 사용하는 프로젝트를 진행하는 도중 진행한 서브 프로젝트입니다.
- 관리자와 사용자 간의 채팅을 통해 문의 또는 답변을 하기 위해 제작했습니다.

- 설명

- '회원으로 가입된 사용자는 관리자 또는 다른 사용자에게 메시지를 전송할 수 있습니다.
- 전송된 메시지는 데이터베이스에 저장되며, 상대방 역시 메시지를 전송하여 1:1 메시지 전송이 가능합니다.
- Dto 는 채팅 방과 채팅 글, 사용자 총 3개로 이루어져 있고 채팅방과 채팅 글은 1:N의 참조 조건을 맺고 있습니다.
- UI는 최근에 사용 중인 채팅 애플리케이션의 디자인을 벤치마킹했습니다.

- 결과

- 등록된 사용자 간의 메세지 전송이 이루어졌습니다.
- 확장성있는 개발로 원래 프로젝트에 잘 이식되었습니다.
- 프로젝트 개발 기간이 길지 않았기 때문에 소켓 통신을 통한 실시간 채팅으로 전환하지 못한 것이 아쉬웠습니다.