

Offertunity Project

이승연

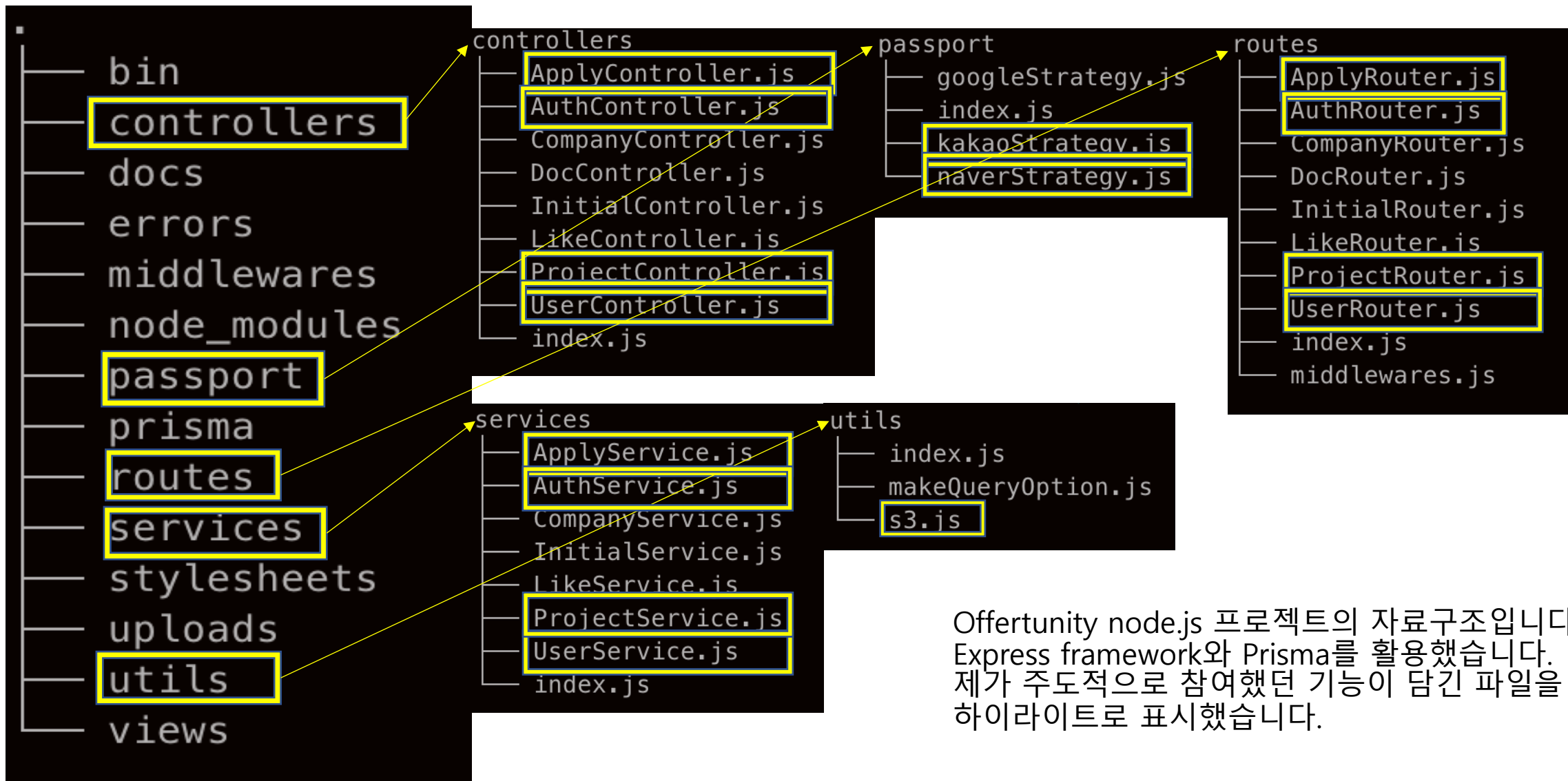
목차

1. Offertunity 프로젝트 개요
2. Offertunity 프로젝트 파일구조
3. 회원정보 수정 기능 관련 코드
4. 지원사업(project) 업로드 및 수정 기능 관련 코드
5. AWS S3 버킷저장소 파일 업로드 기능 관련 코드
6. 회원정보 삭제 cascade 기능 관련 코드

1. Offertunity 프로젝트 개요

- 스타트업과 파트너기업 간 투자유치 활동 도모를 위한 플랫폼을 node.js로 구현했습니다.
- 일주일 반 가량 자바스크립트와 node.js를 학습하고 3주일 동안 구현한 프로젝트입니다.
- Express framework와 Prisma를 활용했습니다.
- 스타트업 엑셀러레이팅을 하고자하는 '파트너'기업이 지원사업에 대한 세부내용을 올리면 '스타트업'기업이 지원할 수 있는 핵심기능을 갖고 있는 웹사이트입니다.
- Express framework를 이용하였고 각 layer에 대한 기능은 다음과 같은 접두사로 분류됩니다:
 - Apply- : 스타트업의 지원현황을 관리합니다.
 - Auth- : 카카오, 네이버, 구글 등 소셜 로그인 기능을 관리합니다.
 - Company- : 기업정보를 관리합니다.
 - Doc- : 기업 간 문서 관리와 전송을 관리합니다.
 - Initial- : 초기 데이터 업로드 자동화를 관리합니다.
 - Like- : '좋아요' 기능을 관리합니다.
 - Project- : 지원사업 내용을 관리합니다.
 - User- : 회원정보를 관리합니다.

2. Offertunity 프로젝트 파일구조



Offertunity node.js 프로젝트의 자료구조입니다.
Express framework와 Prisma를 활용했습니다.
제가 주도적으로 참여했던 기능이 담긴 파일을
하이라이트로 표시했습니다.

3. 회원정보 생성 및 수정 기능 관련 코드

- POST로 수정요청이 들어오면 데이터를 전개구문을 활용해 회원정보를 업데이트했습니다.
- (코드) userinfo_edit/UserRouter.js, userinfo_edit/UserController.js, userinfo_edit/UserService.js

4. 지원사업(project) 업로드 및 수정 기능 관련 코드

- 스타트업 엑셀러레이팅을 하고자하는 '파트너'기업이 지원사업에 대한 세부내용을 올리고 수정할 수 있는 기능을 구현했습니다.
- path parameter로 projectId가 들어오면 해당 프로젝트 데이터를 반환하고 들어오지 않으면 새로운 프로젝트가 생성됩니다.
- 프로젝트에 foreign key로 묶여있는 related_documents 조건이 수정되었을 때 초기화하고 해당 데이터를 재생성하는 resetChoices 함수를 구현했습니다.
- (코드) project_upload_edit/ProjectRouter.js, project_upload_edit/ProjectController.js, project_upload_edit/ProjectService.js

5. AWS S3 버킷저장소 파일 업로드 기능 관련 코드

- multer 미들웨어를 활용하여 프로젝트 전반에 걸쳐 사용되는 모든 이미지 및 파일을 s3 버킷에 업로드 가능하게 구현했습니다.
- 제 [블로그](#)에 자세한 코드와 내용을 정리해놓았습니다([클릭](#)해서 이동).

6. 회원정보 삭제 cascade 기능 관련 코드

- 회원정보 삭제 기능을 구현하려는 중, 회원의 아이디로 활동한 기록이 있으면 자동으로 foreign key로 묶인 데이터까지 삭제되는 cascade on delete 기능을 prisma로 구성한 데이터베이스에 반영하지 않았다는 사실을 알았습니다.
- 데이터베이스 속성을 변경하기에는 이미 프로젝트를 마무리해야하는 단계였고 모든 팀원이 프론트와 백엔드를 맞춰보기 위해 데이터베이스를 활발하게 활용 중이었습니다.
- 따라서 정식으로 cascade on delete 구현하기는 다음 리팩토링 때 하기로 결정하고 본 기능을 직접 구현했습니다.
- (코드) cascade_on_delete/UserService.js: 실제 회원정보를 삭제하기 전 foreign key로 엮인 startup_likes, project_likes 등 테이블에 담김 해당 유저의 정보를 선제적으로 삭제했음을 볼 수 있습니다.
- (코드) cascade_on_delete/UserRouter.js, cascade_on_delete/UserController.js