

한 번에 끝내는 블록체인 개발 A to Z

Chapter 5

Lottery v2 컨트랙트 API 서버 개발

Chapter 5

Lottery v2 컨트랙트 API 서버 개발

Express를 사용한 Lottery v2 컨트랙트 API 서버 구조 파악하기



API 서버로 Express를 사용하는 이유

Express 프레임워크란

Express 프레임워크

- Node.js 웹 애플리케이션 프레임워크
- http 통신 요청(Request; GET, POST, DELETE 등)에 대한 핸들러 제공
- 템플릿에 데이터를 넣어 응답(response)을 만들기 위해 view의 렌더링 엔진과 결합
- 요청을 처리하는 파이프라인 중 필요한 곳에 추가적인 미들웨어 처리 요청 추가

컨트랙트 API 서버로 Express 프레임워크를 사용하는 이유

- Node.js를 사용하는 가벼운 웹 애플리케이션 프레임워크
- API 서버 내에서 언어의 교환 없이 가장 많이 사용되는 web3.js를 사용하여 컨트랙트와의 통신 가능

Express 프레임워크 구조 생성하기

- express generator 사용하기
- <https://www.npmjs.com/package/express-generator>
- 설치 명령어: npm install express-generator -g
- 구조 생성 명령어: express --view=<view 템플릿 엔진명> <프로젝트명>
 - ex) express --view=pug lottery-v2-api-server

```

  bin
  www
  public
    images
    javascripts
    stylesheets
      style.css
  routes
    index.js
    users.js
  views
    error.pug
    index.pug
    layout.pug
  app.js
  package.json
```




Lottery v2 컨트랙트 API 서버 구조 파악하기

Express 프레임워크

기본 구조

- bin/www: express 서버를 구동하는 역할
- public: 정적 페이지를 구성하는 곳
- routes: API 설정하는 곳
- views: 클라이언트에 응답 보낼 때 html 템플릿으로 변환해서 반환하는 파일을 정의하는 곳
- app.js: express 서버 설정 파일
- package.json: 서버에서 사용되는 모듈 리스트 및 버전, 서버 실행 명령어 설정 파일

```

  bin
  ├── www
  public
  ├── images
  ├── javascripts
  ├── stylesheets
  │   └── style.css
  routes
  ├── index.js
  ├── users.js
  views
  ├── error.pug
  ├── index.pug
  ├── layout.pug
  ├── app.js
  └── package.json
```


Lottery v2 API

서버 구조

- config: 환경변수 등 서버 설정 파일 모음
- constants: 서버에서 공통적으로 사용하는 상수 모음
- contractAbis: 컨트랙트 ABI 파일 모음
- controllers: 각 API에 대해 로직적인 부분을 처리하는 부분
- data: 도커 컨테이너 DB 내용을 로컬에 저장해놓은 부분
- migrations: 테이블 구조를 update, rollback 하는 내용을 작성하는 부분
- models: DB 세팅 및 테이블 정의하는 곳
- services: DB 및 컨트랙트와 직접적으로 상호작용하는 부분, controllers에서 이용
- .sequelizerc: sequelize 디폴트 config 파일 path 조정하는 부분
- docker-compose.yml: 도커 컴포즈로 DB 컨테이너 생성

▼ LOTTERY-V2-API-SERVER

- > bin
- > config
- > constants
- > contractAbis
- > controllers
- > data
- > migrations
- > models
- > node_modules
- > public
- > routes
- > seeders
- > services
- > views
- ≡ .sequelizerc
- JS app.js
- 🐳 docker-compose.yml
- { } package-lock.json
- { } package.json

Lottery v2 API

서버 전반적 흐름

- API 서버로 API 요청이 들어오면, routes를 통해서 매핑되는 controller로 전송
- controller는 API 요청 처리에 대해 전반적으로 관리하는 컨트롤타워 역할
- controller는 필요에 따라 services의 interactor와 통신
 - 컨트랙트와의 통신 필요시 Contract Interactor와 통신
 - DB와의 통신 필요시 DB Interactor와 통신
- 각 interactor는 실제 contract 또는 DB와 통신하는 매개체 역할
- controller는 services와 상호작용하며 API response를 만들어 리턴

