

[README] Main Page Build & Deploy Manual

📋 순번	110
🔗 구분	Basic
🌟 개발 상태	완료
🌟 front 연결	완료

Carebot Project Build & Deploy

1. Frontend - Main Platform

사용된 기술



분야	사용한 기술
Program Language	Javascript
Language Engine	Node.js 22.13.1 - Alpine
Web SPA Library	React 18.3.18
Bundling Tool	Vite 6.0.5
Code Analysis Tool	ESLint 9.17.0

사용되는 환경 변수

```
// .env file
REACT_APP_DEV_API=https://dev-api.itdice.net
```

```
REACT_APP_IMAGE_API=https://image.itdice.net
NODE_ENV=production
```

빌드 방법



이 프로젝트는 Docker를 이용한 빌드 과정을 수행합니다. **사전에 Docker와 Docker Compose를 설치하는 과정이 필요합니다.**

1. 우선, 해당 Repository를 Clone하여 모든 파일을 받습니다.
2. `/FE/main-user/express-server` 로 이동합니다.
3. `.env` 파일을 생성하고 해당 내용을 추가합니다.

```
// .env file
REACT_APP_DEV_API=https://dev-api.itdice.net
REACT_APP_IMAGE_API=https://image.itdice.net
NODE_ENV=production
```

4. 그 다음, `/FE/main-user` 로 이동합니다.
5. 이미 실행 중인 Container가 있으면, 컨테이너를 끄고 이미지를 삭제하기 위해 아래의 명령어를 입력합니다. 없다면 건너 뛰어도 됩니다.

```
docker compose -f "docker-compose-dev.yml" down --rmi all
```

6. 이후에 Docker Image를 Build하고, Container를 실행하기 위해 다음의 명령어를 입력합니다.

```
docker compose -f "docker-compose-dev.yml" up -d --build
```

7. Container가 생성되었다는 아래의 콘솔 로그를 확인하고, Host의 `127.0.0.1:4580` 에서 Main Page를 접근할 수 있게 됩니다.

```
dev-main Built
Network main-user_default Creating
Network main-user_default Created
Container dev-main Creating
```

```
Container dev-main Created
Container dev-main Starting
Container dev-main Started
```

유의 사항

1. User API Server, AI Process Server, Image Provider가 실행되고 있어야 서비스
가 정상 작동됩니다.
2. 환경 변수 정보는 배포 환경에 따라 다음과 같이 변경합니다.

변수 이름	내용
REACT_APP_DEV_API	User API Server의 접근 주소
REACT_APP_IMAGE_API	Image Provider의 접근 주소

3. Nginx 설정 시, SSL 인증서를 이용한 **HTTPS로 접근하는 방식**을 추천합니다. (민감한
개인정보가 Body에 담겨서 API에 보내지고 있기 때문입니다)

테스트 계정

ID	main-test@itdice.net
PW	Main!test0123