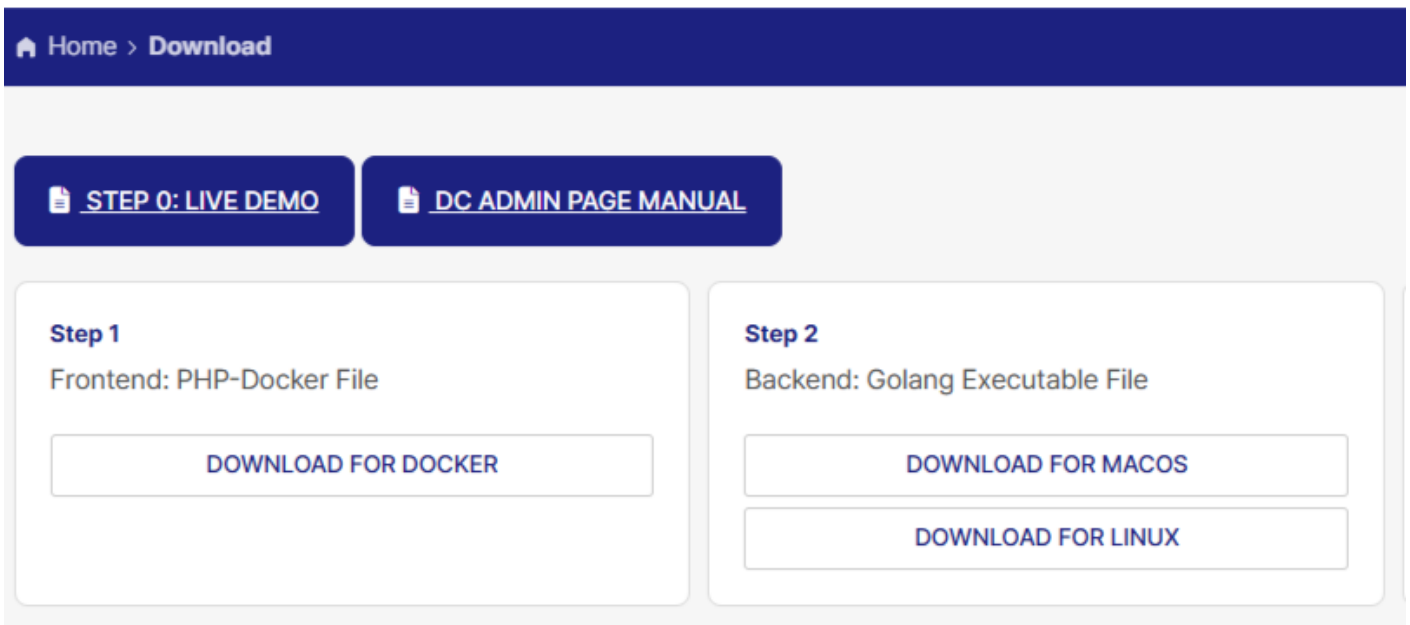


DC-Demo 배포 파일 다운로드 (Local)

DC 배포 파일 다운로드 (Mac OS, Linux 배포)

1. DC 배포 파일 다운로드 Backend: Golang Executable File 다운로드.



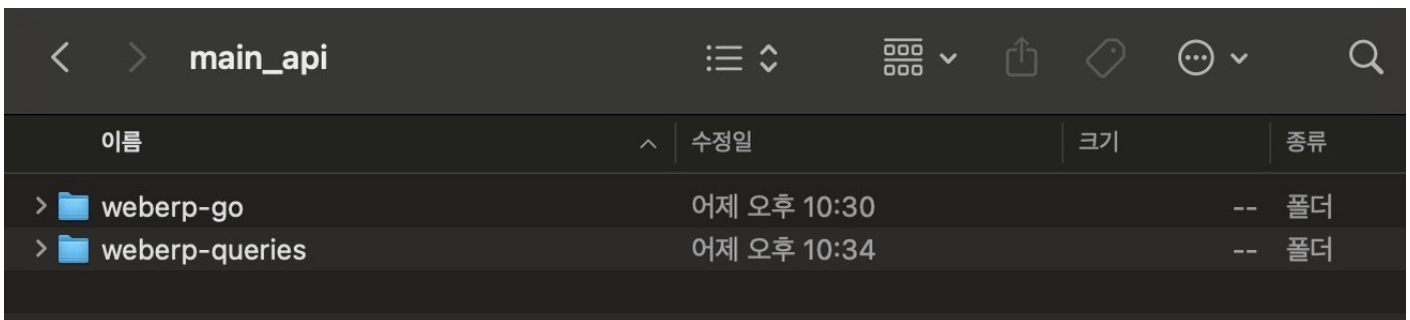
2. `mv ~/Downloads/dc_macos_main_api.tar.gz /path/to/~` 다운로드 파일 다운로드.

3. `tar` 명령어 사용.

`cd /path/to/~`

`tar -xvzf dc_macos_main_api.tar.gz`

4. `main_api` 폴더 생성 `weberp-go`, `weberp-queries` 폴더 생성.



weberp-go (weberp-go) : GateToken을 사용하여 인증을 받습니다.

weberp-go / conf : weberp-go의 설정 파일을 관리합니다.

weberp-queries : 쿼리 파일을 관리합니다.

Conf / local_config.json

Kafka 설정

- **KafkaOn** : Kafka를 사용할지 여부 (Yes/No).
- **KafkaTopic** : Kafka의 주제명 (Topic Name).
- **KafkaAddr** : Kafka의 주소 (IP 주소).
- **KafkaPort** : Kafka의 포트 번호.

gRPC 설정

- **gRpcOn** : gRPC를 사용할지 여부 (Yes/No).
- **AppName** : 애플리케이션 이름.
- **SiteName** : 사이트 이름.
- **gRpcProtocol** : gRPC의 프로토콜 (http/tcp).
- **gRpcAddr** : gRPC의 주소 (IP 주소).
- **gRpcPort** : gRPC의 포트 번호.

Rest 설정

- **RestOn** : REST API를 사용할지 여부 (Yes/No).
- **RestConnect** : REST API의 연결 주소 (IP 주소:8080).
- **RestUri** : REST API의 URI (/api).

개발 모드 설정 (개발 모드, 테스트 모드 등)

- **IsDevMode** : 개발 모드 여부 (Yes/No).
- **IsSqlDebugMode** : SQL 디버그 모드 여부 (Yes/No).
- **IsNormalDebugMode** : 일반 디버그 모드 여부 (Yes/No).

DB 설정

- **IsYDBFixed** : YDB 고정 모드 여부 (Yes/No).
- **DBOptionString** : DB 옵션 문자열.

XDB 설정

- **XDBOn** : XDB를 사용할지 여부 (Yes/No).
- **XDBConnString** : XDB의 연결 문자열.

Crystal 설정

- **CrystalClientId** : Crystal 账号 ID
- **CrystalBB64** : Crystal 账号 Base64 值
- **IsCrystalKeyPair** : Crystal 是否 Yes/No)
- **CrystalKeyPair** :

账号 名称

- **IsCacheKeyPair** : CacheKeyPair 是否
- **CacheKeyPairDir** : CacheKeyPair 目录
- **LocalKeyPair** :

SSO 配置

- **SsoConnString** : SSO 连接字符串
- **SsoAppBase64** : SSO 应用 Base64 值

DBU 配置

- **DbuConnString** : DBU 连接字符串
- **DbuAppBase64** : DBU 应用 Base64 值
- **DeviceAuthOn** : 设备认证是否
- **DbuByForceOn** : 是否 DBU 强制

主题 名称

- **IsQryFromQDB** : QDB 是否 主题 名称
- **QueryDir** : 主题 名称 (/Weberp-quary 主题 名称)
- **ThemeQryDir** : 主题 名称 (/Weberp-quary/themes 主题 名称)

主题 名称

- **DbuEncryptCode** : DBU 加密代码
- **SslMode** : SSL 模式
- **SslConnect** : SSL 连接
- **SslFullChain** : SSL 完整链
- **SslPrivate** : SSL 私钥

Blockchain 配置

- **GethConnString** : Geth(以太坊) 连接字符串
- **XrpConnBridge** : XRP(瑞波) 连接字符串

数据库 名称

- **DbType** : 数据库类型 (mysql/postgresql)

Kafka consumer 配置

- **IsKafkaConsumer**: Kafka Consumer 是否 是
- **KafkaConnString**: Kafka 连接字符串 (例: 123.456.789.101:9092)
- **MainProduceTopic**: Kafka 主生产主题
- **ConsumerTopics**: Kafka 消费者主题
- **KafkaTimeout**: Kafka 超时时间 (秒)

ElasticSearch 是否 是

- **ElasticConnString**: ElasticSearch 连接字符串

Conf / Config_select.json

config_select.json : 配置选择文件

weberp-queries (weberp 查询)

weberp-queries 文件

1. 查询数据库表名 (表名)
2. 查询数据库表名和表结构 (表名, 表结构)
3. 查询数据库表名和表数据 (表名, 表数据)

(表名 = 数据库名.表名, 表结构 = 表名.表结构)

Weberp-queries / themes

weberp-queries 文件

ex) Weberp-queries / themes / erp / eye / eyetest.sql

SELECT dbr_sorder.* FROM dbr_sorder WHERE dbr_sorder.sorder_no=?

Cache-key-pair

缓存键对 (Dabory Composable) 键对 (key pair) 缓存键对

- 缓存键对
 - 公共/私有键 (Public/Private Key) 缓存键对

- 在本地配置文件中配置缓存键对
- 使用缓存键对
 - 在本地配置文件中配置缓存键对
- 使用缓存键对
 - 在本地配置文件中配置缓存键对

Cache-key-pair 使用

- 在本地配置文件中配置缓存键对
- 在本地配置文件中配置缓存键对
- 在本地配置文件中配置缓存键对

5. weberp-go 在 `./weberp-go` 目录下使用。

在 **18080** 端口上运行。 (`local_config.json` 文件中 `RestConnect` 配置项)