

# 1. 環境構築

- APP 実行環境
- Owner key 作成
- Window 環境構築
- env.dabory 作成
- 外部 DB API と DB 接続
- コンテナ環境 (Docker)
- 開発/テスト環境構築
- Dummy(Dummy-Theme)環境構築
- Production(Production-Theme)環境構築
- Dabory Linux 環境 SSH 接続
- xampp 環境構築
- [window] WSL 環境構築

# APP 如何 实现

“

实现 SSO

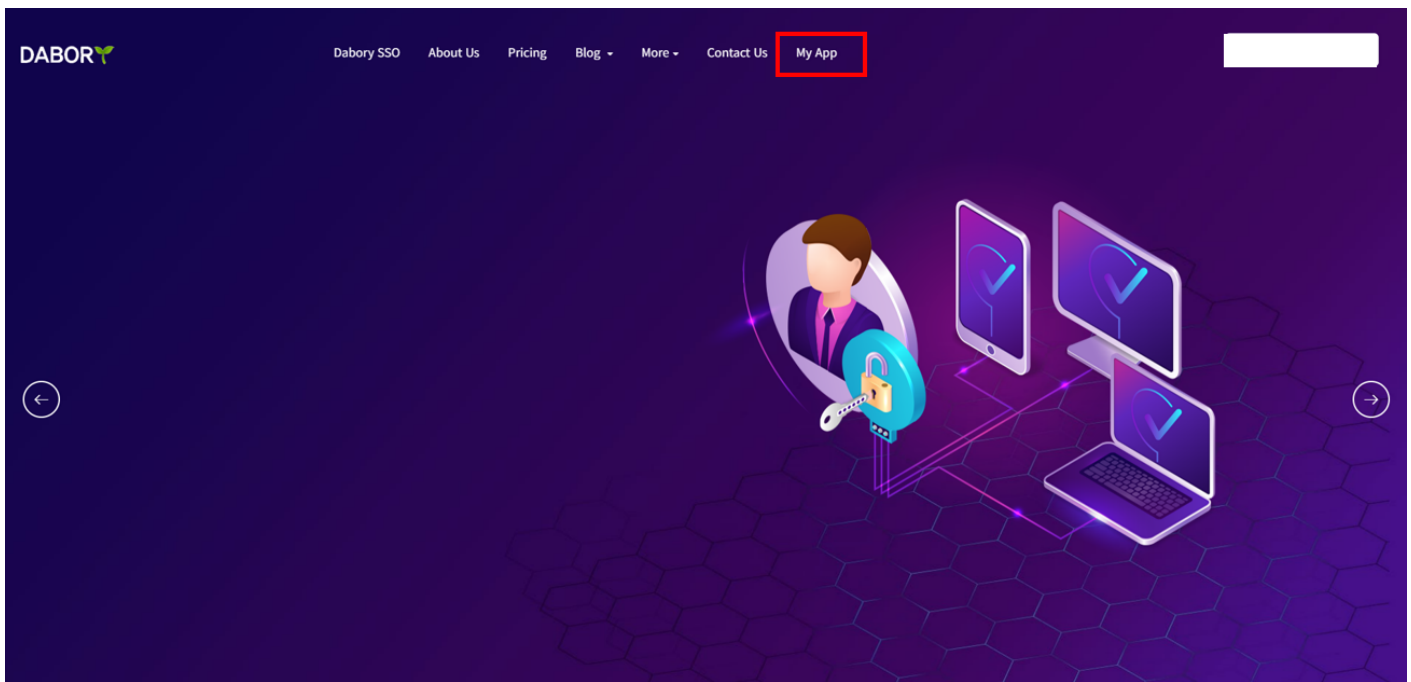
实现 SSO 需要 OAuth 2.0 协议。

“

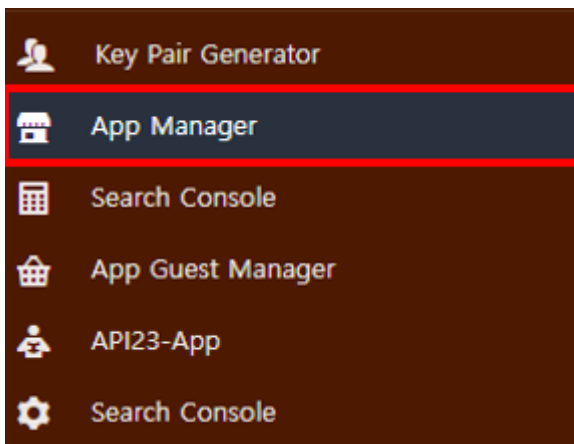
app 如何

实现 api 需要 app 实现。 app 实现 api 需要实现 (Authorization) 实现

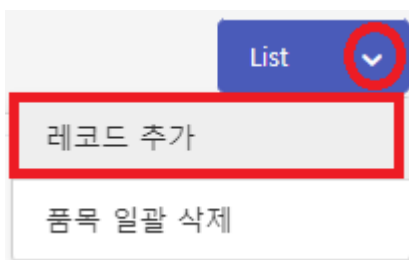
- 1. 实现 SSO 实现 app 实现。
- 2. 实现 My App 实现。



3. ☐☐ ☐☐☐ App Manager☐ ☐☐☐☐.



4. ☐☐ ☐☐ List ☐ ☐☐☐ ☐☐☐☐ -> ☐☐☐☐ ☐☐☐☐☐☐☐.



5. ☐☐ ☐☐☐☐☐ a~h☐☐ ☐☐☐☐☐☐☐.

SSO App input/Update/Delete

MainApp GuestApp

### App Info and Single-Sign-On

App Type\*  
Web App

App Name\*  
xerox\_obong\_final

Redirect URI\* ex) http://localhost:8080/dabory/ssologin/callback - must be https in real domain  
http://localhost:8006/social/member-dabory/callback

Client ID  
YyXgZgaOxX2-bdqdf

Client Secret  
vQMprb5jrrD9bbOiZ~

### API/DB connection Info - generate .env.dabory and Dabory Keypair

API Host:Port\* ex) http://123.124.125.126:18080 - you can update it after download.  
=Enter manually=

DB Host:Port\* ex) localhost:3306 or 13.124.2.254:3306 (do NOT ADD http:// )

DB User\* - DB connection info will be encrypted into BeforeBase64 Key in .env.dabory.

DB Name\* - must delete cache-key-pair folder in MAIN\_API after changing DB connection info.

DB Password\* - DB connection info can not be updated after download .env.dabory.

☐ change API/DB connection info - replace .env.dabory in public\_html folder after download it.

Key Pair\*  
opticalpos\_stage

## App info and Single-Sign-On

App Type : app type

App Name : app name

Redirect URI : app Redirect\_URI (account server callback)

Client ID, Client Secret : app key

## API/DB connection info - generate .env.dabory and Dabory Keypair

API Host : main api host

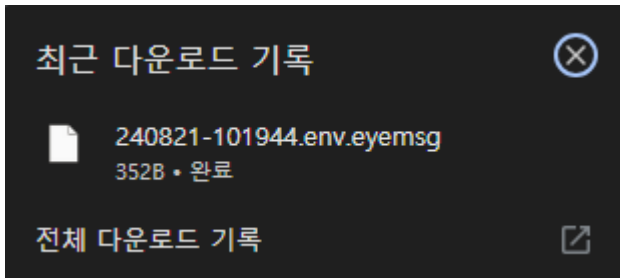
DB Host : db host

DB User, DB Name : `db` username `db name`

DB Password : `db server` password

Key Pair : key pair   .

6. save     .



7.      .

.

```
MAIN_API_URL=""
MAIN_API_CLIENT_ID=""
MAIN_API_CLIENT_SECRET=""
MAIN_API_BEFORE_BASE64=""
```

8. FRONTEND  config     .

\*     account url  url  (

ex : [https://accounts.dabory.com/o/oauth2/authorize?client\\_id=ysTHfKT4noL-](https://accounts.dabory.com/o/oauth2/authorize?client_id=ysTHfKT4noL-xajkbc&redirect_uri=https://visionnote.eyerecord.co.kr/wp-login.php&response_type=code&scope=all&state=tUhPBnu5RFOPcPm8IRBMayCSe5FKBRuG)

[xajkbc&redirect\\_uri=https://visionnote.eyerecord.co.kr/wp-](https://accounts.dabory.com/o/oauth2/authorize?client_id=ysTHfKT4noL-xajkbc&redirect_uri=https://visionnote.eyerecord.co.kr/wp-login.php&response_type=code&scope=all&state=tUhPBnu5RFOPcPm8IRBMayCSe5FKBRuG)

[login.php&response\\_type=code&scope=all&state=tUhPBnu5RFOPcPm8IRBMayCSe5FKBRuG](https://accounts.dabory.com/o/oauth2/authorize?client_id=ysTHfKT4noL-xajkbc&redirect_uri=https://visionnote.eyerecord.co.kr/wp-login.php&response_type=code&scope=all&state=tUhPBnu5RFOPcPm8IRBMayCSe5FKBRuG))

\* app  app\_manager  Redirect\_URI  callback

\*    MAIN\_API\_URL   dabory main\_api server  token  (token   )

\*   token    request header  .



# Owner key 1111

“

Owner key?

1111 main app 1 guest app 1111.

owner key 111 main app 1 guest app 1 owner key 11 11 api 111 1 1111.

“

Owner Key 11 11

main app : 11 A

guest app : 11 B

11 B 11 A 11 api 111 1 1111.

- 1. 11 B 11 1111 111 11 A 111111 11111.
- 2. 11 B 11 1111 1111 11 A 111111 11111. (111111 11 A 11 DB 11 111 11)
- 3. 11 A 11 11 11111 11 B 111111 11111. (111111 11 A 11 DB 11 111 11)

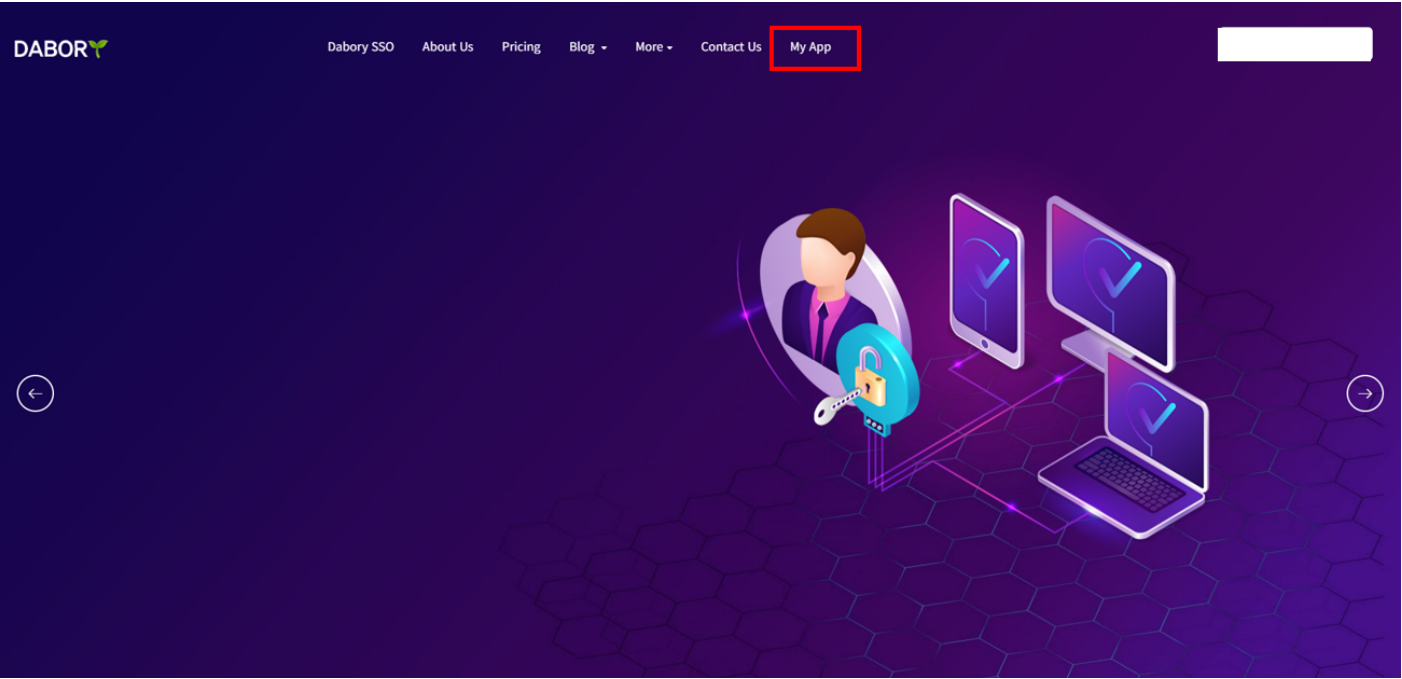
“

Owner Key 1111

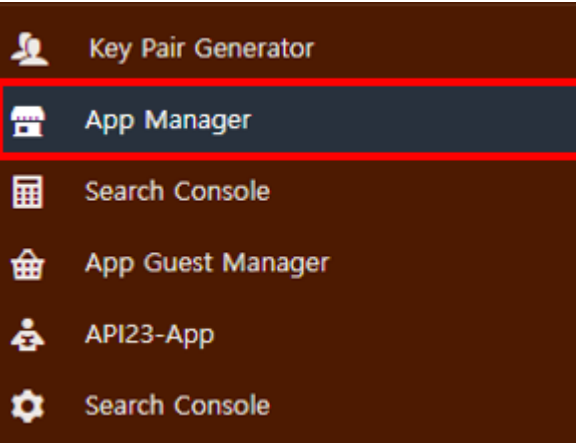
Owner key 1111 1111 guest app 11 1111 Owner key 11111 1111.

- 1. main app 1111 dabory 1111 1111111.

2. guest app 新建 新建 My App 新建 新建.



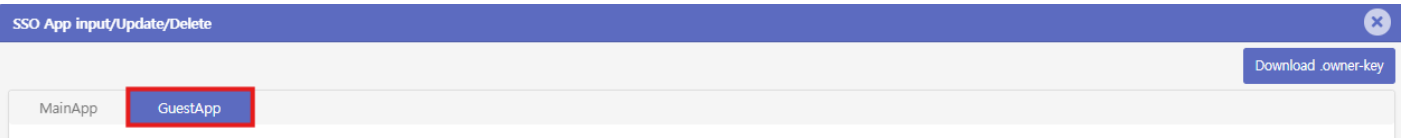
3. 新建 新建 App Manager 新建 新建.



4. 新建 app 新建 Owner key 新建 app 新建.

<input type="checkbox"/>	No	App Name*	App Type	Client Id	Redirect URI	Created	Updated
<input type="checkbox"/>	1	api_test			https://www.daborysso.com	24-04-11 16:42:18	70-01-01 09:00:00

5. GuestApp 新建 新建.



6. a 新建 c 新建 新建 新建.



## Generating GuestApp Ownery Key

Guest App Name\* (ex:credit-strong-api-app)

OwnerCode\*

AppType\* (web, js, native-app)

□ Guest App Name : Owner Key□ □□

□ OwnerCode : Owner Key□ □□□□

- owner : □□ api □□
- keypair :
- dbupdate :
- limited : □□ api□ □□

□ AppType : guest app□ type

7. Download .owner-key □□□ □□□□ Owner key□ download □□□.

8. MainApp □□ □□□ save □□□□□ □□□□□.

9. □□□ .owner-key□ □□□□ frontend □□□□□ open□□□□.

10. .env □□□ MAIN\_API\_OWENER\_KEY □□□ □□□□ □□□ owner key□ □□ □□□□□.

# Window 配置

本文介绍了在 Windows 上配置 Git 的方法。

“

## 1. 配置

### 1. 配置

Windows 上配置 Git 时，`core.autocrlf` 默认为 `true`。

当 `core.autocrlf` 为 `true` 时，Git 会将 LF 转换为 CRLF。

检查 `core.autocrlf` 配置：

```
git config --get core.autocrlf
```

### 2. 配置

将 `core.autocrlf` 设置为 `true`，Git 会将 LF 转换为 CRLF。

```
git config --global core.autocrlf input
```

在 Windows 上，Git 会将 LF 转换为 CRLF。

### 3. Push 配置

Push 时，Git 会将 LF 转换为 CRLF。

在 Windows 上，Git 会将 LF 转换为 CRLF。

当 `core.autocrlf` 为 `true` 时，Git 会将 LF 转换为 CRLF。

**git config --get core.autocrlf**

在 Windows 上，Git 会将 LF 转换为 CRLF。

## git config --global core.autocrlf input

在 git server 上 push 时 LF -> CRLF 的转换是自动的。

push 时 本地文件 的 LF 会被自动转换为 CRLF 再 push 到服务器。

# env.dabory

“  
**env.dabory**

env.dabory는 Laravel의 env 파일을 API 서버 db 서버에 배포할 수 있도록  
제공함.

## 2. db 서버

Frontend는 env.dabory의 api url, API 서버의 GateToken, API 서버의  
GateToken은 ClientId, ClientSecret, BeforeBase64, SSO, GateToken, DB  
GateToken은 API 서버에 배포함.  
GateToken은 main app, guest app 서버에 배포함 API 서버에 배포함.

## 3. env.dabory

MAIN\_API\_URL : API 서버 URL  
MAIN\_API\_CLIENT\_ID : app의 API ID.  
MAIN\_API\_CLIENT\_SECRET : app의 API secret.  
MAIN\_API\_BEFORE\_BASE64 : app의 API Base64

# DB API DB

2 Restfull api 1

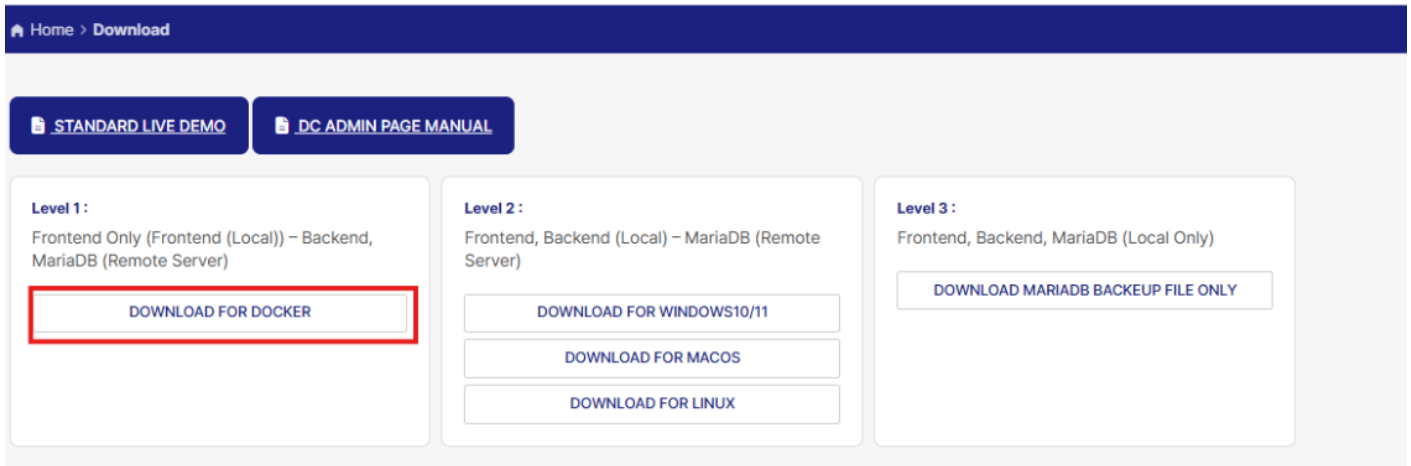
2 1 clientId beforebase64 json type API api response ga

gate token .env.dabory db .env.dabory App Manger

# 部署 DC (Docker)

部署 DC 的步骤:

1. DC 部署包 (Docker) `dc-demo-frontend-docker.tar.gz` 下载。



2. 将 `dc-demo-frontend-docker.tar.gz` 下载到 DC 的 `/path/to/install` 目录, 并执行以下命令。

```
mv ~/Downloads/dc-demo-frontend-docker.tar.gz /path/to/install
```

3. 使用 `tar` 命令解压:

```
cd /path/to/install
tar -xzf dc-demo-frontend-docker.tar.gz
```

进入 `demo` 目录。

```
demo/php-docker/conf.d/
demo/php-docker/conf.d/docker-php-ext-gettext.ini
demo/php-docker/conf.d/docker-php-ext-grpc.ini
demo/php-docker/conf.d/docker-php-ext-mysqli.ini
demo/php-docker/conf.d/docker-php-ext-pdo_mysql.ini
demo/php-docker/conf.d/docker-php-ext-sodium.ini
demo/php-docker/cron.d/
demo/php-docker/cron.d/laravel-cron
demo/php-docker/Dockerfile
demo/php-docker/Dockerfile--d
demo/php-docker/Dockerfile--v8.0
demo/php-docker/lib/
demo/php-docker/lib/dstkt.so
demo/php-docker/php.ini
demo/php-docker/php.ini-pma-upload
demo/php-docker/php.ini-pma-upload--d
demo/php-docker/php.ini-pma-upload-v8.0

HP 15@DESKTOP-5472AJG MINGW64 /c/workspace/baekSan/front_task/dc-demo
$ ls
dc-demo-frontend-docker.tar.gz demo/
```

4. demo 폴더로 이동:

```
cd demo
```

5. demo 폴더에 db, docker-compose, php-docker 폴더 생성:

**db** : DC에서 ERP를 실행할 폴더. ERP를 실행할 때 필요한 폴더.

**docker-compose.yml** : DC에서 Docker를 실행할 폴더. 이 폴더에 docker-compose.yml, docker-compose.yml, docker-compose.yml 폴더를 생성.

**php-docker** : PHP를 실행할 Dockerfile을 실행할 폴더. 이 폴더에 PHP를 실행할 Dockerfile을 실행.

```
HP 15@DESKTOP-5472AJG MINGW64 /c/workspace/baekSan/front_task/dc-demo/demo
$ ls
db/ docker-compose.yml php-docker/
```

5. docker-compose.yml 파일을 Docker를 실행할 DC에서 실행할 폴더. 이 폴더에 docker-compose.yml 파일을 생성:

```
docker-compose up -d
```

6. Docker에서 docker-compose.yml 파일을 실행할 때 실행할 폴더. 이 폴더에 docker-compose.yml 파일을 실행:

- **Pro** (Pro)
  - http://localhost:8028
- **db**
  - http://localhost:8028/dabory/erp

1. DC 电压源 10V，电阻 10Ω，电容 10μF，电感 10mH，求稳态响应。



□□□/□□□□□□ □□□□ □□

“

□□□ □□□□□

0. dabory git server □ □□□ git server □□□

가입하기

사용자명 \*

이메일 \*

비밀번호 \*

재입력 \*



보안 문자 \*

새 계정 생성

이미 계정을 가지고 계신가요? 로그인하세요!

- 1. .env □□
- 2. .env.dabory □□
- 3. php□ mariadb□ □□


□□□□

- □□□ □□□ □□

- “
- /□□□□□□ □□ □□□□

- eyemsg.daboryhostcom
- kloxoSCRIPT
- public\_html

4. `git mybin`



- 
dabory / mybin

Watch 2
Star 0
포크 0

파일
이슈 0
풀 리퀘스트 0
위키

설명 없음

406 커밋
1 브랜치
0 릴리즈

브랜치: master
mybin
HTTP SSH http://git.daboryhost.com:



-

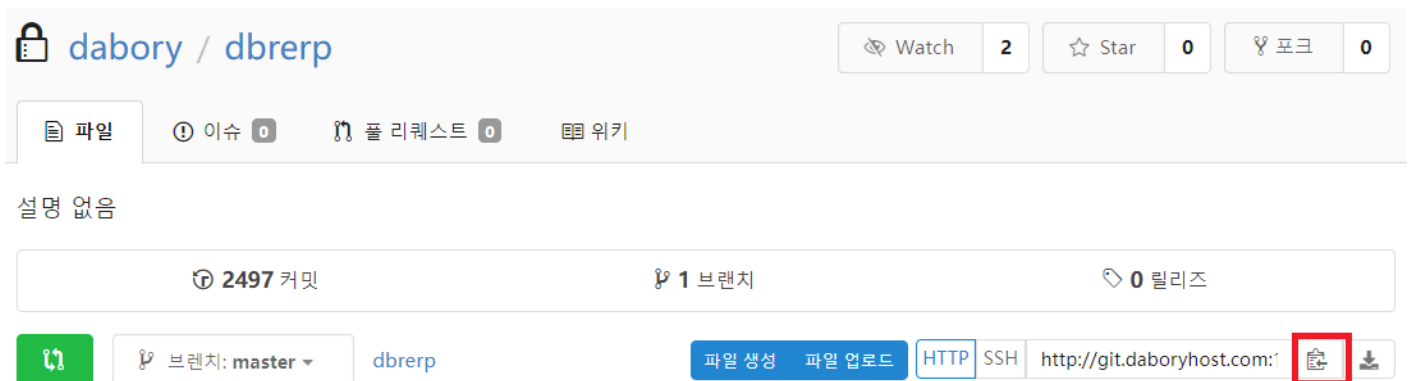
```
[eyemsg@:1 eyemsg]$ ls
backups  eyemsg.daboryhost.com  kloxoscript  mybin  public_html
```

5. `eyemsg.daboryhost.com` 폴더 생성.

```
cd eyemsg.daboryhost.com
```

6. `dbrrerp` 폴더 생성.

7. HTTP url 복사.



8. `git clone` 명령어 사용.

```
git clone http://git.daboryhost.com:10880/dabory/dbrrerp.git
```

9. `dbrrerp` 폴더를 `eyemsg.daboryhost.com` 폴더로 이동.

```
mv dbrrerp/* .
```

10. `.env` 파일을 `.env.dabory` 폴더로 이동.

`.env` 파일을 `.env.dabory` 폴더로 이동.

```
vi .env
vi .env.dabory
```

11. dabory [ ] [ ].

```
cd dabory
```

12. [ ] [ ] theme [ ] [ ] 13[ ] [ ] [ ].

[ ] [ ]

13. [ ] git [ ] vendor [ ] [ ].

14. git clone [ ] [ ] vendor [ ] [ ].

```
git cloen http://git.daboryhost.com:10880/dabory/vendor.git
```

15. storage [ ] Update [ ]


 /home/eyemsg/eyemsg.daboryhost.com/

InvalidArgumentException  
Please provide a valid cache path.  
<https://eyemsg.daboryhost.com/>

16. [eyemsg.daboryhost.com](https://eyemsg.daboryhost.com/)[ ] [ ] [ ] [ ] [ ] [ ].

백산광학 문자중전 서비스

테스트탑


 logo


Search Anything...

Q

Login


Join


 0

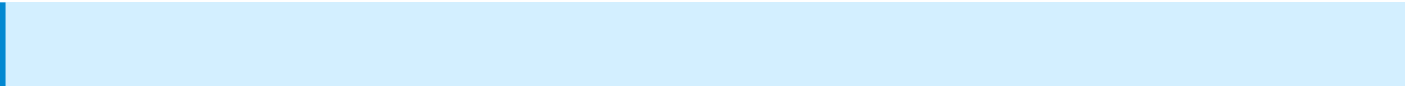
 0

문자중전

고객센터

 Call Us  
(02)871-4134

 Email Us  
kimhi65@gmail.com





- |  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

--	--

--	--	--	--
- |  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

# 安装(Dummy-Theme)的步骤

1. 在 .env 文件中设置 DBR\_THEME 为 erponly。erponly 是 /pro 目录下的 user page 的 redirecti  
/dabory/themes 目录下的 [erponly](#) 主题。安装步骤如下：

#####(Production-Theme)#####

.#####(**Production-Theme**)#####

.env##### ERP\_THEME, PRO\_THEME #####. #####. /themes/pro ##### (

# Dabory Linux SSH로 서버 접속하기

“

서버

1. .ussh (ssh로 서버 접속하기 위해 .ussh 파일을 생성한다.)

2. dbr\_list.txt (서버 목록을 저장한다.)

2-1. 서버 목록을 저장한다.

```
#서버명          #IP          #비밀번호
dbr02=13.124.2.254    #erics-and-kloxo
```

\* **dbr02** 서버 접속. 서버 접속을 위해 dbr02 서버를 선택한다.

2-2. ussh (서버 접속) dbr\_list.txt (서버 목록) 2-1에 저장된 서버 목록을 사용하여 서버에 접속한다.

“

ussh로 서버 접속

1. ussh (서버 접속) dbr\_list.txt (서버 목록)

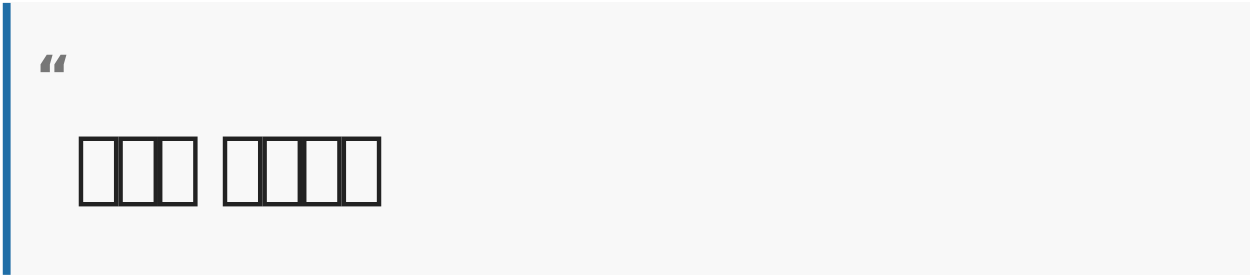
2. 서버 접속을 위해 dbr02 서버를 선택한다.

```
./ussh dbr02 eyemsg
```

```
# ./ussh (서버 접속)
```



# xampp




1. [xampp](#) [xampp](#) [xampp](#).

2. More Downloads

[Apache Friends](#) [Download](#) [Hosting](#) [Community](#) [About](#)  [Search](#) [EN](#)

## Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using [InstallBuilder](#).

 **XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12**

Version	Checksum	Size
8.0.30 / PHP 8.0.30 <a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 144 Mb
8.1.25 / PHP 8.1.25 <a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 148 Mb
8.2.12 / PHP 8.2.12 <a href="#">What's Included?</a>	<a href="#">md5</a> <a href="#">sha1</a>	<a href="#">Download (64 bit)</a> 149 Mb

[Requirements](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

### Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the [Forums](#) or [Stack Overflow](#).

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)

3. [xampp](#) OS [xampp](#) [xampp](#). [xampp](#) window [xampp](#).

An easy to install Apache distribution containing MySQL, PHP, and Perl  
Brought to you by: [beltranrueda](#), [bitnami](#), [koswalds](#), [kvogelgesang](#)

An easy to install Apache distribution containing MySQL, PHP, and Perl  
Brought to you by: [beltranrueda](#), [bitnami](#), [koswalds](#), [kvogelgesang](#)

Code

**xampp-windows-x64-8.2.12-0-VS16-installer.exe (157.6 MB)**

Downloads / Week 19,562 

391,847 

21,442 7 35 ☐5 ☐5 ☐

4. 7.3 ~8.0 php 64bit 64bit 64bit. php 7.3.33 64bit 64bit.



An easy to install Apache distribution containing MySQL, PHP, and Perl  
Brought to you by: [beltranrueda](#), [bitnami](#), [koswalds](#), [kvogelgesang](#)

Summary

Files

Reviews

Support

Wiki

Code

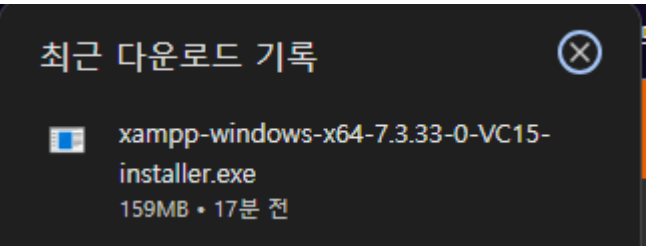
**Download Latest Version**  
xampp-windows-x64-8.2.12-0-VS16-installer.exe (157.6 MB)

Get Updates

Home / XAMPP Windows / 7.3.33

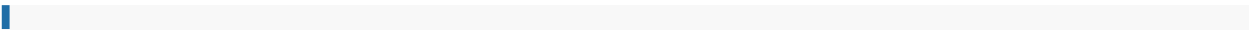
Name	Modified	Size	Downloads / Week
Parent folder			
<a href="#">xampp-portable-windows-x64-7.3.33-0-VC15.7z</a>	2021-11-25	90.2 MB	24
<a href="#">xampp-portable-windows-x64-7.3.33-0-VC15.zip</a>	2021-11-25	202.1 MB	38
<a href="#">xampp-portable-windows-x64-7.3.33-0-VC15-installer.exe</a>	2021-11-25	145.7 MB	50
<a href="#">xampp-windows-x64-7.3.33-0-VC15.7z</a>	2021-11-25	110.5 MB	7
<a href="#">xampp-windows-x64-7.3.33-0-VC15.zip</a>	2021-11-25	231.0 MB	20
<a href="#">xampp-windows-x64-7.3.33-0-VC15-installer.exe</a>	2021-11-25	166.7 MB	243
Totals: 6 Items		946.2 MB	382

6. 다운로드 받은 파일을 설치합니다.



7. env/env.dabory 파일을 실행합니다

- env.dabory 파일을 실행합니다
- .env 파일을 실행합니다

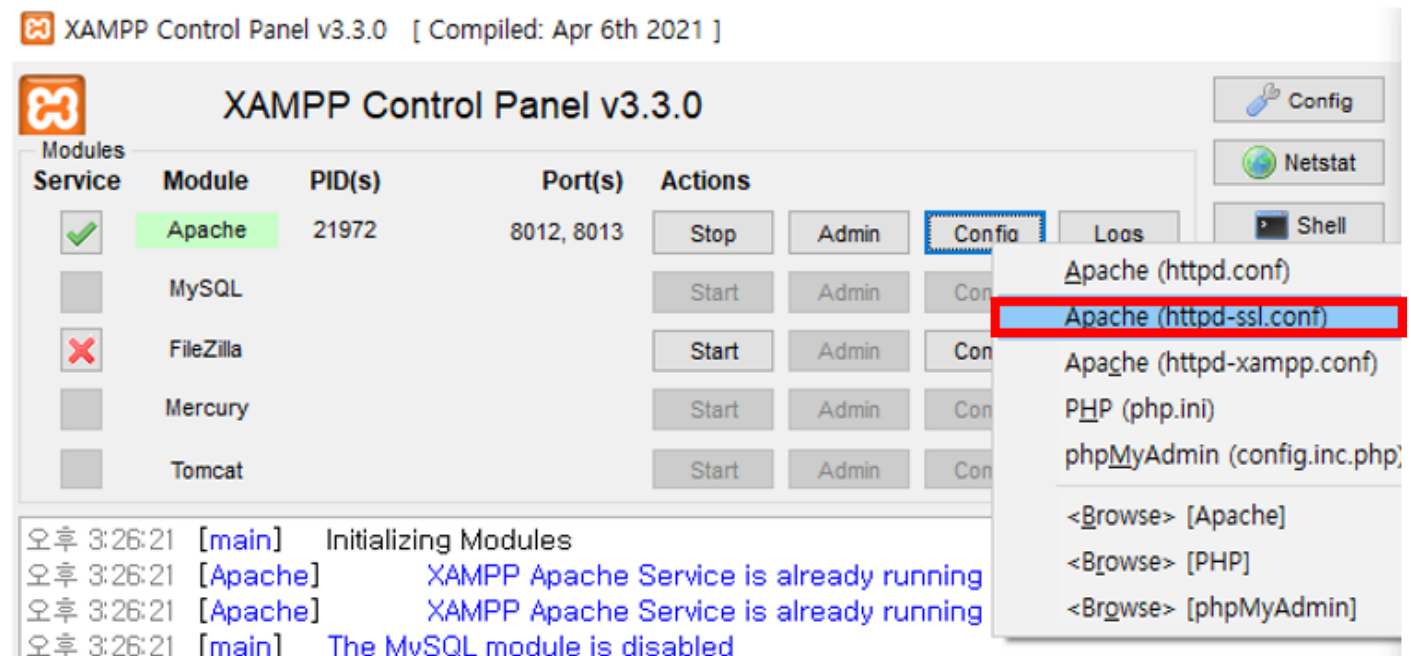


# xampp

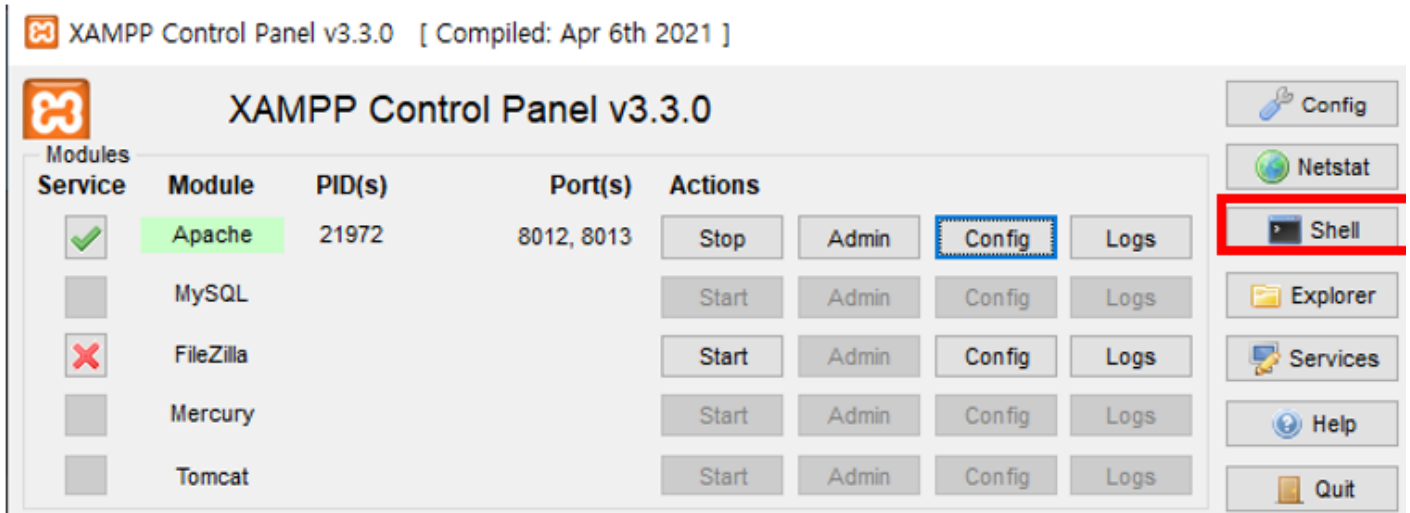
window

1. XAMPP Control Panel

2. xampp 80 / 443 httpd.conf / httpd-ssl.conf / Config







5.

5. 在终端中运行以下命令：

```
cdhtdocs
gitclonehttp://git.daboryhost.com:10880/dabory/db ERP
```

6. 在终端中运行以下命令：

```
# vendor
cd ERP
gitclonehttp://git.daboryhost.com:10880/dabory/vendor.git

# erponly
cd dabory
mkdir themes
gitclonehttp://git.daboryhost.com:10880/db ERP-themes/erponly.git
```

7. 在终端中运行以下命令：

8. 在终端中运行以下命令：

- env 在 env.dabory 中
- vendor 在 storage 中
- xampp apache 在 public 中
- 在终端中运行以下命令：

在终端中运行以下命令：





# [window] WSL 安装教程

“

安装

- wsl 安装 git 安装 教程。

“

git clone 安装

- WSL 在 Windows 上运行，Windows 上 Linux 的安装教程。
- [git clone] 安装教程，在 Windows 上安装 Linux 教程。

wsl 安装 git clone 安装 git push 安装 教程 issue 安装。

安装 教程 安装 教程 安装 教程。

1. wsl 安装 教程 wsl.conf 安装 教程。

```
sudo nano /etc/wsl.conf
```

2. 安装 教程。

```
[automount]
```

```
options = "metadata"
```

3. 安装 wsl 安装 restart 安装。

```
wsl --shutdown
```

4. git clone 的 命令 如下。

5. 命令 如下。

```
ls -l
```

在 wsl 中 运行 git clone 或 git push 时 可能会 遇到 permission 777 的问题。

git clone 或 运行 命令 时 遇到 777 问题 时 可 使用 git clone / push 命令 解决。