

## DB 接続 流れ

### 1 .env の準備

#### 例

DB\_CONNECTION=mysql・・・接続先の sql の指定。デフォルトだと mysql になっている。

DB\_HOST=127.0.0.1・・・db のホスト番号を指定。開発環境でやる場合は自分の pc が元になるので 127.0.0.1 を指定。

DB\_PORT=8889・・・接続したい DB のポート番号を指定。

DB\_DATABASE=laravel\_sample・・・接続したい DB 名を指定。

DB\_USERNAME=root・・・接続先のユーザーネームを指定。何も設定していない場合 root にする。

DB\_PASSWORD=root・・・接続先のパスワードを指定。何も設定していない場合 root にする。

\*本番環境に移す場合は APP\_DEBUG=true を false に変更する。

### 2 config/database で設定する。

-1 //デフォルトの sql の指定。初期設定は mysql が指定されている。

```
'default' => env('DB_CONNECTION', 'mysql'),
```

-2 //接続先の sql の詳細設定

```
'connections' => [  
    'mysql' => [  
        'driver' => 'mysql',  
        'url' => env('DATABASE_URL'),  
        'host' => env('DB_HOST', '127.0.0.1'),・・・ここを変更(開発環境の場合は  
127.0.01 で ok)  
        'port' => env('DB_PORT', '3306'),・・・ここを変更(Mysql の場合は基本  
3306 で OK,MAMP を経由する場合は 8805 以降になっている事が多い)  
        'database' => env('DB_DATABASE', 'forge'),  
        'username' => env('DB_USERNAME', 'root'),  
        'password' => env('DB_PASSWORD', 'root'),・・・ここを変更(特に何もし  
てないなら root で ok。SHOW VARIABLES LIKE '%sock%';)  
        'unix_socket' => env('DB_SOCKET',  
'/Applications/MAMP/tmp/mysql/mysql.sock'),・・・ここを変更(DBSOCKET の登  
録)  
        'charset' => 'utf8mb4',  
        'collation' => 'utf8mb4_unicode_ci',  
        'prefix' => '',  
        'prefix_indexes' => true,  
        'strict' => true,  
        'engine' => null,
```

```
'options' => extension_loaded('pdo_mysql') ? array_filter([
    PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
]) : [],
],
```

http://localhost:8000/

```
cd /Applications/MAMP/htdocs/laravel
cd /Applications/MAMP/htdocs/typeingPractice_app
```

mysql command 関係

(mamp 使っている時限定の sql 文使用)  
cd /Applications/MAMP/Library/bin/

mysql を使う為のコマンド  
後ろのオプションはログイン時の入力を省く為につけている。  
-u . . . userpass  
-p . . . password

```
./mysql -u root -p;
```

(ログイン時の pass は特に変更していない為 root で ok)

初期のみ  
(database 作成)  
CREATE DATABASE データベース名;

できたか確認  
(DATABASE 一覧表示)  
SHOW DATABASES;

扱う DB の選択  
use (DB 名);

テーブル一覧を表示  
show tables;

table の情報(カラムなど)の詳細表示  
describe (対象 table 名);

(db 削除)  
drop database データベース名;

(dbsocket 表示コマンド)  
SHOW VARIABLES LIKE '%sock%';

DB から出る。  
quit;

sql の条件文は` (バッククォート) で囲む。

```
composer create-project laravel/laravel=5.8.*"  
sample_laravel --prefer-dist
```

ポートの確認

デフォルトでは MySQL が TCP ポート 3306 番で Listen しているので、ポートが開いているか確認。

```
$ netstat -tln
```

0.0.0.0:3306 ~ LISTEN xxxx/mysqld の表示があれば OK。

マイグレーションコマンドを実行した場合に出てくるやつについて。

```
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table (0.06 seconds)
```

~ing まで進んだ場合は migrationFile を実行するまでに関係のある部分(routing 関係のファイルなど)に問題がない。

Migrating に進む前に他ファイルのバリテーションが入っている。

Migrated はマイグレーションファイルを元にテーブルが作成された事を伝えている。

Migrating まで進んでエラーが出ていたらマイグレーションファイルに問題がある。

validation 関係はコントローラーで管理する。

errormessage 等はデフォルトだと英語なので日本語化の設定もする。