

1.Django-mysqlのインストール

DjangoでMySQL用のJSONfieldを使えるようにしてくれるモジュール、[Django-mysql](#)をインストールします。

```
1. pip install django-mysql
```

2.モデル追加

自分のモデル定義にJSONfieldを追加。modelの継承元もdjango_mysqlのものにする。

```
1. from django_mysql.models import JSONField, Model
2.
3. class ShopItem(Model):
4.     name = models.CharField(max_length=200)
5.     attrs = JSONField()
6.
7.     def __str__(self):
8.         return self.name
```

3.settings.pyの編集

settings.pyのINSTALLED_APPSに以下を追加する。

(ちなみに、自分の環境では何故かこの設定を行わなくてもJSON型のカラムが生成されてしまったが、[公式ドキュメントにある通り](#)きちんと指定した方が無用なトラブルを避けられると思う。)

```
1. INSTALLED_APPS = (  
2.     ...  
3.     'django_mysql',  
4. )
```

公式ドキュメントによると以上で作業は完了なのだが、自分の環境では

```
1. ?:(django_mysql.W003) The character set is not utf8mb4 for database connection  
   'default'  
2.     HINT: The default 'utf8' character set does not include support for all Unicode  
   characters. It's strongly recommended you move to use 'utf8mb4'. See: https://django-  
   mysql.readthedocs.io/en/latest/checks.html#django-mysql-w003-utf8mb4
```

というエラーが出てしまった。これを解決するため、settings.pyのDB接続情報に'OPTIONS'項目を追加し、utf8mb4という文字コードを指定する。

```
1. DATABASES = {
2.     'default': {
3.         'ENGINE': 'django.db.backends.mysql',
4.         'USER': 'xxxxx',
5.         'PASSWORD': 'xxxxx',
6.         'HOST': 'localhost',
7.         'OPTIONS': {
8.             'charset': 'utf8mb4', # <--- Use this
9.         }
10.    }
11. }
```

utf8mb4とは何なのか

通常のutf-8は1~4バイトで1文字を表しますが、何故かMySQLにおいてutf-8と名付けられているcharsetは1~3バイトまでしか対応していないため、普段あまり使われないような漢字や絵文字が正しく登録されません。

この問題に対応できるMySQLでのcharsetがutf8mb4で、1~4バイトまで対応しているため、基本的にはこちらを使うのが良いようです。

最初、自分の環境では特にデフォルトのcharsetを指定していなかったのですが、その場合はcharsetがutf8mb4になり、今回のエラーに繋がったようです。

デフォルトの文字コードを指定する

やはり文字コードは明示的に指定した方が良いので、MySQLの設定ファイルに設定を追加します。自身の環境のmy.cnfに以下を追記します。[引用](#)

1. `[mysqld]`
2. `...`
3. `character-set-server=utf8mb4`
- 4.
5. `[client]`
6. `default-character-set=utf8mb4`

自身の環境のmy.cnfの場所が不明な場合、

1. `mysql --help | grep my.cnf`

で調べられます。[引用](#)

今回は以上です。

関連する記事

