

INIAD CS Essentials 2

11-1: ViewとModelの連携





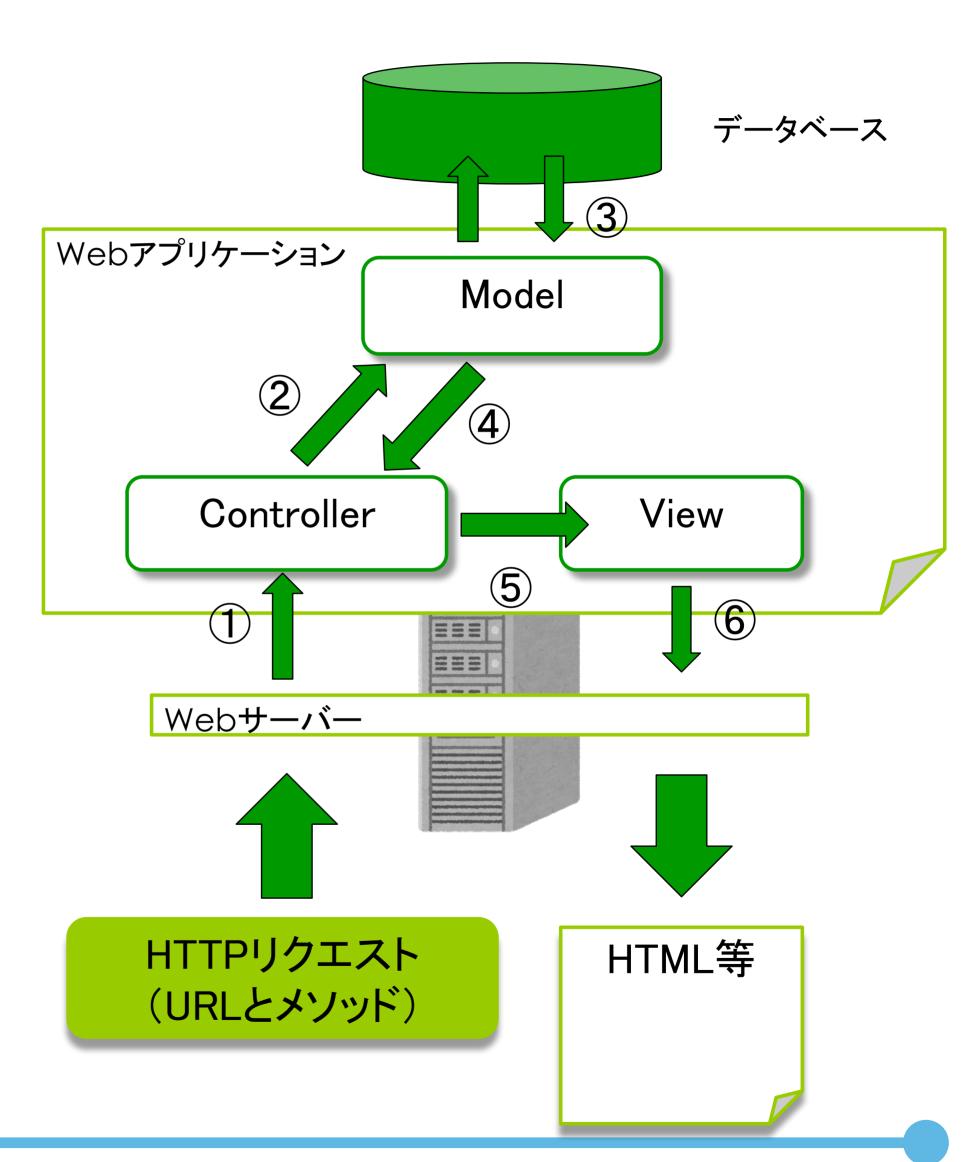
View と Modelの連携

今回の講義までで、DjangoのMVCの一連の流れを学習します。



これまでの復習

- 今回の講義までで、DjangoのMVCの一連の 流れを学びます
 - 10-1: Controller, Viewとその連携
 - 1, 2, 5, 6
 - 10-2 : Model
 - 3
 - 11-1: ViewとModelの連携
 - 4
- DjangoのMVCは、以下のように対応していると見なすことができる
 - Model → blog/models.py
 - Controller → blog/views.py
 - View → blog/templates/blog/***.html



INIAD

Djangoプログラムの動作順序

- Djangoプログラムは、主に以下の順序で動作します。
- URLディスパッチャでリクエストを受け付け、対応したメソッド(これが Controller となる) を呼び出す
 ⇒ 今回の例では blog/urls.py
- 2. Controller内で、リクエストを処理する
 → 今回の例では blog/views.py
- 3. Controller内で、Modelクラスを通じてデータベースにアクセス
 → 今回の例では blog/models.pyのArticle class
- 4. Controller内で、テンプレートエンジンを呼び出しHTMLを生成
 → 今回の例では blog/templates/blog/***.html



オブジェクト関係マッパ (ORマッパ)

- データベースのレコードを、プログラミング言語(今回の場合はPython)のオブジェクトとして扱う仕組み
- 通常関係データベースのアクセスを行うためには、SQLという言語を利用する必要があります
 - SQLは、CS概論2の最後に学習します。
- オブジェクト関係マッパを使うと、SQLを書かずに Pythonでの変数の代入やメソッド呼び出しにより、データベースを操作することができる

INIAD

Djangoのオブジェクト関係マッパ

- blog/models.pyに記述したArticleのような「Modelクラス」を定義することで、データベースのレコードを、Pythonオブジェクトとして扱える
- 例えば、以下のような事ができる
 - データベースの検索結果を、Pythonオブジェクトのリストとして取得する
 - Pythonオブジェクトを生成し、保存メソッドを呼ぶ → データベースにレコードを登録する
 - Pythonオブジェクトの変数に値を代入し、保存メソッドを呼ぶ → データベースのレコードを編集する
 - Pythonオブジェクトの削除メソッドを呼ぶ → データベースのレコードを削除する



まだ完成していないところは?

- ・現段階では以下を学習しました
 - ✔ URLディスパッチャを用いて、urls.py の定義に応じて、views.py の適切な関数を呼び出す手順
 - ✓ テンプレートエンジンを用いて、views.py から渡された情報をテンプ レートに当てはめて出力する手順
 - ✓ オブジェクト関係マッパを用いて、 models.py の定義に応じて、データ ベースからデータを取り出す手順
- あとは、views.py が適切にデータベースと連携をするようになれば、アプリケーションの機能が完成します

ここから先の進め方



- 前回まででオブジェクト関係マッパを用いて、Pythonの プログラムからデータベースを操作することができるよう になりました。
- Viewとオブジェクト関係マッパを連携させましょう
 - レコードの検索
 - レコードの作成
 - レコードの編集
 - レコードの削除



遷移を整理すると…

