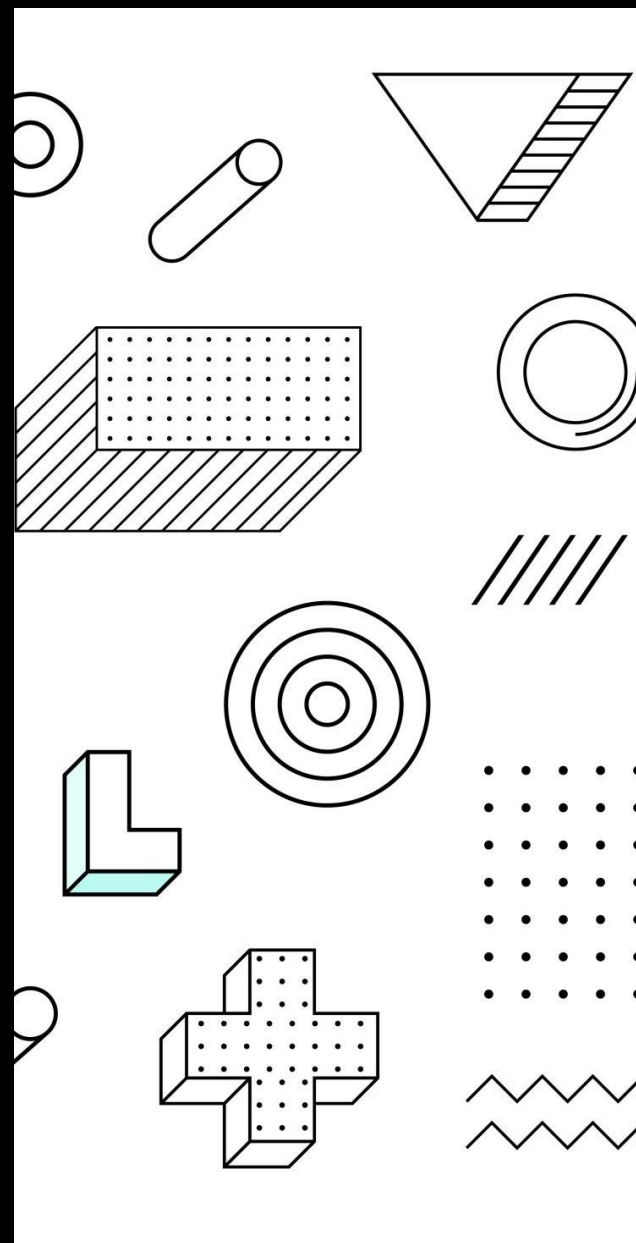


Django framework

Antoni Lasoń
Wojciech Dec





Czym jest framework

W tłumaczeniu z angielskiego:
struktura, szkielet

Platforma programistyczna

Dostarcza komponentów i
bibliotek

Nie jest biblioteką

Zalety i wady stosowania frameworków

Zalety

Efektywność (pisanego kodu)

Poprawa jakości kodu

Niezawodność

Wady

Złożoność

Wydajność

Co to jest Django framework

- „Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design”

Cytat ze strony Django

Krótką historia

- Stworzony w 2003 roku
- Miał umożliwić szybkie tworzenie aplikacji internetowych
- W lipcu 2005 roku opublikowany na licencji BSD
- Tworzony przez programistów związanych z Lawrence Journal-World
- Twórcy: Adrian Holovaty, Simon Willison i inni

Najważniejsze cechy



Szybkość
implementacji



Bezpieczeństwo



Skalowalność



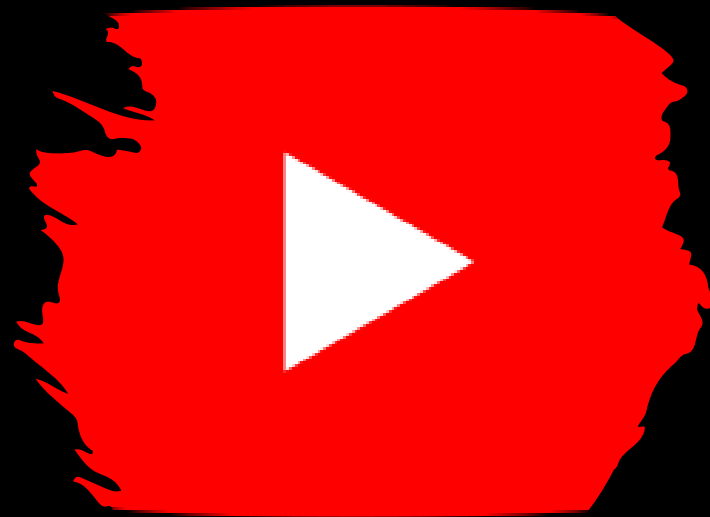
Wszechstronność

Zastosowania

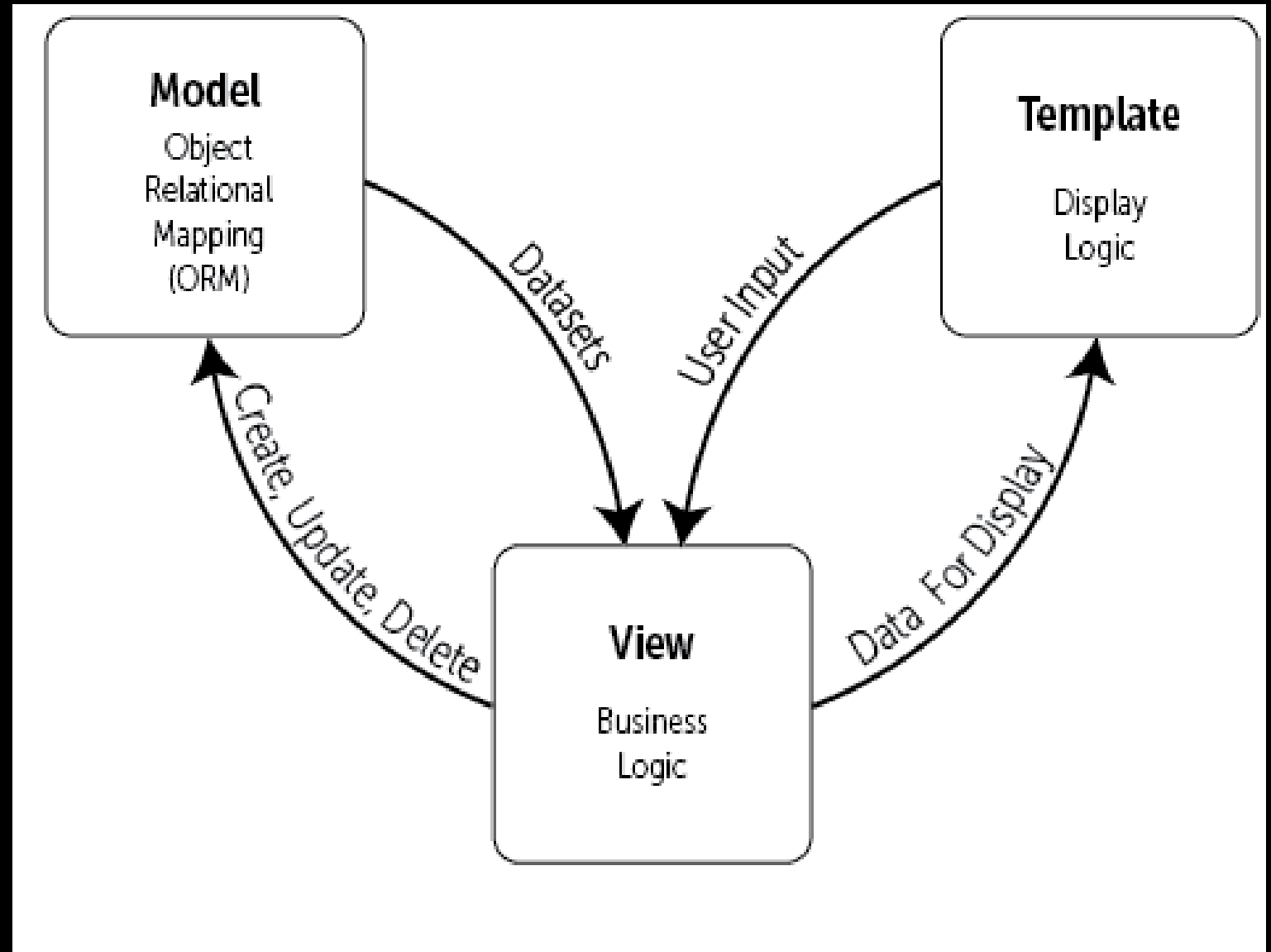
- Trasowanie adresów URL
- Pobieranie danych z baz danych
- Renderowanie szablonów
- Obsługa danych wejściowych od użytkowników, uwierzytelnianie

Kto i dlaczego z tego korzysta

- Szybkie prototypowanie
- Skalowalność
- Bezpieczeństwo
- Aktywna społeczność



Architektura Frameworka Django: Model-View- Template (MVT)



Modele i migracje

- Migracja - przekształcanie kodu Pythona w struktury bazodanowe
- ORM – Object Relational Mapper
- Model – klasa Pythona, zawiera szkielec, na podstawie którego można utworzyć tabelę w bazie danych

Widoki

- Są to funkcje, która pobierają żądanie w postaci obiektu Pythona (HTTP Request platformy Django)
 - Wykonują zapytania do bazy danych
 - Widok może wyrenderować szablon
-
- ```
def result(request):
 search = request.GET.get("search")
 return render(request, "search.html",
 {"search": search})
```

# Szablony

- Pliki HTML
- `<i>`
- Funkcja **render**, argumenty:

```
def index(request):
 name = request.GET.get("name") or
 "świecie"
 return render(request, "base.html",
 {"name": name})
```

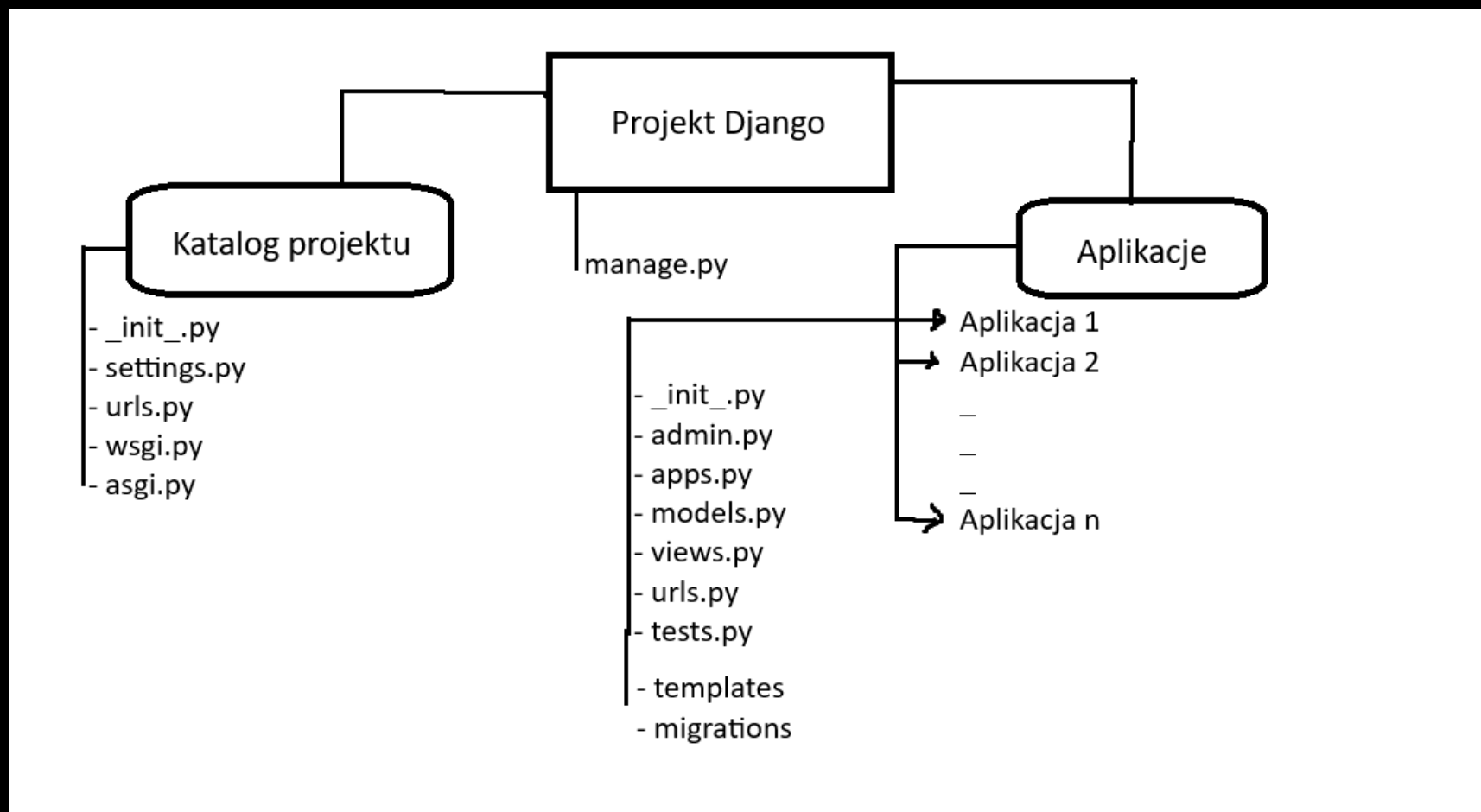
- Przekazywanie **kontekstu** renderowania przez widok

# Mapowanie adresów URL

- Powiązanie widoku i URL

```
urlpatterns = [
 path('admin/', admin.site.urls),
 path('', reviews.views.index),
]
```

# Struktura projektu Django



# Atrybuty HTTP Request Django



method



GET



POST



headers



path

# QueryDict

```
request.GET = QueryDict("val1=a&val2=b&val2=c&val3")
```

```
request.GET["val1"]
```

```
request.GET["val4"]
```

-



# DTL

---

- Django Template Language
- Narzędzie do dynamicznego generowania treści aplikacji webowej
- Skupienie na wyglądzie, nie logice

```
{% extends "base_generic.html" %}

{% block title %}{{ section.title }}{% endblock %}

{% block content %}

 <h1>{{ section.title }}</h1>

 {% for story in story_list %}

 <h2>

 {{ story.headline|upper }}

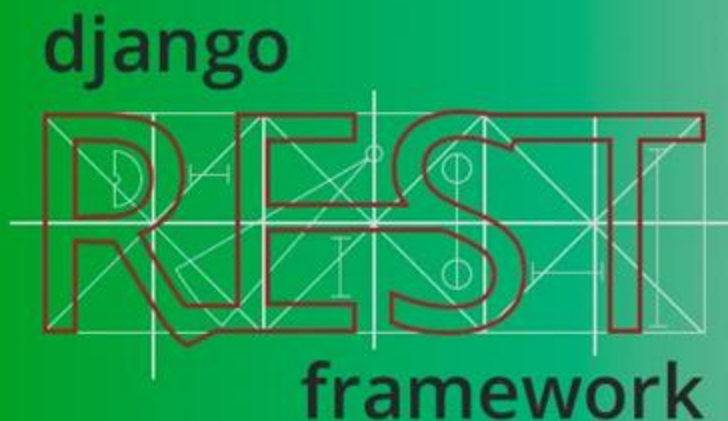
 </h2>

 <p>{{ story.tease|truncatewords:"100" }}</p>

 {% endfor %}

{% endblock %}
```

# Django REST framework



- Narzędzie do tworzenia API
- Dostarcza mechanizmów Uwierzytelniania żądań
- Ułatwia przetwarzanie i obsługę złożonych danych przez konwertowanie ich do postaci typów danych obsługiwanych przez Python

# Źródła i bibliografia

- [www.djangoproject.com](http://www.djangoproject.com)
- [www.django-rest-framework.org](http://www.django-rest-framework.org)
- [www.askpython.com](http://www.askpython.com)
- [www.w3schools.com](http://www.w3schools.com)
- Wikipedia.org
- "Django. Tworzenie nowoczesnych aplikacji internetowych w Pythonie"