



Django

“The web framework for perfectionists with deadlines.”



What is a web framework?

- Also known as web **application** framework.
- It is a software framework that is designed to support the development of web sites and online software applications
- Web application includes (web services, web resources and web APIs).
- It is designed to simplify and accelerate the technical development process for web developers.



Introduction to

django



“Django makes it easier to build better Web apps more quickly and with less code.”

<https://www.djangoproject.com/>



Web Frameworks for Python:

- Django
- Flask
- Web2py
- TurboGear
- Bottle..... etc.....



Why Django?

- Flask (micro-framework) allows freedom.
- Django is high level. (Less code, more productive)
- Flask (own way, but may not be the right way)
- Django (Django way, the best way)
- Open Source (<https://github.com/django/>)
- Quick way to build an fully functional data driven apps
- 1000's of contributors and millions of user
- Active development/ well tested/ popular (31k stars in GitHub)



Why Django?

- Scalable
 - In terms of user
 - In terms of adding new features
- Architecture:
 - MVT
 - Contains main project
 - Project is composed of many apps



Django users

- Pinterest is a very high traffic, distributed Django system.
- Instagram's web application is written on Django.
- The Washington Post, Rdio, The Onion
- NASA also use Django.
- Bitbucket and EventBrite also run on Django

In Nepal:

- Janaki Tech (Khalti)
- MeroJob



Django – Design Philosophies

- Elements independent of each others. (Loosely Coupled)
- Less Coding and DRY
- Fast Development
- Clean Design – follows best web-development practices.



Advantages

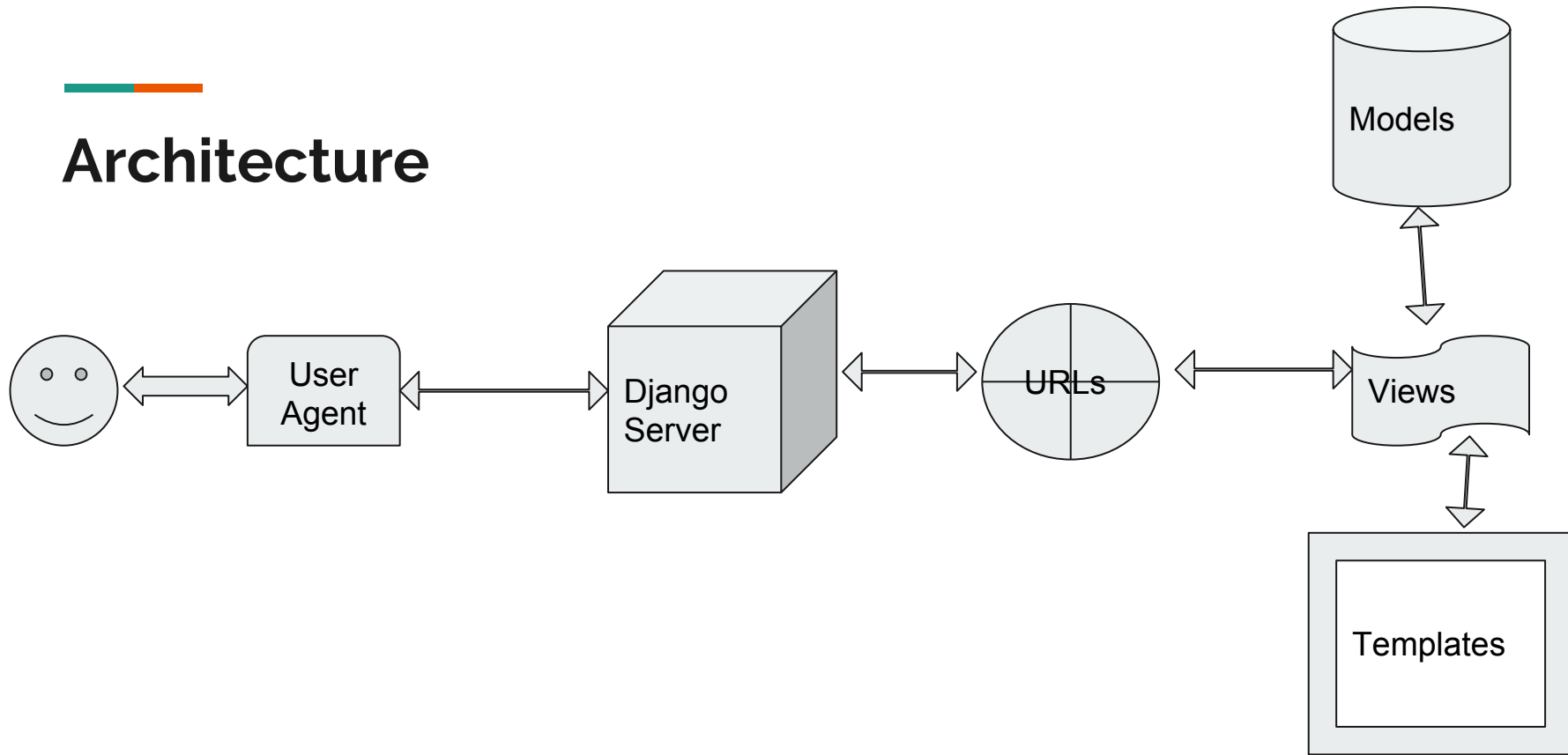
- **Object-Relational Mapping (ORM) Support** – Provides a bridge between the data model and the database engine,(sqlite3, MySQL, Oracle, Postgres) and NoSQL db (MongoDB / google app engine)
- **Multilingual Support**
- **Framework Support** – Django has built-in support for Ajax, RSS, Caching, session, auth and various other frameworks.
- **Administration GUI**
- **Lightweight Development Environment**



Architecture

- Follows MVT pattern
 - **Models** - Django ORM
 - **Views** - Python function/class, Request in Response out.
 - **Templates** - Django Template Engine
 - ***URL Patterns - Regular expression based***

Architecture





Requirements:

- Python 3.6.3
- Pip
- VirtualEnv (Not required but best to have)
- Django 1.11
- PyCharm IDE professional edition (Not necessary, but makes easier)
 - Can use sublime, vscode, notepad++
- GitHub account (To store the code)
- Ability to search for solution when something happens



Getting started:

```
$ django-admin startproject <project_name>
```

```
$ cd <project_folder>
```

```
$ python3 manage.py runserver
```