

TEAM LEAD VERSION (Backend-2)



CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

- ▶ Icebreaking
- ▶ Questions
- ▶ Interview Questions
- ▶ Coding Challenge
- ▶ Video of the week
- ▶ Retro meeting
- ▶ Case study / project

Teamwork Schedule

Ice-breaking

5m

- Personal Questions (Stay at home & Corona, Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Team work

5m

- Ask what exactly each student does for the team, if they know each other, if they care for each other, if they follow and talk with each other etc.

Ask Questions

15m

1. Which one is the reason of the using virtual environment?

- A.** Resolve dependency issues by allowing you to use different versions of a package for different projects. For example, you could use Package A v2.7 for Project X and PackageA v1.3 for Project Y.
- B.** Make your project self-contained and reproducible by capturing all package dependencies in a requirements file
- C.** Install packages on a host on which you do not have admin privileges
- D.** All of them

Answer: D

2. Which architectural design pattern does django use?

- A.** Model-view-presenter(MVP)
- B.** Hierarchical model-view-controller(HMVC)
- C.** Model – View – Template (MVT)
- D.** Model – Viewer – Controller (MVC)

Answer: C

3. Which command is the create django project?

- A. django-admin createproject -projectName-
- B. django-admin addProject -projectName-
- C. py manage.py createproject -projectName-
- D. django-admin startproject -projectName-

Answer: D

4. Which one is not default file in project folder?

- A. Urls.py
- B. Settings.py
- C. **init.py**
- D. Models.py

Answer: D

5. _____ is a place where we put the "logic" of our application. It will request information from the _____ you created before and pass it to a _____ .

- A. Url-View-Template
- B. Model-View-Template
- B. View-Url-Template
- B. View-Model-Template

Answer: D

Interview Questions

15m

1. Explain Django Architecture?

Answer:

Django follows the MVT (Model View Template) pattern which is based on the Model View Controller architecture. It's slightly different from the MVC pattern as it maintains its own conventions, so, the controller is handled by the framework itself. The template is a presentation layer. It is an HTML file mixed with Django Template Language (DTL). The developer provides the model, the view, and the template then maps it to a URL, and finally, Django serves it to the user.

2. Explain the django project directory structure?

Answer:

- `manage.py` - A command-line utility that allows you to interact with your Django project
- `__init__.py` - An empty file that tells Python that the current directory should be considered as a Python package
- `settings.py` - Comprises the configurations of the current project like DB connections.
- `urls.py` - All the URLs of the project are present here
- `wsgi.py` - This is an entry point for your application which is used by the web servers to serve the project you have created.

3. What are models in Django?

Answer:

A model in Django refers to a class that maps to a database table or database collection. Each attribute of the Django model class represents a database field. They are defined in `app/models.py`

4. What are templates in Django or Django template language?

Answer:

Templates are an integral part of the Django MVT architecture. They generally comprise HTML, CSS, and js in which dynamic variables and information are embedded with the help of views.

The main ones are variables and tags. A template is rendered with a context. Rendering just replaces variables with their values, present in the context, and processes tags. Everything else remains as it is. The syntax of the Django template language includes the following four constructs:

- Variables
- Tags
- Filters
- Comments

5. What is Django ORM?

Answer:

This ORM (an acronym for Object Relational Mapper) enables us to interact with databases in a more pythonic way like we can avoid writing raw queries, it is possible to retrieve, save, delete and perform other operations over the database without ever writing any SQL query. It works as an abstraction layer between the models and the database.

Coding Challenge

35m

- No



Coffee Break

10m



Video of the Week

10m

- [What is Django Capable of? | Companies That Use It](#)

Retro Meeting on a personal and team level

5m

Ask the questions below:

- What went well?
- What went wrong?
- What is the improvement areas?

Case study/Project

0m

- [Django ORM \(SQL to ORM\)](#)

Closing

5m

-Next week's plan

-QA Session
