

TEAM LEAD VERSION (Backend-4)



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 (Economic situation, Study Environment, Family&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 of these is not a logging framework message level?

- A.** Error
- B.** Critical
- C.** Important
- D.** Debug
- E.** Info

Answer: C

2. What is the difference between media and static files settings?

- A.** The media settings holds videos and images while static files hold CSS, js.
- B.** The media setting manages files uploaded by the user. Static settings manages the static assets of the website.

Answer: B

3. Suppose you want to count the number of books in Django.

books = Book.objects.all()

Which implementation would be fastest?

- A.** Database level Implementation – books.count()
- B.** Python Implementation – len(books)
- C.** Template Language Implementation – {{ books | length }}

Answer: A

4. Which of these variables are the settings for django.contrib.staticfiles app? Please select 4 correct answers

- A. STATIC_URL
- B. STATIC_ROOT
- C. MEDIA_URL
- D. MEDIA_ROOT
- E. STATICFILES_DIRS
- F. STATICFILES_FINDERS

Answer: A, B, E, F

5. Which of the following Password Validators are not provided by default in Django?

- A. NumericPasswordValidator
- B. CommonPasswordValidator
- C. MinimumLengthValidator
- D. MaximumLengthValidator

Answer: D

6. What is the benefit of unit testing?

- A. It helps protest against the introduction of bugs.
- B. It can help organize programming projects.
- C. It allows programmers to identify problem early in the development process.
- D. All of these answers are correct

Answer: D

Interview Questions

15m

1. Can you customize Django's admin interface?

Answer:

Yes, you can customize Django's admin interface. Django's admin is another entirely customizable application. It enables you to download another third-party application for a different view. You can create your own admin application to have complete control over it. Also, for customizing the Django admin site, you can change the settings of the admin site object.

Also, you can make the desired changes to your models and then apply them in the Django admin for adding specific applications, such as the search bar. You can customize even a smaller detail of your Django admin interface. Still, it is advisable to create a new admin rather than making so many changes at a lower level.

2. What happens when the Django website receives a request?

Answer:

Whenever a user enters the URL in the browser, the Django server receives the request. The server looks for the URL in its URL-config. If the server finds the match there, it will return the corresponding view function.

Then the request is made to the model of an application to get the data. If there is any data to be passed, pass it to the corresponding template. After, the template renders in the browser. If the process does not work as expected, the user will get a "404" error page.

3. Why is Django considered a loosely coupled framework?

Answer:

Django is considered a loosely coupled framework, as it is based on the MVT architecture, a variant of the MVC architecture. The MVT architecture is useful because it entirely separates the server code from the client's machine.

Models and views are available on a client machine. However, the client only receives the template — the HTML and CSS code — along with the data from the models.

Since these components are different, the front-end and back-end developers can work together on the same project. Making changes to a project by both the teams will not impact each other, thus making Django a loosely coupled framework.

4. What is the purpose of middleware in Django?

Answer:

In Django, middleware is the component that works on request and transfers it to the view, and before it passes it to the template engine, it starts operating on a response. It serves several purposes, including session management and user authentication.

5. What are signals in Django?

Answer:

Signals are pieces of code that hold information regarding what is happening. You can use a dispatcher for sending the signals and listening to those signals.

The signals become very useful when we want to do something with the data, before or after a certain event occurs.

Here is the list of events on which we can use the Django signals:

- pre_save() trigger before save().
- post_save() trigger after save().
- pre_delete() trigger before delete().
- post_delete() trigger after delete().

- m2m_changed() triggers when there is a change in ManyToMany Field.
- request_started() trigger when the django starts the HTTP request.
- request_finished() trigger when the django finishes the HTTP request.

Coding Challenge

0m

- No



Coffee Break

10m



Video of the Week

5m

- The Great Journey of a Backend Developer

Retro Meeting on a personal and team level

5m

Ask the questions below:

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

Case study/Project

30m

Pizza App

Closing

5m

-Next week's plan

-QA Session
