Django Serializer Interview Questions & Answers

1. What is a Serializer in Django REST Framework?

A serializer converts complex data types like Django model instances to native Python data types and JSON.

It also supports deserialization for data validation and saving.

2. Whats the difference between Serializer and ModelSerializer?

Serializer: Fields are defined manually.

ModelSerializer: Fields are automatically generated from a model.

3. How do you create a custom serializer?

Example:

```
class BookSerializer(serializers.Serializer):

    title = serializers.CharField(max_length=100)

    author = serializers.CharField()
```

4. How does ModelSerializer work?

Automatically generates fields from a model and includes default create() and update() methods.

5. What is the purpose of the Meta class in a serializer?

To specify model-related metadata like the model and fields to include or exclude.

6. How do you validate data in a serializer?

Use validate_<fieldname>() for field-level validation and validate() for object-level validation.

7. How do you make a field read-only?

Set read_only_fields in the Meta class.

8. What is partial=True in serializers?

Allows partial updates (used in PATCH requests).

9. How do you exclude a field from a serializer?

Use exclude = ['fieldname'] in Meta class.

10. How can you serialize nested relationships?

Use nested serializers by embedding one serializer inside another.

11. How do you create custom serializer fields?

Use SerializerMethodField and define get_<fieldname>(self, obj) method.

12. What is the use of to_representation()?

Customizes how the object is serialized to a native datatype.

13. What is to_internal_value() used for?

Defines how incoming data is converted into native types.

14. How can you override the default .create() and .update() methods?

Define custom create() and update() inside the serializer class.

15. How can you handle many-to-many relationships in serializers?

Use many=True in nested serializers.

16. What is depth in serializers?

Automatically nests related objects up to a certain level.

17. How to handle file uploads in serializers?

Use serializers.FileField() or serializers.ImageField().

18. What are HyperlinkedModelSerializers?

They use hyperlinks instead of primary keys to represent relationships.

19. Can you serialize querysets and single objects with the same serializer?

Yes. Use many=True for lists, and many=False (default) for single instances.

20. How do you validate unique fields using serializers?

Use UniqueValidator with the field definition.