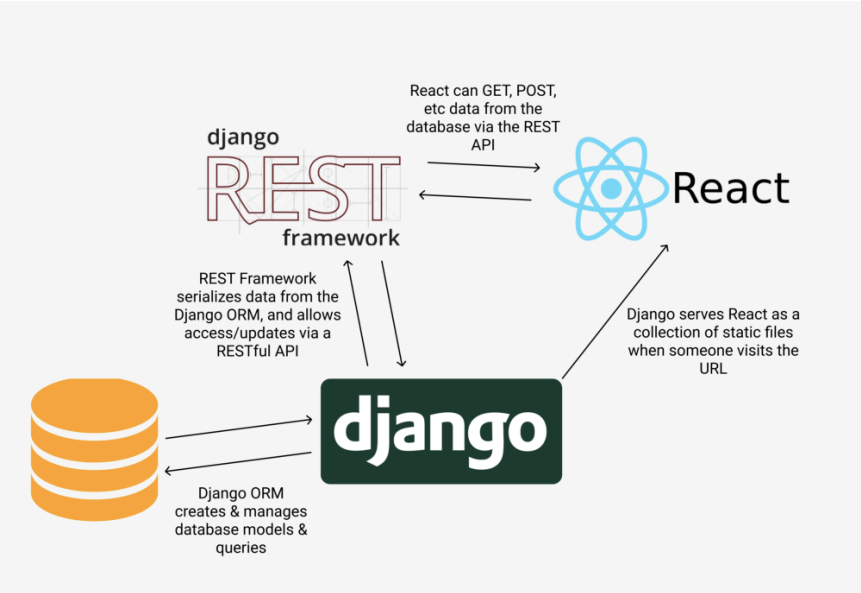

Why you would want to build an API?

- A REST API is a standardized way to provide data to other applications. Those applications can then use the data however they want.
- Sometimes, APIs also offer a way for other applications to make changes to the data.

- Typically, an API is a window into a database.
- The API backend handles querying the database and formatting the response.
- What you receive is a static response, usually in JSON format, of whatever resource you requested.

API for Web Applications

- Many web applications rely on REST APIs to allow the front end to talk to the back end.
- If you're deploying a React application atop Django,
  - you'll need an API to allow React to consume information from the database

Serialization

- The process of querying and converting tabular database values into JSON or another format is called **serialization**.
- When you're creating an API, correct serialization of data is the major challenge.

Why Django REST Framework?

The biggest reason to use Django REST Framework is because it makes **serialization** so easy!

Django ORM:

- The Django ORM handles all the database migrations and queries
- You define your models for your database using Python
- Think of the Django ORM like a librarian, pulling the information you need for you,
  - so you don't have to go get it yourself

The Django REST Framework, then, plays nicely with the Django ORM that's already doing all the heavy lifting of querying the database. Just a few lines of code using Django REST Framework, and you can serialize your database models to REST-ful formats.

Steps in Creating a REST API with Django

1. Set up Django
2. Create a model in the database that the Django ORM will manage
3. Set up the Django REST Framework
4. Serialize the model from *Step 2*
5. Create the URI endpoints to view the serialized data