REST using Flask

Duration:

- 3 days

Detailed course outline:

First REST API

- Installing Flask
- Access the code for this section here
- Your first Flask application
- HTTP Verbs
- REST Principles
- Creating our application endpoints
- Returning a list of stores
- Implementing other endpoints
- Calling the API from JavaScript
- Using Postman for API testing

Flask-Restful For more Efficient Development

- Virtualenvs and setting up Flask-RESTful
- Access the code for this section here
- Test-first API design—what is that?
- Creating our Item Resource
- The ItemList and creating Items
- Improving code and error control
- Authentication and logging in part 1
- Authentication and logging in part 2
- DELETE to delete Items
- PUT to create or update Items

- Advanced request parsing with Flask-RESTful
- Optimizing our final code and request parsing

Storing Resources in SQL Database

- Setting up our project
- Access the code for this section here
- Running a SQLite database and interacting with it from Python
- Logging in and retrieving Users from a database
- Signing up and writing Users to a database
- Preventing duplicate usernames when signing users up
- Retrieving our Item resources from a database
- Writing our Item resources to a database
- Deleting our Item resources from the database
- Refactoring insertion of items
- The PUT method with database interaction
- Retrieving many items from the database
- Advanced Flask-JWT Configuration

Simplifying storage with FLASK-SQLAlchemy

- Access the code for this section here
- Setting up this section's project
- Improving the project structure and maintainability
- Creating User and Item models
- Verifying the app works after our changes
- Advanced Postman: environments and tests
- Errata: small mistake in code in the next video
- Telling SQLAlchemy about our tables and columns
- Implementing the ItemModel using SQLAlchemy
- Implementing the UserModel using SQLAlchemy
- Easily displaying the ItemList resource with SQLAlchemy
- No more creating tables manually—telling SQLAlchemy to create tables
- Creating a new model: StoreModel
- Creating the Store Resource