10.3 Lesson Summary - Introduction to Flask & Serving Data with APIs

Data is often stored in a database with the purpose of sharing some of it with other people. As we have previously discussed the Internet provides an excellent way to share data with other people. Python can access data shared over the internet but it can also be used to share data over the Internet.

Concept: SQLAlchemy can be used to combine data from multiple different data model classes approximating the behavior of a **SQL Join**. Unlike a SQL Join the result of the query isn't a new table but rather a list of object pairings. If we had Customer and Person data model classes, we could use the following query to get the list of people represented in both tables:

*customer\_peoples = session.query(Customer, Person).filter(Customer.name == Person.name).limit(10).all()*

* Activity: 01-Ins\_Joins

Concept: To handle dates in Python you usually use the datetime module. To get the current date or 1/1/2000 you could use the following code:

*import datetime as dt*

*current\_date = dt.date.today()*

*first\_of\_the\_millennium = dt.date(2000, 1, 1)*

SQLAlchemy will store datetimes in the SQL database as dates.

* Activity: 02-Ins\_Dates, 03-Stu\_Dates
* Suppl link: <https://www.w3schools.com/python/python_datetime.asp>

Concept: In the section on APIs we defined the server-client model as a method of exchanging data across a network where the client is the consumer of the data and the server is the provider of the data. In the section on APIs we focused on the client. **Flask** is a Python tool that allows us to create a server to share our data with clients. The *@app.route* code before our methods defines the URL that the client can use to access the data defined in the method. To create a Flask server, you can use the following code:

*from flask import Flask*

*app = Flask(\_\_name\_\_)*

*@app.route("/")*

*def home():*

*return "Hello World!"*

* Activity: 04-Ins\_First\_Steps\_with\_Flask, 05-Stu\_Hello\_Web, 07-Stu\_Justice\_League

Concept: Data from APIs is often organized in **json** format. To convert text into json format you can use the *jsonify* method from the flask module. For example:

*from flask import Flask, jsonify*

*jsonify({"Hello": "World"})*

* Activity: 06-Ins\_Jsonify

Concept: **Variables** can be extracted from the URL used to specify the flask route. If you wanted to read the name of a person specified in a lookup route you could use the following code:

*@app.route("/who-is-that/<person\_name>")*

*def who\_is\_that\_person(person\_name):*

*return “that person is named ” + person\_name*

* Activity: 08-Ins\_Variable\_Rule, 09-Stu\_Variable\_Rule

Concept: The data Flask provides to the user is often stored in a **database**.

* Activity: 10-Ins\_Flask\_with\_ORM, 11-Stu\_Chinook