Use Case: Web Application

Aspects of a Web Request

The Life-Cycle of a Request

1 Send HTTP Request

Request Method, **Headers**, and **Body**

2 Route and Handle

Utilize the **Model**

3 Return Response

Render the **Resource**

HTTP Request Methods

- GET – Requests a representation of a specific resource. This type of request should only receive data.
- HEAD – Like a GET request, but only receiving the headers (no response body)
- POST – Submits data to the application, usually causing a change in the application's state
- PUT – Replaces the current representation of the target resource
- DELETE – Deletes the specified resource
- PATCH – Partially modifies a resource

Application Architecture

- Model – It holds onto the data and business logic. Ideally, the model holds onto “the brains” of our application.
- View – This is the representation of our business logic. For our application, the view portion will be the HTML templates that we render and return.
- Controller – This is the input/output interface for the application. The controller receives requests, then utilizes the model and view to display the proper response. In Flask, there are a few different ways to create a “controller,” but in its simplest form the controller will actually be the Flask application itself.