Research Week 13

3. What does REST stand for and what are some of the key concepts that identify it?

REST stands for REpresentational State Transfer. REST is a style of architecture or design pattern that provides communication standards between computers. It is not a protocol, but a set of constraints or standards that are used for providing web services, such as Application Programming Interfaces, or APIs.

REST most commonly uses HTTP methods or verbs as the medium for making requests and receiving responses between clients and servers. REST allows clients to request resources from servers to display, create, modify, or delete those resources. A resource is any object a server can provide information about, such as a document, a photo, a collection of photos, or even another client.

REST uses six constraints or guiding principles to define the architecture of communications between server and client.

1. Uniform Interface – has four parts:
    a. Requests must include a unique resource identifier for every resource requested.
    b. Responses must contain enough information about the state of the resource to allow the client to perform actions on the resource.
    c. Requests from the client and responses from the server must contain all the information required by the other to understand the request or response.
    d. The engine of application state is hypermedia, meaning that once the request from a client is made, the server returns hyperlinks to the client to allow the client to perform whatever actions on the resource(s) the client wants to do.

2. Separation of client and server:
    a. REST allows the client and server to be implemented independently of each other. Changes in code on either side have no effect on the other side.

3. Stateless interaction:
    a. The server and client each know nothing about the state of the other, so any request for resources made must contain all the information required to fulfill the needs of the request.

4. Layered System
    a. A hierarchical system composed of layers between the client and server. For example, the request from a client might pass through multiple servers before it reaches the server with the resource it is requesting. The client does not need to know anything about the layers between it and the final server.

5. Cacheable
    a. The server tells the client if a response can be cached for later reuse or to check for expired versions of previously cached data.

6. Optional Code on Demand
    a. The client may request code to be sent with a resource request that the client can run to implement features based on the resource.

Research Week 13

References

What is REST. (2022, April 7). Retrieved February 18, 2023 from https://restfulapi.net/

What is REST — A Simple Explanation for Beginners, Part 1: Introduction. (2017, September 5). Retrieved February 18, 2023 from https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1-introduction-b4a072f8740f

What is REST — A Simple Explanation for Beginners, Part 2: REST Constraints. (2017, September 5). Retrieved February 18, 2023 from https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-2-rest-constraints-129a4b69a582

What is REST? (n.d.). Retrieved February 18, 2023 from https://www.codecademy.com/article/what-is-rest

6. What is your favorite thing you learned this week?

I am excited to finally dig into Spring Boot and all that it can provide for programmers. When I first started thinking about enrolling in Promineo Tech's Back End Boot Camp, I wondered what Spring Boot was and why I had never heard of it before. Then I heard that it is like Java on steroids, which made me even more intrigued.

Now the day is finally here, and I can start applying Spring Boot tools on top of the Java I have already learned. I am looking forward to learning more and more about Spring Boot, along with Maven, Swagger, and Lombok as tools for developing robust applications in a more powerful, and perhaps quicker, way.