

RESTFul Services In Java Using Jersey

HTTP PUT Request

Bryan Hansen
twitter: bh5k

<http://www.linkedin.com/in/hansenbryan>



pluralsight 
hardcore developer training

PUT



PUT

- Early on some browsers didn't support PUT (or DELETE)
- REST often gets over simplified
 - Developers use POST to CREATE and UPDATE
 - Theoretically PUT can be used for both functions
- **PUT is idempotent**
 - We can call PUT multiple times and it shouldn't affect the app
 - Think of it like a SQL update, we can update the database multiple times and it won't affect the app
 - PUT should be used if you are supplying the ID for your Object
 - POST should be use if the server is supplying the ID



URLs

- **POST URL should look like:**

- <http://localhost:8080/exercise-services/webapi/activities/activity>
- Notice no ID
- POST to a generic URL
- Each call will create a new Object

- **PUT URL should look like:**

- <http://localhost:8080/exercise-services/webapi/activities/1234>
- Notice the ID
- PUT to a specific URL
- A call will either create or update an Object (Idempotent)

PUT Client

- Unit Test
- Client Code
- Server
- Repository/Stub

PATCH

- Used to do partial updates
- Not built into Jersey yet
- Wouldn't need to send the entire object

Summary

- Client update
- PUT vs POST
- Client creation
- Unit Testing
- PATCH