UCSC Silicon Valley Extension

Designing, Building, and Integrating RESTful API

Dr. Min Wu
mwu63@ucsc.edu

History of Web Services

- Local network vs. Internet
- Page driven vs. data driven
  - Synchronous vs. Asynchronous
- SOAP vs. REST
  - XML vs. JSON
  - Verb centric vs. noun centric
  - Limited and tightly coupled vs. encouraging creativity

HTTP Basics

- HTTP request and response
- A start line
  - Request line: `GET /index.html HTTP/1.1`
  - Response line: `HTTP/1.1 200 OK`
- Zero or more header fields followed by CRLF
- An empty line indicating the end of the header fields
- Optionally a message body

POST

- The POST verb is utilized to **create** new subordinate resources.
- On successful creation, return HTTP status 201
  - With newly-created resource in the response body or
  - A Location header with a link to the newly-created resource
- POST is neither safe nor idempotent.

GET

- The HTTP GET method is used to **read** (or retrieve) a representation of a resource.
- In a successful case, GET returns a representation in XML or JSON and an HTTP response code of 200 (OK).
- In a failure case, it returns a 404 (NOT FOUND) or 400 (BAD REQUEST).
- GET is considered safe. That is, they can be called without risk of data modification or corruption.
- Additionally, GET is idempotent, which means that making multiple identical requests ends up having the same result as a single request.

PUT

- PUT is utilized for **update** capabilities, PUT-ing to a known resource URI with the request body containing the newly-updated representation of the original resource.
- On successful update,
  - return 200 with the updated resource in response body or
  - return 204 without the response body
- PUT is not a safe operation, in that it modifies (or creates) state on the server,
- PUT is idempotent.
  - Use POST for non-idempotent requests.

PATCH

- PATCH is used for **modify** capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource.
- PATCH resembles PUT. PATCH body content should be in some kind of patch language, e.g., JSON Patch
- PATCH is neither safe nor idempotent.

DELETE

- DELETE is to **delete** a resource identified by a URI.
- On successful deletion,
  - return HTTP status 204 (NO CONTENT) with no response body or
  - return HTTP status 200 (OK) along with a response body, the representation of the deleted item
- DELETE is idempotent.