Spring RESTful Web Services

Lesson 02

Agenda

- REST Web Services design principles
- HTTP Methods
- Web services URI design Guidelines

RESTful Web Service Design Principles

- **Four basic design principles:**
  - Use HTTP methods explicitly.
  - Be stateless.
  - Expose directory structure-like URIs.
  - Representation oriented - Transfer XML, JavaScript Object Notation (JSON), or both.

HTTP-REST Vocabulary

HTTP Methods supported by REST:

- GET – Requests a resource at the request URL
- POST – Submits information to the service for processing
- PUT – Add a new resource at the request URL
- DELETE – Removes the resource at the request URL
- OPTIONS – Indicates which methods are supported
- HEAD – Returns meta information about the request URL

HTTP-Methods

- **HTTP-GET**
  - GET is a read-only operation.
  - It is used to query the server for specific information.
  - It is both an idempotent operation'
  - It is safe

- **HTTP-PUT**
  - It is modeled as an insert or update.
  - It is idempotent.
  - PUT requests that the server store the message body sent with the request under the location provided in the HTTP message.
  - When using PUT, the client knows the identity of the resource it is creating or updating.

HTTP-Methods

- HTTP-DELETE
  - DELETE is used to remove resources.
  - It is idempotent as well.

- HTTP-POST
  - POST is the only non idempotent and unsafe operation of HTTP.
  - Each POST method is allowed to modify the service in a unique way.
  - You may or may not send information with the request.
  - You may or may not receive information from the response.

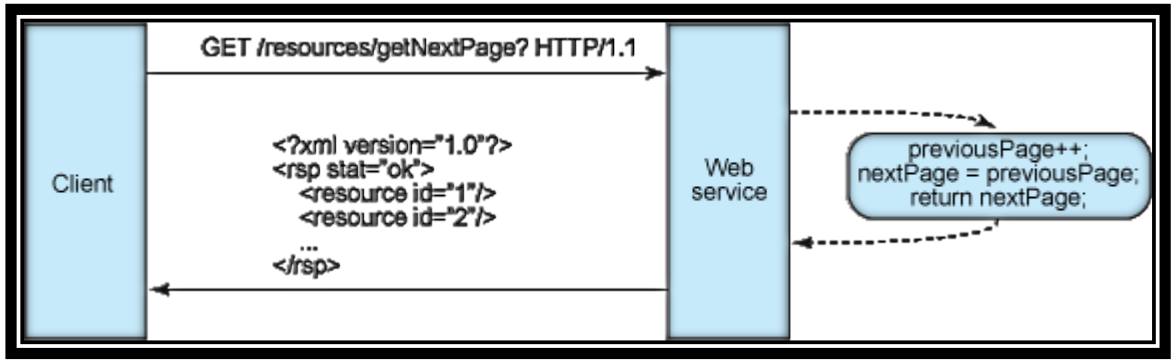
Use HTTP methods explicitly

- REST design principle establishes a one-to-one mapping between create, read, update, and delete (CRUD) operations and HTTP methods.
- According to this mapping:
  - To **create** a resource on the server, use **POST**.
  - To **retrieve** a resource, use **GET**.
  - To **change** the state of a resource or to update it, use **PUT**.
  - To **remove** or delete a resource, use **DELETE**.

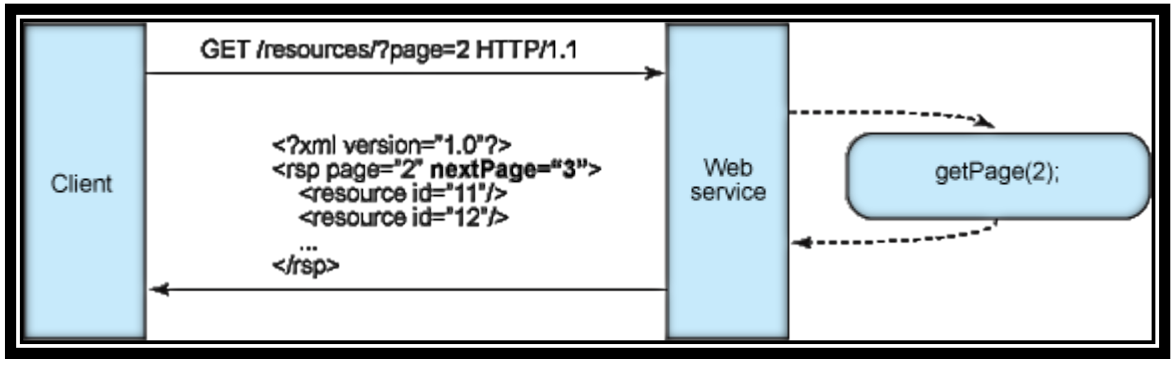
When applying REST over HTTP, stick to the methods provided by the protocol

Stateless

Stateful



Stateless



Stateless

- Stateless means that there is no client session data stored on the server.
- The server only records and manages the state of the resources it exposes.
- If there needs to be session-specific data, it should be held and maintained by the client and transferred to the server with each request as needed.

Expose directory structure-like URIs

- For REST, define directory structure-like URIs.
- This type of URI is hierarchical, rooted at a single path, and branching from it are sub-paths that expose the service's main areas.

http://www.igate.com/project/employee/{id}

Guidelines about URI structure

- Hide the server-side scripting technology file extensions (.jsp, .php, .asp)
- Keep everything lowercase.
- Substitute spaces with hyphens or underscores (one or the other).
- Avoid query strings as much as you can.
- Instead of using the 404 Not Found code if the request URI is for a partial path, always provide a default page or resource as a response.
- URIs should also be static so that when the resource changes or the implementation of the service changes, the link stays the same

Representation oriented

- An object referenced by one URI can have different formats available.
- Different platforms need different formats.
  - AJAX needs JSON.
  - A Java application needs XML.

Common MIME types used by RESTful services

MIME-Type	Content-Type
JSON	application/json
XML	application/xml
XHTML	application/xhtml+xml

Summary

In this lesson we have seen :-

- What are REST Web Services design principles
- What are diff HTTP Methods supported by REST
- Web services URI design Guidelines