RESTful Web Services

API,REST, RESTfull API

Week 10

What is an API?

- API stands for Application Programming Interface.
- API is the code (Interface) that allows two software components talk to each other.
- The simplest example of API is the Command Line Interface (CLI). It allows computer users (on Terminal) to talk to the Operating system. For example:
  - C:\> dir => is a request to display the directory contents
  - C:\> mkdir temp => is a request to create a directory names ‘temp’
  - C:\> copy test.txt temp\ => is a command to copy the file test.txt to the temp directory
- You use APIs every day, even if you are not aware of them. When you create a new Facebook or Twitter post, or like a post, or comment on it you are using APIs.

What is an API?

- API stands for Application Programming Interface.

- REST stands for Representational State Transfer.
- REST is Web-standard based architecture that uses HTTP protocol.
- In this architecture, every component is a resource and all resources are accessed by a common interface.
- In the REST architecture, you need a REST Server to provide access to resources and a REST client to access and manipulate resources.
- Each resource is identified by URI (which global ID).
- REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

What is JSON?

- JSON stands for JavaScript Object Notation.
  - Lightweight data-interchange format
  - Despite the name, JSON is a language-independent way of specifying objects as name-value pairs
  - Structured representation of data object
  - Can be parsed with most modern languages
- JSON Schema can be used to validated a JSON file

JSON Syntax Rules

- JSON is uses Javascript object notation { key: value }
- Uses key/value pairs: {“name”: “Ali”}
- Uses double quotes around KEY and VALUE
- File type is “.json”
- A *value* can be: string, number, true, false, null, object, or array
- *Strings* are enclosed in double quotes, and can contain the usual assortment of escaped characters

JSON Example

```
{
        "name": “Ali Salem",
        "age": 35,
        "address": {
                "street": "5 main St.",
                "city": “Benghazi"
        },
        "children": [“Salem", "Abed"]
}
```

JSON Schema

- A JSON Schema allows you to specify what type of data can go into your JSON files.
- It allows you to restrict the type of data entered.

```
{   "type": "object",
  "properties": {
        "name": {
          "type": "string"
        },
        "age": {
          "type": "integer"
        }
    }
    }
```

Validating JSON file

- The following website can be used to validate a JSON file against a schema

  https://www.jsonschemavalidator.net/

- Paste both the schema and the corresponding JSON file

RESTful Web Services

- A Web service is a collection of open protocols and standards used for exchanging data between applications or systems.
- Software applications written in various programming languages (Javascript and Python) and running on various platforms (Win, Linux) can use web services to exchange data over the Internet.
- This interoperability is made possible because of using open standards.
- Web services based on REST Architecture are known as RESTful web services.
- These web services uses HTTP methods to implement the concept of REST architecture.
- A RESTful web service usually defines a Uniform Resource Identifier (URI).
- A service provides resource representation such as JSON and set of HTTP Methods.

HTTP Methods

- There are four HTTP methods commonly used in REST based architecture:

HTTP Method	Description
GET	Provides a read only access to a resource.
POST	Used to update an existing resource or create a new resource.
DELETE	Used to remove a resource.
PUT	Used to create a new resource.

Creating a RESTful Web Server

- We will create a REST Web Server with Web Service to for weather Information.
- The web service will perform the following functions:

URI	HTTP Method	POST body	Result
/data/	GET	Empty	Show list of all cities
/data/:id	GET	Empty	Show details for 1 city.

- Create a RESTful Web Server inside **weather-api** folder (Use express.js and SQLite3).
- Steps to set up RESTful Web Server:
  - Create a SQLite database to hold weather data.
  - Create a REST Web Server to handle requests from clients.
  - Test the API using a testing tool.

References

- Node.js the Right Way, Jim R. Wilson, The Programatic Bookshelf, ISBN 978-1-937785-73-4
- NodeJS: Practical Guide for Beginners, Matthew Gimson, ISBN 978-1-519354-07-5
- Express.js: Node.js Framework for Web Application Development, Daniel Green, ISBN 978-1-530204-06-9
- Express.com
- Tutorials Point
- The Node Beginner Book, Manuel Kiessling, Leanpub, Link.