**Building RESTful APIs Using Express**

**1) Introduction**:

A REST API defines a set of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST.

**2) RESTful Services**:

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

REST stands for Representational State Transfer. REST is web standards-based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/ global IDs. REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

**HTTP methods**:

Following four HTTP methods are commonly used in REST based architecture.

1. GET − Provides a read only access to a resource.
2. POST − Used to create a new resource.
3. DELETE − Used to remove a resource.
4. PUT − Used to update an existing resource or create a new resource.

**3) Introducing Express**:

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the Node.js foundation.

Express was developed by TJ Holowaychuk and is maintained by the Node.js foundation and numerous open source contributors.

**Install Express**:

For install express run the following command in console

npm i express

**4) Building Your First Web Server (Using Express)**:

Create a file name index.js in our project root directory. In this file write the following code.

const express = require("express");

const app = express();

app.get("/", (req, res) => {

res.send("Hello World");

});

app.listen(3000, () => console.log("Listening on port 3000..."));

Here, first we have to lode the module "express"

const express = require("express");

Here, require("express") return a function. express() function return an object and by convention we store it in app variable.

const app = express();

Now we have to define a route.

app.get("/", (req, res) => {

res.send("Hello World");

});

* Here, get() method have two parameter "/" and a callback function.
* "/" represent the root of the website
* callback function has two parameters (req, res). req -> Request object have a branch of useful properties that gives us information about incoming request. For more info we have to see Express documentation <http://expressjs.com/> . res -> Response is use to response something to the client.

Finally, we need to listen on a given port.

app.listen(3000, () => console.log("Listening on port 3000..."));

Now go to the console and run "node index.js". Program will run and print " Listening on port 3000..." in console. Go to browser and in the address bar write <http://localhost:3000/> and press enter. In the browser we will see "Hello World"

Now another routes in our application.

const express = require("express");

const app = express();

app.get("/", (req, res) => {

res.send("Hello World");

});

app.get("/api/courses", (req, res) => {

res.send([1, 2, 3, 4, 5]);

});

app.listen(3000, () => console.log("Listening on port 3000..."));

**5) Nodemon**:

**6) Environment Variables**:

**7) Route Parameters**:

**8) Handling HTTP GET Requests**:

**9) Handling HTTP POST Requests**:

**10) Calling Endpoints Using Postman**:

**11) Input Validation**:

**12) Handling HTTP PUT Requests**:

**13) Handling HTTP Delete Requests**:

**14) Project- Build the Genres API**:

04 Building RESTful APIs Using Express