Read SOAP

Simple Restful Webservice

Restful web service with Node.js

Node.js – RESTful API

- **REST** stands for **RE**presentational **S**tate **T**ransfer. REST is web standards based architecture and uses HTTP protocol. In this architecture, 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.
- Each resource is identified by URIs (which are global IDs).
- REST uses various representation to represent a resource like text, JSON, XML. JSON is the most popular one.

HTTP Methods

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

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

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 and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.
- This interoperability e.g. between java and Python, or Windows and Linux applications is due to the use of 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 URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

Creating a RESTful Web Server

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

URI	HTTP Method	POST body	Result
/users/	GET	Empty	Show list of all the users
/users/:id	GET	Empty	Show details of a user.

- Create a RESTful Web Server inside **myapp** folder (in which contains all the programs that using express.js). (See app10.js)
- Steps to set up RESTful Web Server:
  - Create a json based database of users. (user.json)
  - Create a REST Web Server to handle requests from clients. (app10.js)

Running the RESTful Web server

- Start the web server: /nodejs_workspace/myapp/node app10.js
- To call the web server from browser:
  - From browser: http://localhost:8081/ to see instructions
  - From browser: http://localhost:8081/users to see all data
  - From browser: http://localhost:8081/1 to see information about id = 1
- To call the REST web service from another server:
  - From another server: /nodejs_workspace/myapp/node app10_client.js
  - From another server: /nodejs_workspace/myapp/node app10_client2.js

user.json


```
{
    "user1": { "name": "Name1", "password": "password1",
        "profession": "teacher", "id": 1 },
    "user2": { "name": "Name2", "password": "password2",
        "profession": "librarian", "id": 2 },
    "user3": { "name": "Name3", "password": "password3",
        "profession": "clerk", "id": 3 }
}
```

app10.js

```
var express = require('express');
var app = express();
var router = express.router();
var fs = require('fs' );
router.get('/', function (req, res, next) {
    console.log('request from root');
    res.send('To list users: http://localhost:8081/users </br>' +
        'To list details: http://localhost:8081/users/1');
});
```

app10.js

```
// GET users listing
router.get('/users', function (req, res, next) {
    fs.readFile(__dirname + "/user.json", "utf8", function (error, data) {
        console.log(data);
        res.end(data);
    });
});
```

app10.js

```
// show the details
app.get('/users/:id', function (req, res, next) {
	// reading the existing users.
	fs.readFile(__dirname + "/user.json", "utf8", function (error, data) {
		var d = JSON.parse(data);
		var user = d["user" + req.params.id];
		if (typeof user === 'undefined') {
			console.log('No such user, id: ' + req.params.id);
			res.end('No such user, id: ' + req.params.id);
		} else {
			console.log(user);
			res.json(user);
		}
	});
});
```

app10.js

```
// mount the router on the app
app.use('/', router);


app.listen(8081, function () {
    console.log('app.js listening to http://127.0.0.1:8081/ or http://localhost:8081/');
});


console.log('Program End.');
```

app10_client.js

```
// http module is required to create a web service client
var http = require('http');

// options are to be used by request
var options = {
    host: 'localhost',
    port: '8081',
    path: '/users'
};
```

app10_client.js

```
var callback = function (response) {
	// continuously update stream with data
	var body = '';
	response.on('data', function (data) {
		body += data;
	});
	response.on('end', function () {
		// data received completely
		console.log(body);
	});
};
// make a request to the server
var req = http.request(options, callback);
req.end();
```

app10_client2.js

```
// http module is required to create a web service client
var http = require('http');

// options are to be used by request
var options = {
    host: 'localhost',
    port: '8081',
    path: '/users/2'
};
```

app10_client2.js

```
// callback function is used to deal with response
var callback = function (response) {
	// continuously update stream with data
	var body = '';
	response.on('data', function (data) {
		body += data;
	});
	response.on('end', function () {
		// data received completely
		console.log(body);
	});
};
// make a request to the server
var req = http.request(options, callback);
req.end();
```