**Assisted Practice: 2.2 Consuming RESTful Web Services**

This section will guide you to:

* Set up Eclipse to work with Spring Boot using the STS plugin
* Create a class to work with a public REST service (<https://gturnquist-quoters.cfapps.io/api/random>)
* Create a class to act as a wrapper for the data sent by the REST Service
* Create a Controller class to consume the REST Service using the RestTemplate class

**Development Environment**

* Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
* Apache Tomcat Server v9.0
* JRE: OpenJDK Runtime Environment 11.0.2
* Spring Boot STS 4
* All other software components are configured automatically by Spring Boot

This lab has nine subsections, namely:

* + 1. Installing the STS plugin in Eclipse
    2. Creating a Spring Boot Starter Project which is web enabled
    3. Creating a class Quote to work with the public REST service
    4. Creating a class Value to act as a wrapper for the REST data
    5. Creating MainController to consume the REST service
    6. Building the project
    7. Publishing and starting the project
    8. Running the project
    9. Pushing the code to your GitHub repositories

**Step 2.2.1:** Installing the Spring Tool Suite plugin in Eclipse

* Spring Tool Suite is already installed as an Eclipse plugin in your practice lab. (Refer FSD: Lab Guide - Phase 3 to verify the installation.)

**Step 2.2.2:** Creating a Spring Boot Starter Project which is web enabled

* Open Eclipse
* Go to the **File** menu. Choose **New->Other**
* In the **Wizard** list, select **Spring Boot->Spring Starter Project**
* In **Name,** enter SpringRESTClient, **Type** as Maven, **Packaging** as Jar, **Group** as com.ecommerce, and **Package** as com.ecommerce
* Click on **Next**
* In the list of **Available** dependencies, scroll down to select **Web->Spring Web Starter**
* Click on **Next**
* Click on **Finish**
* This will create the project files in the Project Explorer

**Step 2.2.3:** Creating a class Quote to work with the public REST service

* In the Project Explorer, expand **SpringRESTClient->src->main>java>com->ecommerce**
* Right click on ecommerce and click on **New->Folder**
* Enter **Folder Name** as **beans** and click on **Finish**
* Right click on **beans** and click on **New->Other**
* In the **Wizard** list, choose **Class** and click on **Next**
* In **Name,** enter **Quote** and click on **Finish**
* Add the following code:

**package** com.ecommerce.beans;

import com.fasterxml.jackson.annotation.\*;

**import** com.fasterxml.jackson.annotation.JsonIgnoreProperties;

**@JsonIgnoreProperties(ignoreUnknown = true)**

**public** **class** Quote {

**private** **String** type;

**private** Value value;

**public** Quote() {

}

**public** **String** getType() {

**return** type;

}

**public** void setType(**String** type) {

**this**.type = type;

}

**public** Value getValue() {

**return** value;

}

**public** void setValue(Value value) {

**this**.value = value;

}

**@Override**

**public** **String** toString() {

**return** "Quote{" +

"type='" + type + '\'' +

", value=" + value +

'}';

}

}

**Step 2.2.4:** Creating a class Value to act as a wrapper for the REST data

* In the Project Explorer, expand **SpringRESTClient->src->main>java>com->ecommerce->beans**
* Right click on **beans** and click on **New->Other**
* In the **Wizard** list, choose **Class** and click on **Next**
* In **Name,** enter **Quote** and click on **Finish**
* Add the following code:

**package** com.ecommerce.beans;

**import** com.fasterxml.jackson.annotation.JsonIgnoreProperties;

**@JsonIgnoreProperties(ignoreUnknown = true)**

**public** **class** Value {

**private** **Long** id;

**private** **String** quote;

**public** Value() {

}

**public** **Long** getId() {

**return** **this**.id;

}

**public** **String** getQuote() {

**return** **this**.quote;

}

**public** void setId(**Long** id) {

**this**.id = id;

}

**public** void setQuote(**String** quote) {

**this**.quote = quote;

}

**@Override**

**public** **String** toString() {

**return** "Value{" +

"id=" + id +

", quote='" + quote + '\'' +

'}';

}

}

**Step 2.2.5:** Creating MainController to consume the REST service

* In the Project Explorer, expand **SpringRESTClient->src->main>java>com->ecommerce->controllers**
* Right click on **controllers** and click on **New->Other**
* In the **Wizard** list, choose **Class** and click on **Next**
* In **Name,** enter **MainController** and click on **Finish**
* Add the following code:

**package** com.ecommerce.controllers;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** org.springframework.web.client.RestTemplate;

**import** com.ecommerce.beans.Quote;

**@Controller**

**public** **class** MainController {

**@RequestMapping("/")**

**@ResponseBody**

**public** **String** index() {

RestTemplate restTemplate = **new** RestTemplate();

Quote quote = restTemplate.getForObject("https://gturnquist-quoters.cfapps.io/api/random", Quote.class);

**return** quote.toString();

}

}

**Step 2.2.6:** Building the project

* From the **Project** menu at the top, click on **Build**
* If any compile errors are shown, fix them as required

**Step 2.2.7:** Publishing and starting the project

* In the Project Explorer, right click **SpringRESTClient->Run As->Spring Boot App**
* Check in the Eclipse Console for the message **Started SpringBootStarterApplication**

**Step 2.2.8:** Running the project

* To run the project, open a web browser and type [**http://localhost:8080**](http://localhost:8080/product/0)

**Step 2.2.9:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**