**Objectives**

* Demonstrate implementation of RESTful Web Service using POST/PUT/DELETE method with input validation
  + HTTP method types (GET, POST, PUT, DELETE), REST service URL naming guidelines, @RequestMapping, @GetMapping, @PostMapping, @PutMapping, @DeleteMapping, setting POST request payload and invoking the REST service in Postman and curl, JSON to bean mapping, @RequestBody, validating input request using javax.validation and hibernate validators, @Size, @NotNull, @NotBlank, @Min, @Max, @JsonFormat, @Valid, global exception handling, handle number formatting errors
    - HTTP Request Methods - https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods
    - RESTful API naming guide - https://restfulapi.net/resource-naming/
    - Request Mapping - https://docs.spring.io/spring/docs/5.2.0.RELEASE/spring-framework-reference/web.html#mvc-ann-requestmapping
    - Validation - https://www.mkyong.com/spring-boot/spring-rest-validation-example/

**Significance of HTTP Method Types in RESTful Web Services**   
  
SME to explain the importance of HTTP Method Types for RESTful Web Services. 

|  |  |
| --- | --- |
| **HTTP Method** | **Usage Scenario** |
| GET | Used to get data about a resource |
| POST | Used to create a resource |
| PUT | Used to update a resource |
| DELETE | Used to delete a resource |

The method type is just a classification and does not actually have the persistence implemented. The respective application is expected to take responsibility in implementing the persistence.

**RESTful Web Service resource naming guidelines**   
  
Find below the guidelines to define a RESTful Web Service URL: 

* Each resource should have a unique and specific URL
* To get all resources provide the resource name in plural
* To get a specific resource provide resource name in plural followed with slash and parameter
* To create a resource the URL should be the resource name in plural and the data to create the resource should be defined in the payload
* To update a resource the URL should be the resource name in plural with data in payload
* To delete a resource the URL should be the resource name in plural followed by slash and the specific resource to delete
* Resource name with multiple words should be separated by hyphen and not with underscore. For example, if the resource it menu item implement the URL as "menu-item".

Refer table below with example for resource as country. 

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Type** | **URL** | **Description** | **Annotation** |
| GET | http://sample.api.com/app-name/countries | Get all countries | @GetMapping |
| GET | http://sample.api.com/app-name/countries/{code} | Get a specific country | @GetMapping("/{id}") |
| POST | http://sample.api.com/app-name/countries | Create country based on data in post | @PostMapping |
| PUT | http://sample.api.com/app-name/countries | Update country based on data in post | @PutMapping |
| DELETE | http://sample.api.com/app-name/countries/{code} | Delete a specific country | @DeleteMapping("/{id}") |

For a particular resource, the URL should be the same for all the methods. Hence in CountryController, the URL can be defined at the class level:

@RequestMapping("/countries")

Find below the method specific annotation definitions:   
  
Get All

@GetMapping

Get specific resource

@GetMapping("/{id}")

Create resource

@PostMapping

NOTE: Payload data should be sent in the body of the request   
  
Update resource

@PutMapping

NOTE: Payload data should be sent in the body of the request   
  
Delete resource

@DeleteMapping("/{id}")

Going forward ensure that this convention is followed when defining a new service.  
  
Modify CountryController to adhere to the above mentioned standards.

**Create RESTful Web Service to handle POST request of Country**   
  
A new RESTful Web Service method to handle POST request of Country. Follow steps below to incorporate the same: 

* Create new method in CountryController based on the following details:
  + Annotation - @PostMapping()
  + Method Signature - public void addCountry()
* Within this method  include "Start" logger.
* Start the web application
* Open Git Bash
* Execute the following curl command, to invoke the web service:
  + -i to display the headers
  + -X to define the HTTP method type
  + -s silent mode, so that performance details are not displayed

curl -i -X POST -s http://localhost:8090/countries

* Check if "Start" is displayed in the console output
* Following is the expected output:

HTTP/1.1 200

Content-Length: 0

Date: Tue, 01 Oct 2019 06:41:49 GMT

* The invocation of web service can also be done using Postman.
* Check the logger if "Start" is logged

**Read country data as a bean in RESTful Web Service**   
  
The country data should be included in the request payload, which should be read by the controller method.  
  
Follow steps below to incorporate the same:

* Include country as parameter to addCountry() method with @RequestBody annotation and country as parameter. Refer method signature below.

public Country addCountry(@RequestBody Country country)

* Include log to display country details
* Return the country. This is to check if country details are populated correctly
* Invoke the service using the following curl command. This can also be tried for execution from Postman.
  + -H denotes inclusion of header. This denotes that we are sending content type in the request header and it mentions that the request payload is of type JSON
  + -d denotes the data payload sent in the request. This represents the country to be added

curl -i -H 'Content-Type: application/json' -X POST -s -d '{"code":"IN","name":"India"}' http://localhost:8090/countries

* Refer the expected HTTP response below:

HTTP/1.1 200

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Tue, 01 Oct 2019 17:23:47 GMT

{"code":"IN","name":"India"}

* Try running the request with minor change and let us see the response. Sample response below. The attribute name is intentionally provided with a spelling mistake.

curl -i -H 'Content-Type: application/json' -X POST -s -d '{"code":"IN","nae":"India"}' http://localhost:8090/countries

* Refer the expected HTTP response below:

HTTP/1.1 200

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Tue, 01 Oct 2019 17:23:47 GMT

{"code":"IN","name":null}

**SME to provide explanation about the following aspects:**

* Explain how spring framework takes care of converting the request payload into country bean
* Spring parses the JSON request payload data using Jackson parser
* For each attribute in JSON, respective method name is constructed by applying initcaps and get prefix. For example, the name attribute is changed with initcaps as Name, then get is prefixed to it which results in getName, based on this the respective method is invoked using Reflection API.
* Spring creates country object and invokes the respective setter method based on JSON data.
* The it invokes the controller method passing the country object created
* Provide explanation regarding bean naming conventions