Developing Spring REST API

Lab78: Building a First RESTful Web Service

Lab78:Working Steps:

A)Setup the Project:

1. **Create Dynamic Web Project**
   a. Project name : Lab78
   b. Tomcat8.5 as target Runtime.

2. **Copy following Jars to WEB-INF/lib folder**
   a. 21 Spring Jars
   b. 6 Jackson Jars

3. **Create the package called com.coursecube.spring**
4. **Write the Following Classes**
   a. JLCWebAppInitializer.java
   b. JLCWebConfig.java

B)Develop RESTful Web Service

5. **Create a Resource Representation Class - Hello**

```
public class Hello {

        String name;
        String messgae;

        //Constrcutors
        //Setters and Getters
}
```

6. **Create a Resource Controller called HelloController**

```
@RestController
public class HelloController {

@GetMapping("/myhello")
public String getMessage1() {
....
}

@GetMapping("/myhello/{name}")
public String getMessage2(@PathVariable String name) {
....
}
```

```
@GetMapping("/myhello/{name}/{email}")
public String getMessage3(@PathVariable String name,@PathVariable String email) {
....
}

@GetMapping(value="/hello1/{name}",produces = "application/json" )
public Hello getMessage4(@PathVariable String name) {
....
}


@GetMapping(value = "/hello2/{name}", produces = "application/xml")
public Hello getMessage5(@PathVariable String name) {

}

@RequestMapping(value = "/hello3/{name}", method = RequestMethod.GET)
public Hello getMessage6(@PathVariable("name") String name) {
....
}

@RequestMapping(value = "/hello4/{name}", method =
RequestMethod.GET,produces = "application/json")
public Hello getMessage7(@PathVariable("name") String name) {
....
}
}
```

C) Test Rest API with Browser.

7. **Hit the following urls from Browser**

http://localhost:5050/Lab78/myhello

http://localhost:5050/Lab78/myhello/srinivas

http://localhost:5050/Lab78/myhello/srinivas/sri@jlc

http://localhost:5050/Lab78/hello1/srinivas

http://localhost:5050/Lab78/hello2/srinivas

http://localhost:5050/Lab78/hello3/srinivas

http://localhost:5050/Lab78/hello4/srinivas

D) Test Rest API with CURL.

A. **Download CURL from https://curl.haxx.se/download.html**
   You will get zip: curl-7.67.0.zip

B. Extract to E: drive.
Now curl home is E:\curl-7.67.0

C. Goto E:\curl-7.67.0 and type the following

curl -v -X GET localhost:5050/Lab78/myhello
curl -v -X GET localhost:5050/Lab78/myhello/vas
curl -v -X GET localhost:5050/Lab78/myhello/vas/vas@jlc
curl -v -X GET localhost:5050/Lab78/hello1/srinivas
curl -v -X GET localhost:5050/Lab78/hello2/srinivas
curl -v -X GET localhost:5050/Lab78/hello3/srinivas
curl -v -X GET localhost:5050/Lab78/hello4/srinivas

E) Test Rest API with Postman

try yourself

Lab78: Files required

1. HelloController.java	2. Hello.java
3. JLCWebConfig.java	4. JLCWebAppInitializer.java

1. HelloController.jsp

```
package com.coursecube.spring;

import org.springframework.web.bind.annotation.*;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@RestController
public class HelloController {

@GetMapping("/myhello")
public String getMessage1() {
System.out.println("HC-getMessage1()");
return "I am getMessage1";
}

@GetMapping("/myhello/{name}")
public String getMessage2(@PathVariable String name) {
System.out.println("HC-getMessage2()");
String msg="Hello "+name +" !!!  - I am getMessage2";
return msg;
}
```

```
@GetMapping("/myhello/{name}/{email}")
public String getMessage3(@PathVariable String name,@PathVariable String email) {
System.out.println("HC-getMessage3()");
System.out.println(name+"\t"+email);
String msg="Hello "+name +" !!!  - I am getMessage3";
return msg;
}

@GetMapping(value="/hello1/{name}",produces = "application/json" )
public Hello getMessage4(@PathVariable String name) {
System.out.println("HC-getMessage4()");
Hello hello=new Hello();
hello.setName(name);
String msg="Hello "+name +" !!!  - I am getMessage4";
hello.setMessgae(msg);
return hello;
}

@GetMapping(value = "/hello2/{name}", produces = "application/xml")
public Hello getMessage5(@PathVariable String name) {
System.out.println("HC-getMessage5()");
Hello hello=new Hello();
hello.setName(name);
String msg="Hello "+name +" !!!  - I am getMessage5";
hello.setMessgae(msg);
return hello;
}

@RequestMapping(value = "/hello3/{name}", method = RequestMethod.GET)
public Hello getMessage6(@PathVariable("name") String name) {
System.out.println("HC-getMessage6()");
Hello hello=new Hello();
hello.setName(name);
String msg="Hello "+name +" !!!  - I am getMessage6";
hello.setMessgae(msg);
return hello;
}

@RequestMapping(value = "/hello4/{name}", method =
RequestMethod.GET,produces = "application/json")
public Hello getMessage7(@PathVariable("name") String name) {
System.out.println("HC-getMessage7()");
Hello hello=new Hello();
hello.setName(name);
```

```
String msg="Hello "+name +" !!!  - I am getMessage7";
hello.setMessgae(msg);
return hello;
}
}
```

2. Hello.java

```
package com.coursecube.spring;

    import javax.xml.bind.annotation.XmlRootElement;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    @XmlRootElement
    public class Hello {
        String name;
        String messgae;
        //Constructors
        //Setters and Getters
        //toString()
    }
```

3. JLCWebConfig.java

```
package com.coursecube.spring;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@EnableWebMvc
@Configuration
@ComponentScan({ "com.coursecube.spring" })
public class JLCWebConfig   {

}
```

Lab79: Building a Complete RESR API + Hibernate

Lab79:Working Steps:

A)Setup the Project:

1. **Copy Lab78 as Lab79-Server**

2. **Make sure that Following Jars are copied to WEB-INF/lib folder**
   a. 21 Spring-5 Jars
   b. 6 Jackson Jars
   c. 20 Hibernate-5 Jars
   d. MySQL Connector Jar

3. **Update JJLCWebConfig.java**
   a. Configure DataSource
   b. Configure SessionFactory
   c. Configure HibernateTemplate
   d. Configure HibernateTxManager

B)Develop RESTful Web Service

4. **Write Resource Representation Class**
   Account.java

5. **Write Utility POJO Class**
   TxInfo.java

6. **Write DAO interface**
   AccountDAO.java

7. **Write Service Layer Components**
   AccountService.java
   AccountServiceImpl.java

8. **Create a Resource Controller called AccountController**
   AccountController.java

9. **Insert One Record in myaccounts table.**
   insert into myaccounts values(101,'SA',25000,'B-9');

10. **Start Lab79-Server app**

11. **Open the browser and hit the following URL**
    http://localhost:54321/**Lab79-Server**/myaccount/101
    http://localhost:54321/**Lab79-Server**/myaccounts

Lab79-Server: Files required

1. **AccountController.java**	2. **Account.java**
3. **AccountService.java**	4. **AccountServiceImpl.java**
5. **AccountDAO.java**	6. **AccountDAOImpl.java**
7. **TxInfo.java**	8. **JLCWebConfig.java**
9. **JLCWebAppInitializer.java**	

1. AccountController.java

```
package com.coursecube.spring;

import java.util.List;
import org.springframework.web.bind.annotation.*;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
    @RestController
    public class AccountController {

        @Autowired
        AccountService accountService;
        @PostMapping("/addAccount")
        public String addAccount(@RequestBody Account acc) {
                System.out.println("AC-addAccount()");
                accountService.addAccount(acc);
                String msg="Account Added successfully";
                return msg;
        }
        @GetMapping("/myaccount/{accno}")
        public Account getAccountByAccno(@PathVariable("accno") int accno) {
        System.out.println("AC-getAccountByAccno()" + accno);
        Account acc= accountService.getAccountByAccno(accno);
        return acc;
        }
        @PutMapping("/mydeposit")
        public void deposit(@RequestBody TxInfo txInfo) {
                System.out.println("AC-deposit()");
                accountService.deposit(txInfo);
        }
        @PutMapping("/mywithdraw")
        public void withdraw(@RequestBody TxInfo txInfo) {
                System.out.println("AC-withdraw()");
                accountService.withdraw(txInfo);
        }
```

```
    @DeleteMapping("/deleteAccount/{accno}")
    public String deleteAccount(@PathVariable  int accno) {
            System.out.println("AC-deleteAccount()");
            accountService.deleteAccount(accno);
            String msg="Account Deleted successfully";
            return msg;
    }
    @GetMapping("/myaccounts")
    public List<Account> getAllAccounts() {
    System.out.println("AC-getAllAccounts()" );
    List<Account> accList= accountService.getAllAccounts();
    return accList;
    }
}
```

2. Account.java

```
package com.coursecube.spring;

import javax.persistence.*;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@Entity
@Table(name = "myaccounts")
public class Account {
    @Id
    @Column(name = "cid")
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int accno;

    @Column(name = "atype")
    private String atype;

    @Column(name = "bcode")
    private String bcode;

    @Column(name = "bal")
    private double bal;

    //Constructors
    //Setters and Getters
    //toString()
}
```

3. AccountService.java

```
package com.coursecube.spring;

import java.util.List;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
public interface AccountService {
	public void addAccount(Account acc);
	public Account getAccountByAccno(int accno);
	public void deposit(TxInfo txInfo);
	public void withdraw(TxInfo txInfo);
	public void deleteAccount(int accno);
	public List<Account> getAllAccounts();
}
```

4. AccountServiceImpl.java

```
package com.coursecube.spring;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@Service
@Transactional
public class AccountServiceImpl  implements AccountService{

	@Autowired
	AccountDAO accountDAO;

	@Override
	public void addAccount(Account acc) {
		System.out.println("ADAO-addAccount()");
		accountDAO.addAccount(acc);
	}

	@Override
	public Account getAccountByAccno(int accno) {
		System.out.println("ADAO-getAccountByAccno()");
		Account acc=accountDAO.getAccountByAccno(accno);
		return acc;
	}
```

```
	@Override
	public void deposit(TxInfo txInfo) {
		System.out.println("ADAO-deposit()");
		Account acc=accountDAO.getAccountByAccno(txInfo.getAccno());
		if(acc!=null) {
			double updatedBal=acc.getBal()+txInfo.getAmount();
			acc.setBal(updatedBal);
			accountDAO.updateAccount(acc);
		}
	}

	@Override
	public void withdraw(TxInfo txInfo) {
		System.out.println("ADAO-withdraw()");
		Account acc=accountDAO.getAccountByAccno(txInfo.getAccno());
		if(acc!=null) {
			double updatedBal=acc.getBal()-txInfo.getAmount();
			acc.setBal(updatedBal);
			accountDAO.updateAccount(acc);
		}
	}

	@Override
	public void deleteAccount(int accno) {
		System.out.println("ADAO-deleteAccount()");
		accountDAO.deleteAccount(accno);

	}
	@Override
	public List<Account> getAllAccounts() {
		System.out.println("ADAO-getAllAccounts()");
		return accountDAO.getAllAccounts();
	}
}
```

5. AccountDAO.java

```
package com.coursecube.spring;

import java.util.List;

public interface AccountDAO {
	public void addAccount(Account acc);
	public Account getAccountByAccno(int accno);
	public void updateAccount(Account acc);
	public void deleteAccount(int accno);
	public List<Account> getAllAccounts();
}
```

6. AccountDAOImpl.java

```
package com.coursecube.spring;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Repository;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@Repository
public class AccountDAOImpl  implements AccountDAO{

	@Autowired
	HibernateTemplate htemp;

	@Override
	public void addAccount(Account acc) {
		System.out.println("ADAO-addAccount()");
		htemp.save(acc);
	}

	@Override
	public Account getAccountByAccno(int accno) {
		System.out.println("ADAO-getAccountByAccno()");
		Account acc=htemp.get(Account.class,accno);
		return acc;
	}

	@Override
	public void updateAccount(Account acc) {
		htemp.update(acc);
	}

	@Override
	public void deleteAccount(int accno) {
		Account acc=htemp.get(Account.class,accno);
		if(acc!=null)
		    htemp.delete(acc);
	}

	@Override
	public List<Account> getAllAccounts() {
		System.out.println("ADAO-getAccountByAccno()");
		return htemp.loadAll(Account.class);
	}
}
```

7. TxInfo.java

```
package com.coursecube.spring;

    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class TxInfo {
        private int accno;
        private double amount;

        //Constructors
        //Setters and Getters

    }
```

8. JLCWebConfig.java

```
package com.coursecube.spring;

import java.util.Properties;
import javax.sql.DataSource;
import org.hibernate.SessionFactory;
import org.springframework.context.annotation.*;
import org.springframework.jdbc.datasource.DriverManagerDataSource;
import org.springframework.orm.hibernate5.*;
import org.springframework.transaction.annotation.EnableTransactionManagement;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
@EnableWebMvc
@Configuration
@EnableTransactionManagement
@ComponentScan({ "com.coursecube.spring" })
public class JLCWebConfig   {

        @Bean
        public DriverManagerDataSource getDS() {
        DriverManagerDataSource ds=new DriverManagerDataSource();
        ds.setDriverClassName("com.mysql.cj.jdbc.Driver");
        ds.setUrl("jdbc:mysql://localhost:3306/myspringdb");
        ds.setUsername("root");
        ds.setPassword("srinivas");
        return ds;
```

```
	}

	@Bean
	public LocalSessionFactoryBean sessionFactory(DataSource dataSource) {

		LocalSessionFactoryBean factory = new LocalSessionFactoryBean();

		Properties props = new Properties();
		props.put("hibernate.show_sql", "true");
		props.put("hibernate.dialect", "org.hibernate.dialect.MySQL8Dialect");
		props.put("hibernate.hbm2ddl.auto", "update");
		props.put("hibernate.transaction.factory_class",
				"org.hibernate.transaction.JDBCTransactionFactory");
		factory.setHibernateProperties(props); //1
		factory.setDataSource(dataSource); //2
		factory.setPackagesToScan("com.coursecube.spring"); //3

	return factory;
	}

	@Bean
	public HibernateTemplate hTemp(SessionFactory sessionFactory) {
	return new HibernateTemplate(sessionFactory);
	}

	@Bean
	public HibernateTransactionManager txManager(SessionFactory sessionFactory) {
	return new HibernateTransactionManager(sessionFactory);
	}
}
```

Lab79-Client: REST API Client

Lab79-Client:Working Steps:

A)Setup the Project:

1. **Copy Lab79-Server as Lab79-Client**

2. **Delete the following files.**
   a) AccountController.java
   b) AccountDAO.java
   c) AccountDAOImpl.java
   d) AccountService.java
   e) AccountServiceImpl.java
   f) JLCWebConfig.java
   g) JLCWebAppInitializer.java

3. **Write Test Programs to Test REST API**
   Test1.java
   Test2.java
   Test3.java
   Test4.java
   Test5.java
   Test6.java
   Test7.java

4. **Run Test Programs One by One.**
   a. While Running Test Programs make sure that Lab79-Server app is running.

Lab79-Clinet: Files required

1. Account.java	2. TxInfo.java
3. Test1.java	4. Test2.java
5. Test3.java	6. Test4.java
7. Test5.java	8. Test6.java
9. Test7.java	

1. Account.java

```
package com.coursecube.spring;
/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
public class Account {
	private int accno;
	private String atype;
	private String bcode;
	private double balance;

	//Constructors
	//Setters and Getters
	//toString()
}
```

2. TxInfo.java

```
package com.coursecube.spring;

/*
* @Author : Srinivas Dande
* @company : Java Learning Center
* */
public class TxInfo {
private int accno;
private double amount;

//Constructors
//Setters and Getters
}
```

3. Test1.java

```
package com.coursecube.spring;

    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test1 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/addAccount";
        Account acc=new Account("SA","B-99",75000);
        String msg=restTemp.postForObject(URL, acc, String.class);
        System.out.println(msg);
        System.out.println("Done");
    }
    }
```

4. Test2.java

```
package com.coursecube.spring;

    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test2 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/myaccount/101";
        Account acc=restTemp.getForObject(URL, Account.class) ;
        System.out.println(acc);
        System.out.println("Done");
    }
    }
```

5. Test3.java

```
package com.coursecube.spring;

    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test3 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/mydeposit";
        TxInfo txInfo=new TxInfo(101, 5000);
        restTemp.put(URL,txInfo) ;
        System.out.println("Done");
    }
    }
```

6. Test4.java

```
package com.coursecube.spring;

    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test4 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/mywithdraw";
        TxInfo txInfo=new TxInfo(102, 10000);
        restTemp.put(URL,txInfo) ;
        System.out.println("Done");
    }
    }
```

7. Test5.java

```
package com.coursecube.spring;

    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test5 {
    public static void main(String[] args) {
        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/deleteAccount/103";
        restTemp.delete(URL);
        System.out.println("Done");
    }
    }
```

8. Test6.java

```
package com.coursecube.spring;

    import java.util.*;
    import org.springframework.web.client.RestTemplate;
    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test6 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/myaccounts";

        List<Map<String, String>> accList=restTemp.getForObject(URL, List.class);
        for(Map<String, String> mymap:accList) {
                System.out.println(mymap);
        }
        System.out.println("Done");
    }
    }
```

9. Test7.java

```
package com.coursecube.spring;

    import java.util.*;
    import org.springframework.http.HttpHeaders;
    import org.springframework.http.ResponseEntity;
    import org.springframework.web.client.RestTemplate;

    /*
    * @Author : Srinivas Dande
    * @company : Java Learning Center
    * */
    public class Test7 {
    public static void main(String[] args) {

        RestTemplate restTemp=new RestTemplate();
        String URL="http://localhost:5050/Lab79-Server/myaccounts";

        ResponseEntity<List>  respEntity=  restTemp.getForEntity(URL, List.class);
        System.out.println(respEntity.getStatusCode());
        System.out.println(respEntity.getStatusCodeValue());

        HttpHeaders myheaders= respEntity.getHeaders();
        System.out.println(myheaders);
        List<Map<String, String>> accList=respEntity.getBody();
        for(Map<String, String> mymap:accList) {
                System.out.println(mymap);
        }
        System.out.println("Done");
    }
    }
```