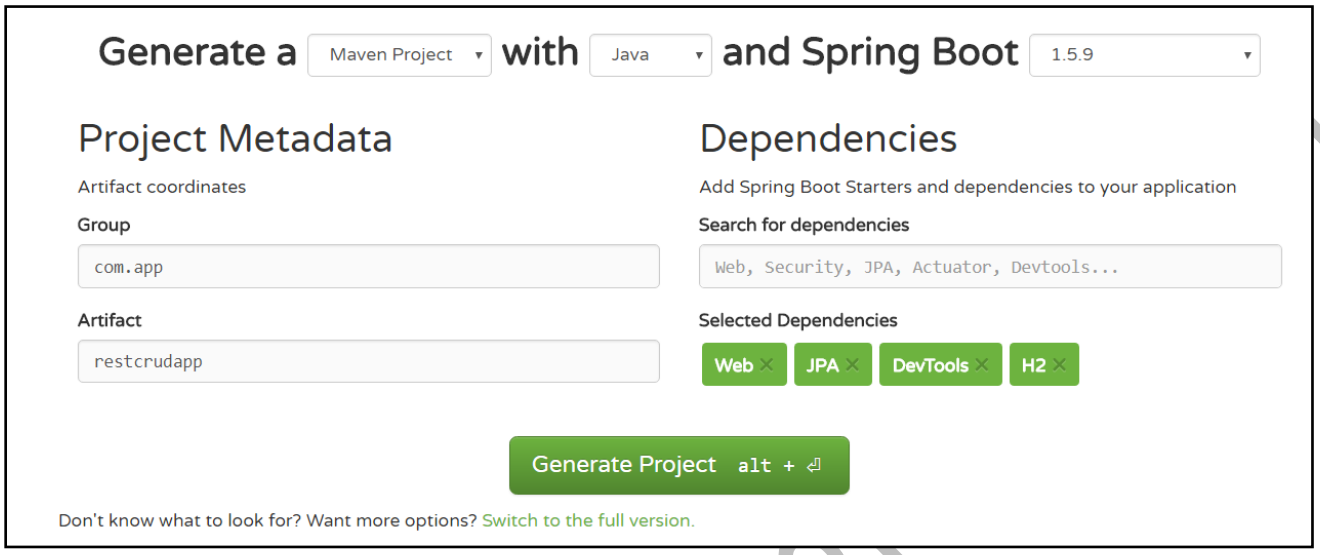
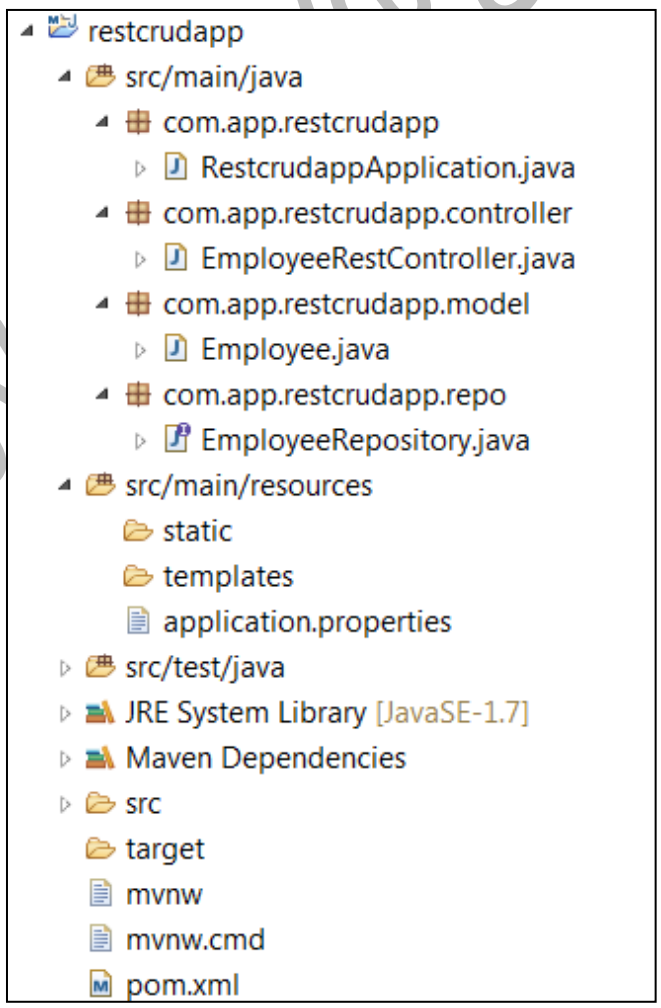
Spring Boot Rest and Crud Operations

EMBEDDED TOMCAT SERVER AND EMBEDDED H2 DATABASE WITH DEV TOOLS

start.spring.io screen:



Eclipse Folder Structure:



1.pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>

	<groupId>com.app</groupId>
	<artifactId>restcrudapp</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<packaging>jar</packaging>

	<name>restcrudapp</name>
	<description>Demo project for Spring Boot</description>

	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>1.5.9.RELEASE</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>

	<properties>
		<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
		<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
		<java.version>1.7</java.version>
	</properties>

	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-jpa</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>
```

```
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-devtools</artifactId>
			<scope>runtime</scope>
		</dependency>
		<dependency>
			<groupId>com.h2database</groupId>
			<artifactId>h2</artifactId>
			<scope>runtime</scope>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>


</project>
```

2.application.properties:

```
server.port=2018
```

3. starter class:

```
package com.app.restcrudapp;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.data.jpa.repository.config.EnableJpaAuditing;
```

```
@SpringBootApplication
@EnableJpaAuditing
public class RestcrudappApplication {
	public static void main(String[] args) {
		SpringApplication.run(RestcrudappApplication.class, args);
	}
}
```

4.Model class:

```
package com.app.restcrudapp.model;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.EntityListeners;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.Digits;

import org.hibernate.annotations.GenericGenerator;
import org.hibernate.validator.constraints.NotBlank;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

@Entity
@Table(name="emptab")
@EntityListeners(AuditingEntityListener.class)
public class Employee {

	@Id
	@GeneratedValue(generator="empgen")
	@GenericGenerator(name="empgen",strategy="increment")
	@Column(name="eid")
	private Long empId;

	@Column(name="ename")
	@NotBlank
```

```
    private String empName;

    @Column(name="esal")
    @Digits(integer=10,fraction=3)
    private Double empSal;

    @Column(name="ecdate")
    @CreatedDate
    private Date createdDate;

    @Column(name="eudate")
    @LastModifiedDate
    private Date updatedDate;

    public Employee() {
    }

    public Long getEmpId() {
        return empId;
    }
    public void setEmpId(Long empId) {
        this.empId = empId;
    }
    public String getEmpName() {
        return empName;
    }
    public void setEmpName(String empName) {
        this.empName = empName;
    }
    public Double getEmpSal() {
        return empSal;
    }
    public void setEmpSal(Double empSal) {
        this.empSal = empSal;
    }
    public Date getCreatedDate() {
        return createdDate;
    }
    public void setCreatedDate(Date createdDate) {
        this.createdDate = createdDate;
    }
```

```
	public Date getUpdatedDate() {
		return updatedDate;
	}
	public void setUpdatedDate(Date updatedDate) {
		this.updatedDate = updatedDate;
	}

}
```

5.Repository Interface:

```
package com.app.restcrudapp.repo;
import org.springframework.data.repository.CrudRepository;
import com.app.restcrudapp.model.Employee;
public interface EmployeeRepository extends CrudRepository<Employee,Long>{ }
```

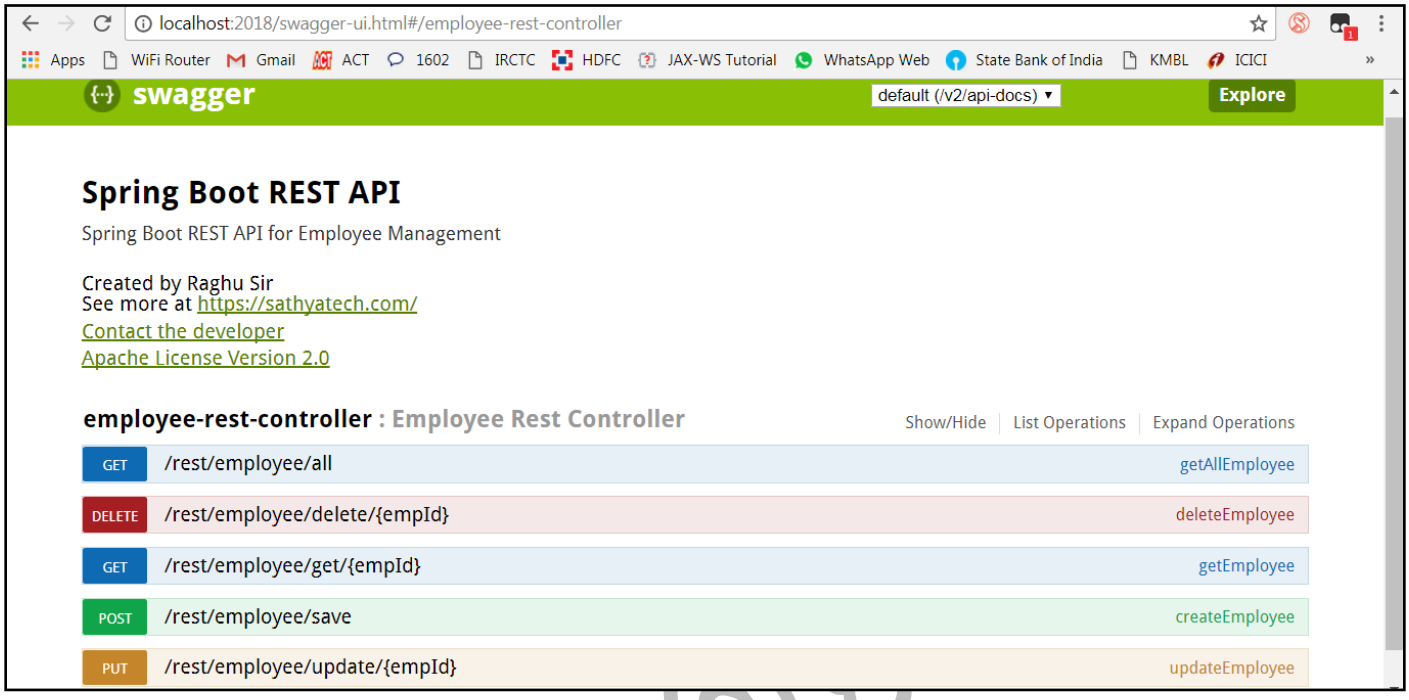
6.Rest Controller:

```
package com.app.restcrudapp.controller;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.Errors;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.app.restcrudapp.model.Employee;
import com.app.restcrudapp.repo.EmployeeRepository;

@RestController
@RequestMapping("/employee")
public class EmployeeRestController {
```

```
	@Autowired
	private EmployeeRepository repo;

	@PostMapping("/save")
	public ResponseEntity<Object> createEmployee(@Valid @RequestBody Employee
employee,Errors errors){

		Object response=null;
		if(errors.hasErrors()) {
			response=errors.getAllErrors();
			return ResponseEntity.badRequest().body(response);
		} else {
			employee=repo.save(employee);
			response="Saved with :"+employee.getEmpId();
			return ResponseEntity.ok(response);
		}
	}

	@GetMapping("/get/{empId}")
	public ResponseEntity<Object> getEmployee(@PathVariable Long empId){
		Object response=null;
		Employee emp=repo.findOne(empId);
		if(emp==null){
			response="Employee("+empId+") Not found";
		}
		else {
			response=emp;
		}
		return ResponseEntity.ok(response);
	}

	@DeleteMapping("/delete/{empId}")
	public ResponseEntity<Object> deleteEmployee(@PathVariable Long empId){
		Object response=null;
		Employee emp=repo.findOne(empId);
		if(emp==null){
			response="Employee("+empId+") Not found";
		}else{
			repo.delete(empId);
			response="Employee("+empId+") Deleted";
		}
```

```
		return ResponseEntity.ok(response);
	}

	@PutMapping("/update/{empId}")
	public ResponseEntity<Object> updateEmployee(@PathVariable Long empId,
			@Valid @RequestBody Employee employee,Errors errors){
		Object response=null;

		if(errors.hasErrors()) {
			response=errors.getAllErrors();
			return ResponseEntity.badRequest().body(response);
		}

		Employee emp=repo.findOne(empId);

		if(emp==null){
			response="Employee("+empId+") Not found";
		}else{
			emp.setEmpName(employee.getEmpName());
			emp.setEmpSal(employee.getEmpSal());
			empId=repo.save(emp).getEmpId();
			response="Employee("+empId+") Updated";
		}
		return ResponseEntity.ok(response);

	}

	@GetMapping("/all")
	public ResponseEntity<Object> getAllEmployee(){
		Object response=null;
		Iterable<Employee> emps=repo.findAll();
		if(emps==null || emps.toString().equals("[]")){
			response="Employees not found";
		}else{
			response=emps;
		}
		return ResponseEntity.ok(response);
	}

}
```

Swagger UI : http://localhost:2018/swagger-ui.html



pom.xml (to be added code)

```
<dependency>
	<groupId>io.springfox</groupId>
	<artifactId>springfox-swagger2</artifactId>
	<version>2.6.1</version>
	<scope>compile</scope>
  </dependency>

  <dependency>
	<groupId>io.springfox</groupId>
	<artifactId>springfox-swagger-ui</artifactId>
	<version>2.6.1</version>
	<scope>compile</scope>
  </dependency>
```

Swagger Configuration class:

```
package com.app.restcrudapp.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
```

```
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.service.ApiInfo;
import springfox.documentation.service.Contact;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;
import static springfox.documentation.builders.PathSelectors.regex;


@Configuration
@EnableSwagger2
public class SwaggerConfig {
    @Bean
    public Docket productApi() {
        return new Docket(DocumentationType.SWAGGER_2)
                .select()

.apis(RequestHandlerSelectors.basePackage("com.app.restcrudapp"))
                .paths(regex("/rest.*"))
                .build()
                .apiInfo(metaData());
    }

    private ApiInfo metaData() {
        ApiInfo apiInfo = new ApiInfo(
                "Spring Boot REST API",
                "Spring Boot REST API for Employee Management",
                "1.0",
                "Terms of service",
              new Contact("Raghu Sir", "http://www.sathyatech.com/",
             "javabyraghu@gmail.com"),
                "Apache License Version 2.0",
                "https://www.apache.org/licenses/LICENSE-2.0");
        return apiInfo;
    }
}
```