```
==================================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
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.0.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ameya</groupId>
    <artifactId>007-RequestValidation</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>007-RequestValidation</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</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-validation</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>

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

</project>
==================================application.properties==============================
server.port=9001
logging.level.web=trace

spring.datasource.url=jdbc:mysql://localhost:3306/nineleapsdb
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.hibernate.ddl-auto=update
```

```
==================================Person.java==========================
package com.ameya.models;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotBlank;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;

@Entity
@Table(name="aj_person")
public class Person {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private long id;
    @NotNull
    @Size(min = 2, message="First Name Must be atleast 2 Characters")
    private String firstName;
    @NotNull
    @Size(min = 2, message="Last Name Must be atleast 2 Characters")
    private String lastName;
    @NotNull
    @NotBlank
    private String emailId;
    @NotNull
    @Size(min = 2, message="Passport Number  Must be atleast 2 Characters")
    private String passportNumber;
    public Person() {

    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
    public String getPassportNumber() {
        return passportNumber;
    }
    public void setPassportNumber(String passportNumber) {
        this.passportNumber = passportNumber;
    }

}
==================================PersonDAO==============================
package com.ameya.daos;

import org.springframework.data.repository.CrudRepository;

import com.ameya.models.Person;

```

```
public interface PersonDAO extends CrudRepository<Person, Long> {

}
==================================ResourceNotFoundException.java====================
package com.ameya.exceptions;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value=HttpStatus.NOT_FOUND)
public class ResourceNotFoundException extends RuntimeException {

    private static final long serialVersionUID = 1L;

    public ResourceNotFoundException(String message) {
        super(message);
    }


}
==============================ErrorDetails.java==============================
package com.ameya.exceptions;

import java.util.Date;

public class ErrorDetails {

    private Date timeStamp;
    private String message;
    private String details;
    public ErrorDetails(Date timeStamp, String message, String details) {
        super();
        this.timeStamp = timeStamp;
        this.message = message;
        this.details = details;
    }
    public Date getTimeStamp() {
        return timeStamp;
    }
    public String getMessage() {
        return message;
    }
    public String getDetails() {
        return details;
    }

}
==================================PersonController.java=========================
package com.ameya.controllers;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
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.ameya.daos.PersonDAO;
import com.ameya.exceptions.ResourceNotFoundException;
import com.ameya.models.Person;

@RestController
```

```
@RequestMapping("/persons")
public class PersonController {
    @Autowired
    private PersonDAO personDao;

    @GetMapping(path="/getAll")
    public List<Person> getAllPersons(){
        List<Person> persons=new ArrayList<>();
        personDao.findAll().forEach(persons::add);
        return persons;
    }
    @GetMapping(path="/{id}")
    public ResponseEntity<Person> getPersonById(@PathVariable("id") long id){
        Person person=personDao.findById(id).orElseThrow(()->
            new ResourceNotFoundException("Person Not Found ["+id+"]")
                );
        return ResponseEntity.ok().body(person);
    }
    @PostMapping("/create")
    public Person createPerson(@Valid @RequestBody Person person) {
        return personDao.save(person);
    }
    @PutMapping("/update/{id}")
    public ResponseEntity<Person> updatePerson(@PathVariable("id") long id,
            @Valid @RequestBody Person person) {
        Person per=personDao.findById(id).orElseThrow(()->
        new ResourceNotFoundException("Person Not Found ["+id+"]")
            );
        per.setEmailId(person.getEmailId());
        per.setFirstName(person.getFirstName());
        per.setLastName(person.getLastName());
        per.setPassportNumber(person.getPassportNumber());
        Person p=personDao.save(per);
        return ResponseEntity.ok(p);
    }
    @DeleteMapping(path="/delete/{id}")
    public Map<String,Boolean> deletePerson(@PathVariable("id") long id){
        Person per=personDao.findById(id).orElseThrow(()->
        new ResourceNotFoundException("Person Not Found ["+id+"]")
            );
        personDao.delete(per);
        Map<String,Boolean> response=new HashMap<>();
        response.put("deleted successfully",Boolean.TRUE);
        return response;
    }
}
==================================GlobalExceptionHandler.java=======================
==
package com.ameya.controllers.advices;

import java.util.Date;

import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.MethodArgumentNotValidException;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.context.request.WebRequest;
import
org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;

import com.ameya.exceptions.ErrorDetails;
import com.ameya.exceptions.ResourceNotFoundException;

@ControllerAdvice
public class GlobalExceptionHandler extends ResponseEntityExceptionHandler{

    @ExceptionHandler(ResourceNotFoundException.class)
    public ResponseEntity<?> resourceNotFoundException(ResourceNotFoundException ex,
            WebRequest request){
        ErrorDetails errorDetails=new ErrorDetails(new Date(),
                ex.getMessage(), request.getDescription(false));
```

```
            return new ResponseEntity<>(errorDetails,HttpStatus.NOT_FOUND);
        }

        @Override
        protected ResponseEntity<Object>
        handleMethodArgumentNotValid(MethodArgumentNotValidException ex,
                HttpHeaders headers, HttpStatus status, WebRequest request) {
            ErrorDetails errorDetails=new ErrorDetails(new Date(),"Validation Failed"
                    , ex.getBindingResult().toString());
            return new ResponseEntity<Object>(errorDetails,HttpStatus.BAD_REQUEST);
        }

    }
    ======================================================

    http://localhost:9001/persons/create
    {
        "firstName":"A",
        "lastName":"Joshi",
        "emailId":"",
        "passportNumber":"ABCD1234"
    }
```