Se crea el proyecto en Spring -> https://start.spring.io/

<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-data-rest</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-validation</artifactId>  
 </dependency>  
  
</dependencies>

Luego se configura application.properties

#crea las entidades en la DB create,create-drop,validate, update, none  
spring.data.rest.base-path=/api  
spring.jpa.hibernate.ddl-auto=update  
spring.datasource.url=jdbc:mysql://localhost:3306/test\_escuela  
spring.datasource.username=root  
spring.datasource.password=Sertec2580$  
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver  
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.format\_sql=true

Lo resaltado en gris es lo que se agrego!

La entidad

@Data  
@AllArgsConstructor  
@NoArgsConstructor  
@Entity  
@Table(name = "cursos")  
public class Curso {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 @Column(name = "title")  
 @NotBlank  
 private String title;  
 @Column(name = "price")  
 @NotNull  
 @Positive  
 private Float price;  
}

El repository

import com.sertec.tallerapires01.entity.Curso;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.rest.core.annotation.RepositoryRestResource;  
  
  
@RepositoryRestResource(path = "cursos",collectionResourceRel = "cursos")  
public interface CursoRepository extends JpaRepository<Curso,Long> {  
}

lo resaltado es lo nuevo y es de data-rest

esto es el control de excepciones

import jakarta.validation.ConstraintViolationException;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ProblemDetail;  
import org.springframework.web.bind.annotation.ExceptionHandler;  
import org.springframework.web.bind.annotation.ResponseStatus;  
import org.springframework.web.bind.annotation.RestControllerAdvice;  
  
@RestControllerAdvice  
public class ExceptionHandling {  
 @ResponseStatus(HttpStatus.*BAD\_REQUEST*)  
 @ExceptionHandler(ConstraintViolationException.class)  
 public ProblemDetail handerConstraintViolationException(){  
  
 return ProblemDetail.*forStatusAndDetail*(HttpStatus.*BAD\_REQUEST*,"La solicitud http presenta errores");  
 }  
}

solicitud desde Inmsonia

GET

POST

<http://localhost:8080/api/cursos>

PUT

DELETE

El 1 representa el id del curso a modificar o eliminar

<http://localhost:8080/api/cursos/1>