



Facultad: Ingeniería y Sistemas

Módulo: DESARROLLO DE APLICACIONES WEB

Docente: CARLOS BORIS MARTINEZ CALZADIA

Grupo: 01

Alumno: Bonilla Rivera José Giovanni. BR100720

Actividad: Laboratorio

Modelos

Camisa

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class Camisa {

    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    no usages
    private Integer numero;
    no usages
    private String color;
}

Jaxxer.Hikari.HikariDataSource : HikariPool-1 - Start completed.
hibernate.dialect.Dialect : hibernate.dialect.PostgreSQL95Dialect
e.t.j.p.i.JpaPlatformInitiator : hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform
localContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
s.o.p.t.s.JetAccessTokenConverter : Unable to create an RSA verifier from verifierKey (ignoreable if using MAC)
s.web.DefaultSecurityFilterChain : Will secure Url [Ant [pattern='/oauth/token'], Ant [pattern='/oauth/token_key'], Ant [pattern='/oauth/check_token']] with [org.s
s.web.DefaultSecurityFilterChain : Will secure org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerConfiguration$NotAuthRequest
s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter@6d7d0946, org.springframework.sec
s.w.embedded.Tomcat.TomcatWebServer : Started MediasBackendApplication in 4.802 seconds (JVM running for 5.206)
c.c.c.Tomcat.[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

Jugador

```
import javax.persistence.*;

@Entity
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class Jugador {

    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    no usages
    private String nombre;
    no usages
    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "camisa_id", referencedColumnName = "id")
    private Camisa camisa;
    no usages
    @ManyToOne
    @JoinColumn(name = "posicion_id", referencedColumnName = "id")
    private Posicion posicion;
}

Jaxxer.Hikari.HikariDataSource : HikariPool-1 - Start completed.
hibernate.dialect.Dialect : hibernate.dialect.PostgreSQL95Dialect
e.t.j.p.i.JpaPlatformInitiator : hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform
localContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
s.o.p.t.s.JetAccessTokenConverter : Unable to create an RSA verifier from verifierKey (ignoreable if using MAC)
s.web.DefaultSecurityFilterChain : Will secure Url [Ant [pattern='/oauth/token'], Ant [pattern='/oauth/token_key'], Ant [pattern='/oauth/check_token']] with [org.s
s.web.DefaultSecurityFilterChain : Will secure org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerConfiguration$NotAuthRequest
s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter@6d7d0946, org.springframework.sec
s.w.embedded.Tomcat.TomcatWebServer : Started MediasBackendApplication in 4.802 seconds (JVM running for 5.205)
c.c.c.Tomcat.[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

Posicion

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
@Getter
@Setter
@NoArgsConstructor
@EqualsAndHashCode
public class Posicion {
    // no se usara
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    // no se usara
    private String nombre;
}
```

```
jakarta.persistence.EntityManagerFactory : HikariPool-1 - Start completed.
hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.PostgreSQL95Dialect
e.t.j.p.i.JpaPlatformInitiator : HHH000490: Using JpaPlatform implementation: [org.hibernate.engine.transaction.jta
LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
s.o.p.t.s.JwtAccessTokenConverter : Unable to create an RSA verifier from verifierKey (ignoreable if using MAC)
s.web.DefaultSecurityFilterChain : Will secure Or [Ant [pattern= '/oauth/token'], Ant [pattern= '/oauth/token_key'], An
s.web.DefaultSecurityFilterChain : Will secure org.springframework.security.oauth2.config.annotation.web.configuratio
s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEmco
j.w.embedded.TomcatTomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
nitcode.WedispBackendApplication : Started WedispBackendApplication in 4.802 seconds (JVM running for 5.205)
s.c.o.Tomcat.[localhost/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

El recorte se guardó en el Portapapeles
Selecciona esta opción para marcar y compartir la imagen

DTOS

Camisa

```
import lombok.NoArgsConstructor;
import java.validation.constraints.NotNull;

import javax.persistence.*;

@Data
@NoArgsConstructor
@EqualsAndHashCode
@EqualsAndHashCode.Include
public class CamisaDTO {

    @NotNull
    private Integer id;

    @NotNull
    private Integer numero;

    @NotNull
    private String color;
}

1. Zaxxer Hikari HikariDataSource : HikariPool-1 - Start completed.
2. org.hibernate.dialect.Dialect : org.hibernate.dialect.PostgreSQL95Dialect
3. org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform : org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform
4. org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
5. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Unable to create an RSA verifier from verifony (ignoreable if using MAC)
6. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Will secure the [Ant [pattern="/oauth/token"], Ant [pattern="/oauth/token.key"], Ant [pattern="/oauth/check.token"]] with [org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerConfiguration$OAuthRequest
7. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Will secure any request with [org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerConfiguration$OAuthRequest
8. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Tomcat started on port(s): 8080 (http) with context path ''
9. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Started MediaplatformApplication in 4.802 seconds (JVM running for 5.205)
10. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Initializing Spring DispatcherServlet 'dispatcherServlet'
11. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Initializing Servlet 'dispatcherServlet'
12. org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter : Completed initialization in 1 ms
```

Jugador

```
package com.mitocode.dto;

import lombok.*;
import java.validation.constraints.NotNull;

import javax.persistence.*;

@Data
@NoArgsConstructor
@EqualsAndHashCode
@EqualsAndHashCode.Include
public class JugadorDTO {

    @NotNull
    private Integer id;

    @NotNull
    private String nombre;

    @NotNull
    private CamisaDTO camisa;

    @NotNull
    private PosicionDTO posicion;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }
}
```

```
private PosicionDTO posicion;

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

no usages
public String getNombre() {
    return nombre;
}

no usages
public void setNombre(String nombre) {
    this.nombre = nombre;
}

no usages
public CamisaDTO getCamisa() {
    return camisa;
}

no usages
public void setCamisa(CamisaDTO camisa) {
    this.camisa = camisa;
}

no usages
public PosicionDTO getPosicion() {
    return posicion;
}

no usages
public void setPosicion(PosicionDTO posicion) {
    this.posicion = posicion;
}
}
```

Posicion

```
package com.mitocode.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.EqualsAndHashCode;
import lombok.NoArgsConstructor;
import lombok.NoArgsConstructor;

import javax.validation.constraints.NotNull;

@Data
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(onlyExplicitlyIncluded = true)
public class PosicionDTO {

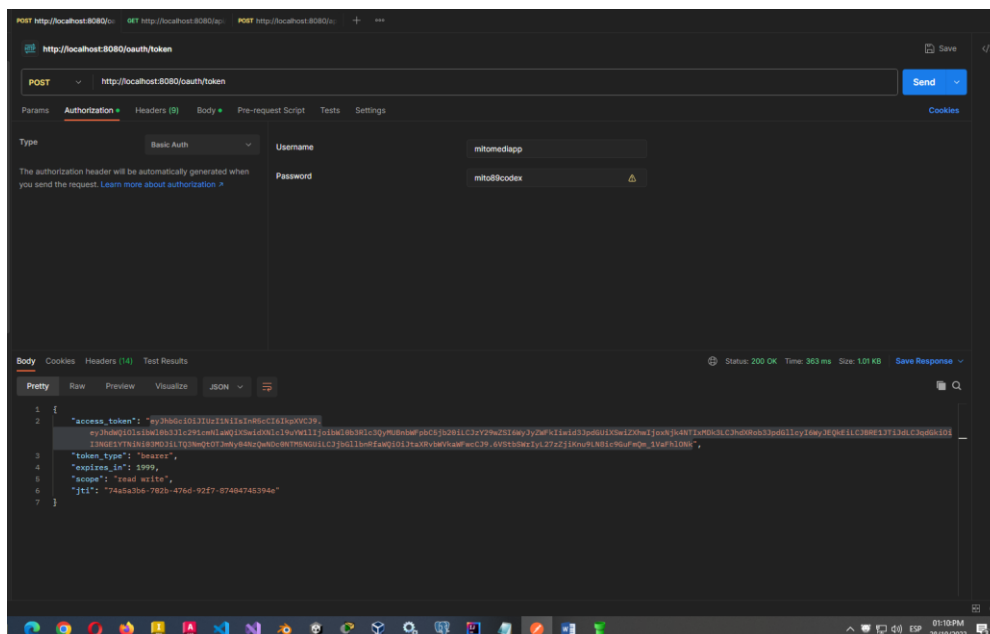
    no usages
    @EqualsAndHashCode.Include
    private Integer id;

    no usages
    @NotNull
    private String nombre;
}
```

El controlador

The image shows a screenshot of an IDE (IntelliJ IDEA) with the file 'JugadorController.java' open. The code is written in Java and includes the following imports:
`import java.net.URI;
import java.util.List;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;
import javax.validation.Valid;
import java.util.List;
import java.util.stream.Collectors;`
The code defines a `@RestController` with `@RequestMapping("/api/jugadores")`. The class `JugadorController` contains two methods:
1. `findAll()`: A `GET` method that returns a `ResponseEntity<List<JugadorDTO>>`. It uses `jugadorRepository.findAll().stream().map(p -> mapper.map(p, JugadorDTO.class)).collect(Collectors.toList());` to retrieve and map the data.
2. `save()`: A `POST` method that takes a `JugadorDTO` as input and returns a `ResponseEntity<Void>`. It uses `@Valid` and `@RequestBody` annotations. The method uses `jugadorRepository.save(mapper.map(dto, Jugador.class))` to save the entity and `ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(p.getId()).toUri()` to generate the URI for the response.
The IDE interface shows the file explorer on the left with a tree view of the project structure. The bottom status bar displays the time as 28:14, CRLF, UTF-8, 4 spaces, and the date 28/10/2023.

Para obtener el JWT



Para hacer el post

