CREO PROGETTO ED INSERISCO LE DIPENDENZE

DATA JPA - DEVTOOLS - WEB - THYMELEAF - LOMBOK

DOPO QUELLI DEL DB SCELTO

AVVIO E MODIFICO LE APPLICATION.PROPERTIES

CREO ENTITIES DEL PROGETTO, RICORDARSI @ENTITY E @TABLE SU PK VA MESSO @ID E E GENERARE GETTERS, SETTERS E CONSTRUCTOR

REPOSITORY CON ENTITYREPOSITORY (INTERFACE) CHE ESTENDE JPAREPOSITORY CHE SERVE AD OTTENERE METODI CRUD

< ENTITY, TIPO PK >
STRING, INTEGER

DENTRO METTO SOLO NOME METODI E PARAMETRI

KEYWORDS CRUD

SOPRA @REPOSITORY

@COMPONENT

CREO SERVICE ENTITYSERVICE, CREO OGGETTO DI TIPO ENTITYREPOSITORY E GLI METTO IL SUO COSTRUTTORE CON IN CIMA @AUTOWIRED

CI POSSO METTERE DEI METODI TIPO GETENTITIES CON UNA LIST E POI RETURN USANDO L'OGGETTO CREATO . FINDALL()

```java
public List<Player> getPlayersByTeamAndPosition(String team, String position){
    return playerRepository.findAll().stream()
        .filter(player -> team.equals(player.getTeam()) && position.equals(player.getPos()))
        .collect(Collectors.toList());
}
```

RESTITUISCE UNA LISTA DI PLAYER CHE GIOCANO IN UN CERTO TEAM IN UN DATO RUOLO

FINDALL RECUPERA TUTTI GLI OGGETTI

STREAM SI USA PER FARE UNA SEQUENZA SU CUI APPLICARE CERTE OPERAZIONI

FILTER SI USA SOLO PER SELEZIONARE ELEMENTI CHE SODDISFANO DEI CRITERI SPECIFICI

PLAYER —> INDICA OGNI ELEMENTO PLAYER DELLA FUNZIONE

TEAM.EQUALS CONFRONTA TEAM GIOCATORE CORRENTE CON QUELLO PASSATO IN INGRESSO

POSITION.EQUALS CONFRONTA POSIZIONE PASSATA CON QUELLA DEL GIOCATORE CORRENTE

COLLECT RACCOGLIE ELEMENTI FILTRATI IN UNA LISTA

```java
    public Player addPlayer(Player player){
        playerRepository.save(player);
        return player;
    }

    public Player updatePlayer(Player updatedPlayer){
        Optional<Player> existingPlayer = playerRepository.findByName(updatedPlayer.getName());

        if (existingPlayer.isPresent()){
            Player playerToUpdate = existingPlayer.get();
            playerToUpdate.setName(updatedPlayer.getName());
            playerToUpdate.setTeam(updatedPlayer.getTeam());
            playerToUpdate.setPos(updatedPlayer.getPos());
            playerToUpdate.setNation(updatedPlayer.getNation());

            playerRepository.save(playerToUpdate);
            return playerToUpdate;
        }
        return null;
    }
}
```

SAVE SALVA PLAYER NEL DATABASE CON JPA - FA SIA INSERT CHE UPDATE

CREO ENTITYCONTROLLER CHE SI OCCUPA DELLA GESTIONE DI CHIAMATE HTTP, PER FARLO BISOGNA INSERIRE ANNOTAZIONE @RESTCONTROLLER COSÌ I METODI RITORNANO UN DOMINIO E NON UNA VISTA

INSERISCO ANCHE @REQUESTMAPPING CHE INDICA URL DI PARTENZA

PER OGNI METODO VA MESSO @GETMAPPING @PUTMAPPING CHE INDICA URL DEL METODO DA AGGIUNGERE A QUELLO BASE

UN METODO ACCETTA PARAMETRI TRAMITE @REQUESTPARAM NELLA SEZIONE PARAMETRI

```java
@PostMapping
public ResponseEntity<Player> addPlayer(@RequestBody Player player) {
    Player createdPlayer = playerService.addPlayer(player);
    return new ResponseEntity<>(createdPlayer, HttpStatus.CREATED);
}

@PutMapping
public ResponseEntity<Player> updatePlayer(@RequestBody Player updatedPlayer) {
    Player resultPlayer = playerService.updatePlayer(updatedPlayer);
    if (resultPlayer != null) {
        return new ResponseEntity<>(resultPlayer, HttpStatus.OK);
    } else {
        return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
}
@DeleteMapping("/{playerName}")
public ResponseEntity<String> deletePlayer(@PathVariable String playerName) {
    playerService.deletePlayer(playerName);
    return new ResponseEntity<>(body:"Player deleted successfully", HttpStatus.OK);
}
```

USA METODO ADDPLAYER DEL SERVICE

SEMPRE METODI SERVICE

RESPONSE ENTITY CLASSE JAVA CHE RAPPRESENTA INTERA RISPOSTA HTTP CON STATUS CODE 200, 400..... EVENTUALI HEADER ED IL CORPO DELLA RISPOSTA

NEL BODY AD ESEMPIO MANDA PLAYER APPENA AGGIUNTO CON MESSAGGIO HTTP 201 DI CONFERMA CREAZIONE RISORSA

OPPURE SOLO MESSAGGIO CHE INDICA RISORSA NON TROVATA

PER TOGLIERE UN GIOCATORE BISOGNA PASSARE IL NOME NELL URL DELLA RICHIESTA

IN QUESTO CASO PUNTANO AGLI STESSI ENDPOINT, DEVO SOLO OCCUPARMI DI CAMBIARE IL TIPO DI RICHIESTA HTTP