

Create more controllers

1. Duplicate previous exercise folder, and continue here
2. Create all the rest of controllers to form a full CRUD functionality:
 - `public ResponseEntity<Movie> getMovie(@PathVariable long id)`
 - `public ResponseEntity<Movie> saveMovie(@RequestBody Movie movie)`
 - `public ResponseEntity<Movie> updateMovie(@PathVariable long id, @RequestBody Movie movie)`
 - `public ResponseEntity<Void> deleteMovie(@PathVariable long id)`
 - The controllers and endpoints are almost identical to yesterday's exercises (you may copy them), except you actually use a real database this time, through a service class `MovieService`, which in turn calls the interface `MovieRepository`
 - From `MovieController` class controllers, you must use `MovieService` field methods to perform database operations:
 - * `findMovieById(long id)`
 - Calls `findById(id)` method of repository
 - * `saveMovie(Movie movie)`
 - Calls `save(movie)` method of repository
 - * `existsMovieById(long id)`
 - Calls `existsById(id)` method of repository
 - * `deleteMovieById(long id)`
 - Calls `deleteById(id)` method of repository
 - 3. Don't forget that each and every controller should return the correct response code as before, as well as validate the needed fields; again, similar to what you have done yesterday, but this time you work with a real database