

- Duplicate the exercise project from yesterday, and continue here
- Refactor your codebase (the backend) to use DTOs in controllers
- Create all DTO related classes in `dto` package
- Do the refactoring little by little, for example:
 - All of the field validation logic should be moved to DTO
 - Start with the controller `getMovies()`. Create a mapper class `MovieMapper`. Inside of it, create static helper method to take in a `List<Movie>`, and convert it to a `List<MovieDTO>`. Change the return type of your `getMovies()` controller to `List<MovieDTO>`. Try calling the endpoint with a client (Postman)
 - * *Tip:* If the refactored method works, commit your code with Git, and move onto the next controller
 - The commit message should make sense. An example of a good commit message, "Refactor `getMovies()` controller to use DTO"
- End result:
 - Each and every controller in your project should end up using a DTO in some way or another. The goal is to receive data as DTO, and return result to client in DTO format - *not* entity