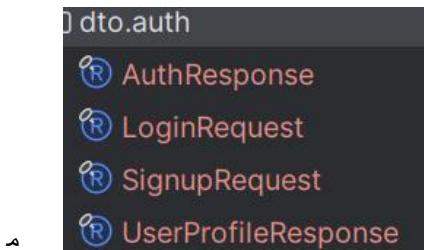


- We have multiple end-points. So need to make different **response** and **request** DTOs.
 - ↪ Previously we were using same DTO for request and response. But now we'll not.



- ↪ **AuthResponse** = **UserProfileResponse**+**token**.
 - ↪ `AuthResponse(String token, UserProfileResponse user) {`
 - ↪ **LoginRequest** and **SignupRequest** are DTOs for POST request payload.
 - ↪ You should create different DTOs for request and response.
 - ↪ For `getProfile` controller, only `UserProfileResponse` is needed, there is no need of token.
- We'll use **MapStruct** for mapping of one class type to another instead of `ModalMapper`.
 - ↪ As we are using **lombok** as well, so we need to make sure that **lombok** will be executed first then **MapStruct**.
 - ↪ Because, both will be executed during *compile time*.
 - ↪ Annotation processors do NOT run sequentially like “Lombok first, MapStruct second”. They run in the same compilation round.
 - ↪ For this **lombok-mapstruct-binding** is required.

```

<properties>
  <java.version>21</java.version>
  <org.mapstruct.version>1.6.3</org.mapstruct.version>
</properties>

```

(variable)

```

<dependency>
  <groupId>org.mapstruct</groupId>
  <artifactId>mapstruct</artifactId>
  <version>${org.mapstruct.version}</version>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok-mapstruct-binding</artifactId>
  <version>0.2.0</version>
</dependency>
<dependencies>

```

- **Dependency vs Plugins**
 - Dependencies are used by your application at compile/runtime.
 - Plugins are used by Maven to build your application.
 - To create mapper (MapStruct),

```
@Mapper(componentModel = "spring")
public interface ProjectMapper {
}
```

- Use the **componentModel="spring"**

➤
➤ F
➤ F
➤ F
➤ F
➤ F
➤