

- Entity relations and Response on the controller may not be match; and also Response may contain data from different entities (might be all fields or a few fields).
 - In case of multiple level of objects; to map those using **MapStruct** use multilevel access.
 - ❖ Lets say **user** is there in **ProjectMember**, to map user's email:

Let's say `user` is there in `ProjectMember`, to map user's email:

```
    @Mapping(target = "email", source = "user.email")
```

```
@Mapping(target = "userId", source = "user.id") 1 usage 1 implementation
@Mapping(target = "email", source = "user.email")
@Mapping(target = "name", source = "user.name")
MemberResponse toMemberResponseFromMember(ProjectMember member);
```

- In case of EmbeddedID, we need to set the ID field by our-self.

```
ProjectMember projectMember = ProjectMember.builder()
    .id(new ProjectMemberId(projectId, userId))
    .user(invitedUser)
    .project(project)
    .invitedAt(Instant.now())
    .projectRole(request.role())
    .build();
```

➤