

案例 3: 枚举和特殊规则的处理

MapStruct Plus 允许在转换过程中加入特定的逻辑，如枚举转换或者特定条件下的属性映射。

```
```
public enum Status {
 ACTIVE, INACTIVE
}

public class ProjectDto {
 private String name;
 private Status status;
}

public class Project {
 private String name;
 private String status;
}

// 枚举转换
@AutoMapper
public interface ProjectMapper {
 ProjectMapper INSTANCE =
 Mappers.getMapper(ProjectMapper.class);

 @Mapping(target = "status", expression =
"java(dto.getStatus().name())")
 Project toEntity(ProjectDto dto);

 @Mapping(target = "status", expression =
"java(Status.valueOf(entity.getStatus()))")
 ProjectDto toDto(Project entity);
}
```

```

案例 4: 处理列表转换

经常需要在DTO和实体类之间转换列表数据，例如在处理用户列表或产品列表

时。

```
```
// 用户实体和DTO
public class User {
 private String name;
 private int age;
}

public class UserDto {
 private String name;
 private int age;
}

// 转换接口
@Autowired
public interface UserListMapper {
 UserListMapper INSTANCE =
Mappers.getMapper(UserListMapper.class);

 List<UserDto> toDtoList(List<User> users);
 List<User> toEntityList(List<UserDto> userDtos);
}
````
```

案例 5: 处理空值和默认值

在实际应用中，经常需要处理空值，或者在数据为空时提供默认值。

```
}

// Usage within a Spring Service
@Service
public class OrderService {
    @Autowired
    private OrderMapper orderMapper; // This bean will be initialized
lazily

    public OrderDto convertToDto(Order order) {
        return orderMapper.toDto(order);
    }
}
```

``` ### 案例 9: 自动生成反向转换逻辑

MapStruct Plus可以自动生成逆向映射的逻辑，简化了将数据转换回其原始形式或替代目标形式的过程，无需手动定义映射。

```
```  

@AutoMapper
public interface UserMapper {
 UserMapper INSTANCE = Mappers.getMapper(UserMapper.class);

 User toEntity(UserDto userDto);

 // Automatically generate reverse mapping
 UserDto toDto(User user);
}

public class UserService {
 @Autowired
 private UserMapper userMapper;

 public User updateUserFromDto(UserDto userDto, User user) {
 userMapper.updateUserFromDto(userDto, user); // Updates the
 user entity with data from DTO
 return user;
 }

 // Example of using the reverse mapping
 public UserDto convertToDto(User user) {
 return userMapper.toDto(user); // Automatically uses the reverse
 conversion logic
 }
}
```
```

这些示例展示了如何在Spring应用程序中利用MapStruct Plus的高级功能，简化配置并减少样板代码，使您的代码库更清晰、更易维护。

原文链接: <https://juejin.cn/post/7390631421328228389>