

MapStruct

Маппинг в режиме low-код

Введение

MupStruct - процессор аннотаций для Java компилятора, основан на стандарте JSR-269 (JDK API для подключаемых процессоров аннотаций).
Зародился в 2012 году из недр Гугела.



Возможности

- Быстрая работа за счет использования прямого вызова (`invoke_virtual`)
- Безопасное преобразования за счет типизации во время компиляции
- Прерывание компиляции при неполном или неправильном сопоставлении



Возможности

- ОТСУТСТВИЕ РЕФЛЕКСИИ



Проблемы

1. Сопоставление полей производится с помощью String переменных
2. Нет поддержки автозавершения в IDE
3. Нет полноценной поддержки Generic-типов



Конвертации

- Между числовыми примитивами и их обочками и обратно (с проверкой на null)
- Любой объект в String
- Любое число в String с поддержкой Decimal Format. `@Mapping(numberFormat = "###.00")`
- String с валидным числом в примитив и его оболочку
- Enum в String и обратно
- Объекты времени и даты в String с поддержкой Simple Date Format.
`@Mapping(dateFormat = "dd.MM.yyyy")`
- Таймштемп в String и обратно (Instant, Duration, Period)
- Currency в String и обратно по ISO-4217



Подключение к проекту MAVEN

```
...
<properties>
  <org.mapstruct.version>1.4.2.Final</org.mapstruct.version>
</properties>
...
<dependencies>
  <dependency>
    <groupId>org.mapstruct</groupId>
    <artifactId>mapstruct</artifactId>
    <version>${org.mapstruct.version}</version>
  </dependency>
</dependencies>
...
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <annotationProcessorPaths>
          <path>
            <groupId>org.mapstruct</groupId>
            <artifactId>mapstruct-processor</artifactId>
            <version>${org.mapstruct.version}</version>
          </path>
        </annotationProcessorPaths>
      </configuration>
    </plugin>
  </plugins>
</build>
...
```



Начало работы

1. Класс маппера должен быть **интерфейсом** или **абстрактным** классом
2. Интерфейс должен быть аннотирован **@Mapper**
3. Сигнатура метода преобразователя должна иметь входной и выходной класс. Дженерик тип не поддерживается
4. Поля имеющие разные имена должны быть сопоставлены с помощью аннотации **@Mapping(source = "field_A", target = "field_B")**




Простой маппер

```
@Mapper
public interface Mapper {

    @Mapping(source = "src", target = "dest")
    DTO toDTO(Model model);

}
```



Инверсия преобразования

Если имеется преобразование одного объекта в другой, то можно использовать инверсия для обратного преобразования. Для создания инвертированного метода необходимо воспользоваться аннотацией

`@InheritInverseConfiguration`

`@Mapper`

`public interface Mapper {`

```
    @Mapping(source = "src", target = "dest")
    DTO toDTO(Model model);
```

```
    @InheritInverseConfiguration
    Model toModel(DTO dto);
```

```
}
```



Несколько входных объектов

Если надо агрегировать данные из нескольких объектов, то в аннотации @Mapping используется путь через точку: <исходный_объект>.<поле>

Конфликтующие поля необходимо явно прописывать в аннотации @Mapping



Несколько входных объектов

```
@Mapper
```

```
public interface Mapper {
```

```
    @Mapping(source = "model.src", target = "dest")
```

```
    @Mapping(source = "entity.src_2", target = "dest_2")
```

```
    DTO toDTO(Model model, Entity entity);
```

```
}
```



Обновление существующего объекта

Задается созданием метода с VOID сигнатурой, который принимает в качестве второго параметра целевой объект. Целевой объект помечается аннотацией @MappingTarget

```
@Mapper
public interface Mapper {

    void updateDTO(Model model, @MappingTarget DTO dto);
}
```



Поддержка Lombok

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.5.1</version>
  <configuration>
    <annotationProcessorPaths>
      <path> <!-- MUST BE FIRST -->
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>${lombok.version}</version>
      </path>
      <path>
        <groupId>org.mapstruct</groupId>
        <artifactId>mapstruct-processor</artifactId>
        <version>${org.mapstruct.version}</version>
      </path>
    </annotationProcessorPaths>
  </configuration>
</plugin>
```

Spring по умолчанию

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.5.1</version>
  <configuration>
    <annotationProcessorPaths>
      ...
    </annotationProcessorPaths>
    <compilerArgs>
      <compilerArg>
        -Amapstruct.defaultComponentModel=spring
      </compilerArg>
    </compilerArgs>
  </configuration>
</plugin>
```



Thank you /-|-\