docker

介绍

dozer入门案例

第一步: 创建项目dozer_demo

第二步:修改pom文件 第三步:创建UserEntity 第四步:创建UserDTO

第五步: 在resources/dozer/目录下创建dozer的全局配置文件global.dozer.xml

第六步:在resources/dozer/目录下创建dozer的映射文件biz.dozer.xml

第七步:编写配置类 第八步:编写单元测试类 自定义spring boot starter

开发starter

第一步:初始化项目 第二步:修改pom文件

第三步:编写工具类DozerUtils

第四步:编写配置类DozerAutoConfiguration第五步:创建并编写spring.factories文件

使用starter

第一步: 导入tools-dozer的依赖 第二步: 拷贝之前编写的User类

第三步: 拷贝配置文件

第四步:编写配置文件application.yml

第五步:编写UserController

第六步: 启动程序 第七步: 访问服务

docker

介绍

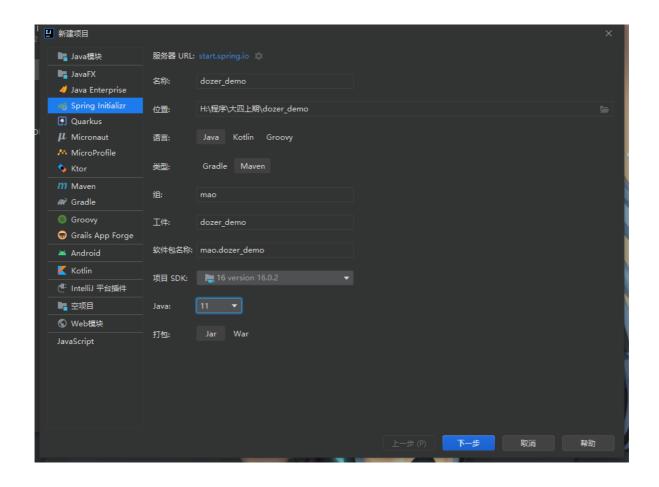
Dozer是Java Bean到Java Bean映射器,它以递归方式将数据从一个对象复制到另一个对象。 dozer是用来对两个对象之间属性转换的工具,有了这个工具之后,我们将一个对象的所有属性值转给另一个对象时,就不需要再去写重复的调用set和get方法了。dozer其实是对我们熟知的beanutils的封装

dozer的maven坐标:

为了简化使用方式, dozer还提供了starter, 其maven坐标为:

dozer入门案例

第一步: 创建项目dozer_demo



第二步:修改pom文件

```
<?xml version="1.0" encoding="UTF-8"?>
 1
 2
    project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0"
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelversion>4.0.0</modelversion>
4
 5
        <parent>
 6
            <groupId>org.springframework.boot
            <artifactId>spring-boot-starter-parent</artifactId>
 7
8
            <version>2.7.1
9
            <relativePath/> <!-- lookup parent from repository -->
10
        </parent>
11
        <groupId>mao</groupId>
        <artifactId>dozer_demo</artifactId>
12
13
        <version>0.0.1-SNAPSHOT</version>
        <name>dozer_demo</name>
14
15
        <description>dozer_demo</description>
16
17
        cproperties>
18
            <java.version>11</java.version>
19
        </properties>
20
21
        <dependencies>
```

```
22
             <dependency>
23
                 <groupId>org.springframework.boot</groupId>
24
                 <artifactId>spring-boot-starter-web</artifactId>
             </dependency>
25
26
27
             <dependency>
28
                 <groupId>org.springframework.boot</groupId>
29
                 <artifactId>spring-boot-starter-test</artifactId>
30
                 <scope>test</scope>
31
             </dependency>
32
33
             <dependency>
34
                 <groupId>com.github.dozermapper</groupId>
35
                 <artifactId>dozer-spring-boot-starter</artifactId>
36
                 <version>6.5.0</version>
37
             </dependency>
38
39
        </dependencies>
40
41
        <build>
            <plugins>
42
43
                 <plugin>
44
                     <groupId>org.springframework.boot</groupId>
                     <artifactId>spring-boot-maven-plugin</artifactId>
45
46
                 </plugin>
             </plugins>
47
        </build>
48
49
50
    </project>
```

第三步: 创建UserEntity

```
1
    package mao.dozer_demo.entity;
2
    import java.util.Date;
3
4
    /**
5
 6
    * Project name(项目名称): dozer_demo
     * Package(包名): mao.dozer_demo.entity
7
8
    * Class(类名): UserEntity
9
    * Author(作者): mao
10
    * Author QQ: 1296193245
11
    * GitHub: https://github.com/maomao124/
     * Date(创建日期): 2022/10/28
12
13
     * Time(创建时间): 19:37
     * Version(版本): 1.0
14
15
     * Description(描述): 无
     */
16
17
18
```

```
19 public class UserEntity
20
    {
        /**
21
        * id
22
        */
23
24
        private String id;
        /**
25
26
        * 名字
         */
27
28
        private String name;
        /**
29
        * 年龄
30
         */
31
32
        private int age;
        /**
33
         * 地址
34
         */
35
36
        private String address;
37
        /**
        * 生日
38
         */
39
40
        private Date birthday;
41
42
        /**
43
        * Instantiates a new User entity.
44
        */
45
46
        public UserEntity()
47
        {
48
49
        }
50
51
52
        * Instantiates a new User entity.
53
         * @param id
                          the id
54
55
         * @param name
                           the name
56
         * @param age
                          the age
57
         * @param address the address
58
         * @param birthday the birthday
59
        public UserEntity(String id, String name, int age, String address, Date
60
    birthday)
61
        {
            this.id = id;
62
63
            this.name = name;
64
            this.age = age;
65
            this.address = address;
            this.birthday = birthday;
66
67
        }
68
        /**
69
70
         * Gets id.
71
         * @return the id
72
73
74
        public String getId()
75
        {
```

```
76
     return id;
 77
         }
 78
         /**
 79
         * Sets id.
 80
 81
         * @param id the id
 82
 83
         public void setId(String id)
 84
 85
 86
           this.id = id;
 87
         }
 88
         /**
 89
         * Gets name.
 90
 91
 92
         * @return the name
 93
 94
         public String getName()
 95
 96
           return name;
 97
         }
98
99
         /**
         * Sets name.
100
101
         * @param name the name
102
103
104
         public void setName(String name)
105
106
           this.name = name;
107
         }
108
109
         /**
110
         * Gets age.
111
112
         * @return the age
         */
113
114
         public int getAge()
115
116
           return age;
117
         }
118
         /**
119
120
         * Sets age.
121
         * @param age the age
122
         */
123
124
         public void setAge(int age)
125
126
            this.age = age;
127
         }
128
         /**
129
         * Gets address.
130
131
132
          * @return the address
          */
133
```

```
134
         public String getAddress()
135
         {
136
              return address;
137
         }
138
         /**
139
140
          * Sets address.
141
142
          * @param address the address
143
144
         public void setAddress(String address)
145
         {
146
             this.address = address;
147
         }
148
         /**
149
          * Gets birthday.
150
151
152
          * @return the birthday
153
         public Date getBirthday()
154
155
         {
156
             return birthday;
         }
157
158
         /**
159
          * Sets birthday.
160
161
          * @param birthday the birthday
162
163
         public void setBirthday(Date birthday)
164
165
         {
             this.birthday = birthday;
166
167
         }
168
         @override
169
170
         @SuppressWarnings("all")
         public String toString()
171
172
173
              final StringBuilder stringBuilder = new StringBuilder();
174
              stringbuilder.append("id: ").append(id).append('\n');
              stringbuilder.append("name: ").append(name).append('\n');
175
              stringbuilder.append("age: ").append(age).append('\n');
176
              string builder.append ("address: ").append (address).append ('\n');\\
177
178
              stringbuilder.append("birthday: ").append(birthday).append('\n');
179
              return stringbuilder.toString();
180
         }
181
    }
```

第四步: 创建UserDTO

```
1
    package mao.dozer_demo.entity;
2
   /**
3
4
    * Project name(项目名称): dozer_demo
    * Package(包名): mao.dozer_demo.entity
 5
6
    * Class(类名): UserDTO
    * Author(作者): mao
    * Author QQ: 1296193245
8
9
    * GitHub: https://github.com/maomao124/
10
    * Date(创建日期): 2022/10/28
11
    * Time(创建时间): 19:39
    * Version(版本): 1.0
12
13
    * Description(描述): 无
14
    */
15
16
    public class UserDTO
17
18
        /**
19
20
        * 用户id
21
        */
22
        private String userId;
        /**
23
24
        * 用户名
        */
25
26
       private String userName;
27
        /**
        * 用户年龄
28
        */
29
30
        private int userAge;
       /**
31
        * 地址
32
        */
33
        private String address;
34
       /**
35
        * 生日
36
37
38
        private String birthday;
39
40
        * Instantiates a new User dto.
41
42
43
        public UserDTO()
44
        {
45
        }
46
47
48
        * Instantiates a new User dto.
49
50
        * @param userId the user id
51
        * @param userName the user name
52
         * @param userAge the user age
53
         * @param address the address
```

```
54
        * @param birthday the birthday
 55
          */
         public UserDTO(String userId, String userName, int userAge, String
 56
     address, String birthday)
 57
 58
             this.userId = userId;
 59
             this.userName = userName;
 60
             this.userAge = userAge;
 61
             this.address = address;
 62
             this.birthday = birthday;
 63
         }
 64
         /**
 65
 66
          * Gets user id.
 67
          * @return the user id
 68
 69
          */
 70
         public String getUserId()
 71
 72
             return userId;
 73
         }
 74
         /**
 75
 76
          * Sets user id.
 77
          * @param userId the user id
 78
 79
          */
 80
         public void setUserId(String userId)
 81
 82
             this.userId = userId;
 83
         }
 84
 85
          * Gets user name.
 86
 87
          * @return the user name
 88
          */
 89
 90
         public String getUserName()
 91
 92
             return userName;
 93
         }
 94
         /**
 95
 96
          * Sets user name.
 97
 98
          * @param userName the user name
99
100
         public void setUserName(String userName)
101
             this.userName = userName;
102
103
         }
104
         /**
105
          * Gets user age.
106
107
          * @return the user age
108
109
110
         public int getUserAge()
```

```
111
112
             return userAge;
113
         }
114
115
         * Sets user age.
116
117
          * @param userAge the user age
118
119
120
         public void setUserAge(int userAge)
121
122
             this.userAge = userAge;
123
         }
124
         /**
125
         * Gets address.
126
127
          * @return the address
128
129
130
         public String getAddress()
131
132
             return address;
133
         }
134
         /**
135
136
         * Sets address.
137
138
          * @param address the address
139
140
         public void setAddress(String address)
141
142
             this.address = address;
143
         }
144
         /**
145
         * Gets birthday.
146
147
         * @return the birthday
148
         */
149
150
         public String getBirthday()
151
152
             return birthday;
153
         }
154
155
156
         * Sets birthday.
157
          * @param birthday the birthday
158
159
         public void setBirthday(String birthday)
160
161
         {
162
             this.birthday = birthday;
163
         }
164
165
         @override
         @SuppressWarnings("all")
166
167
         public String toString()
168
         {
```

```
final StringBuilder stringbuilder = new StringBuilder();
169
170
             stringbuilder.append("userId: ").append(userId).append('\n');
             stringbuilder.append("userName: ").append(userName).append('\n');
171
             stringbuilder.append("userAge: ").append(userAge).append('\n');
172
173
             stringbuilder.append("address: ").append(address).append('\n');
174
             stringbuilder.append("birthday: ").append(birthday).append('\n');
175
             return stringbuilder.toString();
176
         }
177
     }
```

第五步:在resources/dozer/目录下创建dozer的全局配置文件global.dozer.xml

```
<?xml version="1.0" encoding="UTF-8"?>
 1
    <mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
 2
 3
              xmlns="http://dozermapper.github.io/schema/bean-mapping"
 4
              xsi:schemaLocation="http://dozermapper.github.io/schema/bean-
    mapping
 5
                                   http://dozermapper.github.io/schema/bean-
    mapping.xsd">
 6
        <!--
 7
        全局配置:
 8
        <date-format>表示日期格式
9
10
        <configuration>
11
            <date-format>yyyy-MM-dd</date-format>
        </configuration>
12
13
14
15
    </mappings>
```

第六步:在resources/dozer/目录下创建dozer的映射文件biz.dozer.xml

```
8
         <mapping date-format="yyyy-MM-dd">
 9
             <class-a>mao.dozer_demo.entity.UserEntity</class-a>
             <class-b>mao.dozer_demo.entity.UserDTO</class-b>
10
11
             <field>
12
                 <a>id</a>
13
14
                 <b>userId</b>
15
            </field>
16
            <field>
17
                 <a>name</a>
18
                 <b>userName</b>
19
            </field>
20
            <field>
21
                 <a>age</a>
22
                 <b>userAge</b>
23
             </field>
24
25
        </mapping>
26
         <mapping date-format="yyyy-MM-dd" map-id="user">
27
28
             <class-a>mao.dozer_demo.entity.UserEntity</class-a>
             <class-b>mao.dozer_demo.entity.UserDTO</class-b>
29
30
31
            <field>
                 <a>id</a>
32
33
                 <b>userId</b>
34
            </field>
35
            <field>
36
                 <a>name</a>
37
                 <b>userName</b>
38
            </field>
39
            <field>
40
                 <a>age</a>
41
                 <b>userAge</b>
42
             </field>
43
44
        </mapping>
45
46
47
48
    </mappings>
```

第七步: 编写配置类

```
@Configuration
 2
    public class DozerMapperConfig
 3
 4
        @Bean
 5
        public DozerBeanMapperFactoryBean
    dozerMapper(@value("classpath:dozer/*.xml") Resource[] resources)
 6
                throws IOException
 7
        {
8
            DozerBeanMapperFactoryBean dozerBeanMapperFactoryBean = new
    DozerBeanMapperFactoryBean();
9
            dozerBeanMapperFactoryBean.setMappingFiles(resources);
10
            return dozerBeanMapperFactoryBean;
11
        }
12
    }
```

第八步:编写单元测试类

```
package mao.dozer_demo;
 2
    import com.github.dozermapper.core.Mapper;
    import mao.dozer_demo.entity.UserDTO;
 5
    import mao.dozer_demo.entity.UserEntity;
 6
    import org.junit.jupiter.api.Test;
7
    import org.slf4j.Logger;
    import org.slf4j.LoggerFactory;
9
    import org.springframework.beans.factory.annotation.Autowired;
10
    import org.springframework.boot.test.context.SpringBootTest;
11
12
    import java.util.Date;
13
    /**
14
15
    * Project name(项目名称): dozer_demo
    * Package(包名): mao.dozer_demo
16
17
    * Class(类名): DozerTest
18
    * Author(作者): mao
     * Author QQ: 1296193245
19
20
    * GitHub: https://github.com/maomao124/
21
    * Date(创建日期): 2022/10/28
22
    * Time(创建时间): 19:50
23
    * Version(版本): 1.0
     * Description(描述): 无
24
25
    */
26
    @SpringBootTest
27
28
    public class DozerTest
29
    {
30
31
        @Autowired
32
        private Mapper mapper;
33
```

```
34     private static final Logger log =
    LoggerFactory.getLogger(DozerTest.class);
35
36
        @Test
37
        void test1()
38
39
            UserDTO userDTO = new UserDTO();
40
            userDTO.setUserId("134");
41
            userDTO.setUserName("张三");
42
            userDTO.setUserAge(19);
            userDTO.setAddress("中国");
43
44
            userDTO.setBirthday("2012-10-13");
45
            log.info(userDTO.toString());
46
47
48
            UserEntity userEntity = mapper.map(userDTO, UserEntity.class);
49
50
            log.info(userEntity.toString());
51
52
        }
53
54
        @Test
55
        void test2()
56
            UserEntity userEntity = new UserEntity("1234", "李四", 14, "中国",
57
58
            UserDTO userDTO = mapper.map(userEntity, UserDTO.class);
59
            log.info(userEntity.toString());
60
61
            log.info(userDTO.toString());
62
        }
63
    }
```

测试1:

```
1 2022-10-28 20:32:37.318 INFO 15068 --- [
                                                     main]
    mao.dozer_demo.DozerTest
                                         : userId: 134
   userName: 张三
 3
   userAge: 19
4
   address: 中国
5
   birthday: 2012-10-13
 6
    2022-10-28 20:32:37.326 INFO 15068 --- [
    mao.dozer_demo.DozerTest
                                         : id: 134
8
   name: 张三
9
   age: 19
10
   address: 中国
11 | birthday: Sat Oct 13 00:00:00 CST 2012
```

测试2:

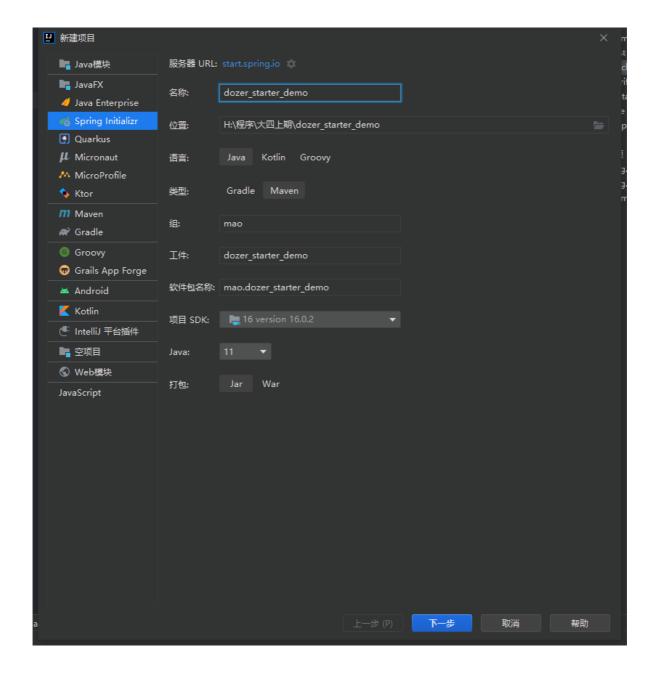
```
1 2022-10-28 20:33:08.781 INFO 1252 --- [ main] mao.dozer_demo.DozerTest : id: 1234
2 name: 李四
3 age: 14
4 address: 中国
5 birthday: Fri Oct 28 20:33:08 CST 2022
6
7 2022-10-28 20:33:08.781 INFO 1252 --- [ main] mao.dozer_demo.DozerTest : userId: 1234
8 userName: 李四
9 userAge: 14
10 address: 中国
birthday: 2022-10-28
```

自定义spring boot starter

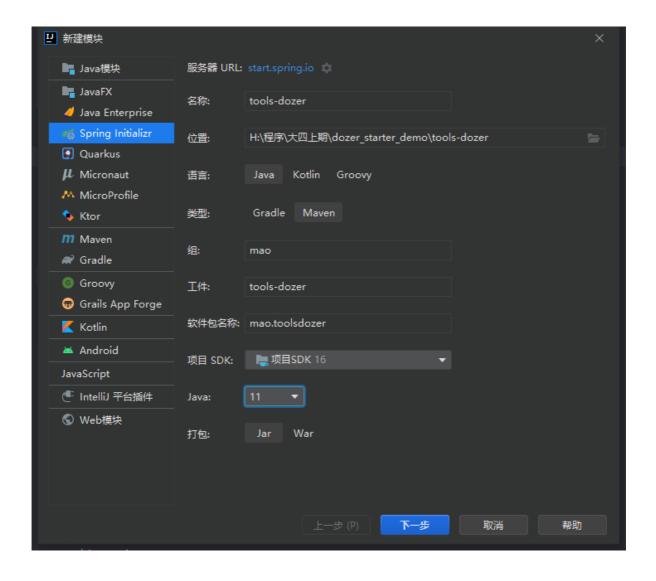
开发starter

第一步: 初始化项目

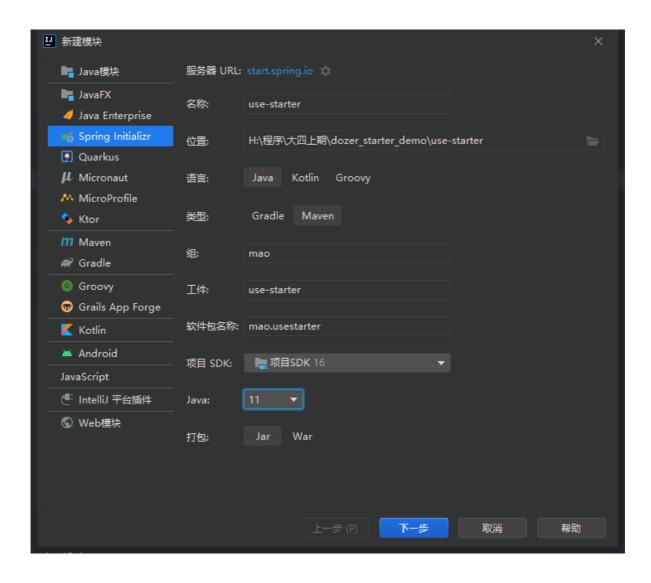
创建父工程dozer_starter_demo



创建子工程tools-dozer



创建子工程use-starter



第二步:修改pom文件

父工程dozer_starter_demo的pom文件:

```
<?xml version="1.0" encoding="UTF-8"?>
1
2
    project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0"
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelversion>4.0.0</modelversion>
4
5
6
        <parent>
7
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-parent</artifactId>
8
9
            <version>2.7.1
10
            <relativePath/> <!-- lookup parent from repository -->
11
        </parent>
12
13
        <groupId>mao
```

```
14
         <artifactId>dozer_starter_demo</artifactId>
15
         <version>0.0.1-SNAPSHOT</version>
16
         <name>dozer_starter_demo</name>
17
         <description>dozer_starter_demo</description>
18
         <packaging>pom</packaging>
19
20
         cproperties>
21
             <java.version>11</java.version>
         </properties>
22
23
24
         <modules>
25
             <module>tools-dozer</module>
26
             <module>use-starter</module>
27
         </modules>
28
29
30
         <dependencies>
31
         </dependencies>
32
33
         <dependencyManagement>
34
35
36
             <dependencies>
37
38
             </dependencies>
39
         </dependencyManagement>
40
41
         <build>
42
43
             <plugins>
44
                 <plugin>
45
                      <groupId>org.springframework.boot</groupId>
                      <artifactId>spring-boot-maven-plugin</artifactId>
46
47
                 </plugin>
48
             </plugins>
49
         </build>
50
51
    </project>
```

子工程tools-dozer的pom文件:

```
<?xml version="1.0" encoding="UTF-8"?>
1
    project xmlns="http://maven.apache.org/POM/4.0.0"
2
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
4
5
6
        <parent>
            <artifactId>dozer_starter_demo</artifactId>
8
            <groupId>mao</groupId>
9
            <version>0.0.1-SNAPSHOT</version>
10
        </parent>
11
```

```
12
        <artifactId>tools-dozer</artifactId>
13
        <version>0.0.1-SNAPSHOT</version>
14
         <name>tools-dozer</name>
15
        <description>tools-dozer</description>
16
17
        cproperties>
18
19
        </properties>
20
21
         <dependencies>
22
23
             <dependency>
24
                 <groupId>org.springframework.boot</groupId>
25
                 <artifactId>spring-boot-starter-web</artifactId>
26
             </dependency>
27
28
             <dependency>
29
                 <groupId>com.github.dozermapper</groupId>
                 <artifactId>dozer-spring-boot-starter</artifactId>
30
31
                 <version>6.5.0</version>
             </dependency>
32
33
34
             <!--spring boot starter开发依赖-->
35
             <dependency>
36
                 <groupId>org.springframework.boot
                 <artifactId>spring-boot-starter</artifactId>
37
             </dependency>
38
39
40
            <dependency>
41
                 <groupId>org.springframework.boot</groupId>
                 <artifactId>spring-boot-autoconfigure</artifactId>
42
43
             </dependency>
44
45
             <dependency>
46
                 <groupId>org.springframework.boot</groupId>
47
                 <artifactId>spring-boot-configuration-processor</artifactId>
             </dependency>
48
49
        </dependencies>
50
51
52
         <build>
53
            <plugins>
54
                 <plugin>
55
                     <groupId>org.springframework.boot</groupId>
56
                     <artifactId>spring-boot-maven-plugin</artifactId>
57
                     <configuration>
58
                         <skip>true</skip>
59
                     </configuration>
60
                 </plugin>
            </plugins>
61
62
        </build>
63
64
    </project>
65
```

```
<?xml version="1.0" encoding="UTF-8"?>
 2
    project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
 4
        <modelversion>4.0.0</modelversion>
 5
 6
        <parent>
 7
             <artifactId>dozer_starter_demo</artifactId>
 8
             <groupId>mao</groupId>
 9
             <version>0.0.1-SNAPSHOT</version>
10
        </parent>
11
12
        <artifactId>use-starter</artifactId>
        <version>0.0.1-SNAPSHOT</version>
13
        <name>use-starter</name>
14
15
        <description>use-starter</description>
16
17
        cproperties>
18
19
        </properties>
20
21
        <dependencies>
22
             <dependency>
23
                 <groupId>org.springframework.boot</groupId>
                 <artifactId>spring-boot-starter-web</artifactId>
24
25
             </dependency>
26
27
             <dependency>
                 <groupId>org.springframework.boot</groupId>
28
29
                 <artifactId>spring-boot-starter-test</artifactId>
30
                 <scope>test</scope>
             </dependency>
31
32
33
        </dependencies>
34
35
        <build>
            <plugins>
36
37
                 <plugin>
38
                     <groupId>org.springframework.boot</groupId>
                     <artifactId>spring-boot-maven-plugin</artifactId>
39
40
                 </plugin>
            </plugins>
41
42
        </build>
43
44
    </project>
```

第三步:编写工具类DozerUtils

```
1
    package mao.toolsdozer.utils;
2
3
   import com.github.dozermapper.core.Mapper;
4
   import java.util.*;
5
    import java.util.stream.Collectors;
6
7
   /**
8
9
    * Project name(项目名称): dozer_starter_demo
10
    * Package(包名): mao.toolsdozer.utils
    * Class(类名): DozerUtils
11
12
    * Author(作者): mao
13
    * Author QQ: 1296193245
14
    * GitHub: https://github.com/maomao124/
15
    * Date(创建日期): 2022/10/28
    * Time(创建时间): 21:57
16
17
    * Version(版本): 1.0
    * Description(描述): 无
18
19
20
21
    public class DozerUtils
22
23
        private final Mapper mapper;
24
25
        public DozerUtils(Mapper mapper)
26
        {
27
            this.mapper = mapper;
28
        }
29
        public Mapper getMapper()
30
31
        {
32
            return this.mapper;
33
        }
34
        /**
35
36
37
        * Constructs new instance of destinationClass and performs mapping
    between from source
38
39
         * @param source
         * @param destinationClass 目标类
40
41
        * @return {@link T}
42
        public <T> T map(Object source, Class<T> destinationClass)
43
44
45
            if (source == null)
46
            {
47
                return null;
            }
48
49
            return mapper.map(source, destinationClass);
50
        }
51
        /**
52
```

```
53
          * map2
 54
 55
          * @param source
 56
          * @param destinationClass 目标类
 57
          * @return {@link T}
 58
          */
 59
         public <T> T map2(Object source, Class<T> destinationClass)
 60
         {
 61
             if (source == null)
 62
             {
 63
                 try
 64
 65
                      return destinationClass.newInstance();
 66
                 }
 67
                 catch (Exception ignored)
 68
                 {
 69
                 }
 70
             }
 71
             return mapper.map(source, destinationClass);
 72
         }
 73
 74
         /**
 75
          * 地图
          * Performs mapping between source and destination objects
 76
 77
          * @param source
 78
 79
          * @param destination 目地
 80
         public void map(Object source, Object destination)
 81
 82
 83
             if (source == null)
 84
 85
                 return;
 86
             }
 87
             mapper.map(source, destination);
 88
         }
 89
         /**
 90
 91
 92
          * Constructs new instance of destinationClass and performs mapping
     between from source
 93
 94
          * @param source
 95
          * @param destinationClass 目标类
 96
          * @param mapId
                                     mapId
 97
          * @return {@link T}
          */
 98
         public <T> T map(Object source, Class<T> destinationClass, String
 99
     mapId)
100
         {
             if (source == null)
101
102
103
                 return null;
104
105
             return mapper.map(source, destinationClass, mapId);
         }
106
107
         /**
108
```

```
109
110
          * Performs mapping between source and destination objects
111
112
          * @param source
113
          * @param destination 目标
114
          * @param mapId mapId
115
          */
         public void map(Object source, Object destination, String mapId)
116
117
118
             if (source == null)
119
             {
120
                 return;
121
             }
             mapper.map(source, destination, mapId);
122
123
         }
124
125
126
          * 将集合转成集合
127
128
         * List<A> --> List<B>
129
          * @param sourceList 源集合
130
131
          * @param destinationClass 待转类型
          * @return {@link List}<{@link T}>
132
133
134
         public <T, E> List<T> mapList(Collection<E> sourceList, Class<T>
     destinationClass)
135
         {
136
             return mapPage(sourceList, destinationClass);
137
         }
138
139
140
         public <T, E> List<T> mapPage(Collection<E> sourceList, Class<T>
     destinationClass)
141
142
             if (sourceList == null || sourceList.isEmpty() || destinationClass
     == null
143
             {
144
                 return Collections.emptyList();
145
             }
             return sourceList.stream()
146
147
                     .filter(Objects::nonNull)
148
                     .map((sourceObject) -> mapper.map(sourceObject,
     destinationClass))
149
                     .collect(Collectors.toList());
150
         }
151
152
         public <T, E> Set<T> mapSet(Collection<E> sourceList, Class<T>
     destinationClass)
153
             if (sourceList == null || sourceList.isEmpty() || destinationClass
154
     == null
155
             {
156
                 return Collections.emptySet();
157
             }
158
             return sourceList.stream().map((sourceObject) ->
     mapper.map(sourceObject, destinationClass)).collect(Collectors.toSet());
159
         }
```

第四步:编写配置类DozerAutoConfiguration

```
1
    package mao.toolsdozer.config;
 2
 3
    import com.github.dozermapper.core.Mapper;
    import com.github.dozermapper.spring.DozerBeanMapperFactoryBean;
 4
    import mao.toolsdozer.utils.DozerUtils;
 5
 6
    import org.springframework.beans.factory.annotation.Autowired;
    import org.springframework.beans.factory.annotation.Value;
8
    import org.springframework.context.annotation.Bean;
9
10
    import org.springframework.context.annotation.Configuration;
11
    import org.springframework.core.io.Resource;
12
13
14
15
    import java.io.IOException;
16
17
    /**
18
     * Project name(项目名称): dozer_starter_demo
    * Package(包名): mao.toolsdozer.config
19
20
     * Class(类名): DozerAutoConfiguration
21
     * Author(作者): mao
     * Author QQ: 1296193245
22
23
     * GitHub: https://github.com/maomao124/
    * Date(创建日期): 2022/10/28
24
25
    * Time(创建时间): 22:03
26
     * Version(版本): 1.0
27
     * Description(描述): 无
     */
28
29
    @Configuration
30
31
    public class DozerAutoConfiguration
32
    {
33
34
        /*public DozerBeanMapperFactoryBean
    dozerMapper(@Value("classpath:dozer/*.xml") Resource[] resources)
35
                throws IOException
36
        {
37
            DozerBeanMapperFactoryBean dozerBeanMapperFactoryBean = new
    DozerBeanMapperFactoryBean();
38
            dozerBeanMapperFactoryBean.setMappingFiles(resources);
39
            return dozerBeanMapperFactoryBean;
        }*/
40
41
        @Autowired
42
43
        private Mapper mapper;
44
45
        @Bean
```

```
public DozerUtils getDozerUtils() throws IOException
{
    return new DozerUtils(mapper);
}

50
51
}
```

第五步: 创建并编写spring.factories文件

```
org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
mao.toolsdozer.config.DozerAutoConfiguration
```

使用starter

第一步: 导入tools-dozer的依赖

```
<?xml version="1.0" encoding="UTF-8"?>
 1
    project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0"
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelversion>4.0.0</modelversion>
 4
 5
 6
        <parent>
            <artifactId>dozer_starter_demo</artifactId>
 7
 8
            <groupId>mao
 9
            <version>0.0.1-SNAPSHOT</version>
10
        </parent>
11
        <artifactId>use-starter</artifactId>
12
13
        <version>0.0.1-SNAPSHOT</version>
        <name>use-starter</name>
14
        <description>use-starter</description>
15
16
17
        cproperties>
18
```

```
19
        </properties>
20
21
        <dependencies>
            <dependency>
22
23
                 <groupId>org.springframework.boot</groupId>
                 <artifactId>spring-boot-starter-web</artifactId>
24
25
            </dependency>
26
27
            <dependency>
28
                 <groupId>org.springframework.boot</groupId>
29
                 <artifactId>spring-boot-starter-test</artifactId>
30
                 <scope>test</scope>
            </dependency>
31
32
            <dependency>
33
                 <groupId>mao</groupId>
34
35
                 <artifactId>tools-dozer</artifactId>
                 <version>0.0.1-SNAPSHOT</version>
36
37
            </dependency>
38
        </dependencies>
39
40
        <build>
41
42
            <plugins>
43
                 <plugin>
                     <groupId>org.springframework.boot</groupId>
44
45
                     <artifactId>spring-boot-maven-plugin</artifactId>
46
                 </plugin>
47
            </plugins>
48
        </build>
49
50
    </project>
```

第二步: 拷贝之前编写的User类

```
we-starter

image: monor with the property of the property of
```

第三步: 拷贝配置文件

```
UseStarterApplication

✓ ■ resources

✓ ■ dozer

ಈ biz.dozer.xml

ಈ global.dozer.xml

■ static
■ templates

ಈ application.yml

> ■ test
```

biz.dozer.xml:

```
1
    <mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"</pre>
2
              xmlns="http://dozermapper.github.io/schema/bean-mapping"
 3
              xsi:schemaLocation="http://dozermapper.github.io/schema/bean-
    mapping
4
                                 http://dozermapper.github.io/schema/bean-
    mapping.xsd">
5
        <!--描述两个类中属性的对应关系,对于两个类中同名的属性可以不映射-->
6
 7
        <mapping date-format="yyyy-MM-dd">
8
            <class-a>mao.usestarter.entity.UserEntity</class-a>
9
10
            <class-b>mao.usestarter.entity.UserDTO</class-b>
11
            <field>
12
                <a>id</a>
13
                <b>userId</b>
14
```

```
</field>
15
16
             <field>
17
                 <a>name</a>
18
                 <b>userName</b>
19
            </field>
            <field>
20
21
                 <a>age</a>
                 <b>userAge</b>
22
23
             </field>
24
25
        </mapping>
26
        <mapping date-format="yyyy-MM-dd" map-id="user">
27
28
             <class-a>mao.usestarter.entity.UserEntity</class-a>
             <class-b>mao.usestarter.entity.UserDTO</class-b>
29
30
31
            <field>
                 <a>id</a>
32
33
                 <b>userId</b>
            </field>
34
            <field>
35
36
                 <a>name</a>
37
                 <b>userName</b>
38
            </field>
            <field>
39
                 <a>age</a>
40
41
                 <b>userAge</b>
42
            </field>
43
        </mapping>
45
46
47
48
    </mappings>
49
```

第四步:编写配置文件application.yml

```
dozer:
mapping-files:
    - classpath:dozer/global.dozer.xml
    - classpath:dozer/biz.dozer.xml
```

第五步:编写UserController

```
package mao.usestarter.controller;
 2
 3
    import mao.toolsdozer.utils.DozerUtils;
    import mao.usestarter.entity.UserDTO;
 4
    import mao.usestarter.entity.UserEntity;
    import org.slf4j.Logger;
 6
    import org.slf4j.LoggerFactory;
 7
 8
    import org.springframework.beans.factory.annotation.Autowired;
 9
    import org.springframework.web.bind.annotation.GetMapping;
    import org.springframework.web.bind.annotation.RestController;
10
11
12
13
    import java.util.Date;
14
    /**
15
    * Project name(项目名称): dozer_starter_demo
16
17
     * Package(包名): mao.usestarter.controller
     * Class(类名): UserController
18
19
     * Author(作者): mao
20
     * Author QQ: 1296193245
21
     * GitHub: https://github.com/maomao124/
     * Date(创建日期): 2022/10/28
22
    * Time(创建时间): 22:18
23
24
     * Version(版本): 1.0
25
     * Description(描述): 无
     */
26
27
28
29
    @RestController
    public class UserController
30
31
    {
32
33
        @Autowired
34
        private DozerUtils dozerUtils;
35
        private static final Logger log =
36
    LoggerFactory.getLogger(UserController.class);
37
        @GetMapping("/test1")
38
39
        public UserDTO test1()
40
        {
41
            log.info("test1");
            UserEntity userEntity = new UserEntity("1234", "张三", 13, "中国",
42
    new Date());
43
            log.info(userEntity.toString());
44
            UserDTO userDTO = dozerUtils.map(userEntity, UserDTO.class);
45
            log.info(userDTO.toString());
46
            return userDTO;
47
        }
48
49
        @GetMapping("test2")
50
        public UserEntity test2()
51
```

```
log.info("test2");
52
53
            UserDTO userDTO = new UserDTO("1234", "张三", 13, "中国", "2011-07-
    23");
54
            log.info(userDTO.toString());
55
            UserEntity userEntity = dozerUtils.map(userDTO, UserEntity.class);
            log.info(userEntity.toString());
56
57
            return userEntity;
58
       }
59 }
```

第六步: 启动程序

```
1
    /\\ / ___'_ _ _ _ _(_)_ _ _ _ _ _ \ \ \ \
 2
 3
    (()\__|'_||'_|\/_\/\
4
    \\/ __)| |_)| | | | | | (_| | ) ) ) )
 5
    ' |___| .__|_| |_|_| |_\__, | / / / /
    ======|_|======|___/=/_/_/
 6
 7
    :: Spring Boot ::
                                   (v2.7.1)
 8
   2022-10-28 23:36:55.652 INFO 1684 --- [
                                                  main]
   mao.usestarter.UseStarterApplication : Starting UseStarterApplication
    using Java 16.0.2 on mao with PID 1684 (H:\程序\大四上期
    \dozer_starter_demo\use-starter\target\classes started by mao in H:\程序\大四
    上期\dozer_starter_demo)
   2022-10-28 23:36:55.654 INFO 1684 --- [
10
                                                   main]
   mao.usestarter.UseStarterApplication : No active profile set, falling
    back to 1 default profile: "default"
11 2022-10-28 23:36:56.285 INFO 1684 --- [
    o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s):
    8080 (http)
12
   2022-10-28 23:36:56.292 INFO 1684 --- [
                                                  main]
    o.apache.catalina.core.StandardService : Starting service [Tomcat]
   2022-10-28 23:36:56.292 INFO 1684 --- [
13
    org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
   Tomcat/9.0.64]
14
   2022-10-28 23:36:56.373 INFO 1684 --- [
                                                 main] o.a.c.c.C.[Tomcat].
    [localhost].[/] : Initializing Spring embedded WebApplicationContext
   2022-10-28 23:36:56.373 INFO 1684 --- [
15
    w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
   initialization completed in 680 ms
   2022-10-28 23:36:56.418 INFO 1684 --- [
16
                                                   main]
    c.g.d.core.DozerBeanMapperBuilder : Initializing Dozer. Version:
    6.5.0, Thread Name: main
   2022-10-28 23:36:56.419 INFO 1684 --- [
    c.g.dozermapper.core.util.RuntimeUtils : OSGi support is false
18
   2022-10-28 23:36:56.422 INFO 1684 --- [
                                                 main]
    d.c.c.r.LegacyPropertiesSettingsResolver : Trying to find Dozer
    configuration file: dozer.properties
```

```
19 | 2022-10-28 23:36:56.426 INFO 1684 --- [ main]
    d.c.c.r.LegacyPropertiesSettingsResolver : Failed to find dozer.properties
    com.github.dozermapper.core.config.resolvers.LegacyPropertiesSettingsResolve
20 2022-10-28 23:36:56.428 WARN 1684 --- [
                                                    main]
    c.g.d.core.el.ELExpressionFactory : javax.el is not supported; Failed
    to resolve ExpressionFactory, com.sun.el.ExpressionFactoryImpl
21 2022-10-28 23:36:56.437 INFO 1684 --- [
    c.g.d.c.b.xml.BeanMappingXMLBuilder
                                        : Using URL
    [file:/H:/%e7%a8%8b%e5%ba%8f/%e5%a4%a7%e5%9b%9b%e4%b8%8a%e6%9c%9f/dozer_star
    ter_demo/use-starter/target/classes/dozer/global.dozer.xml] to load custom
    xml mappings
c.g.d.c.b.xml.SchemaLSResourceResolver : Trying to resolve XML entity with
    public ID [null] and system ID [http://dozermapper.github.io/schema/bean-
    mapping.xsd]
   2022-10-28 23:36:56.570 INFO 1684 --- [
23
    c.g.d.c.b.xml.SchemaLSResourceResolver : Resolved public ID [null] and
    system ID [http://dozermapper.github.io/schema/bean-mapping.xsd]
    2022-10-28 23:36:56.589 INFO 1684 --- [
24
    c.g.d.c.b.xml.BeanMappingXMLBuilder : Successfully loaded custom xml
    mapping.
25 | 2022-10-28 23:36:56.590 INFO 1684 --- [
    c.g.d.c.b.xml.BeanMappingXMLBuilder
                                         : Using URL
    [file:/H:/%e7%a8%8b%e5%ba%8f/%e5%a4%a7%e5%9b%9b%e4%b8%8a%e6%9c%9f/dozer_star
    ter_demo/use-starter/target/classes/dozer/biz.dozer.xml] to load custom xml
    mappings
26 | 2022-10-28 23:36:56.594 INFO 1684 --- [
                                                    main1
    c.g.d.c.b.xml.SchemaLSResourceResolver : Trying to resolve XML entity with
    public ID [null] and system ID [http://dozermapper.github.io/schema/bean-
    mapping.xsd]
27 2022-10-28 23:36:56.594 INFO 1684 --- [
                                                     main]
    c.g.d.c.b.xml.SchemaLSResourceResolver : Resolved public ID [null] and
    system ID [http://dozermapper.github.io/schema/bean-mapping.xsd]
28 2022-10-28 23:36:56.605 INFO 1684 --- [
    c.g.d.c.b.xml.BeanMappingXMLBuilder : Successfully loaded custom xml
    mapping.
29 2022-10-28 23:36:56.829 INFO 1684 --- [
                                                    mainl
    \hbox{o.s.b.w.embedded.tomcat.} Tomcat webServer : Tomcat started on port(s): 8080
    (http) with context path ''
30 2022-10-28 23:36:56.841 INFO 1684 --- [
    mao.usestarter.UseStarterApplication : Started UseStarterApplication in
    1.472 seconds (JVM running for 1.952)
```

第七步: 访问服务

{"userId": "1234", "userName": "张三", "userAge": 13, "address": "中国", "birthday": "2022-10-28"}

{"id":"1234", "name":"张三", "age":13, "address":"中国", "birthday":"2011-07-22T16:00:00.000+00:00"}

```
1 | 2022-10-28 23:38:32.015 INFO 1684 --- [nio-8080-exec-1]
    m.usestarter.controller.UserController : test1
    2022-10-28 23:38:32.015 INFO 1684 --- [nio-8080-exec-1]
    m.usestarter.controller.UserController : id: 1234
    name: 张三
 4
    age: 13
 5
    address: 中国
6
    birthday: Fri Oct 28 23:38:32 CST 2022
    2022-10-28 23:38:32.021 INFO 1684 --- [nio-8080-exec-1]
    m.usestarter.controller.UserController : userId: 1234
9
    userName: 张三
10
    userAge: 13
11
    address: 中国
    birthday: 2022-10-28
12
13
14
    2022-10-28 23:38:43.110 INFO 1684 --- [nio-8080-exec-2]
    m.usestarter.controller.UserController : test2
    2022-10-28 23:38:43.110 INFO 1684 --- [nio-8080-exec-2]
15
    m.usestarter.controller.UserController : userId: 1234
16
    userName: 张三
17
    userAge: 13
    address: 中国
18
    birthday: 2011-07-23
19
20
    2022-10-28 23:38:43.111 INFO 1684 --- [nio-8080-exec-2]
21
    m.usestarter.controller.UserController : id: 1234
22
    name: 张三
23
    age: 13
24
    address: 中国
25 | birthday: Sat Jul 23 00:00:00 CST 2011
```

end	
by mao	
2022 10 28	