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：

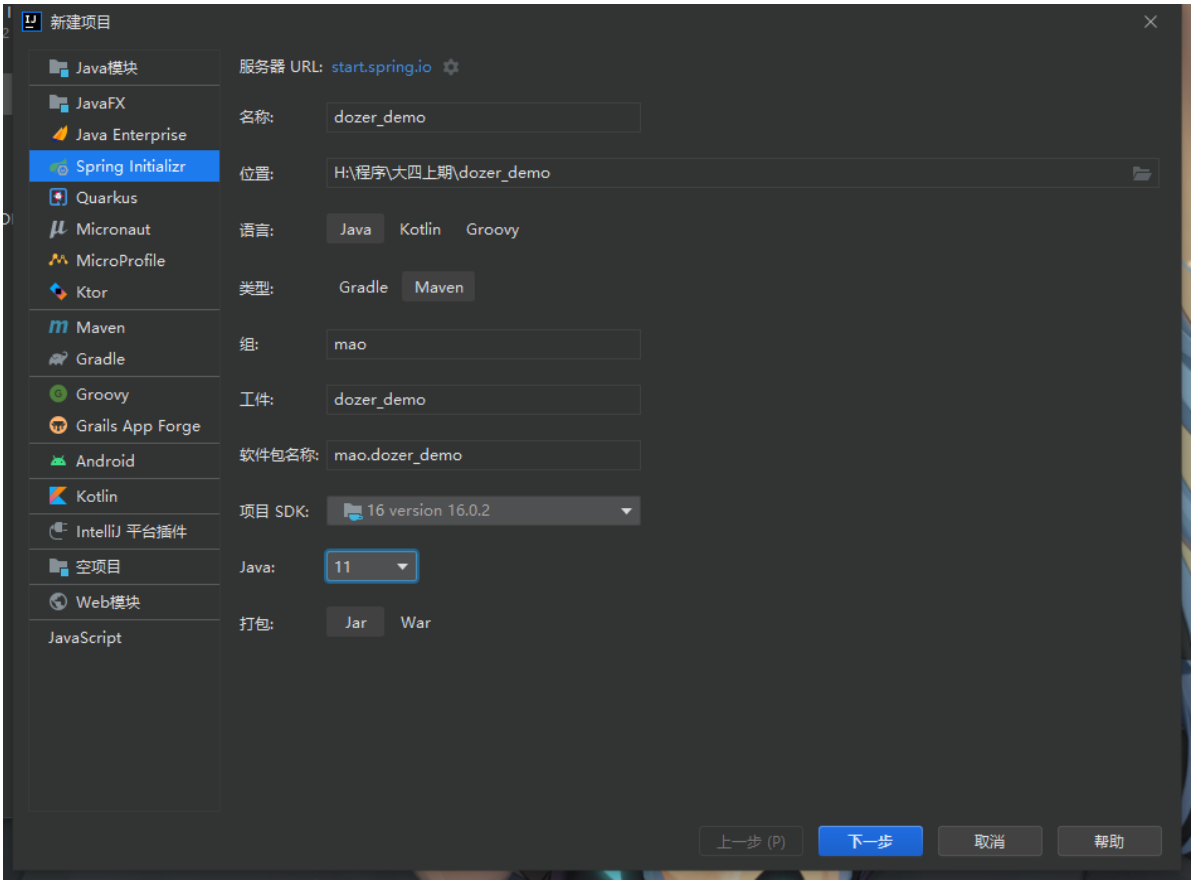
```
<dependency>
    <groupId>com.github.dozermapper</groupId>
    <artifactId>dozer-core</artifactId>
    <version>6.5.0</version>
</dependency>
```

为了简化使用方式，dozer还提供了starter，其maven坐标为：

```
<dependency>
    <groupId>com.github.dozermapper</groupId>
    <artifactId>dozer-spring-boot-starter</artifactId>
    <version>6.5.0</version>
</dependency>
```

dozer入门案例

第一步：创建项目dozer_demo



第二步：修改pom文件

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         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>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>mao</groupId>
    <artifactId>dozer_demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>dozer_demo</name>
    <description>dozer_demo</description>

    <properties>
        <java.version>11</java.version>
    </properties>

    <dependencies>
```

```
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>com.github.dozermapper</groupId>
            <artifactId>dozer-spring-boot-starter</artifactId>
            <version>6.5.0</version>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

第三步：创建UserEntity

```
package mao.dozer_demo.entity;

import java.util.Date;

/**
 * Project name(项目名称)：dozer_demo
 * Package(包名): mao.dozer_demo.entity
 * Class(类名): UserEntity
 * Author(作者）: mao
 * Author QQ：1296193245
 * GitHub：https://github.com/maomao124/
 * Date(创建日期)： 2022/10/28
 * Time(创建时间)： 19:37
 * Version(版本): 1.0
 * Description(描述)： 无
 */


```

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

```
76          return id;
77      }
78
79      /**
80       * Sets id.
81       *
82       * @param id the id
83       */
84      public void setId(String id)
85      {
86          this.id = id;
87      }
88
89      /**
90       * Gets name.
91       *
92       * @return the name
93       */
94      public String getName()
95      {
96          return name;
97      }
98
99      /**
100      * Sets name.
101      *
102      * @param name the name
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      *
122      * @param age the age
123      */
124     public void setAge(int age)
125     {
126         this.age = age;
127     }
128
129     /**
130      * Gets address.
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     /**
150      * Gets birthday.
151      *
152      * @return the birthday
153      */
154     public Date getBirthday()
155     {
156         return birthday;
157     }
158 
159     /**
160      * Sets birthday.
161      *
162      * @param birthday the birthday
163      */
164     public void setBirthday(Date birthday)
165     {
166         this.birthday = birthday;
167     }
168 
169     @Override
170     @SuppressWarnings("all")
171     public String toString()
172     {
173         final StringBuilder stringbuilder = new StringBuilder();
174         stringbuilder.append("id: ").append(id).append('\n');
175         stringbuilder.append("name: ").append(name).append('\n');
176         stringbuilder.append("age: ").append(age).append('\n');
177         stringbuilder.append("address: ").append(address).append('\n');
178         stringbuilder.append("birthday: ").append(birthday).append('\n');
179         return stringbuilder.toString();
180     }
181 }
```

第四步：创建UserDTO

```java
package mao.dozer_demo.entity;

/**
 * Project name(项目名称)：dozer_demo
 * Package(包名): mao.dozer_demo.entity
 * Class(类名): UserDTO
 * Author(作者）: mao
 * Author QQ：1296193245
 * GitHub：https://github.com/maomao124/
 * Date(创建日期)： 2022/10/28
 * Time(创建时间)： 19:39
 * Version(版本): 1.0
 * Description(描述)： 无
 */


public class UserDTO
{
    /**
     * 用户id
     */
    private String userId;
    /**
     * 用户名
     */
    private String userName;
    /**
     * 用户年龄
     */
    private int userAge;
    /**
     * 地址
     */
    private String address;
    /**
     * 生日
     */
    private String birthday;

    /**
     * Instantiates a new User dto.
     */
    public UserDTO()
    {
    }

    /**
     * Instantiates a new User dto.
     *
     * @param userId   the user id
     * @param userName the user name
     * @param userAge  the user age
     * @param address  the address
```

```
54          * @param birthday the birthday
55          */
56         public UserDTO(String userId, String userName, int userAge, String
    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          *
68          * @return the user id
69          */
70         public String getUserId()
71         {
72             return userId;
73         }
74
75         /**
76          * Sets user id.
77          *
78          * @param userId the user id
79          */
80         public void setUserId(String userId)
81         {
82             this.userId = userId;
83         }
84
85         /**
86          * Gets user name.
87          *
88          * @return the user name
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        {
102            this.userName = userName;
103        }
104
105        /**
106         * Gets user age.
107         *
108         * @return the user age
109         */
110        public int getUserAge()
```

```
111     {
112         return userAge;
113     }
114 
115     /**
116      * Sets user age.
117      *
118      * @param userAge the user age
119      */
120     public void setUserAge(int userAge)
121     {
122         this.userAge = userAge;
123     }
124 
125     /**
126      * Gets address.
127      *
128      * @return the address
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     /**
146      * Gets birthday.
147      *
148      * @return the birthday
149      */
150     public String getBirthday()
151     {
152         return birthday;
153     }
154 
155     /**
156      * Sets birthday.
157      *
158      * @param birthday the birthday
159      */
160     public void setBirthday(String birthday)
161     {
162         this.birthday = birthday;
163     }
164 
165     @Override
166     @SuppressWarnings("all")
167     public String toString()
168     {
```

```
        final StringBuilder stringbuilder = new StringBuilder();
        stringbuilder.append("userId: ").append(userId).append('\n');
        stringbuilder.append("userName: ").append(userName).append('\n');
        stringbuilder.append("userAge: ").append(userAge).append('\n');
        stringbuilder.append("address: ").append(address).append('\n');
        stringbuilder.append("birthday: ").append(birthday).append('\n');
        return stringbuilder.toString();
    }
}
```

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"?>
<mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://dozermapper.github.io/schema/bean-mapping"
          xsi:schemaLocation="http://dozermapper.github.io/schema/bean-mapping
                              http://dozermapper.github.io/schema/bean-mapping.xsd">
    <!--
    全局配置:
    <date-format>表示日期格式
     -->
    <configuration>
        <date-format>yyyy-MM-dd</date-format>
    </configuration>


</mappings>
```

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

```
<mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://dozermapper.github.io/schema/bean-mapping"
          xsi:schemaLocation="http://dozermapper.github.io/schema/bean-mapping
                              http://dozermapper.github.io/schema/bean-mapping.xsd">

    <!--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-->
```

```
    <mapping date-format="yyyy-MM-dd">
        <class-a>mao.dozer_demo.entity.UserEntity</class-a>
        <class-b>mao.dozer_demo.entity.UserDTO</class-b>

        <field>
            <a>id</a>
            <b>userId</b>
        </field>
        <field>
            <a>name</a>
            <b>userName</b>
        </field>
        <field>
            <a>age</a>
            <b>userAge</b>
        </field>

    </mapping>

    <mapping date-format="yyyy-MM-dd" map-id="user">
        <class-a>mao.dozer_demo.entity.UserEntity</class-a>
        <class-b>mao.dozer_demo.entity.UserDTO</class-b>

        <field>
            <a>id</a>
            <b>userId</b>
        </field>
        <field>
            <a>name</a>
            <b>userName</b>
        </field>
        <field>
            <a>age</a>
            <b>userAge</b>
        </field>

    </mapping>



</mappings>
```

第七步：编写配置类

```java
@Configuration
public class DozerMapperConfig
{
    @Bean
    public DozerBeanMapperFactoryBean
dozerMapper(@Value("classpath:dozer/*.xml") Resource[] resources)
            throws IOException
    {
        DozerBeanMapperFactoryBean dozerBeanMapperFactoryBean = new
DozerBeanMapperFactoryBean();
        dozerBeanMapperFactoryBean.setMappingFiles(resources);
        return dozerBeanMapperFactoryBean;
    }
}
```

第八步：编写单元测试类

```java
package mao.dozer_demo;

import com.github.dozermapper.core.Mapper;
import mao.dozer_demo.entity.UserDTO;
import mao.dozer_demo.entity.UserEntity;
import org.junit.jupiter.api.Test;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import java.util.Date;

/**
 * Project name(项目名称)：dozer_demo
 * Package(包名): mao.dozer_demo
 * Class(类名): DozerTest
 * Author(作者）: mao
 * Author QQ：1296193245
 * GitHub：https://github.com/maomao124/
 * Date(创建日期)： 2022/10/28
 * Time(创建时间)： 19:50
 * Version(版本): 1.0
 * Description(描述)： 无
 */

@SpringBootTest
public class DozerTest
{

    @Autowired
    private Mapper mapper;

```

```
    private static final Logger log =
LoggerFactory.getLogger(DozerTest.class);

    @Test
    void test1()
    {
        UserDTO userDTO = new UserDTO();
        userDTO.setUserId("134");
        userDTO.setUserName("张三");
        userDTO.setUserAge(19);
        userDTO.setAddress("中国");
        userDTO.setBirthday("2012-10-13");

        log.info(userDTO.toString());

        UserEntity userEntity = mapper.map(userDTO, UserEntity.class);

        log.info(userEntity.toString());

    }

    @Test
    void test2()
    {
        UserEntity userEntity = new UserEntity("1234", "李四", 14, "中国",
new Date());
        UserDTO userDTO = mapper.map(userEntity, UserDTO.class);

        log.info(userEntity.toString());
        log.info(userDTO.toString());
    }
}
```

测试1：

```
2022-10-28 20:32:37.318  INFO 15068 --- [           main]
mao.dozer_demo.DozerTest                 : userId：134
userName：张三
userAge：19
address：中国
birthday：2012-10-13

2022-10-28 20:32:37.326  INFO 15068 --- [           main]
mao.dozer_demo.DozerTest                 : id：134
name：张三
age：19
address：中国
birthday：Sat Oct 13 00:00:00 CST 2012
```

测试2：

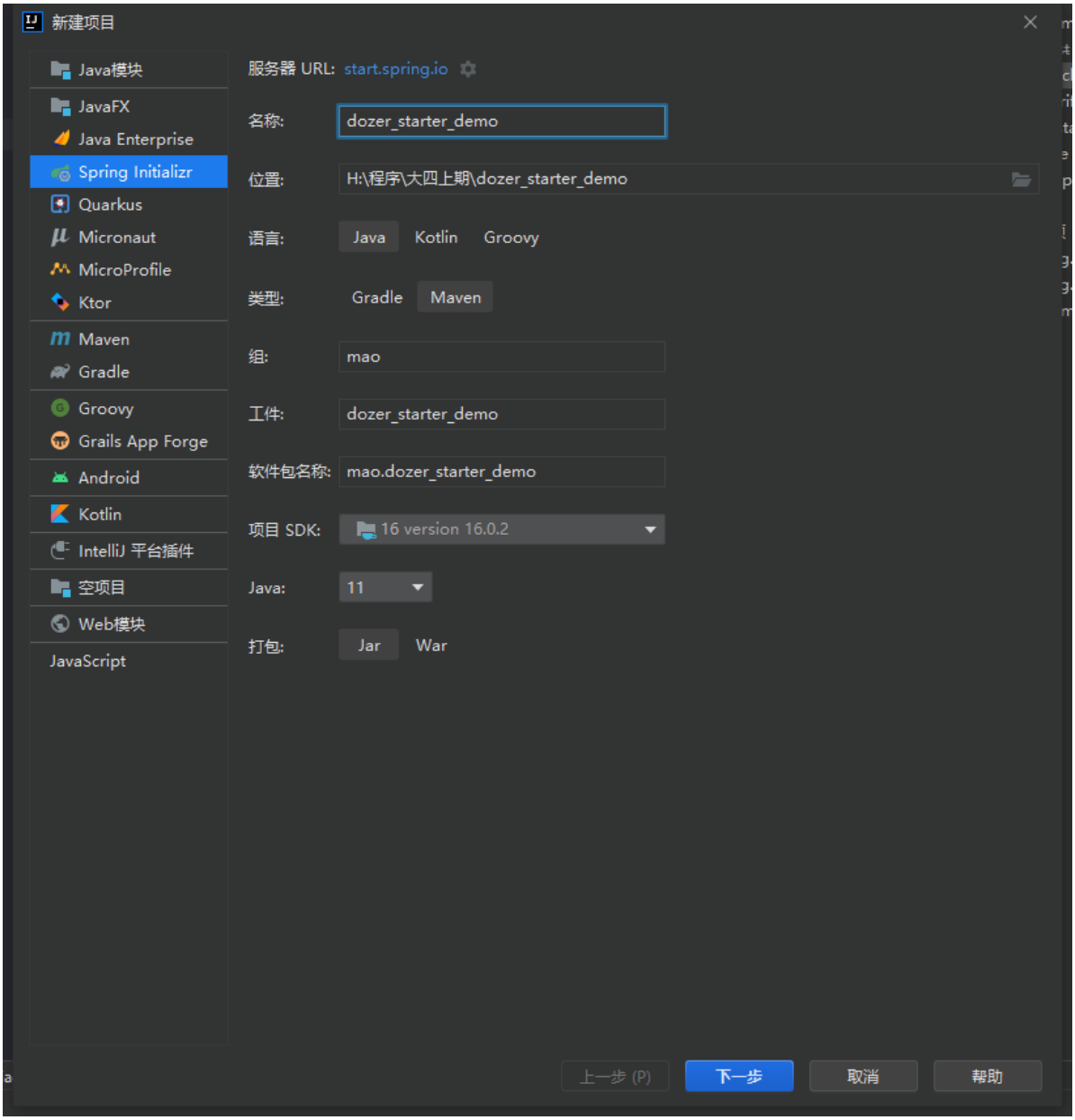
```
2022-10-28 20:33:08.781  INFO 1252 --- [           main] mao.dozer_demo.DozerTest                 : id：1234
name：李四
age：14
address：中国
birthday：Fri Oct 28 20:33:08 CST 2022

2022-10-28 20:33:08.781  INFO 1252 --- [           main] mao.dozer_demo.DozerTest                 : userId：1234
userName：李四
userAge：14
address：中国
birthday：2022-10-28
```

自定义spring boot starter

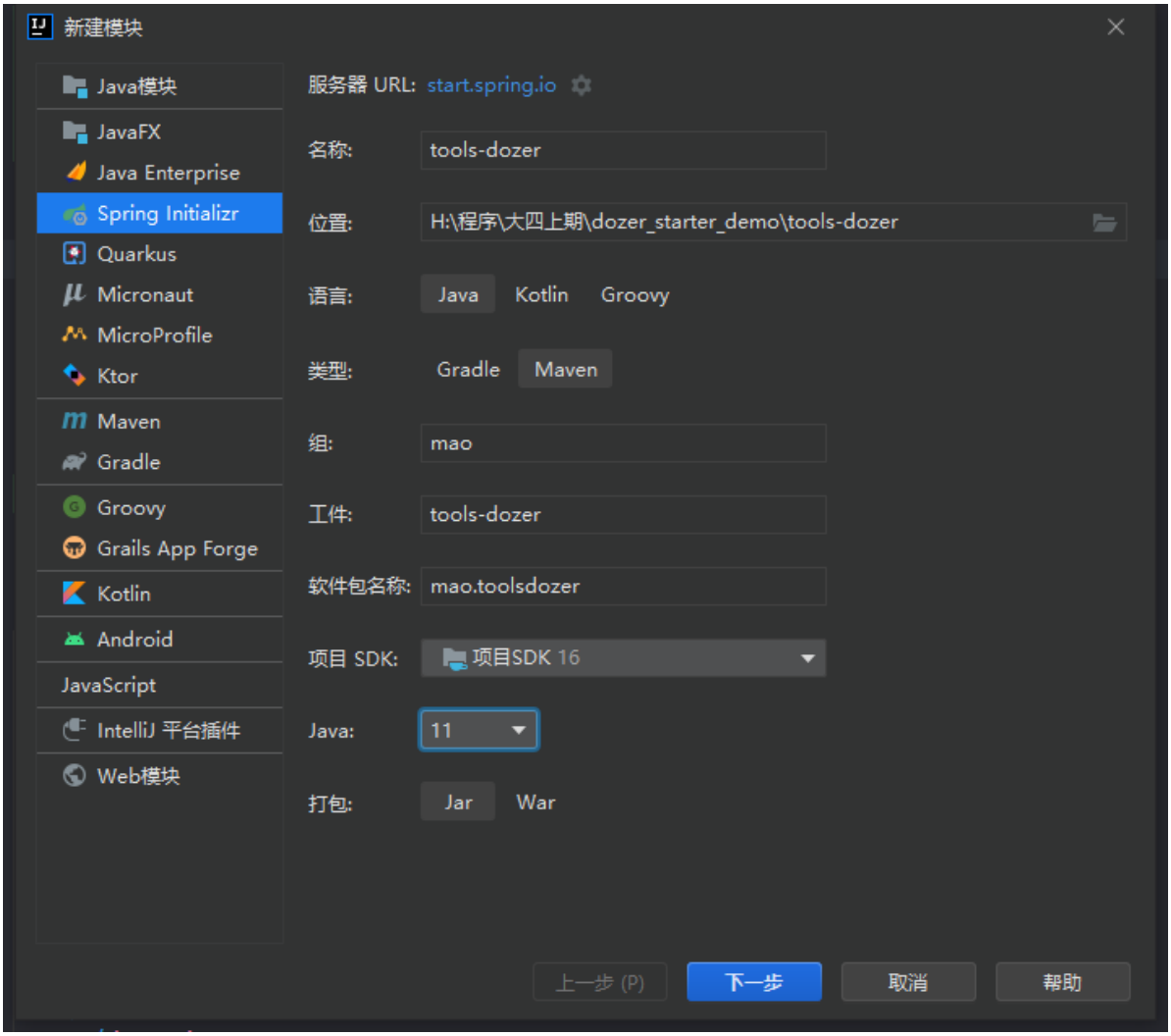
开发starter

第一步：初始化项目

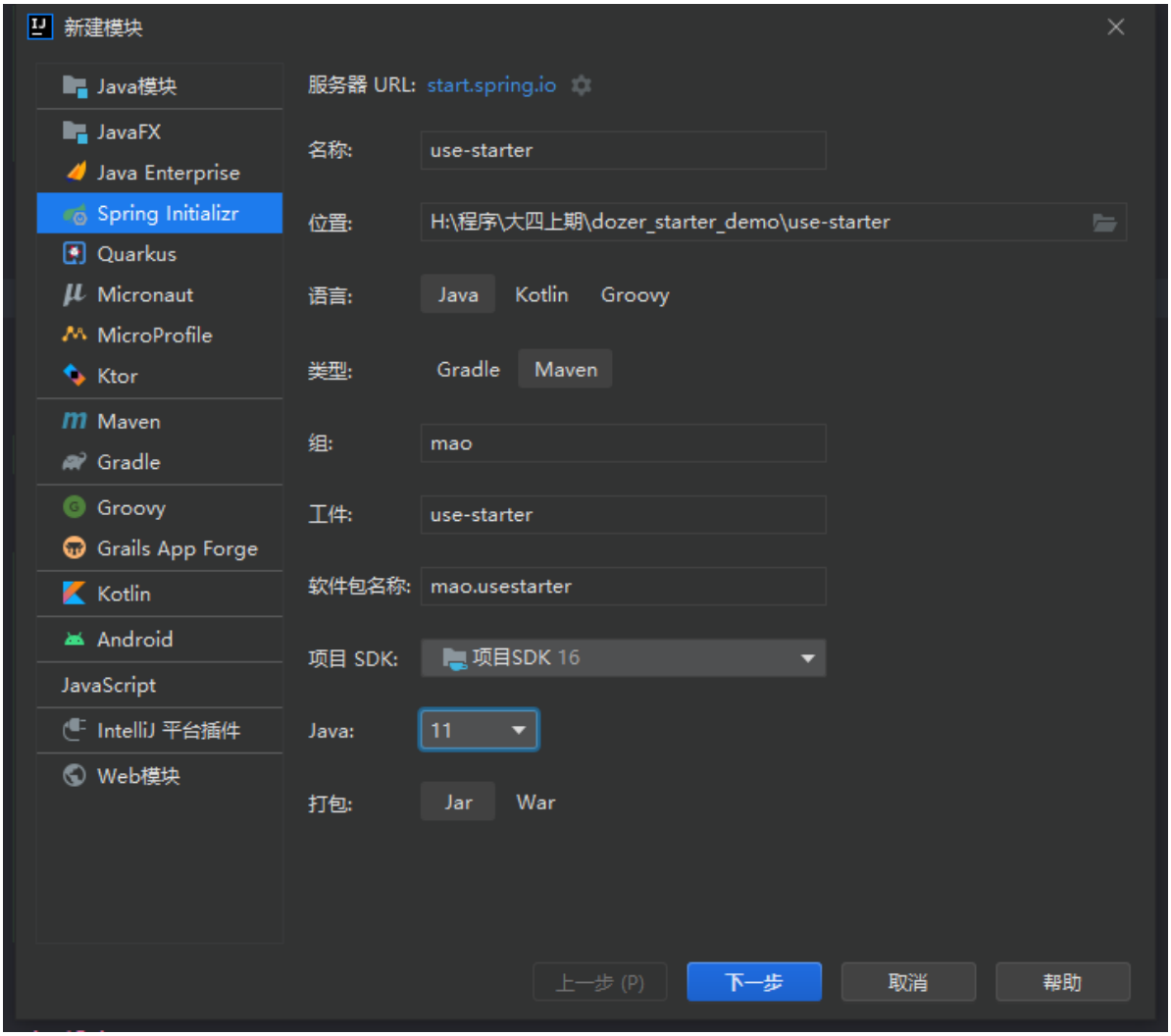
创建父工程dozer_starter_demo



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"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         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>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <groupId>mao</groupId>
```

```
    <artifactId>dozer_starter_demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>dozer_starter_demo</name>
    <description>dozer_starter_demo</description>
    <packaging>pom</packaging>

    <properties>
        <java.version>11</java.version>
    </properties>

    <modules>
        <module>tools-dozer</module>
        <module>use-starter</module>
    </modules>


    <dependencies>

    </dependencies>

    <dependencyManagement>

        <dependencies>

        </dependencies>

    </dependencyManagement>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

子工程tools-dozer的pom文件:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         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>

    <parent>
        <artifactId>dozer_starter_demo</artifactId>
        <groupId>mao</groupId>
        <version>0.0.1-SNAPSHOT</version>
    </parent>

```

```xml
    <artifactId>tools-dozer</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>tools-dozer</name>
    <description>tools-dozer</description>

    <properties>

    </properties>

    <dependencies>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>com.github.dozermapper</groupId>
            <artifactId>dozer-spring-boot-starter</artifactId>
            <version>6.5.0</version>
        </dependency>

        <!--spring boot starter开发依赖-->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-autoconfigure</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-configuration-processor</artifactId>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <skip>true</skip>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>

```

子工程use-starter的pom文件：

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         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>

    <parent>
        <artifactId>dozer_starter_demo</artifactId>
        <groupId>mao</groupId>
        <version>0.0.1-SNAPSHOT</version>
    </parent>

    <artifactId>use-starter</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>use-starter</name>
    <description>use-starter</description>

    <properties>

    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

第三步：编写工具类DozerUtils

```java
package mao.toolsdozer.utils;

import com.github.dozermapper.core.Mapper;

import java.util.*;
import java.util.stream.Collectors;

/**
 * Project name(项目名称)：dozer_starter_demo
 * Package(包名): mao.toolsdozer.utils
 * Class(类名): DozerUtils
 * Author(作者）: mao
 * Author QQ：1296193245
 * GitHub：https://github.com/maomao124/
 * Date(创建日期)： 2022/10/28
 * Time(创建时间)： 21:57
 * Version(版本): 1.0
 * Description(描述)： 无
 */

public class DozerUtils
{
    private final Mapper mapper;

    public DozerUtils(Mapper mapper)
    {
        this.mapper = mapper;
    }

    public Mapper getMapper()
    {
        return this.mapper;
    }

    /**
     * 地图
     * Constructs new instance of destinationClass and performs mapping between from source
     *
     * @param source           源
     * @param destinationClass 目标类
     * @return {@link T}
     */
    public <T> T map(Object source, Class<T> destinationClass)
    {
        if (source == null)
        {
            return null;
        }
        return mapper.map(source, destinationClass);
    }

    /**
```

```
     * map2
     *
     * @param source           源
     * @param destinationClass 目标类
     * @return {@link T}
     */
    public <T> T map2(Object source, Class<T> destinationClass)
    {
        if (source == null)
        {
            try
            {
                return destinationClass.newInstance();
            }
            catch (Exception ignored)
            {
            }
        }
        return mapper.map(source, destinationClass);
    }

    /**
     * 地图
     * Performs mapping between source and destination objects
     *
     * @param source      源
     * @param destination 目地
     */
    public void map(Object source, Object destination)
    {
        if (source == null)
        {
            return;
        }
        mapper.map(source, destination);
    }

    /**
     *
     * Constructs new instance of destinationClass and performs mapping between from source
     *
     * @param source           源
     * @param destinationClass 目标类
     * @param mapId            mapId
     * @return {@link T}
     */
    public <T> T map(Object source, Class<T> destinationClass, String mapId)
    {
        if (source == null)
        {
            return null;
        }
        return mapper.map(source, destinationClass, mapId);
    }

    /**
```

```
109      *
110      * Performs mapping between source and destination objects
111      *
112      * @param source      源
113      * @param destination 目标
114      * @param mapId       mapId
115      */
116     public void map(Object source, Object destination, String mapId)
117     {
118         if (source == null)
119         {
120             return;
121         }
122         mapper.map(source, destination, mapId);
123     }
124 
125     /**
126      *
127      * 将集合转成集合
128      * List<A> -->  List<B>
129      *
130      * @param sourceList       源集合
131      * @param destinationClass 待转类型
132      * @return {@link List}<{@link T}>
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         }
146         return sourceList.stream()
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     {
154         if (sourceList == null || sourceList.isEmpty() || destinationClass
== null)
155         {
156             return Collections.emptySet();
157         }
158         return sourceList.stream().map((sourceObject) ->
mapper.map(sourceObject, destinationClass)).collect(Collectors.toSet());
159     }
```

```
160 }
```

第四步：编写配置类DozerAutoConfiguration

```
1  package mao.toolsdozer.config;
2
3  import com.github.dozermapper.core.Mapper;
4  import com.github.dozermapper.spring.DozerBeanMapperFactoryBean;
5  import mao.toolsdozer.utils.DozerUtils;
6  import org.springframework.beans.factory.annotation.Autowired;
7  import org.springframework.beans.factory.annotation.Value;
8  import org.springframework.context.annotation.Bean;
9
10 import org.springframework.context.annotation.Configuration;
11
12 import org.springframework.core.io.Resource;
13
14
15 import java.io.IOException;
16
17 /**
18  * Project name(项目名称)：dozer_starter_demo
19  * Package(包名): mao.toolsdozer.config
20  * Class(类名): DozerAutoConfiguration
21  * Author(作者）: mao
22  * Author QQ：1296193245
23  * GitHub：https://github.com/maomao124/
24  * Date(创建日期)： 2022/10/28
25  * Time(创建时间)： 22:03
26  * Version(版本): 1.0
27  * Description(描述)： 无
28  */
29
30 @Configuration
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
42     @Autowired
43     private Mapper mapper;
44
45     @Bean
```

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

}

```

第五步：创建并编写spring.factories文件

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

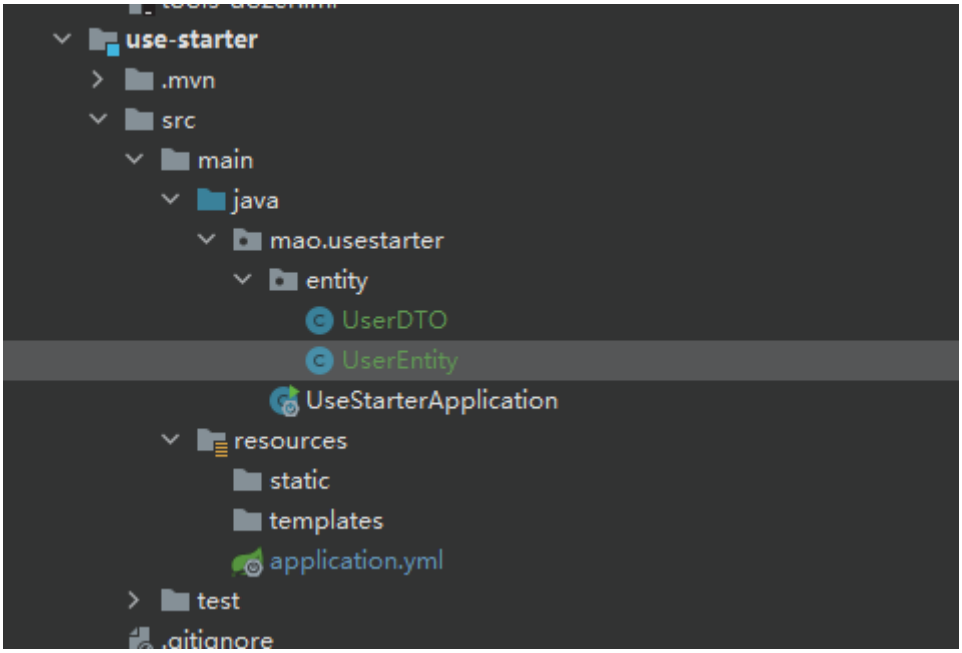
使用starter

第一步：导入tools-dozer的依赖

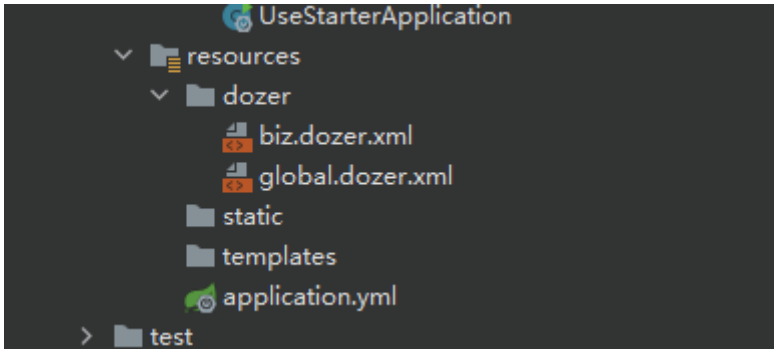
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <parent>
        <artifactId>dozer_starter_demo</artifactId>
        <groupId>mao</groupId>
        <version>0.0.1-SNAPSHOT</version>
    </parent>

    <artifactId>use-starter</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>use-starter</name>
    <description>use-starter</description>

    <properties>

```

```
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>mao</groupId>
            <artifactId>tools-dozer</artifactId>
            <version>0.0.1-SNAPSHOT</version>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

第二步：拷贝之前编写的User类



第三步：拷贝配置文件



biz.dozer.xml：

```
<mappings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://dozermapper.github.io/schema/bean-mapping"
          xsi:schemaLocation="http://dozermapper.github.io/schema/bean-mapping
                              http://dozermapper.github.io/schema/bean-mapping.xsd">

    <!--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-->

    <mapping date-format="yyyy-MM-dd">
        <class-a>mao.usestarter.entity.UserEntity</class-a>
        <class-b>mao.usestarter.entity.UserDTO</class-b>

        <field>
            <a>id</a>
            <b>userId</b>
```

```
        </field>
        <field>
            <a>name</a>
            <b>userName</b>
        </field>
        <field>
            <a>age</a>
            <b>userAge</b>
        </field>

    </mapping>

    <mapping date-format="yyyy-MM-dd" map-id="user">
        <class-a>mao.usestarter.entity.UserEntity</class-a>
        <class-b>mao.usestarter.entity.UserDTO</class-b>

        <field>
            <a>id</a>
            <b>userId</b>
        </field>
        <field>
            <a>name</a>
            <b>userName</b>
        </field>
        <field>
            <a>age</a>
            <b>userAge</b>
        </field>

    </mapping>



</mappings>

```

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

```java
package mao.usestarter.controller;

import mao.toolsdozer.utils.DozerUtils;
import mao.usestarter.entity.UserDTO;
import mao.usestarter.entity.UserEntity;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;


import java.util.Date;

/**
 * Project name(项目名称)：dozer_starter_demo
 * Package(包名): mao.usestarter.controller
 * Class(类名): UserController
 * Author(作者）: mao
 * Author QQ：1296193245
 * GitHub：https://github.com/maomao124/
 * Date(创建日期)： 2022/10/28
 * Time(创建时间)： 22:18
 * Version(版本): 1.0
 * Description(描述)： 无
 */


@RestController
public class UserController
{

    @Autowired
    private DozerUtils dozerUtils;

    private static final Logger log =
LoggerFactory.getLogger(UserController.class);

    @GetMapping("/test1")
    public UserDTO test1()
    {
        log.info("test1");
        UserEntity userEntity = new UserEntity("1234", "张三", 13, "中国",
new Date());
        log.info(userEntity.toString());
        UserDTO userDTO = dozerUtils.map(userEntity, UserDTO.class);
        log.info(userDTO.toString());
        return userDTO;
    }

    @GetMapping("test2")
    public UserEntity test2()
    {
```

```
        log.info("test2");
        UserDTO userDTO = new UserDTO("1234", "张三", 13, "中国", "2011-07-
23");
        log.info(userDTO.toString());
        UserEntity userEntity = dozerUtils.map(userDTO, UserEntity.class);
        log.info(userEntity.toString());
        return userEntity;
    }
}
```

第六步：启动程序

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.1)

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 --- [           main]
mao.usestarter.UseStarterApplication     : No active profile set, falling
back to 1 default profile: "default"
2022-10-28 23:36:56.285  INFO 1684 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s):
8080 (http)
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 --- [           main]
org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
Tomcat/9.0.64]
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 --- [           main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 680 ms
2022-10-28 23:36:56.418  INFO 1684 --- [           main]
c.g.d.core.DozerBeanMapperBuilder        : Initializing Dozer. Version:
6.5.0, Thread Name: main
2022-10-28 23:36:56.419  INFO 1684 --- [           main]
c.g.dozermapper.core.util.RuntimeUtils   : OSGi support is false
2022-10-28 23:36:56.422  INFO 1684 --- [           main]
d.c.c.r.LegacyPropertiesSettingsResolver : Trying to find Dozer
configuration file: dozer.properties
```

```
2022-10-28 23:36:56.426  INFO 1684 --- [           main] d.c.c.r.LegacyPropertiesSettingsResolver : Failed to find dozer.properties via com.github.dozermapper.core.config.resolvers.LegacyPropertiesSettingsResolver.
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
2022-10-28 23:36:56.437  INFO 1684 --- [           main] 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_starter_demo/use-starter/target/classes/dozer/global.dozer.xml] to load custom xml mappings
2022-10-28 23:36:56.569  INFO 1684 --- [           main] 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 --- [           main] 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 --- [           main] c.g.d.c.b.xml.BeanMappingXMLBuilder      : Successfully loaded custom xml mapping.
2022-10-28 23:36:56.590  INFO 1684 --- [           main] 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_starter_demo/use-starter/target/classes/dozer/biz.dozer.xml] to load custom xml mappings
2022-10-28 23:36:56.594  INFO 1684 --- [           main] 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 23:36:56.594  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]
2022-10-28 23:36:56.605  INFO 1684 --- [           main] c.g.d.c.b.xml.BeanMappingXMLBuilder      : Successfully loaded custom xml mapping.
2022-10-28 23:36:56.829  INFO 1684 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2022-10-28 23:36:56.841  INFO 1684 --- [           main] 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"}

```
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
age：13
address：中国
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
userName：张三
userAge：13
address：中国
birthday：2022-10-28

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] m.usestarter.controller.UserController   : userId：1234
userName：张三
userAge：13
address：中国
birthday：2011-07-23

2022-10-28 23:38:43.111  INFO 1684 --- [nio-8080-exec-2] m.usestarter.controller.UserController   : id：1234
name：张三
age：13
address：中国
birthday：Sat Jul 23 00:00:00 CST 2011
```

end

by mao

2022 10 28