

# Mapper.xml

MyBatis 框架主要有两个配置文件：config.xml 和 Mapper.xml，这两个配置文件都可以自定义文件名。

config.xml 是全局环境配置文件，主要配置 MyBatis 框架的数据源(DataSource)，事务管理(TransactionManager)，是否打印 SQL 语句，开启二级缓存，设置实体类别名等功能。

Mapper.xml 的作用？

MyBatis 框架是一个“半自动”的 ORM 框架，SQL 语句需要开发者自定义，MyBatis 框架的关注点在于 Java Bean 于 SQL 之间的映射关系，自定义的 SQL 语句定义在 Mapper.xml 中。

Mapper.xml 常用属性：

- parameterType 参数数据类型
  - 基本数据类型，通过 id 查询 User。

UserRepository

```
//通过 id 查询 User
public User findById(int id);
```

UserRepository.xml

```
<select id="findById" parameterType="int"
resultType="com.southwind.entity.User">
    select * from t_user where id = #{id}
</select>
```

- String 类型，通过 username 查询 User。

UserRepository

```
//通过 username 查询 User
public User findByUsername(String username);
```

UserRepository.xml

```
<select id="findByUsername" parameterType="java.lang.String"
resultType="com.southwind.entity.User">
    select * from t_user where username = #{username}
</select>
```

- 包装类，通过 age 查询 User。

UserRepository

```
//通过 age 查询 User
public User findByAge(Integer age);
```

UserRepository.xml

```
<select id="findByAge" parameterType="java.lang.Integer"
resultType="com.southwind.entity.User">
    select * from t_user where age = #{age}
</select>
```

- 多个参数，通过 username 和 age 查询 User。两个参数分别是 String 类型和 int 类型，类型不一致，所以此时 parameterType 可以省略，通过参数下标取出参数值。

UserRepository

```
//通过username和age查询User
public User findByUsernameAndAge(String username,int age);
```

UserRepository.xml

```
<select id="findByUsernameAndAge"
resultType="com.southwind.entity.User">
    select * from t_user where username = #{param1} and age = #{param2}
</select>
```

通过下标取出对应的参数，下标从1开始，#{param1} 表示第一个参数，#{param2} 表示第二个参数，以此类推。

- JavaBean，当同时存在多个参数时，通过下标一个一个写很麻烦，这时候可以将参数列表进行封装，将封装的对象作为 parameterType 的值。

UserRepository

```
//通过User封装对象查询User
public User findByUser(User user);
```

UserRepository.xml

```
<select id="findByUser" parameterType="com.southwind.entity.User"
resultType="com.southwind.entity.User">
    select * from t_user where username = #{username} and age = #{age}
</select>
```

- resultType: 结果类型
  - 基本数据类型，统计 User 总数。

## UserRepository

```
//统计User总数  
public int count();
```

## UserRepository.xml

```
<select id="count" resultType="int">  
    select count(*) from user  
</select>
```

- 包装类，统计 User 总数。

## UserRepository

```
//统计User总数  
public Integer count();
```

## UserRepository.xml

```
<select id="count" resultType="java.lang.Integer">  
    select count(*) from t_user  
</select>
```