**静态文件生成文档**

# 一、添加依赖

|  |
| --- |
| <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.37</version>  </dependency>  <dependency>  <groupId>org.mybatis</groupId>  <artifactId>mybatis</artifactId>  <version>3.2.8</version>  </dependency>  <dependency>  <groupId>com.google.code.gson</groupId>  <artifactId>gson</artifactId>  <version>2.2.4</version>  </dependency> |

# 二、Javabean



# 三、db.properties

|  |
| --- |
| driver=com.mysql.jdbc.Driver  url=jdbc:mysql://localhost:3306/atguigu\_bis  username=root  password=root |

# 四、mybatis\_config.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE configuration SYSTEM "http://mybatis.org/dtd/mybatis-3-config.dtd" >  <configuration>  <properties resource=*"db.properties"* />  <environments default=*"development"*>  <environment id=*"development"*>  <transactionManager type=*"JDBC"* />  <dataSource type=*"POOLED"*>  <property name=*"driver"* value=*"${driver}"* />  <property name=*"url"* value=*"${url}"* />  <property name=*"username"* value=*"${username}"* />  <property name=*"password"* value=*"${password}"* />  </dataSource>  </environment>  </environments>  <mappers>  <package name=*"com.atguigu.mapper"* />  </mappers>  </configuration> |

# 五、XXXMapper.java & XXXMapper.xml

|  |
| --- |
| package com.atguigu.mapper;  import java.util.List;  import com.atguigu.bean.T\_MALL\_CLASS\_1;  public interface ClassMapper {  public List<T\_MALL\_CLASS\_1> select\_class\_1();  } |

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE mapper SYSTEM "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >  <mapper namespace=*"com.atguigu.mapper.ClassMapper"*>  <select id=*"select\_class\_1"* resultType=*"com.atguigu.bean.T\_MALL\_CLASS\_1"*>  select \* from  t\_mall\_class\_1  </select>  </mapper> |

# 六、创建MySqlSessionFactory

|  |
| --- |
| package com.atguigu.factory;  import java.io.IOException;  import java.io.InputStream;  import org.apache.ibatis.io.Resources;  import org.apache.ibatis.session.SqlSessionFactory;  import org.apache.ibatis.session.SqlSessionFactoryBuilder;  public class MySqlSessionFactory {  public static SqlSessionFactory getMyFactory() {  String resource = "mybatis-config.xml";  InputStream inputStream = null;  try {  inputStream = Resources.getResourceAsStream(resource);  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);  return sqlSessionFactory;  }  } |

# 七、创建Test类 ，生成静态js

|  |
| --- |
| package com.atguigu.test;  import java.io.File;  import java.io.FileOutputStream;  import java.util.List;  import org.apache.ibatis.session.SqlSessionFactory;  import com.atguigu.bean.T\_MALL\_CLASS\_1;  import com.atguigu.factory.MySqlSessionFactory;  import com.atguigu.mapper.ClassMapper;  import com.google.gson.Gson;  public class Test01 {  public static void main(String[] args) {  SqlSessionFactory myFactory = MySqlSessionFactory.getMyFactory();  ClassMapper mapper = myFactory.openSession().getMapper(ClassMapper.class);  List<T\_MALL\_CLASS\_1> select\_class\_1 = mapper.select\_class\_1();  Gson gson = new Gson();  String json = gson.toJson(select\_class\_1);  try {  FileOutputStream fos = new FileOutputStream(new File("d:/class\_1.js"));  fos.write(json.getBytes("utf-8"));  } catch (Exception e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  System.out.println(select\_class\_1);  }  } |