# 添加druid的连接池

|  |
| --- |
| <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>druid</**artifactId**>  <**version**>1.0.25</**version**> </**dependency**> |

# 添加application.properties

|  |
| --- |
| **spring.datasource.type**=**com.alibaba.druid.pool.DruidDataSource  spring.datasource.url** = **jdbc:mysql://localhost:3306/edu1?useUnicode=true&characterEncoding=utf-8 spring.datasource.username** = **root spring.datasource.password** = **123456 spring.datasource.driverClassName** = **com.mysql.jdbc.Driver** *#连接池的配置信息* **spring.datasource.initialSize**=**5 spring.datasource.minIdle**=**5 spring.datasource.maxActive**=**20 spring.datasource.maxWait**=**60000 spring.datasource.timeBetweenEvictionRunsMillis**=**60000 spring.datasource.minEvictableIdleTimeMillis**=**300000 spring.datasource.validationQuery**=**SELECT 1 FROM DUAL spring.datasource.testWhileIdle**=**true spring.datasource.testOnBorrow**=**false spring.datasource.testOnReturn**=**false spring.datasource.poolPreparedStatements**=**true spring.datasource.maxPoolPreparedStatementPerConnectionSize**=**20 spring.datasource.filters**=**stat,wall,log4j spring.datasource.connectionProperties**=**druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000** |

# 使用注解来配置数据源

|  |
| --- |
| **package** com.gyf.dbconfig;  **import** com.alibaba.druid.pool.DruidDataSource; **import** org.apache.log4j.Logger; **import** org.slf4j.LoggerFactory; **import** org.springframework.beans.factory.annotation.Value; **import** org.springframework.context.annotation.Bean; **import** org.springframework.context.annotation.Configuration; **import** org.springframework.context.annotation.Primary;  **import** javax.sql.DataSource; **import** java.sql.SQLException;  @Configuration **public class** DruidDBConfig {  **private** Logger **logger** = Logger.*getLogger*(DruidDBConfig.**class**);   @Value(**"${spring.datasource.url}"**)  **private** String **dbUrl**;   @Value(**"${spring.datasource.username}"**)  **private** String **username**;   @Value(**"${spring.datasource.password}"**)  **private** String **password**;   @Value(**"${spring.datasource.driverClassName}"**)  **private** String **driverClassName**;   @Value(**"${spring.datasource.initialSize}"**)  **private int initialSize**;   @Value(**"${spring.datasource.minIdle}"**)  **private int minIdle**;   @Value(**"${spring.datasource.maxActive}"**)  **private int maxActive**;   @Value(**"${spring.datasource.maxWait}"**)  **private int maxWait**;   @Value(**"${spring.datasource.timeBetweenEvictionRunsMillis}"**)  **private int timeBetweenEvictionRunsMillis**;   @Value(**"${spring.datasource.minEvictableIdleTimeMillis}"**)  **private int minEvictableIdleTimeMillis**;   @Value(**"${spring.datasource.validationQuery}"**)  **private** String **validationQuery**;   @Value(**"${spring.datasource.testWhileIdle}"**)  **private boolean testWhileIdle**;   @Value(**"${spring.datasource.testOnBorrow}"**)  **private boolean testOnBorrow**;   @Value(**"${spring.datasource.testOnReturn}"**)  **private boolean testOnReturn**;   @Value(**"${spring.datasource.poolPreparedStatements}"**)  **private boolean poolPreparedStatements**;   @Value(**"${spring.datasource.maxPoolPreparedStatementPerConnectionSize}"**)  **private int maxPoolPreparedStatementPerConnectionSize**;   @Value(**"${spring.datasource.filters}"**)  **private** String **filters**;   @Value(**"{spring.datasource.connectionProperties}"**)  **private** String **connectionProperties**;   @Bean *//声明其为Bean实例* @Primary *//在同样的DataSource中，首先使用被标注的DataSource* **public** DataSource dataSource(){  DruidDataSource datasource = **new** DruidDataSource();   datasource.setUrl(**this**.**dbUrl**);  datasource.setUsername(**username**);  datasource.setPassword(**password**);  datasource.setDriverClassName(**driverClassName**);   *//configuration* datasource.setInitialSize(**initialSize**);  datasource.setMinIdle(**minIdle**);  datasource.setMaxActive(**maxActive**);  datasource.setMaxWait(**maxWait**);  datasource.setTimeBetweenEvictionRunsMillis(**timeBetweenEvictionRunsMillis**);  datasource.setMinEvictableIdleTimeMillis(**minEvictableIdleTimeMillis**);  datasource.setValidationQuery(**validationQuery**);  datasource.setTestWhileIdle(**testWhileIdle**);  datasource.setTestOnBorrow(**testOnBorrow**);  datasource.setTestOnReturn(**testOnReturn**);  datasource.setPoolPreparedStatements(**poolPreparedStatements**);  datasource.setMaxPoolPreparedStatementPerConnectionSize(**maxPoolPreparedStatementPerConnectionSize**);  **try** {  datasource.setFilters(**filters**);  } **catch** (SQLException e) {  **logger**.error(**"druid configuration initialization filter"**, e);  }  datasource.setConnectionProperties(**connectionProperties**);   **return** datasource;  } } |

# Application.properties

|  |
| --- |
| *#spring.freemarker.suffix=.ftl #spring.freemarker.templateEncoding=UTF-8 #spring.freemarker.templateLoaderPath=classpath:/templates/ #spring.freemarker.content-type=text/html* **spring.freemarker.request-context-attribute**=**request** |

# 页面引用js/jpg/css



# 打包添加

|  |
| --- |
| <**resources**>  <**resource**>  <**directory**>src/main/java</**directory**>  <**includes**>  <**include**>\*\*/\*.xml</**include**>  </**includes**>  </**resource**>  <**resource**>  <**directory**>src/main/resources</**directory**>  <**includes**>  <**include**>\*\*/\*.\*</**include**>  </**includes**>  </**resource**> </**resources**> |