

```
1.jQuery是一个js的工具库, 里面有很多function
                          ~2.作用:使用jQuery简化对dom的操作,执行事件处理,动画,ajax请求。
                          ~ 3.获取jquery:从官网获取jquery的文件
                  概述
                                                使用传统的js代码获取的是dom对象, document.getElementById("txt"); —— 要使用dom的属性或者方法
                                               使用jquery的语法获取的对象,就是jquery对象 —— 目的是要使用jquery库中的函数
                           4.dom对象和jquery对象
                                               dom对象转为jquery: $(dom对象)
                                                jquery转dom: 从jquery数组中获取成员,每个成员都是dom对象
                                     1.获取jquery的js文件
                           基本使用
                                     2.在你的文件要加入js文件的地址, <script type="text/javascript" src="js/jquery-3.4.1.js"> </script>
                                     3.获取jquery对象,调用它的函数
                                                1.id选择器:使用dom的id值定位dom对象 —— $("#id")
                                               ~ 2.class选择器:使用css样式名称定位dom对象 —— $(".class名")
                                    基本选择器
                                               - 3.标签选择器:使用标签名称定位dom对象 —— $("标签名称")
                                               4.所有选择器:选择页面中的所有的dom对象 —— $("*")
                           选择器
                                               5.组合选择器:使用id, class,标签的选择器,多个dom对象的定位 —— $("#id,.class,标签名")
                                               根据type的属性值定位dom对象的,页面中的组件一定有type属性才可以使用。
                                    表单选择器 一一 表单选择器和 form标签无关。
                                               语法: $(":type属性的值), 例如 , 所有的单行文本框 , $(":text")
                                             基于选择器,进行过滤筛选的。过滤器就是一个字符串,筛选条件
                                    概念:
                                               使用dom对象在数组中的位置,筛选dom对象
                                               - $("选择器:first")第一个dom对象
                                               - $("选择器:last"):最后一个dom对象
jQuery
                                    基本过滤器
                                               - $("选择器:eq(下标)"):定位到指定下标的dom对象
                                               $("选择器:lt(下标)"):定位小于指定下标的dom对象
                           过滤器
                                                $("选择器:gt(下标)"):定位大于指定下标的dom对象
                  核心
                                                  使用dom对象的状态作为筛选条件, 定位dom对象
                                                  - $("选择器:enabled"): 可用的dom对象
                                    表单属性过滤器
                                                  $("选择器:disabled"):不可用的dom对象
                                                  $("选择器:checked"):选中的dom对象
                                                  $("选择器>option:selected"):下拉列表框选中的值
                                  val(): 操作的value属性值
                                  ~text():操作的标签的文本内容
                                  ~attr(): 操作value , 文本以外的属性的
                                  remove():删除dom和他的子对象
                           函数 ——— empty():删除dom的子对象
                                  append():添加dom对象
                                  html():操作标签文本的
                                                                           1.静态函数调用:$.each( 要遍历的对象, function(index,element) { 处理程序 } )
                                  each():循环函数,可以循环数组,json对象,jquery对象
                                                                           2.使用jquery对象调用: jQuery 对象.each( function( index, element ) { 处理程序 } )
                                   1.绑定事件, $("选择器").事件名称(事件处理函数)
                           事件
                                   2.on绑定事件: $(选择器) .on(event, function)
                                  1.$.ajax(): 最核心的处理ajax请求的
                                  2.$.get(): 做get请求的ajax
                           ajax
                                  3.$.post():做post请求的ajax
```

1.maven是apache基金会的项目构建工具 2.项目构建:是一个项目编译,测试,打包,安装,部署这些过程 概述和安装 ⁻ 3.maven主要功能:1)项目构建;2)管理依赖,也就是项目中使用jar包 4.安装:1)创建一个环境变量M2_HOME, 它的值maven工具的安装目录; 2.为了方便maven命令的使用, 需要把%M2_HOME%\bin目录加入到path 之中。 1.maven工具可以单独使用:独立使用maven编译代码,测试代码,打包,安装等等。 maven使用的使用方式 2.可以在其他的开发工具中,使用maven。把maven和idea等工具结合使用。 src/main/java —— 主程序功能代码, service, dao, controller等 - src/main/resources —— 主程序要使用的配置文件 1.约定的目录结构 src/test/java —— 测试程序 , junit的代码放在这里 src/test/resources —— 测试程序使用的配置文件 -个xml格式的文件:maven工具各种配置都在这个文件中 - 1.自己项目的坐标 / 2.pom文件 [·] 2.property属性 · 3.依赖dependencies , 其它工具库 , 框架的gav坐标 4.构建build,控制maven构建项目时一些操作,比如设置编译阶段,使用的jdk等 唯一标识,表示这个maven项目的 groupId:组织名称,域名的倒写 3.gav:坐标 artifactId:项目名称 version: 版本号 maven ~4.依赖dependencies -·依赖类似导入,把其它的jar加入到项目中 1.第三方的jar maven核心 存放的内容 2.你自己项目的jar 3.maven运行时需要的jar 5.仓库 1.本地仓库 仓库分类 2.中央仓库和镜像 3.私服 1.mvn clean:清理 ✓ 2.mvn compile : 编译src/main/java目录中的java代码为class文件 - 3.mvn test-compile:编译src/test/java目录中的测试代码的 6.maven声明周期,命令,插件~ 4.mvn test:执行src/test/java目录下的测试程序 5.mvn package:打包, javase程序是jar , web程序是war 6.mvn install:安装,把jar安装到maven仓库 1.compile默认的 , 在maven构建项目的各个阶段都需要的依赖 2.test,只有在执行test测试是需要的依赖 7.依赖范围 3.provider,在代码编写,运行,测试阶段需要的依赖,打包,部署由服务器提供,比如tomcat有servlet的jar 1.先定义 <标签名就是变量名>变量值</...> 8.定义全局变量 2.使用变量: \${标签名} 使用,在pom.xml的<build>加入 <resouces>插件的声明 9.创建resource 作用是把src/main/java目录中的文件拷贝到target/classess目录中

```
1.视图层,界面层 —— 接收用户的请求,调用业务层处理请求 —— springmvc
                                     2.业务逻辑层 —— 计算数据,处理请求,对应的service包 —— spring
                            三层架构
                                      3.持久层 —
                                               操作数据的,对数据执行增删改查 —— mybatis
                   介绍
                                        主要功能:执行数据库的操作,完成对数据的增删改查;看做是jdbc的升级版本
                            mybatis框架
                                               1.执行数据库的增删改查,可以只关心sql语句如何编写
                                        特色
                                               1.sql mapps: sql映射, mybatis可以把表中的数据转为一个java对象。 把表和java对象对应起来
                                                1.通过sql的id,使用SqlSession对象,执行sql语句
                                                2.使用自定义的dao对象。
                                  框架的使用方式
                                                                     使用SqlSession.getMapper()获取一个dao对象,通过这个dao对象执行方法就能访问数据库了
                                                3.使用mybatis 的代理对象
                                                                     核心是mybatis创建的dao,也是调用SqlSession的方法执行sql语句
                                                       1.实体类:使用实体类表示表中的数据:表中的一行数据就相当于是一个实体类对象
                                                      2.定义dao接口:定义方法,方法表示要执行的数据操作 ,方法名称应该和mapper文件中的id一样
                                                      - 3.定义mapper文件,是一个xml文件,编写sql语句
                                  mybatis框架的使用步骤:
                                                      · 4.定义mybatis主配置文件:1)定义数据源DataSource;2)指定mapper文件的位置
                                                       5.mybatis执行sql语句的核心对象:SqlSession,使用SqlSession的方法执行sql语句 ,使用SqlSessionFactory的openSession()方法获取SqlSession对
                                                 从java程序把数据传入到mapper的文件的sql语句
                                                \sim 1.dao方法有一个简单类型的参数,在mapper文件中,使用#{任意字符串}
                                                 2.使用@Param注解给参数命名 ,在mapper文件 ,使用#{自定义的名称}
                                  r mybatis的传参数 ·
                                                 3.使用java对象传参,在mapper文件中,使用#{java对象属性名称}
                                                 4.使用参数的位置 ,在mapper文件 ,#{arg位置值 ,从0开始}
                                                 5.使用map , #{map的key}
                                             1.#是占位符,使用?做占位符合,mybatis内部使用PreparedStatement对象执行sql语句,效率高
                                            2.#表示列的值 , 一般是放在等号的右侧使用
                                            - 3.$是字符串代理连接 ,把sql语句和 ${}位置的值连接在一起
                                  #和$区别
                                            4.$所表示的内容是原样替换的
                                            5.$使用的sql语句, mybatis使用的是Statement对象执行sql的
                                            6.$一般是替换表名,列名,部分sql语句的
                                                          mybatis执行sql语句后,把ResultSet转为java对象,mybatis负责创建对象,给属性赋值
mybatis
                                                                                                            - 1.java类型的全限定名称
                                                         1.resultType: 在<select>标签中使用的,把查询的结果转为java对象的类型
                                                                                                             2.别名
                                  ~ mybatis封装sql语句的执行结果
                                                                                            1.先使用 < result Map > 定义列和属性对应关系
                                                          2.resultMap: 自定义列名和属性名称的对应关系
                   mybatis实战
                                                                                            2.在<select resultMap="定义好的名称">
                                                              1.使用resultMap指定列名和属性名称的对应
                                  表的列名和java对象的属性名称不一样
                                                              2.使用resultType通过别名的方式,指定别名和属性名称对应一致
                                                 1.在java代码中指定模糊查询的内容,例如 "%李%" ,然后把这个值传入到sql语句中
                                  模糊查询Like使用
                                                 2.在sql语句中拼接like的内容 , 方式 where name like "%"空格#{name}空格"%"
                                           通过在mapper中,使用标签,通过条件生成不同的sql语句,主要是控制where 条件部分
                                          <if test="boolean条件">: 当条件为true,把sql语句加入到主sql中
                                  动态sql
                                           <where> : where里面是if ,当if条件有一个满足,在主sql语句加入 where 关键字和条件
                                                       1.循环基本类型,例如List<Integer>
                                            <foreach>
                                                       2.循环对象类型,例如List < Student >
                                                          1.settings中可以配置日志
                                                                              1.typeAlias : 一次只能给一个java类型配置别名
                                                         2.配置别名<typeAliases>
                                                                              2.package : 把一个包中的所有类作为别名。 类名就是别名
                                                         ~3.dataSource:配置数据源
                                                                              1.使用<mapper resource="com/bjpowernode/dao/StudentDao.xml"/>
                                  mybatis主配置文件的配置内容
                                                          4.配置mapper文件的位置
                                                                              3.使用<package name="包名,例如com.bjpowernode.dao"/>
                                                                                    1.定义属性配置文件,在resources目录下创建jdbc.properties
                                                                                    2.属性配置文件中,定义key=value
                                                         5.使用属性配置文件 ( .properties )
                                                                                    · 3.在mybatis主配置文件中 ,使用<proper resource="类路径之下的jdbc.properties"/>
                                                                                    4.在需要使用值的地方, 语法 ${key}
                                             是一个插件,给sql语句增加分页功能的。 在我们的sql语句之后,通过插件加入 limit这样的语句
                                                        1.加入依赖
                                                        2.在mybatis主配置文件中,加入plugin:
                                  PageHelp
                                                        <plugins>
                                             使用步骤:
                                                        <plugin interceptor="com.github.pagehelper.PageInterceptor" />
                                                        </plugins>
                                                        3.在你的查询方法之前加入 PageHelper.startPage(第一页,每页几行数据)
```

```
_ spring是java的开源框架 ,核心技术是ioc , aop。 spring也是一个容器 , 容器放的是java对象。
                         1.什么是spring -
                                      通过spring框架可以实现解耦合。 解决业务对象之间的耦合。也可以解决业务对象和非业务对象之间的耦合
                                      把对象的创建,属性赋值,依赖关系都交给代码之外的容器,由容器实现对象的管理。
                 概念
                                      把控制对象的权利转移给代码之外的容器。IoC是一个理论,概念,思想
                         2.IoC:控制反转 -
                                      · IoC的重要技术实现DI(依赖注入): 通过名称就可以获取对象 ,对象创建,属性赋值都由容器内部实现
                                      Spring使用DI的技术, 底层实现是反射机制
                                        ·控制反转:把创建对象new 对象,给属性赋值交给容器实现。 spring是一个框架也叫做容器
                                        r spring是容器:存放对象的,可以存放service,dao,工具类等对象。
                                        怎么使用spring:把程序中要使用的java对象都交给spring,让spring作为一个工厂加工对象。 我们在程序中通过名称获取要使用的对象。
                                                        - 1.spring默认调用无参数构造方法,创建对象。
                                        创建对象的注意地方
                                                        · 2.创建spring容器ApplicationContext时,会把配置文件中所有对象都创建,备用。
                                                   '依赖注入:就是给类中的简单类型,引用类型的属性赋值
                                                                     - 1.在spring的xml配置文件中,使用标签和属性, <bean>和<property>
                                                   di给属性赋值有两种语法:
                                                                     2.在java的源代码中,使用注解
                                                                                     spring框架调用类中的set方法,通过set方法可以完成属性赋值
                                                                1.set注入,也叫做设值注入 ——
                                                                                     - 1.简单的属性赋值   <property name="set方法set之后的部分" value="此属性值" />
                                                                                     2.引用类型赋值 <property name="set方法set之后的部分" ref="bean的id" />
                                                                         c spring调用类中的有参数构造方法,通过构造方法完成属性赋值
                                                                                                                 1.name : 构造方法的形参名
                                                                ~ 2.构造注入 —
                                                                           <constructor-arg>: 一个标签表示构造方法的 一个参数 一
                                                                                                                 - 2.index:构造方法参数的位置,从0开始
                                                   di分类(方式)
                                                                                                                 3.省略index属性
                                                                                ~spring框架根据byName,byType规则给引用类型赋值
                                                                                <sup>-</sup> 1.byName:按名称注入 ,把引用类型的属性名称当做是bean的id使用 , 从spring容器中获取这个bean赋值给引用类型
                                                                                <sup>-</sup> 2.byType:按类型注入,把和引用类型的数据类型同源的对象,从容器中获取,赋值给引用类型
                                                                3.引用类型自动注入 -
                                                                                          1.类型一样
                                                                                 同源关系 —— 2.父子类关系的
                                                                                          3.接口和实现类关系的
                                        di依赖注入 一
                                                                     包含关系的配置文件:一个总的文件,包含其他的多个配置文件
                                                   多配置文件的使用方式
                                                                     <import resource="classpath:其他配置文件的路径"/>
                                                                     使用通配符 , <import resource="classpath: spring-*.xml"/>
                                                               使用注解完成对象创建,属性赋值
                                                                                - 1.在类中加入注解,例如@Component
                                                               使用注解的基本步骤: 一
                                                                                 2.在spring的配置文件中加入注解驱动的标签 <context:component-scan="包名" />
                                                                             1.@Component:创建对象,调用类的无参数构造方法
                                                                            - 2.@Repository:创建dao对象的,访问数据库
                                                               创建对象的注解:
                                                                            ~ 3.@Service: 创建service对象 , service是业务类 ,能够执行业务方法 ,可以有事务的功能
                                                                             4.@Controller:创建控制器对象,控制器能够接受用户的请求,显示请求的处理结果
                                                   基于注解的di -
                                                                          1.简单类型,使用@Value
                                                                                                              ~ 1.byType: 默认是byType,从容器中找到同源关系的对象,进行赋值
                                                                                                                         · 1.@Autowired: spring框架中提供的 ·
                                                                                                               2.byName: -
                                                               给属性赋值
                                                                                                                          2.@Qualifier: 指定对象的名称,使用这个名字的对象,赋值给引用类型
                                                                          2.引用类型
                                                                                                  是jdk中的提供的注解
                                                                                                 - 1.byType: @Resource默认是byName,如果byName失败,再使用byType方式。
                                                                                     · 2.@Resource –
                                                                                                 2.byName:需要指定@Resource的属性name,name是对象的名称
                                                              1.什么是aop —— 要给业务方法增加功能,在原有的业务方法代码不改变的情况下,增加一些功能。
                                                                        1.实现解耦合: 业务功能和其它的非业务功能的解耦合
                                                                       - 2.减少重复的代码
                                                              · 2.aop作用 —
                                                                        3.专注业务方法的实现,不用考虑其他的代码
                                                       概念 -
                                                              · 3.Aspect:切面 —— 给业务方法增加的功能 ,切面通常是非业务功能 , 例如事务功能 , 日志 , 权限管理
                                                              · 4.JoinPoint:连接点 —— 业务方法,表示这个方法执行时,要增加切面的功能
                                                              5.Pointcut:切入点 —— 是多个连接的集合,表示切面在这些方法执行时,要增强切面的功能
                                                             6.Advice:通知,增强 —— 表示切面功能的执行时间
                                                                     1.spring框架:本身实现了aop的功能, spring框架自身的一些功能, 用的是自己的aop
                                                       实现aop的框架 ·
                                                                     2.aspectj框架:可以注解,xml配置文件实现aop
                                                                                        1.切面的功能
spring
                                                                       作用aop,切面的要素
                                                                                        2.切面的执行位置
                                                                                                      - 切入点表达式 execution(方法的定义)
                                                                                        3.切面执行的时间
                                                                                                       使用注解表示切面代码的执行时间
                                                                                                          <dependency>
                 spring核心技术
                                 ioc –
                                                                                                          <groupId>org.springframework</groupId>
                                                                                                          - <artifactId>spring-aspects</artifactId>
                                                                                       1.加入一个aspectj的依赖
                                                                                                          <version>5.2.5.RELEASE</version>
                                                                                                          </dependency>
                                                                      实现aop的基本步骤:
                                                                                       2.定义业务方法
                                                                                                   1类的上面加入@Aspect
                                                                                       3.定义切面类。
                                                       aspectj框架的使用:
                                                                                                   2类的里面加入@Before等注解
                                                                                       4.在spring的配置文件加入对象的声明
                                                                      ~1.@Before:前置通知
                                                                                       - 在目标方法之前先执行的
                                                                      `2.@AfterReturning:后置通知 —— 在目标方法执行之后执行的 ,能够获取到目标方法的执行结果
                                                                      ^{\circ} 3.^{\circ} 3.^{\circ} Around:环绕通知 —— 在目标方法的前和后都能增强功能 ,可以修改目标方法的执行结果 ,控制目标方法是否执行。
                                                                      4.@AfterThrowing:异常通知 —— 在目标方法抛出异常时执行的,通过Exception类型的参数,能够获取到异常信息
                                                                      5.@After:最终通知 —— 总是会被执行的。
                                                                      6.@Pointcut:定义和管理切入点的注解
                                                                             —— 代理对象是指由框架创建的一种内存中的对象,通过这个对象执行方法时,可以执行切面的功能代码。实现功能的增强
                                                                       使用spring的ioc技术,把mybatis框架中使用的对象交给spring创建,管理,赋值
                                                                       · 1.使用业务比较有名连接池: 连接池也是一个类,使用的阿里的DruidDataSource —— 创建这个类的对象,使用<bean>
                                                       <sup>-</sup> spring集成mybatis <sup>-</sup>
                                                                       2.创建SqlSessionFactory对象,使用SqlSessionFactoryBean在内部创建的SqlSessionFactory
                                                                       3.创建dao的代理对象:访问数据库的操作 ,使用 MapperScannerConfigurer 在内部创建dao对象 ,执行SqlSession.getMapper(接口.class) 。
                                                                       事务:一系列的操作步骤,多个sql语句的集合。
                                        aop面向切面编程
                                                                       在程序中事务是放在service类的方法上面的 ,因为在service方法里面会使用多个dao执行数据库的操作
                                                                                                                                   把事务的处理委托给事务管理器: PlatformTransactionManager和他的实现类
                                                                                                                                                 1.使用jdbc或者mybatis框架:DataSourceTransactionManager
                                                                       ⁄spring框架中的事务的处理: 是把事务处理抽象为一个通用的模型,对大多数项目都是使用的'
                                                                                                                                   事务管理器类: 一
                                                                                                                                                 2.使用hibernate框架,HibernateTransactionManager
                                                                                                                                        1.DEFAULT:采用 DB 默认的事务隔离级别
                                                                                                                                       - 2.READ_UNCOMMITTED:读未提交
                                                                                                                        1.隔离级别(5个) —
                                                                                                                                       - 3.READ_COMMITTED:读已提交
                                                                                                                                       - 4.REPEATABLE_READ:可重复读
                                                                                                                                       5.SERIALIZABLE: 串行化
                                                                                                                                     1.PROPAGATION_REQUIRED:默认的,需要事务,可以使用已经存在的,没有时,创建新事务
                                                                       使用事务定义接口(TransactionDefinition)的常量,控制方法的事务行为
                                                                                                                                    ✓ 2.PROPAGATION_REQUIRES_NEW:需要新事务
                                                                                                                                   ✓ 3.PROPAGATION_SUPPORTS:支持事务,没有事务也可以
                                                                                                                        ~2.传播行为(7) -
                                                                                                                                    - 4.PROPAGATION_MANDATORY
                                                                                                                                    5.PROPAGATION_NESTED
                                                                                                                                    6.PROPAGATION_NEVER
                                                                                                                                    7.PROPAGATION_NOT_SUPPORTED
                                                       spring框架处理事务
                                                                                                                        3.超时时间(1) —— 业务方法的最长执行时间
                                                                                                      1.它是spring框架自己的实现方式 ,底层是aop环绕通知
                                                                                                                 ~ 1.加入依赖:spring-tx
                                                                                                                 - 2.在配置文件中,声明事务管理器 <bean id="transactionManager" class="...DataSourceTransactionManager">
                                                                                                      2.实现步骤: -
                                                                                 1.使用注解@Transactional
                                                                                                                 · 3.开启事务注解驱动:告诉spring框架现在是用注解处理事务, <tx:annotation-driven transaction-manager="transactionManager" />
                                                                                                                  4.在service的实现类,public方法的上面加入@Transactional
                                                                                                      注解方式:适合中小项目 , 需要事务的方法不多的情况。
                                                                       事务实现
                                                                                                       1.使用aspectj框架的实现aop,管理事务,底层环绕通知
                                                                                                                   1.加入依赖:spring-aspects
                                                                                                                  - 2.在配置文件中,声明事务管理器对象
                                                                                                       2.实现步骤:
                                                                                 2.使用aspectj框架的aop方式
                                                                                                                   - 3.配置事务的通知(切面):使用<tx:advice>配置方法的事务属性
                                                                                                                   4.配置增强器:配置切入点表达式
                                                                                                       3.适合的是大型项目,代码和事务的配置是完全分离的,声明式事务
                                                                                         使用监听器创建容器对象
                                                                                                               1.创建容器对象WebApplicationContext
                                                                       web项目中使用spring
                                                                                         ContextLoaderListener作用 —— 2.把容器对象放入到ServletContext作用域
                                                                                                               3.创建容器对象,把spring配置文件所有对象都创建好
```

