1. 准备工具：eclipse EE/IntelliJ IDEA、tomcat、mysql、Navicat
2. 创建maven项目，设置setting.xml，配置pom.xml配置jar包依赖

* Spring：spring-jdbc、spring-test、spring-aspects
* SpringMVC：spring-webmvc
* MyBatis：mybatis-sprig、mybatis-generator-core
* 数据库连接池、驱动包：c3p0、mysql-connect-java
* 其他：jstl、junit、javax.servlet-api、pagehelper、jackson-databind、hibernate-validator

1. 引入bootstrap前端框架，下载后放入webapp/static下

引入jQuery，下载后放入webapp/js下

1. 编写SSM整合的关键配置文件(eclipse需要安装spring IDE，从market中找)
   1. Web.xml：
   2. Spring-MVC:dispatcher-servlet.xml:
   3. Spring：applicationContext.xml:
   4. Mybatis：mybatis-config.xml
   5. dbconfig.properties
2. 使用mybatis的逆向工程生成对应的bean、dao以及mapper
   1. 创建generator.xml
   2. 创建MBGTest.java，运行
3. 根据实际需要，修改Mapper文件，新增方法

注意resultMap、sql id的书写

1. 测试Mpper，自动注解mapper对象，增删改查，查看连通性
2. 创建新bean——Msg，属性：状态码code，返回提示msg，返回内容extend
3. 搭建基础静态页面，方便后期ui开发
4. 请求列表：GET 查询员工；POST 保存员工；PUT 修改员工；DELETE 删除员工
5. **查询**：

* 引入pagehelper插件，查询之前调用，传入页码及每页显示大小，返回json格式字符串(需要注解ResponseBody).
* 测试分页请求，方式：使用模拟mvc测试类MockMvc、控制台输出、网页查看(响应参数responsebody)
* 界面发送ajax请求，解析显示员工数据，分页信息，分页条信息。

1. **新增：**

* 创建模态框，每弹出时清空任何内容以及样式
* 构建查询部门controller及service，查出部门信息显示与下拉列表中
* 校验表单数据，包括前端校验（字符校验）、后端校验（信息校验）、bean校验（JSR303校验）。
* 数据合格，发送ajax请求，保存信息，关闭页面去到最后一页；数据不合格，界面显示校验结果的提示信息。

1. **修改：**

* 创建模态框，注意：编辑按钮是动态生成的，为其添加事件需要使用on函数
* 通过前期构建的controller及service，查出部门信息，显示在模态框中的下拉列表中
* 为编辑按钮添加自定义属性edit-id，构建通过id查询员工姓名的controller和service，查询结果显示在模态框中
* 校验修改数据的合格性，不合格回显错误信息，合格发送ajax请求。
* 注意：tomcat不会封装PUT请求，直接PUT请求导致后台拿不到数据。
* 方式一：type可以发POST请求，然后data连接“&\_method=PUT"，把put请求转成post发出
* 方式二：type直接发PUT请求，data不必连接其他字符串，但需要在web.xml中引入HttpPutFormContentFilter过滤机制。
* 关闭模态框，回到本页面

1. **删除：**

* 为UI界面添加复选框按钮，表格内的需要动态生成，完成自动及手动全选功能
* 构建删除controller和service，完成批量删除和单个删除二合一，批量参数{id1-id2-id3…}，单个参数{id}
* 单个删除按钮绑定id，通过id查询姓名，显示确认框，提示删除，确认删除发送ajax请求，通过单个id执行删除
* 批量删除需要判断循环提取复选框id，查姓名显示确认框提示删除，确认删除发送ajax，通过多个id执行删除。

1. 详细项目情况访问：[https://github.com/NigelBoxing/SSM-CRUD-Internet](https://github.com/NigelBoxing/SSM-) 或者是
2. 项目结构图：

