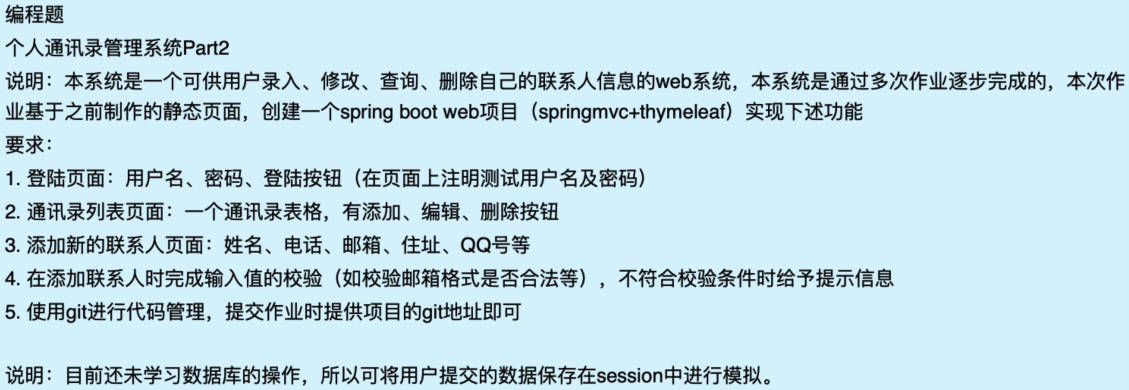
班级：2018211313 姓名：梁奥博 学号：2018211265



目录

[1. 登录页面：loginController 1](#_Toc56361292)

[a. 初始登录请求(“/”, “/login”) 1](#_Toc56361293)

[b. 判断用户名密码是否正确请求(“/checklogin”) 2](#_Toc56361294)

[c. 其他访问类型(“/checklogin”, “/list”) 2](#_Toc56361295)

[2. 通讯录列表页面：changeController，addController，del 2](#_Toc56361296)

[a. 添加请求(“/add”) 2](#_Toc56361297)

[b. 修改/删除请求 3](#_Toc56361298)

[3. 附加：退出登录：loginController 4](#_Toc56361299)

1. 登录页面：loginController

用户名位admin，密码为123456，已经在页面左上角标注

1. 初始登录请求(“/”, “/login”)

在登录时，该服务就会判断Session中的登录成功标志位是否存在，存在则自动重定向到联系人列表；若不存在，spring服务器就会创建一个LoginInfo类，该Java类包含了登录用户的昵称、密码信息，controller指示的服务将把这个类写入到模型中，并交给login.html进行渲染。



1. 判断用户名密码是否正确请求(“/checklogin”)

当点击登录按钮后，form表单会有一个POST形式向checklogin发出的request，同样，由loginController捕捉。模型中的LoginInfo类此时就是用户名密码的载体，该服务就是获取model中的这个LoginInfo类，并判断用户名和密码是否和“admin”与“123456”匹配。如果匹配则重定向到列表“/list”，否则将LoginInfo类的message设置为“用户名或密码错误”，并重新执行初始登录请求的服务。

没有进行重定向，因为这会导致创建一个新的LoginInfo类，无法回传message这一错误信息。因此只是执行了服务而已，在地址栏，也会发现此时停留在了/checklogin。

1. 其他访问类型(“/checklogin”, “/list”)

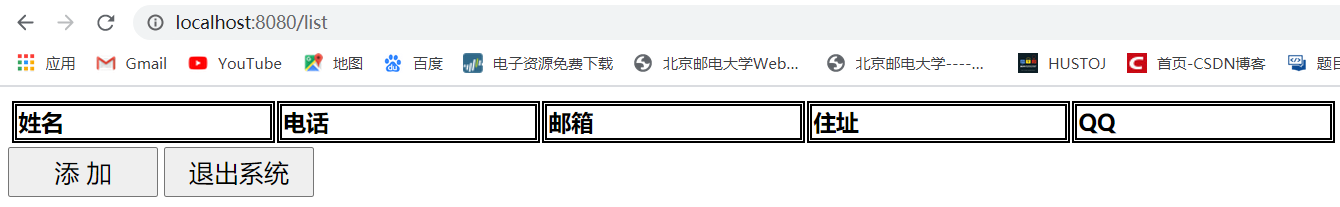
服务会根据Session中的登陆成功标志位判断当前是否已经登录，已登录则重定向到联系人列表，否则重定向到登录界面

1. 通讯录列表页面：changeController，addController，del
2. 添加请求(“/add”)

点击“添加”按钮，会执行一个action，以POST形式提交/add的request，并重定向到/add页面。这一过程中，spring服务器会创建一个联系人对象linkman作为Model传给add.html进行渲染。

在/add页面中，会正则检查输入是否合法。如果所有输入都合法，就会成功创建一个linkman联系人对象。该对象中包含所有联系人信息，包括姓名、电话、邮箱、住址、QQ。

在联系人列表中，是使用th:each="m:${session.table.getTable()}"这个可迭代对象进行数据更新。添加成功时，会在一个Vector<linkman>类型的向量列表中假如当前创建的linkman，之后，list.html的可迭代对象就会将更新后的向量列表展现出来。





1. 修改/删除请求

点击编辑按钮，会以POST的形式提出/change的request。由于每次添加、删除都会更新列表内容，所以根据th:each="m:${session.table.getTable()}"这个可迭代对象进行数据更新。点击编辑按钮时，可以利用JavaScript获取当前元素所在的行，并作为提交对象的参数。@ModelAttribute即可以从model中获取指定键值对，从而在Controller调用服务时获得参数，修改/删除向量列表中指定位置的元素。

由于修改操作和添加操作非常相近，所以对于这两个html来说，采用的css样式和js都是相同的。

**修改**







**删除**





1. 附加：退出登录：loginController

(“/exit”)被loginController捕捉，从Session中删除登陆成功标志位，并重定向到“/login”请求，重新跳转到登录界面

本系统只允许一个用户登录，所以一个用户就对应了这一次Session，因此重新登陆后，其联系人数据也都会保持不变