* 学习目标

  * 能够掌握Web监听器

  * 能够使用Web监听器开发简单的案例

  * 能够掌握Ajax的概述

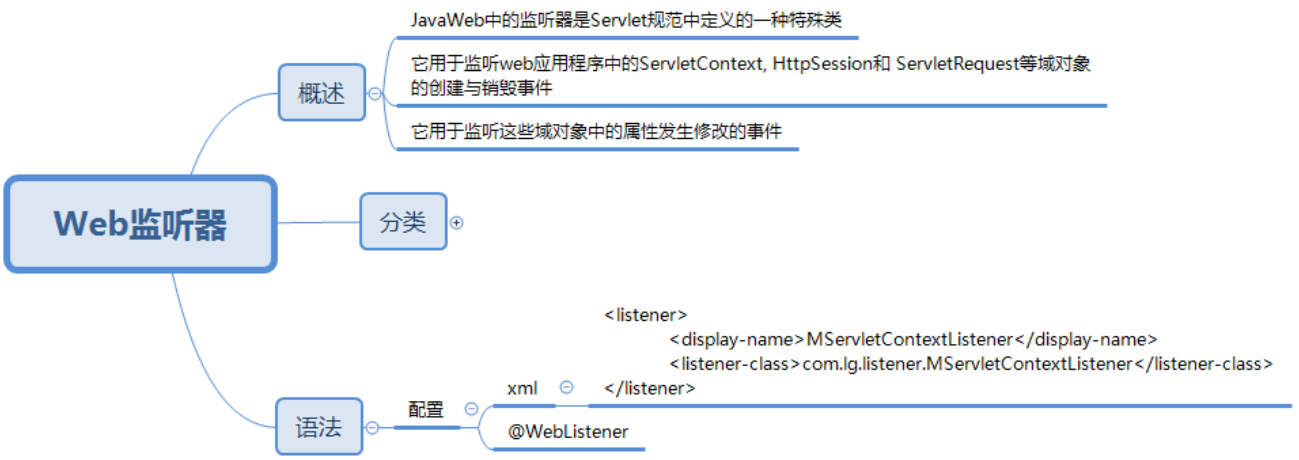
  * 能够掌握Ajax的开发

  * 能够结合BootStrap实现列表分页显示的效果

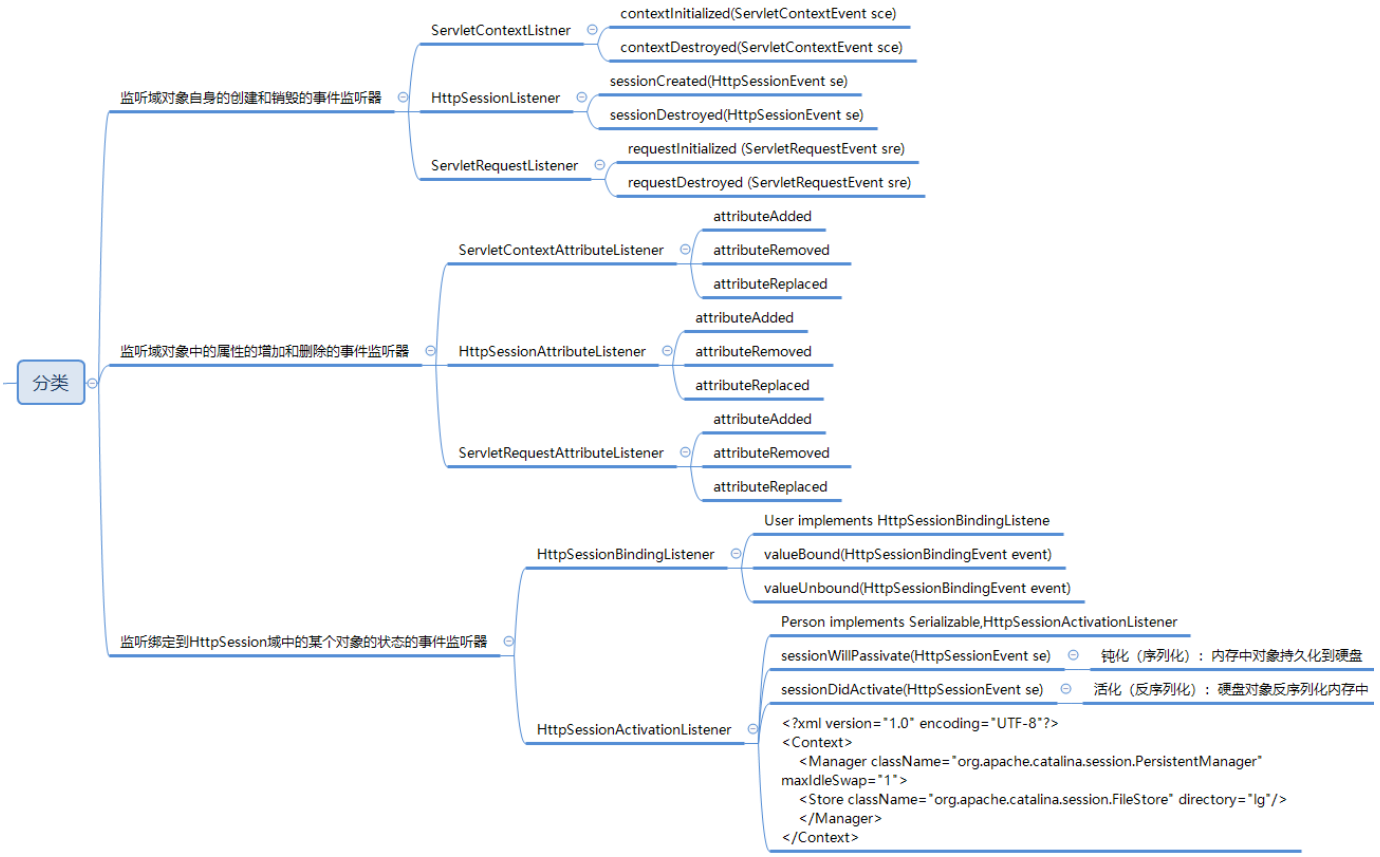
---

  * 能够掌握Web监听器

    * Web监听器的概述



    * 分类




```
* 开发案例
@WebListener()
public class MListener implements ServletContextListener,
        HttpSessionListener, HttpSessionAttributeListener {
    public MListener() {
    }

    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("MListener:contextInitialized");
    }

    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("MListener:contextDestroyed");
    }

    public void sessionCreated(HttpSessionEvent se) {
        HttpSession session = se.getSession();
        System.out.println("MListener:sessionCreated");
```

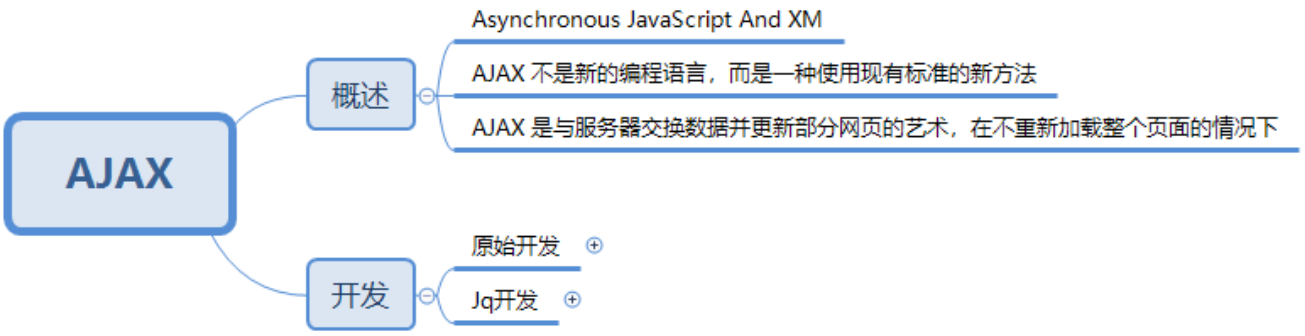
```
        System.out.println(session+"被创建");
    }

    public void sessionDestroyed(HttpSessionEvent se) {
        HttpSession session = se.getSession();
        System.out.println("MListener:sessionDestroyed");
        System.out.println(session+"被创建销毁");
    }

    public void attributeAdded(HttpSessionBindingEvent sbe) {
        System.out.println("HttpSessionAttibuteLitener:attributeAdded:"+sbe.get
    }

    public void attributeRemoved(HttpSessionBindingEvent sbe) {
        System.out.println("HttpSessionAttibuteLitener:attributeRemoved:"+sbe.g
    }

    public void attributeReplaced(HttpSessionBindingEvent sbe) {
        System.out.println("HttpSessionAttibuteLitener:attributeReplaced:"+sbe.
    }
}

```

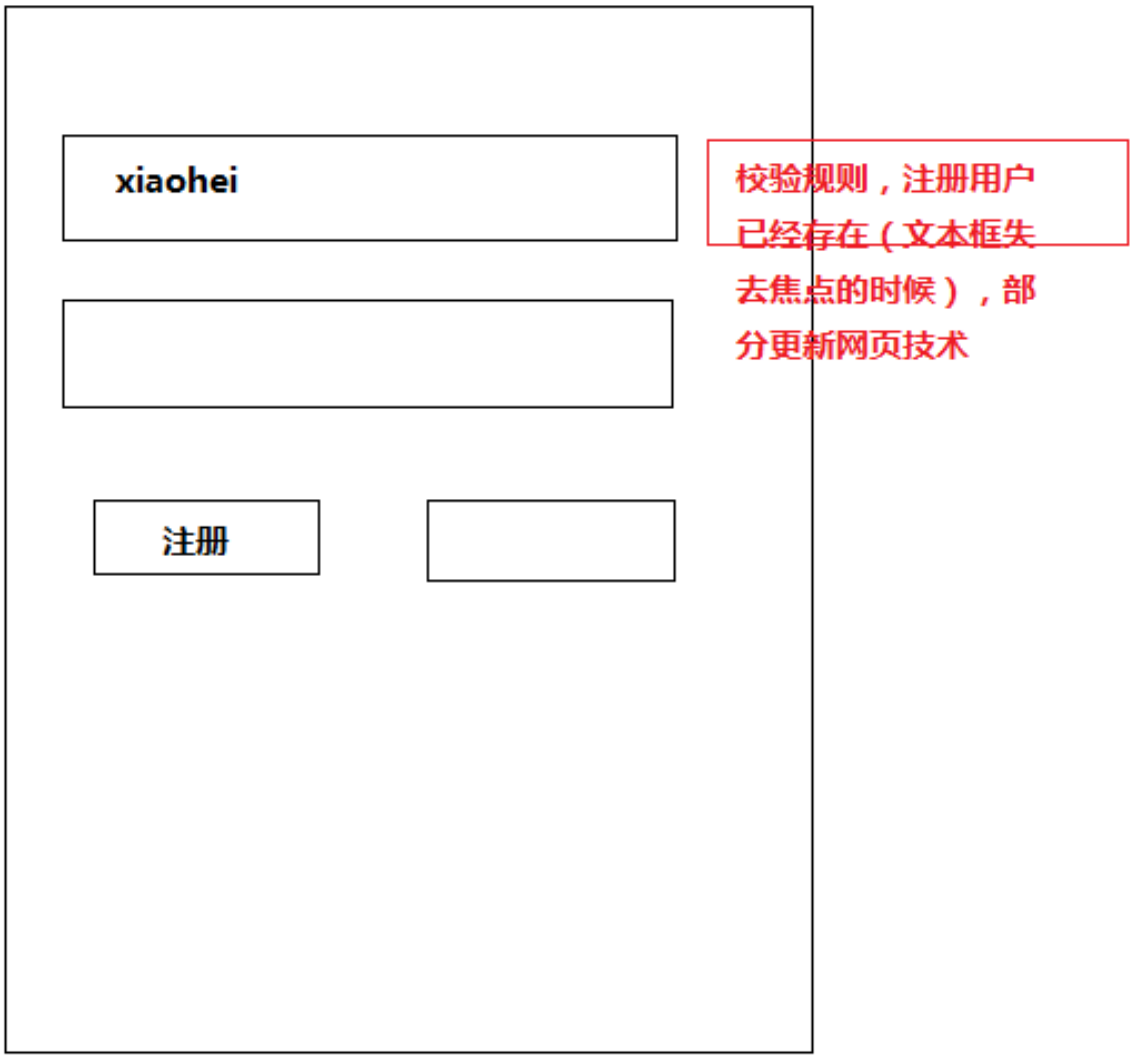
* 能够使用Web监听器开发简单的案例

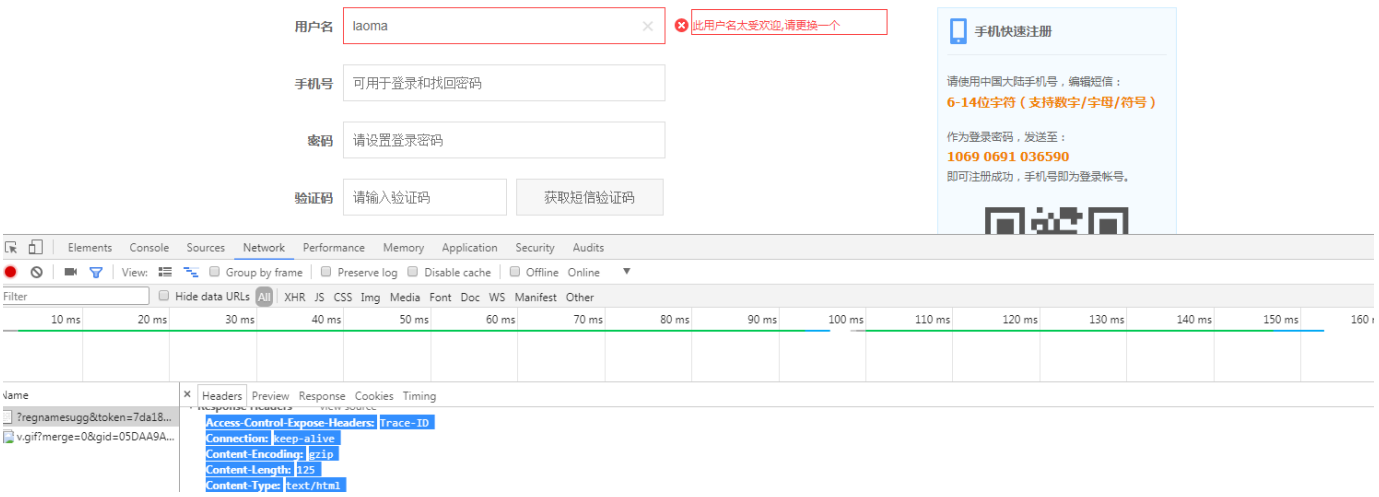
```
* 开发案例：
  * 统计当前在线人数
    *  当用户访问我们网站：分配一个Session
    *  当Session被创建的时候，计数器加一，当Session被销毁的时候，计数器减一
    *  计数器存在ServletContext域中
@WebListener
public class OnLineCountLinstener implements HttpSessionListener {
    private ReentrantLock lockCreated=new ReentrantLock();
    private ReentrantLock lockDestroyed=new ReentrantLock();
    @Override
    public void sessionCreated(HttpSessionEvent se) {
        lockCreated.lock();
```

```
        HttpSession session = se.getSession();
        ServletContext context = session.getServletContext();
        Integer count= (Integer) context.getAttribute("count");
        if(count==null){
            context.setAttribute("count",1);
        }else{
            count++;
            context.setAttribute("count",count);
        }
        lockCreated.unlock();
    }

    @Override
    public void sessionDestroyed(HttpSessionEvent se) {
        lockDestroyed.lock();
        HttpSession session = se.getSession();
        ServletContext context = session.getServletContext();
        Integer count= (Integer) context.getAttribute("count");
        if(count==null){
            context.setAttribute("count",1);
        }else{
            count--;
            context.setAttribute("count",count);
        }
        lockDestroyed.unlock();
    }
}

* 在首页加上
  * <h1 style="color: maroon;">当前在线人数：${count }</h1>
```

* 能够掌握Ajax的概述

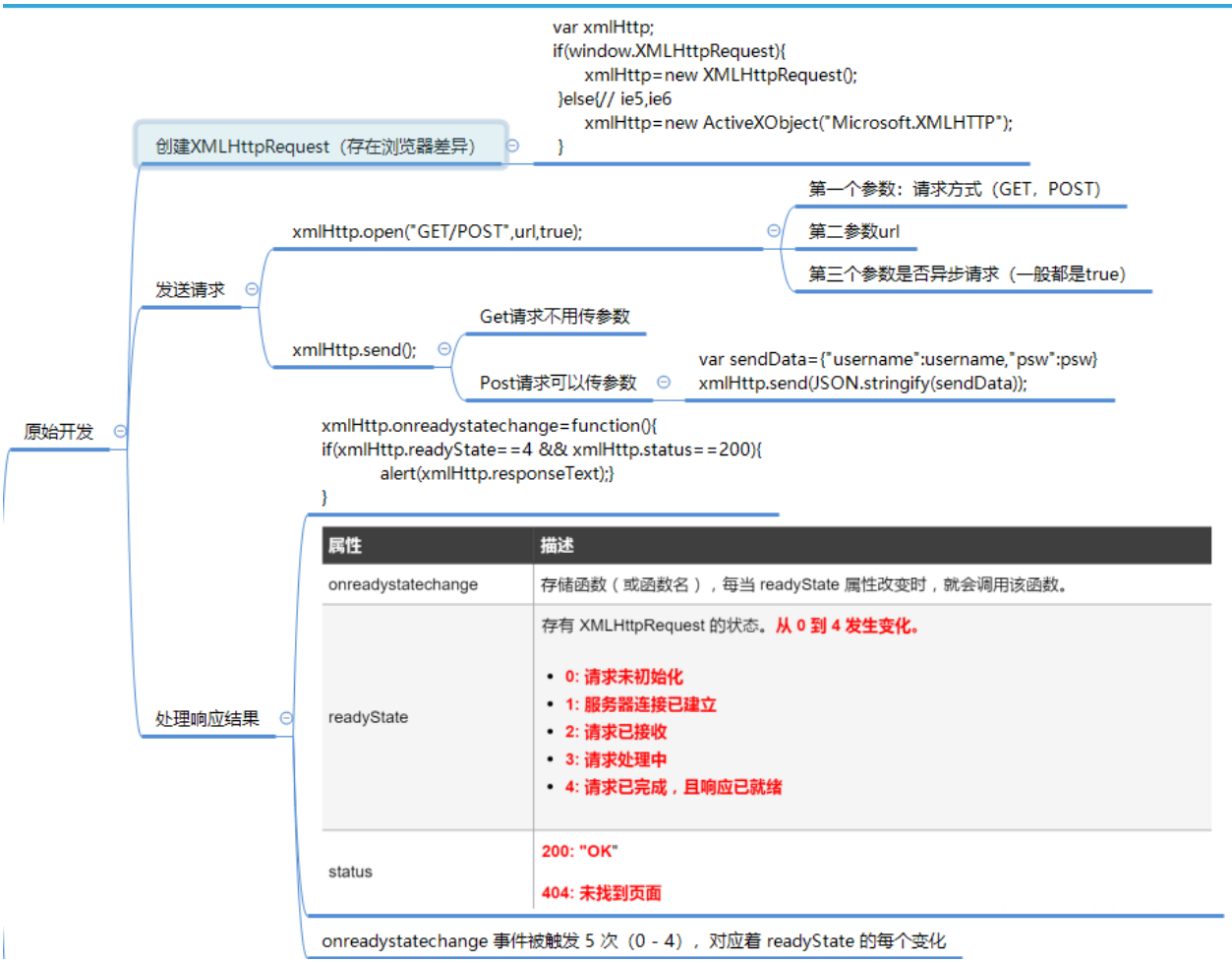






* 能够掌握Ajax的开发

  * 原始开发




```
* 案例
 * 注册页面
  * Get方式提交
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>注册</title>
    <script>
        function checkName() {
            // 1 创建XMLHttpRequest（存在浏览器差异）
            var xmlHttp;
            if(window.XMLHttpRequest){
                xmlHttp=new XMLHttpRequest();
            }else{//ie5,ie6
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")
            }
            // 2 发送请求
            var param=document.getElementById("iname").value;
            var url="/lgweb02/reg?username="+param;
            xmlHttp.open("GET",url,true);
            xmlHttp.send();
            // 3 处理结果
            xmlHttp.onreadystatechange=function (ev) {
                if(xmlHttp.readyState==4 && xmlHttp.status==200){
                    document.getElementById("sname").innerHTML=xmlHttp.response
                }
            }
        }
    </script>
</head>
<body>
    <form action="/lgweb02/reg" method="get">
        <input id="iname" type="text" name="username" onblur="checkName();"/>&r
        <span id="sname"></span>
    </form>
</body>
</html>

* Post 提交
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

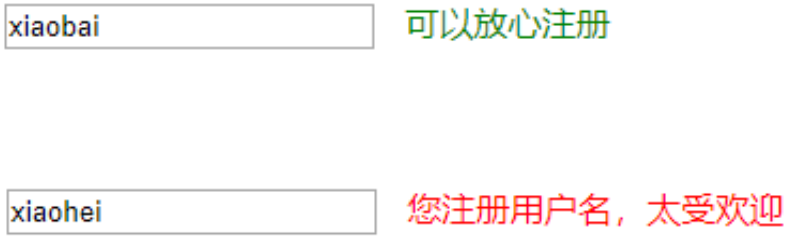
```
    <title>注册</title>
    <script>
        function checkName() {
            // 1 创建XMLHttpRequest（存在浏览器差异）
            var xmlHttp;
            if(window.XMLHttpRequest){
                xmlHttp=new XMLHttpRequest();
            }else{//ie5,ie6
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")
            }
            // 2 发送请求
            var param="username="+document.getElementById("iname").value;
            var url="/lgweb02/reg";
            xmlHttp.open("POST",url,true);
            xmlHttp.setRequestHeader("content-type","application/x-www-form-url
            xmlHttp.send(param);
            // 3 处理结果
            xmlHttp.onreadystatechange=function (ev) {
                if(xmlHttp.readyState==4 && xmlHttp.status==200){
                    document.getElementById("sname").innerHTML=xmlHttp.response
                }
            }
        }
    </script>
</head>
<body>
    <form action="/lgweb02/reg" method="post">
        <input id="iname" type="text" name="username" onblur="checkName();"/>&r
        <span id="sname"></span>
    </form>
</body>
</html>
* Servlet
@WebServlet("/reg")
public class RegServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        String username = req.getParameter("username");
        if ("xiaohei".equals(username)) {
            resp.getWriter().write("<font color='red'>您注册用户名，太受欢迎</for
```

```
        } else {
            resp.getWriter().write("<font color='green'>可以放心注册</font>");
        }
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
        doGet(req, resp);
    }
}

 * 案例二（JSON数据提交和响应数据）：
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>注册</title>
    <script>
        function checkName() {
            // 1 创建XMLHttpRequest（存在浏览器差异）
            var xmlHttp;
            if(window.XMLHttpRequest){
                xmlHttp=new XMLHttpRequest();
            }else{//ie5,ie6
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")
            }
            // 2 发送请求
            var username=document.getElementById("iname").value;
            var sendData={"username":username}
            var url="/lgweb02/check";
            xmlHttp.open("POST",url,true);
            xmlHttp.setRequestHeader("content-type","application/json");
            xmlHttp.send(JSON.stringify(sendData));
            // 3 处理结果
            xmlHttp.onreadystatechange=function (ev) {
                    if(xmlHttp.readyState==4 && xmlHttp.status==200){
                        var content= JSON.parse(xmlHttp.responseText);
                        document.getElementById("sname").innerHTML=content.msg;
                    }
            }
```

```
        }
    </script>
</head>
<body>
    <form action="/lgweb02/reg" method="post">
        <input id="iname" type="text" name="username" onblur="checkName();"/>&r
        <span id="sname"></span>
    </form>
</body>
</html>

* 代码
@Data
@AllArgsConstructor
@NoArgsConstructor
public class RespResult {
    private Object msg;
    private int code;
}

@WebServlet("/check")
public class ChekcServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) thro
        resp.setContentType("application/json;charset=utf-8");
        // 获取json数据
        ServletInputStream is = req.getInputStream();
        byte[] buffer=new byte[1024];
        StringBuilder sb=new StringBuilder();
        while((is.read(buffer))!=-1) {
            sb.append(new String(buffer));
        }
        String json=sb.toString().trim();
        System.out.println(json);
        Gson gson=new Gson();
        User user=gson.fromJson(json, User.class);
        RespResult result=new RespResult();
        if ("xiaohei".equals(user.getUsername())) {
            result.setCode(401);// 401代表用户名已经存在
            result.setMsg("<font color='red'>您注册用户名，太受欢迎</font>");
```

```
165         } else {
166             result.setCode(200);// 200 代表用户不存在
167             result.setMsg("<font color='green'>可以放心注册</font>");
168         }
169         String resultStr = gson.toJson(result);
170         resp.getWriter().write(resultStr);
171     }
172 
173     @Override
174     protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
175         doGet(req, resp);
176     }
177 }
178 
```

* 效果图



* JQ开发


```
* 案例
 * 引入jquery库
 * Get的请求
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>注册</title>
    <script src="/lgweb02/js/jquery-3.3.1.min.js"></script>
    <script>
       $(function () {
           $("#iname").blur(function () {
               var param=$("#iname").val();
               var url="/lgweb02/reg?username="+param;
               $.get(url,function (data) {
                   $("#sname").html(data);
               });
           });
       })
    </script>
</head>
```

```
<body>
    <form action="/lgweb02/reg" method="get">
        <input id="iname" type="text" name="username"/>  
        <span id="sname"></span>
    </form>
</body>
</html>
 * Post请求
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>注册</title>
    <script src="/lgweb02/js/jquery-3.3.1.min.js"></script>
    <script>
       $(function () {
           $("#iname").blur(function () {
               var param=$("#iname").val();
               var url="/lgweb02/reg";
               $.post(url,{"username":param},function (data) {
                   $("#sname").html(data);
               });
           });
       })
    </script>
</head>
<body>
    <form action="/lgweb02/reg" method="post">
        <input id="iname" type="text" name="username"/>  
        <span id="sname"></span>
    </form>
</body>
</html>

 * AJAX请求
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>注册</title>
```

```
    <script src="/lgweb02/js/jquery-3.3.1.min.js"></script>
    <script>
       $(function () {
           $("#iname").blur(function () {
               var param=$("#iname").val();
               var url="/lgweb02/check";
               var sendData={"username":param};
               $.ajax({
                   type:"POST",
                   url:url,
                   contentType:"application/json;charset=utf-8",
                   dataType:"json",
                   data:JSON.stringify(sendData),
                   processData:false,//默认是true，是否转换为查询字符串
                   success:function(result,status){
                       var content = JSON.parse(JSON.stringify(result));
                       $("#sname").html(content.msg);
                   },
                   error:function(result){
                       alert("请求数据失败")
                   }
               });
           });
       })
    </script>
</head>
<body>
    <form action="/lgweb02/check" method="post">
        <input id="iname" type="text" name="username"/>  
        <span id="sname"></span>
    </form>
</body>
</html>
```

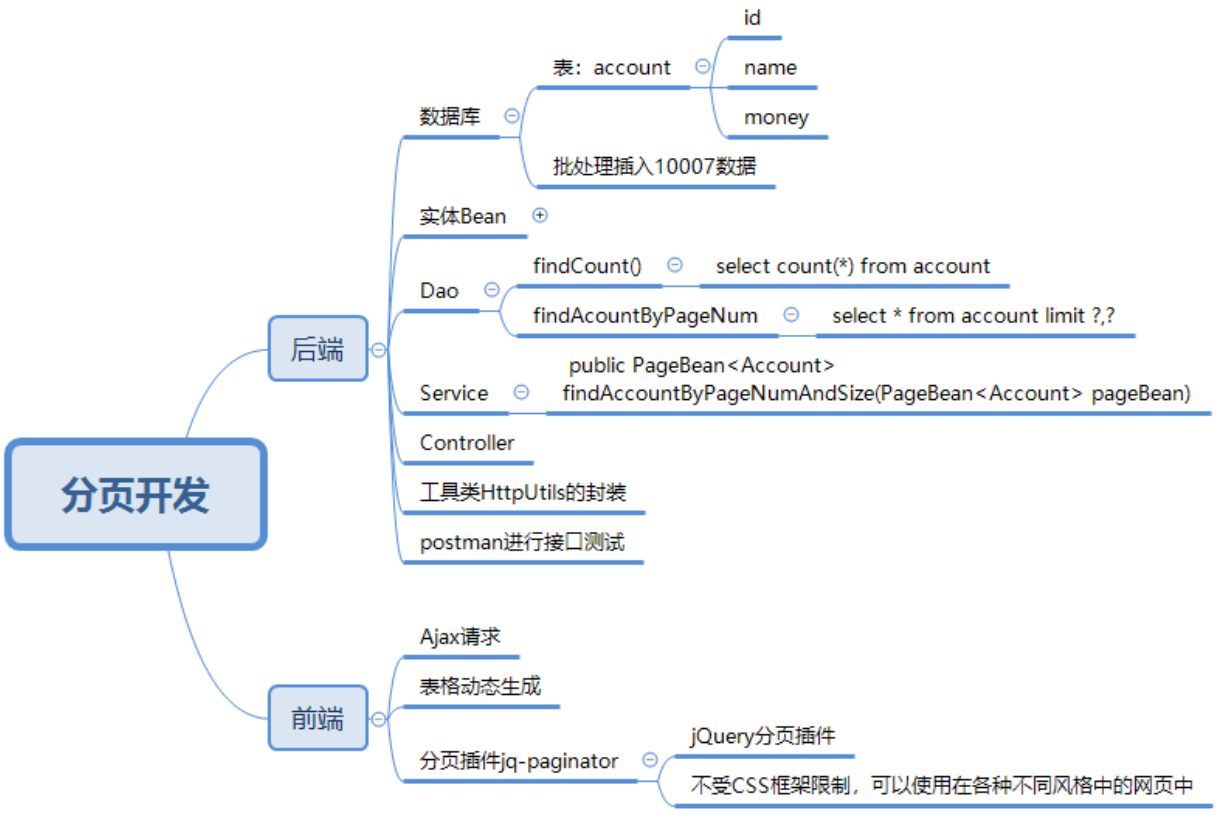
* 能够结合BootStrap实现列表分页显示的效果

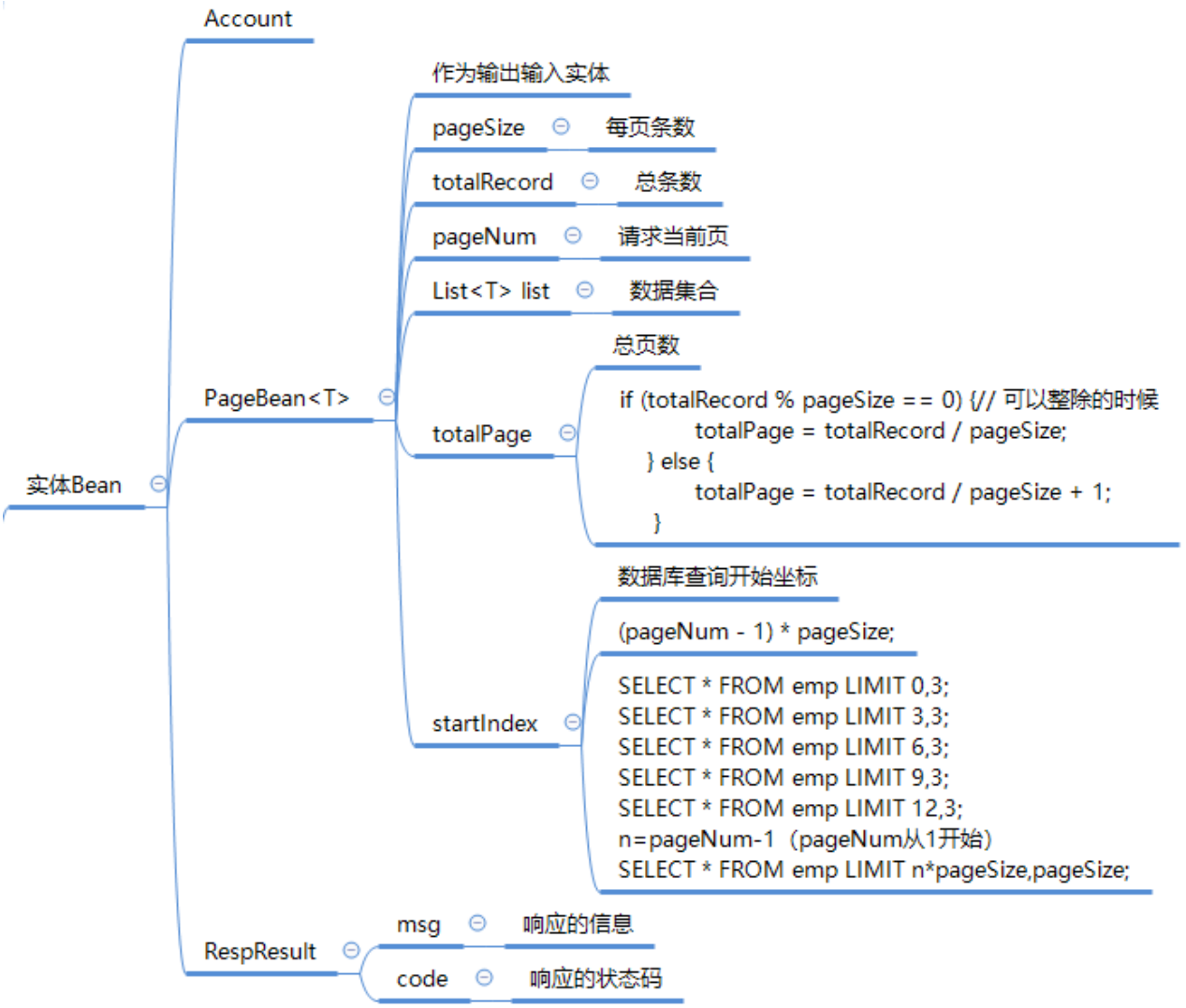
* 分页效果

亮哥教育

员工编号	员工姓名	薪水
1	xiaobai1	2001
2	xiaobai2	2002
3	xiaobai3	2003
4	xiaobai4	2004
5	xiaobai5	2005
6	xiaobai6	2006
7	xiaobai7	2007
8	xiaobai8	2008
9	xiaobai9	2009
10	xiaobai10	2010

首页 上一页 1 2 3 4 5 6 7 8 9 10 下一页 末页





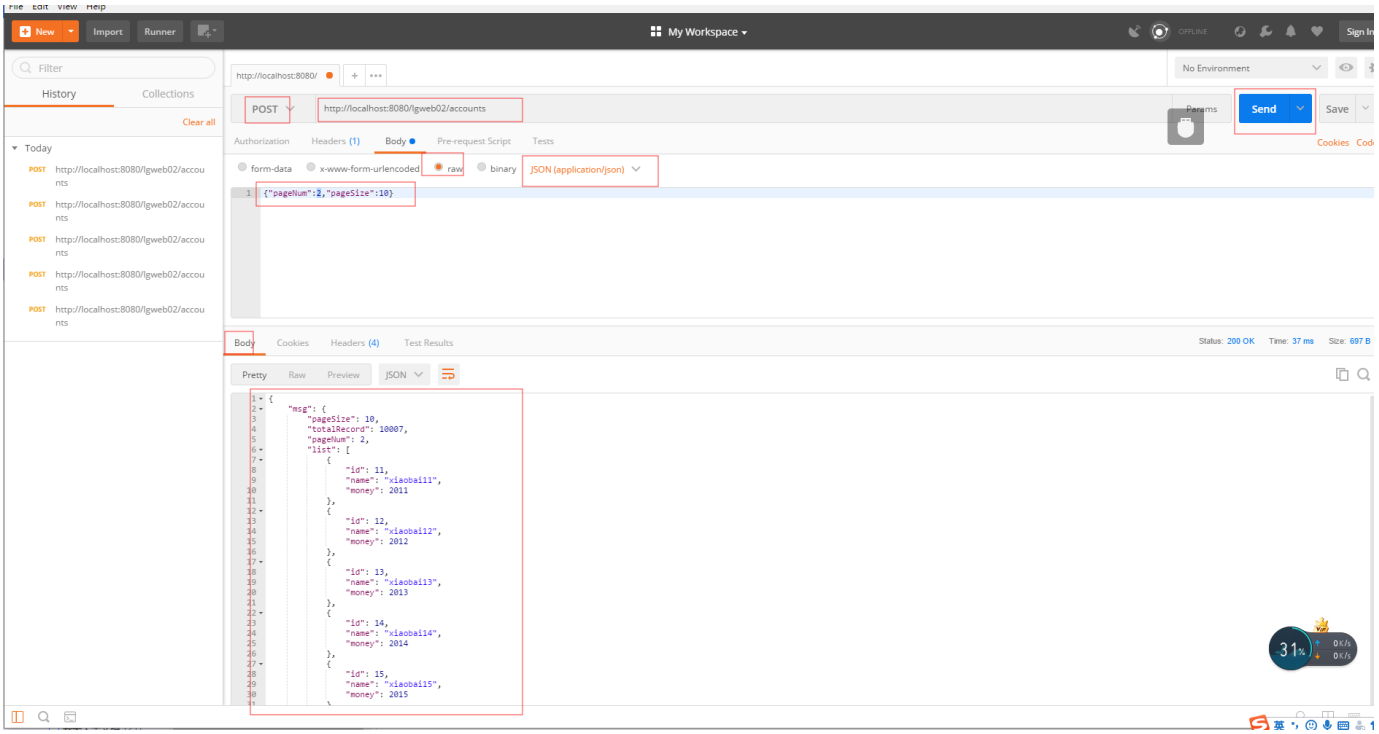

```
* 开发案例
 * Account表的创建（id,name,money）,批量插入数据
 * 后端:
  * 持久层
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Account {
    private int id;
    private String name;
    private double money;
}
public interface AccountDao {
    // 查询所有记录
     int findCount() throws SQLException;
    // 根据下标和每页条数，查询账户列表
     List<Account> findAcountByPageNum(int startIndex, int pageSize) throws SQL
```

```
}
public class AccountDaoImpl implements AccountDao {
    @Override
    public int findCount() throws SQLException {
        QueryRunner runner=new QueryRunner(DataSourceUtils.getDataSource());
        String sql = "select count(*) from account";
        Long count = runner.query(sql, new ScalarHandler<Long>());
        return count.intValue();
    }

    @Override
    public List<Account> findAcountByPageNum(int startIndex, int pageSize) thro
        QueryRunner runner=new QueryRunner(DataSourceUtils.getDataSource());
        String sql="select * from account limit ?,?";
        List<Account> accounts = runner.query(sql, new BeanListHandler<Account>
        return accounts;
    }
}
public class AccountTest {
    @Test
    public void test1() throws SQLException {
        AccountDao accountDao=new AccountDaoImpl();
        int count = accountDao.findCount();
        System.out.println(count);
    }
    @Test
    public void test2() throws SQLException {
        AccountDao accountDao=new AccountDaoImpl();
        List<Account> accounts = accountDao.findAcountByPageNum(0, 10);
        System.out.println(accounts);
    }

}
  * 服务层
@Data
@NoArgsConstructor
public class PageBean<T> {
    // 已知
    // 每页条数：pageSize
    private int pageSize;
```

```
    // 总条数：totalRecord
    private int totalRecord;
    // 请求当前页：pageNum
    private int pageNum;
    // 集合
    private List<T> list;
    // 计算出来
    // 总页数：totalPage
    private int totalPage;
    // 起始坐标：startIndex
    private int startIndex;
    public void setTotalRecord(int totalRecord) {
        this.totalRecord = totalRecord;
        // * 总页数：totalPage
        // * 当可以整除的时候：totalPage=totalRecord/pageSize
        // * 当不可以整除的时候：totalPage=totalRecord/pageSize+1
        if (totalRecord % pageSize == 0) {// 可以整除的时候
            totalPage = totalRecord / pageSize;
        } else {
            totalPage = totalRecord / pageSize + 1;
        }
    }
    public int getStartIndex(){
        return (pageNum - 1) * pageSize;
    }
}
public interface AccountService {
     PageBean<Account> findAccountByPageNumAndSize(PageBean<Account> pageBean)
}
public class AccountServiceImpl implements AccountService {
    @Override
    public PageBean<Account> findAccountByPageNumAndSize(PageBean<Account> page
        AccountDao accountDao=new AccountDaoImpl();
        pageBean.setTotalRecord(accountDao.findCount());
        pageBean.setList(accountDao.findAcountByPageNum(pageBean.getStartIndex(
        return pageBean;
    }
}
 * 控制层
@Data
```

```java
@AllArgsConstructor
@NoArgsConstructor
public class RespResult {
    private Object msg;
    private int code;
}
public class HttpUtils<T> {
    public String getJson(HttpServletRequest req, HttpServletResponse resp) thr
        resp.setContentType("application/json;charset=utf-8");
        // 获取json数据
        ServletInputStream is = req.getInputStream();
        byte[] buffer=new byte[1024];
        StringBuilder sb=new StringBuilder();
        while((is.read(buffer))!=-1) {
            sb.append(new String(buffer));
        }
        String json=sb.toString().trim();
        return json;
    }
    public T getBean(HttpServletRequest req, HttpServletResponse resp,Class<T>
        String json = getJson(req, resp);
        Gson gson=new Gson();
        T t = gson.fromJson(json, clazz);
        return t;
    }

    public T getBean(HttpServletRequest req, HttpServletResponse resp,Type type
        String json = getJson(req, resp);
        Gson gson=new Gson();
        T t = gson.fromJson(json, type);
        return t;
    }

    public String toJson(Object obj){
        return new Gson().toJson(obj);
    }
}

@WebServlet("/accounts")
public class AccountServlet extends HttpServlet {
```

```
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throc
        HttpUtils<PageBean<Account>> httpUtils=new HttpUtils<>();
        PageBean<Account> pageBean = httpUtils.getBean(req, resp, new TypeToker
        AccountService accountService=new AccountServiceImpl();
        RespResult result=new RespResult();
        try {
            PageBean<Account> page = accountService.findAccountByPageNumAndSize
            result.setCode(200);
            result.setMsg(page);
        } catch (SQLException e) {
            e.printStackTrace();
            result.setCode(500);
            result.setMsg("服务器出错");
        }
        String resultStr = httpUtils.toJson(result);
        resp.getWriter().write(resultStr);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) thr
        doGet(req, resp);
    }
}
 * 使用postman测试接口
```




```
* 前端开发
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>分页显示案例</title>
    <link rel="stylesheet" href="/lgweb02/css/bootstrap.css">
    <script src="/lgweb02/js/jquery-3.3.1.js"></script>
    <script src="/lgweb02/js/jq-paginator.js"></script>
    <script src="/lgweb02/js/popper.min.js"></script>
    <script src="/lgweb02/js/bootstrap.js"></script>
    <script>
        var url="/lgweb02/accounts";
        var sendData={"pageNum":1,"pageSize":10};
        $(function () {
            // 当前页：1，第一次发送
            sendRequest(1,true)
        });
        function sendRequest(num,isfirst) {
            sendData.pageNum=num;
            $.ajax({
                type:"POST",
```

```
                url:url,
                contentType:"application/json;charset=utf-8",
                dataType:"json",
                data:JSON.stringify(sendData),
                processData:false,//默认是true，是否转换为查询字符串
                success:function(result,status){
                    var content = JSON.parse(JSON.stringify(result));
                    if(content.code==200){
                        console.log(content.msg.totalRecord);
                        var tbody=document.getElementsByTagName("tbody")[0];
                        tbody.innerHTML="";
                        for(var index in content.msg.list){
                            var account=content.msg.list[index];
                            console.log(account);
                            tbody.innerHTML+="<tr>\n" +
                                "                <td>"+account.id+"</td>\n" +
                                "                <td>"+account.name+"</td>\n" +
                                "                <td>"+account.money+"</td>\n"
                                "            </tr>";
                        }
                        if(isfirst){
                            $('#pagination1').jqPaginator({
                                totalPages: content.msg.totalPage,
                                visiblePages: 10,
                                currentPage: num,
                                first:'<li class="page-item"><a class="page-lin
                                prev: '<li class="page-item"><a class="page-lin
                                next: '<li class="page-item"><a class="page-lin
                                last: '<li class="page-item"><a class="page-lin
                                page: '<li class="page-item"><a class="page-lin
                                onPageChange: function (num) {
                                    sendRequest(num,false);
                                }
                            });
                        }
                    }
                },
                error:function(result){
                    alert("请求数据失败")
                }
```

```
            });
        }
    </script>
</head>
<body>
    <div class="container-fluid">
        <h1>亮哥教育</h1>
        <table class="table table-bordered table-striped">
            <thead>
            <tr>
                <th>员工编号</th>
                <th>员工姓名</th>
                <th>薪水</th>
            </tr>
            </thead>
            <tbody>
            <!--<tr>-->
                <!--<td>100001</td>-->
                <!--<td>刘备</td>-->
                <!--<td>2000</td>-->
            <!--</tr>-->
            </tbody>
        </table>
        <ul class="pagination" id="pagination1">

        </ul>
    </div>
</body>
</html>

```

亮哥教育

员工编号	员工姓名	薪水
9991	xiaobai9991	11991
9992	xiaobai9992	11992
9993	xiaobai9993	11993
9994	xiaobai9994	11994
9995	xiaobai9995	11995
9996	xiaobai9996	11996
9997	xiaobai9997	11997
9998	xiaobai9998	11998
9999	xiaobai9999	11999
10000	xiaobai10000	12000

首页 上一页 992 993 994 995 996 997 998 999 1000 1001 下一页 末页