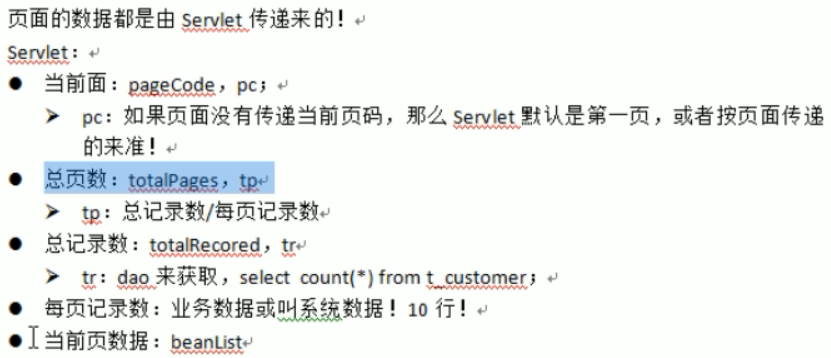
# Jdbc（04）

## 分页（客户关系管理系统）

### 优点

只查询一页，不用查询所有的页。

### 分页数据



### 封装数据

将与分页相关的参数封装到PageBean中，便于操作。

public class PageBean<T> {

private int pageNum;//当前页码

// private int totalPage;//总页数(经过计算得到)

private int totalRecords;//总记录数

private int pageSize;//每页的记录数

private List<T> beanList;//当前页的记录

}

### 分页在各层中的处理

* 页面

给出分页相关的链接们。

传递PageNum参数，没有默认为1.

* Servlet

创建PageBean对象，给PageBean所有的属性赋值，然后转发到页面中。

* 需要给Dao传PageNum,PageSize.
* Service

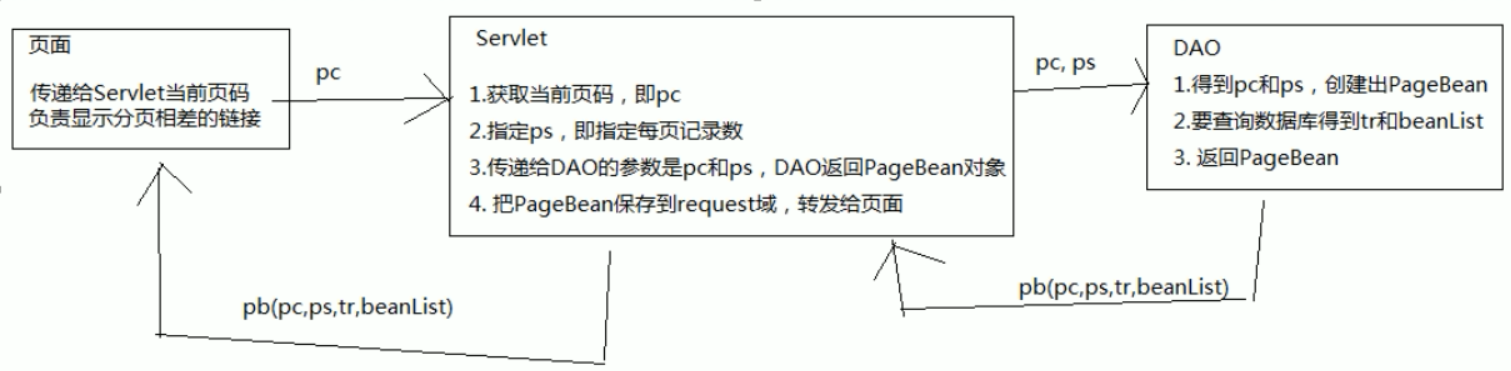
调用Dao中的方法

* Dao

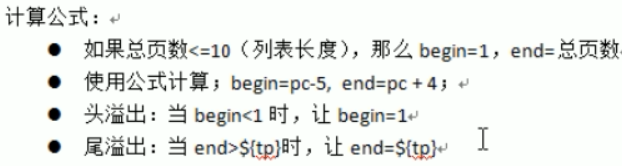
totalRecords: SELECT COUNT(\*) form customer;

beanList : SELECT \* FROM customer LIMIT “开始行数 - 1”,”需要的行数”;

### 流程图



### 显示分页的页码列表



Pc:当前页码

Tp:总页数

## 监听器

### JavaWeb三大组件

* Servlet
* Listener
* Filter

注意：都需要在 web.xml 中进行配置，Listener 中的两个感知监听器不需要配置

### 监听器概述

* 它是一个接口，内容由自己实现；
* 它需要注册，例如注册在按钮上；
* 监听器中的方法，会在特殊事件发生时被调用。

### 观察者概述

* 事件源
* 小偷
* 事件
* 偷东西
* 监听器
* 警察
* 监听器中的方法：抓捕！

### JavaWeb中的监听器

* 事件源：三大域！
* ServletContext
* 生死（生命周期）监听

ServletContextLister,有两个方法，一个在出生时调用，一个在死亡时调用。

void contextInitialized(ServletContextEvent sce)

ServletContextEvent中包含方法：

ServletContext getServletContext()

* 属性监听

ServletContextAttributeLister，有三个方法，一个在添加属性时调用，一个在替换属性时调用，一个在移除属性时调用。

* HttpSession
* 生死监听
* HttpSessionLister,有两个方法，一个在出生时调用，一个在死亡时调用。
* 属性监听

HttpSessionAttributeLister，有三个方法，一个在添加属性时调用，一个在替换属性时调用，一个在移除属性时调用。

* ServletRequest
* 生死监听
* ServletRequestLister,有两个方法，一个在出生时调用，一个在死亡时调用。
* 属性监听

ServletRequestAttributeLister，有三个方法，一个在添加属性时调用，一个在替换属性时调用，一个在移除属性时调用。

* 在JavaWeb中完成编写监听器
* 创建一个监听器类：要求必须实现某个监听器接口；
* 注册，在web.xml中配置来完成注册。

### 感知监听（都与HttpSession相关）

* 它用来添加到JavaBean上，而不是添加到三大域上；
* 这两个监听器都不需要在web.xml中注册。
* HttpSessionBindingListener：添加到JavaBean上，JavaBean就知道自己是否添加到JavaBean中了。

## Session序列化

### Session persistence

服务器关闭之后，将session写入硬盘，重启之后读入内存，以保证会话的持久性。

### Session钝化 & 活化

* 钝化

对于长期无活动的session，将其移出内存，保存在硬盘中。

* 活化

当被移出内存的session有活动时，将其session从硬盘中添加到内存中。

### 注意

* Session属性值为字符串：

由于字符串实现了 序列化 接口，会与session一起保存在硬盘上

* Session属性值为JavaBean对象：

还要添加 序列化 接口：

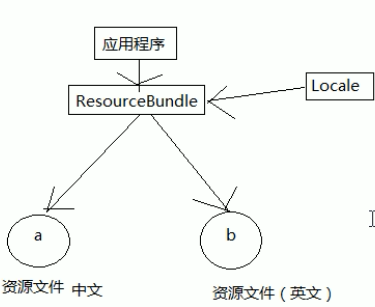


HttpSessionActivationListener接口：

监听JavaBean对象是否随session一起钝化或活化。

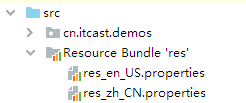
## 国际化

### 流程



应用程序使用ResourceBundle类来加载两个资源文件中的一个，到底加载谁，是通过给ResourceBundle的 Locale来决定的，如果Locale是中文的，则加载a文件，英文的则加载文件b.

### 资源文件



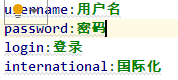
资源文件名称格式：基本名称+ Locale部分+.properties(扩展名)

\* 例如：res\_zh\_CN.properties ,res是基本名称，而zh\_CN是 Locale 部分

\* 注意： 所有资源文件打额基本名称要相同，不同之处就是 Locale 部分

资源文件要放在 src 文件中

文件内容：



### 实例代码

public class Demo1 {  
 @Test  
 public void fun1() throws UnsupportedEncodingException {  
 Locale locale = Locale.CHINA;  
 //得到 ResourceBundle  
 /\*  
 \* 第一个参数：基本名称  
 \* 第二个参数：Locale  
 \* \*/  
 ResourceBundle rb = ResourceBundle.getBundle("res",locale);  
 //使用 ResourceBundle 来获取资源信息  
 String username\_CN = new String(rb.getString("username").getBytes("ISO-8859-1"),"UTF-8");  
 System.out.println(username\_CN);  
  
  
  
 }  
}