

第 3 章 Web 应用开发与 Web 测试

在本书的前两章，笔者已经对测试的两个重要组成部分：软件测试工程师、测试用例及其编写的方法进行了简要介绍。从第 3 章开始，本书将进入 Web 测试的世界。首先，3.1 节将讲解 Web 应用的一些知识，进而介绍有关 Web 测试的诸多分类，比如功能测试、性能测试、兼容性测试等。由于在实际工作中，Web 性能测试工程师往往也要在项目的各个阶段从事功能测试的工作，因此除去本书的主要内容——性能测试之外，相比其他测试，对于 Web 功能测试做了更为丰富的介绍。

在经历以上这些基础知识的铺垫后，在第 4 章小白将与我们一起，进行具体的性能测试知识学习。

3.1 Web 应用的基本知识

“万丈高楼平地起”。在展开 Web 应用的性能测试之前，首先需要了解有关 Web 应用的若干基本概念，并熟悉建立网站，以及浏览网站的基本过程，这些知识和经验对于理解和解决实际发生的 Web 应用性能问题会有很大帮助。

本节通过“请客吃饭”这一场景类比用户浏览网站的过程，使读者能够熟悉服务和 Web 服务的概念。

3.1.1 什么是服务

在 Web 应用中的网页文件与 Word 文档等其他类型的文件一样，都存放于电脑的硬盘之中。但是，与供用户自己使用的文档不同，绝大部分的网页都是供非编写者浏览的。为了让使用者能够通过网络看到 Web 应用开发人员辛辛苦苦编写出来的网页文件，首先须把存放网页文件的电脑变成一台能够提供网页浏览服务的服务器。

通俗地讲，IT 世界里的“服务”与日常生活中所遇到的“服务”这两个词语在含义上没有什么本质区别。假设读者要请自己的朋友吃饭，那么一般会有两种选择：在家里亲自下厨招待或者去外面的饭店。下面笔者不妨详细地比较一下这两个选择，从中引申出 IT 术语“服务”的概念和相关含义。

在自己家每天做饭吃，可能不需要太过讲究，只需考虑营养问题、做饭是否快捷、饭菜是否合乎自己的胃口等几个要点就行了。但是，如果请朋友到家里吃饭，主人就必须得有一点服务的观念，除了之前考虑的那几个要素，更要充分考虑朋友的喜好和要求，一句话，请朋友吃饭，是为了朋友能够吃好，这样才能增进彼此的感情。

开发 Web 应用也是同样的道理：技术人员制作出来的网页是要交由使用者进行操作、浏览者查看信息，因此要尽可能地满足最终用户方方面面的需求，可以说，IT 行业中所说的服务，与请朋友吃饭非常类似，具备同样的出发点。满足用户需求就是“服务”这个词语的主要含义。

【服务的 IT 定义】

服务，是一个或多个程序的集合，为其他程序或最终用户提供所需要的信息和实现相关功能。

3.1.2 服务的场所

无论日常生活还是在电脑世界里，服务总是有个请求→反馈的过程，这个过程构成了服务这个词的具体内容。

“民以食为天”，前文提到过请朋友到家里吃饭的选择，在当今的时代，更多的是请朋友到外面的饭馆去吃要更普遍。这是因为饭馆的服务更专业：由专业的厨师进行烹饪，由专业的服务员端茶、送水、上菜，更有比家庭更适合用餐的专业环境等。同样地，在 Web 应用领域，为了满足用户的需求，也要专业、到位的服务。类似“饭馆”这样可以提供专业服务的场所，在 IT 领域是由硬件与软件相结合共同实现的，提供服务的硬件设备就是“服务器”（Server），提供服务的软件并没有专门名称，这里就称其为“服务”（Service）。

在本节，将主要介绍提供服务的硬件场所——服务器。如图 3-1 是某品牌服务器的外观。

可以看到，服务器与普通台式电脑差别不大，其在硬件上同样包含各种各样的接口、插槽、连线等。但是，为了满足多种用户的不同需求，服务器对普通的台式机、笔记本电脑进行了优化与改进，具备如下特点。



图 3-1 某品牌服务器的外观

1. 更大更强的能力

这个特点主要是指服务器具有速度更快的 CPU，容量更大的硬盘，速度更快、容量更大、缓存更多的内存等。为了解释这一点，笔者再次举出“请客吃饭”的例子。

一般来说，饭馆都要比家里地方宽敞很多：有不少餐桌，这样才能容纳更多的顾客来就餐。有时候，当用餐者来到某家很火的饭馆，如果位子不够，也就是饭馆容量不够大，服务人员会提示需要等待，一旦时间长了用餐者就会产生抱怨的情绪，从而影响服务的效果。同理，服务器如果硬件条件不好，对于网民的浏览页面请求无法完全满足，一样会导致服务效果的下降：网站打开很慢、出现内部错误等。所以说服务器的硬件设施一定要好，才能够提供优质的服务。

2. 服务器中运行着服务

这里的 service 是指提供服务的软件，即 Service。

饭馆里只有大间的屋子，众多的餐桌，如果服务员态度恶劣，那么这个饭馆也坚持不

了多久。对于服务器来说，也是同样的道理。除了良好的硬件条件以外，服务器上还必须运行一些服务程序。我们先来看一下服务员是如何工作的：他们要随时查看每个桌子顾客的需要，然后采取相应的措施：上茶、上菜还是结账等。更好的状况是，他们中间有一部分人随时查看饭馆门口的情况，如果客人满了，要给没有座位的顾客以安抚，好的饭馆还会发给一些号码，以提示用户等待的时间等。

好的服务程序也是一样，能够按请求服务的程序或者用户的需要反馈相应的合适信息。它相对于坏的服务程序还有一个优点：能够在有限的容量内尽量实现更多的请求。（想想同样是 10 张桌子的饭馆，一个手脚麻利的服务员和一个呆若木鸡的服务员工作效果是明显不同的）。当然，如果顾客盈门，实在太多，好的服务是应该给“没有座位的顾客”排队号码，从而在有空余的时候尽早对他们进行服务，以避免因为不满的顾客太多，导致饭馆的声誉下降甚至出现崩溃等严重的危机。

本节用饭馆类比服务器，服务员类比服务程序，充分说明计算机世界和日常生活是很相像的，不少道理都是相通的。今后在实际工作中遇到难以理解的问题时，不妨联系日常生活进行一番思考，可能会加深对问题的理解。

从 3.1.3 节起将开始介绍为 Web 应用创建服务场所的具体操作：在服务器上，使开发完毕的软件代码能够满足用户需求而服务的过程。这个过程简单地说，就是网站的建立。

3.1.3 创建服务场所——建立网站

建立网站就是一个创建服务场所的过程。

继续以“请客吃饭”为例。大家都知道，开饭馆需要考虑很多种要素：要考虑市场（价位如何）、面对的群体（商务宴请还是小区居民）、菜系（四川还是广东或其他）、场所（租金地段是个大问题）和厨师等人员（工资福利）。网站的建立也是一样的辛苦：要考虑网站的规模定位（可以是大而全的门户网站，也可以是小而精的专业信息网站等）、面对的群体（个人网络购物，个人交友社区还是作为展示单位资讯的窗口，让潜在客户获得有价值的信息）、“菜系”（实现网站所采用的技术是什么？），场所（网页存放的服务器性能如何，用户流量大不大，方便不方便用户访问等诸多问题）、技术人员的分工等。当然，对于普通的性能测试工程师来说，不必考虑那么多的方方面面，但心中必须明确的观念是：网站是众多人员合作的产物。

在技术方面，测试工程师则需要知道一个网站是如何被建立的。创建网站，笼统地划分可以有如下 3 个步骤。

(1) 建立网页的过程。技术人员需要利用多种编辑工具（记事本、写字板、VI 等程序）或者专业的网站开发工具（Dreamweaver 和 Visual Studio 等）将设计师设计的网页效果图变为现实。同时，网站还需要数据库等多方面的支持才能更好地满足用户需求。

(2) 规划服务器空间的过程。这个过程视网站规模大小而有所差别。如果是个人主页或者是博客，则可以申请网络服务提供商（Internet Service Provider，简称 ISP）提供免费的个人主页空间，以及利用某些网站提供的模板自定义；如果是小型网站，可能要到 ISP 购买相应的服务器空间；对于大中型网站，一般则是把自己的服务器直接放到网络服务提供商的机房（又称为“互联网数据中心”，IDC）中，利用 IDC 所提供的网络出口对外提供服务。

【提高服务质量的方法】

对于大中型网站，往往有多台服务器可以提供带冗余的服务（一台服务器出现问题，马上由另一台服务器接替工作）以保证服务质量，这个方法有点类似增加候补队员以提高“板凳深度”。另外，由于客观现实的要求，会有不少网站在各大 ISP 和网络（比如教育网、电信网、甚至局域网）中都会放置服务器，存储相同页面，提供相同服务，因此可以保证处于各个网络内的用户都能正常顺畅地浏览和使用。这是另一种提高服务质量的方法，有点类似大饭馆在全国各地开的连锁店面，使得各地的食客不必远行。

(3) 发布网页的过程。有了存放网页文件的服务器空间，还需要一个发布网页的过程，这样网站才能最终建立。简要地说，要通过各种方式，把本地（指开发 Web 应用代码的电脑）编辑好的网页代码传输到服务器的硬盘空间。之后，还需要开启在服务器上运行的网页发布服务，从而完成发布网页的过程。这一过程将在 3.1.4 节中介绍。

3.1.4 网站文件的上传

编辑好的网页代码从本地电脑转移到服务器空间的过程就是上传（Upload）。上传可以通过多种方式进行，比如文件传输协议（FTP），共享文件夹复制与粘贴等。其中最常用的则是 FTP 方式。

FTP，全称是 File Transfer Protocol，文件传输协议。我们将在下一部分讲述协议的故事，这里读者只需要知道它是一个可以用来上传文件的国际标准就可以了，而且不需要了解它的具体内容，因为有很多支持这个标准的软件可以供我们使用，比如 LeapFTP、CuteFTP 等。利用这些软件的功能，我们就可以实现网页的上传。

下面以 CuteFTP 为例来，说明如何从本地电脑将网页文件上传到服务器的过程。

(1) 运行 CuteFTP，按下 Ctrl+N 键，按照图 3-2 所示，设置申请服务器空间时 ISP 提供的 FTP 上传地址、用户名和密码。

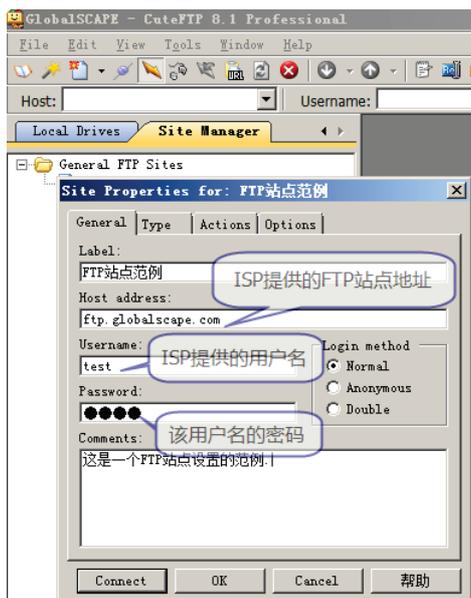


图 3-2 在 CuteFTP 8.1 中设置一个新的 FTP 站点信息

(2) 单击 OK 按钮后，可以看到 CuteFTP 主界面分为两个部分，如图 3-3 所示。这两个部分分别代表本地计算机和服务器的空间。具体上传步骤很简单：在“站点管理”（Site Manager）选项卡内双击新建立的 FTP 站点，连接成功后将本地硬盘上已经完成的网页文件拖动到服务器空间的相应文件夹中就可以了。

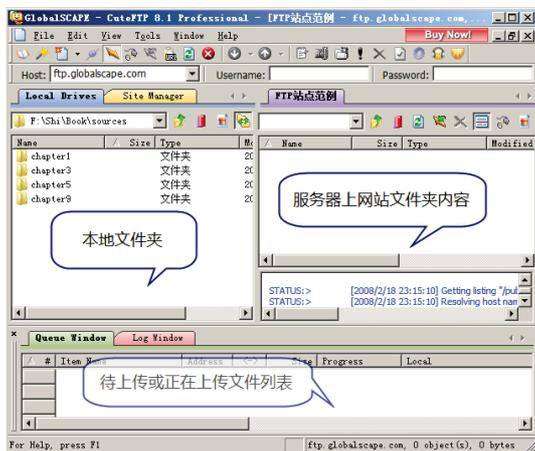


图 3-3 CuteFTP 界面

其他 FTP 软件基本上都是类似的界面和操作，不再详细介绍。

还有一些网页开发/编辑软件集成了网页的上传功能。下面以业内使用比较普遍的 Dreamweaver 和 Visual Studio 2005 为例说明网站集成开发环境（也叫做 IDE）中的上传操作。

Dreamweaver 中首先要建立一个站点，按图 3-4 和图 3-5 所示分别设置好本地与远程服务器空间的信息，比如服务器空间的用户名和密码，然后就可以进行编辑网页的过程。编辑结束后，可以直接选中文件并右击，在弹出的快捷菜单中选择 Put（上传）命令即可。

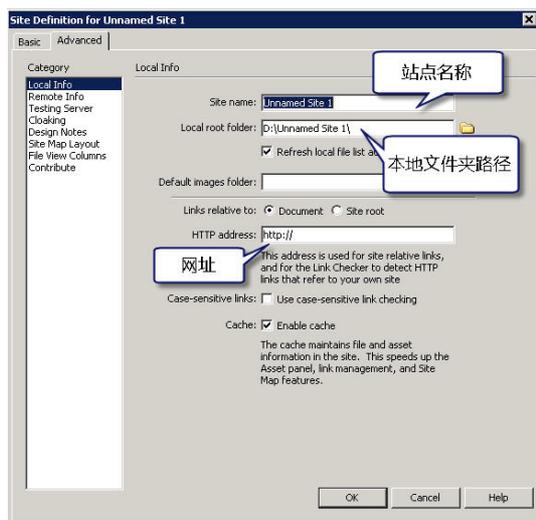


图 3-4 Dreamweaver 8 中的站点本地设置



图 3-5 Dreamweaver 8 中的站点服务器信息设置

要完成同样的上传工作,在 Visual Studio 2005 中的操作有所不同,本书以其 Team Suite 版本为例。首先需要新建一个网站(依次选择“文件”|“新建”|“网站”命令),在完成网页的编写后,选择 Build/Publish Web Site(生成/发布网站)命令,Visual Studio 2005 会弹出如图 3-6 所示的对话框,在 Target Location(目标地址)文本框中填写正确的域名并单击“确定”按钮后,若当前登录账户对于服务器文件夹有写入的权限,网页就随之上传到了服务器硬盘的相应目录中。

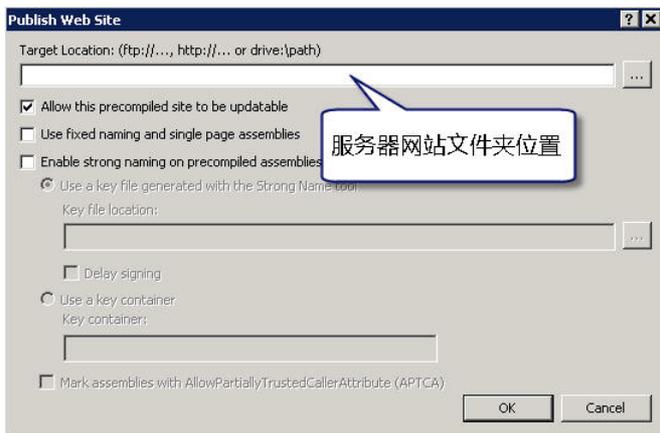


图 3-6 在 Visual Studio Team Suite 中发布网站

3.1.5 开启网页发布服务

在服务器上有了网页文件就相当于饭馆的厨师做好了饭菜,还需要服务员将其端到食客的餐桌上。这个“服务员”的功能是由运行在服务器上的网页发布服务程序来完成的。由于历史和商业的原因,这样的程序有很多,比如 Windows 平台下的 Internet Information Services(简称 IIS),以及支持多种操作系统的 Apache 等。如图 3-7 显示了在 Windows 服务器中打开 IIS 管理器后的界面。当然,IIS 管理器也可以通过单击“开始”按钮,在弹出的菜单中选择“运行”命令,并在文本框中输入 inetmgr 来启动。

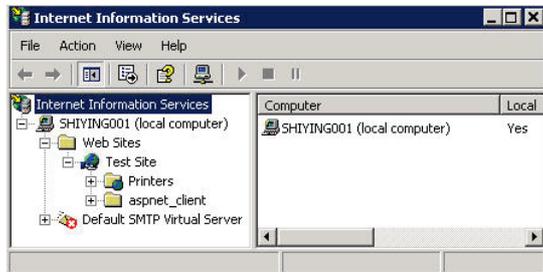


图 3-7 IIS 管理器界面

(1) 如图 3-7 所示,在 IIS 管理器中会包含服务器的名字,其后面的括号中还有 local computer(本地计算机)的字样。单击该服务器名称,右击 Web Sites(网站)节点,在弹出的快捷菜单中选择“新建”|“站点”命令,建立新站点向导的界面就会出现。单击 Next

按钮后，界面如图 3-8 所示。

(2) 在图 3-8 中的 Description (描述) 文本框中输入待建立站点的说明，比如：“XXXX 公司网站”这样的文字即可。从这一设置页开始，剩下的几个页面均可保留系统提供的默认值，不做任何修改，连续单击多个 Next 按钮，直到图 3-9 所示界面的出现，整个创建过程基本完成。



图 3-8 建立新网站向导：输入新建网站的描述信息



图 3-9 建立新网站向导：输入网站文件在服务器中的位置

(3) 根据实际情况在图 3-9 所示界面上设置好网站文件所在服务器的具体位置（单击 Path (路径) 文本框右边的 Browse (浏览) 按钮，选择网页文件的根目录即可），继续单击多个 Next 按钮，直到最后向导提示完成。这样，一个网站就建设好了。

(4) 除此之外，还要确认新建的网站是否启动了发布：选择新建立的网站，观察 IIS 管理器工具栏中的“运行”按钮是否为虚，如图 3-10 所示。

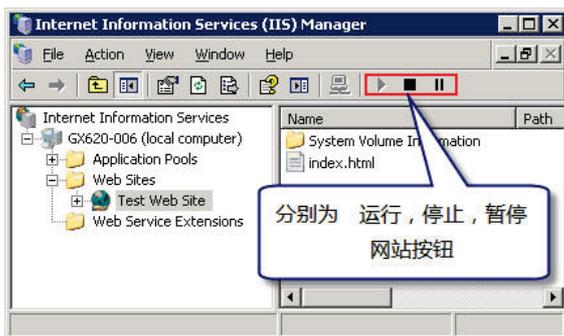


图 3-10 在 IIS 管理器中察看网站是否运行

建立好了网站，也开启了服务，那么如何确定我们编辑好的网页显示符合我们的期望呢？方法也很简单，在 IIS 管理器左边的树形目录中选择刚刚建立好的网站右击，在弹出的快捷菜单中选择 Browse (浏览) 命令，就可以在右边的内容区域中看到劳动成果了。

除了 IIS 之外，前文提到还有很多的网页发布服务可供使用，比如 Apache 等，其需要设置的网站属性与 IIS 区别不大，但必须手工修改配置文件。因此，在实际操作方面，IIS 提供了更丰富细致的管理界面，初学者更易直观地理解。

3.1.6 用户浏览网站的过程

运行在服务器端的网站截至目前已经架设完毕，也就是说，饭菜已经端上了餐桌，剩下的就是品尝了。本节将介绍在对“网站”的品尝当中，用户端所经历的过程。如图 3-11 是一幅示意图，显示了用户浏览网站的粗略过程。



图 3-11 用户浏览网站的基本过程

下面把整个过程分成几个步骤来详细地解释。

(1) 用户的操作：用户打开本地的浏览器，在地址栏中输入要访问的网址（假设是 www.sohu.com）。然后按下键盘的 Enter 键，将获取网页的任务交给浏览器去完成。

(2) 浏览器的工作：浏览器首先要去寻找目标网站的 IP 地址。这个 IP 地址就好比我们要去某餐馆吃饭，那个餐馆的门牌号码。让我们记住门牌号码是非常困难的，特别是在好吃的饭馆很多的情况下。因此我们往往记住的是饭馆的名字，一般来说，名字比门牌号码更容易记忆，字数也没有那么长。在网络世界中，网址就相当于饭馆的名字，而 IP 地址则是饭馆的门牌号码。

【网址与 IP 地址的关系】

网址和 IP 地址是对应的，这种对应关系由一种名为 DNS 的协议和系统来记录。浏览器在接到访问网址的请求后，首先查询 DNS 服务器，利用如上的对应关系，把该网址翻译成某个具体的 IP 地址，再向绑定该 IP 地址的某台服务器请求相关网页文件并显示在窗口中，从而完成网页浏览的整个过程。

(3) 网站处理浏览器发来的用户请求，并给予反馈：经过前面的两个步骤，用户的请求就被传递到了网站的门口，在服务器上运行的网页发布服务首先向用户返回一个预先设置好的首页面。这个首页面就好比饭馆中的菜单一样，上面列出了网站都有哪些栏目，然后用户就可以根据自己的喜好，进入各个频道浏览了（很像点菜的过程）。假设一个用户非常喜欢看娱乐新闻，就可以单击首页上的娱乐新闻链接。网页发布服务就像一个服务员一样，得到了该用户的请求，将需要的网页发送回去，这样就完成了一次如图 3-11 所示的请求与反馈的过程。

在整个过程中，用户所使用的浏览器可能不同，服务器上运行的网页发布服务也可能不同，那么，浏览器和网页发布服务是靠什么来沟通的呢？答案就是协议。

3.1.7 网络沟通标准——协议

所谓协议，英文叫做 Protocol，是一组在网络上发送和接收信息的规则与约定。这些规则控制在网络设备和程序之间交换消息的内容、格式、定时、顺序以及错误控制。通俗

地讲，协议就是不同网络设备、程序之间交流沟通的语言。

【协议与方言】

中国有很多种方言，如果不特别学习，各个地方的人相互交流就可能存在一些障碍，你听不懂我说的，我也听不懂你说的。因此，有必要制定一种大家都能够听懂的中国话的标准，这就是普通话产生的原因。普通话也可以说是一种协议。

前文提到的 FTP，与本节提到的 HTTP 都是协议的名称，是为了应用在不同的场合下方便程序之间交流的标准。

浏览网站正是由很多个请求——反馈这样的过程组成的。在一个用户浏览的同时，其他用户也可以同时浏览该网站，正如饭店里不止一张桌子的人在吃饭。能够同时容纳很多人进行网络浏览而不“手忙脚乱”，这是考验网页服务程序的一个重要指标之一。当然，人再忙也有个限度，如果访问量太大，就得增加饭店的服务员，扩大饭店的营业面积来应对。也就是说，如果网站遇到类似的情况，为了达到同样的服务质量，就需要增加服务器的数量和质量，以及提高服务器的性能来实现了。

【HTTP 协议】

前文提到了 HTTP 协议，这里多介绍一下。实际上，互联网上的核心之一就是 HTTP 协议。它在构成互联网的各个部分：客户端、服务器中都得到实现，这样才能保证信息沟通、交流的进行。HTTP 协议有一些很复杂的定义，对行走在网络上的消息文本的结构以及客户端、服务器如何交换消息的方式、方法进行了规定。具体内容，读者可以参阅有关互联网和 TCP/IP 的书籍。

3.1.8 前台页面与后台数据库

服务很重要。本节依然用饭馆来类比，简单说，一般的网站实际上都是由前台页面和后台数据库组成的。

1. 前台页面

前台页面实际上就是网站的最终用户在浏览器中看到的页面，相当于我们在饭馆中就餐的场所。不言而喻，前台页面的地位绝对重要：

- ❑ 前台页面上的各个元素要干净、整洁，正如餐馆就餐区的环境卫生。
- ❑ 前台页面的全部显示要尽可能快速，正如餐馆服务员上菜的速度。
- ❑ 前台页面实现的功能要满足用户的需求，正如餐馆服务员对于客户要求的满足程度。

本书所讲述的网站性能，很重要的一个指标就是上面的第二点，前台页面的显示速度。

2. 后台数据库

餐馆光有就餐场所是不行的，必须有一个强大的厨房、库房和采购部门的支持。这个后勤部门在网站上就是后台数据库的作用。后台数据库为前台页面提供了：

- ❑ 前台页面丰富多彩的内容，正如厨房为就餐者制作精美的菜肴。
- ❑ 保存用户在前台页面上操作的过程、结果，正如餐馆服务员常问的“菜有没有忌口”等类似的话，厨房再根据这些用户的偏好进行调整。有些高级餐馆甚至会记

录并保留客户的更多信息（比如生日和口味、菜系、食材的偏好等），就餐者下次再来就会享受更贴心的服务。

3. 前、后台的物理位置

前台页面和后台数据库出于性能的考虑，一般都不被安装在一台服务器中。这其实很容易理解：在餐馆中，厨房和就餐区域一般来说是分开的（餐饮术语分别为“前场”和“后场”，可见使用了类似的称谓方法），如果在一起，厨师做菜、服务员上菜难免会磕磕碰碰，乱作一团。

后台数据库的这两个功能是由以下两个子部分来实现的：

- ❑ 一系列供网站内部编辑人员增加、删除、更新数据的程序或者网页，一般称之为后台管理程序。这部分相当于厨师的作用。
- ❑ 一个或者多个数据库：用于保存网站的内容、后台管理程序与前台用户操作的结果。相当于餐馆的库房。

3.2 Web 开发的基础技术

还记得小白吗？在经过短短几天的学习之后，他已经有了一些测试的经验。虽然目前他需要做的只是黑盒测试，但为了尽快熟悉被测试的网站，还是很希望学习一下目前主流的开发网站的方法。本节就简单介绍目前流行的几种 Web 开发技术。

- ❑ Java Web 开发技术。
- ❑ .NET 开发技术。
- ❑ PHP 开发技术与其他开源平台。
- ❑ AJAX 开发技术。

【AJAX 的特别之处】

最后一种 AJAX 开发技术与其他 3 种并不属于同一种范畴，严格说来它更类似一种具体的技术实现方法，可被包含于前面 3 种开发技术的任何一个里。AJAX 技术的简单原理将在 3.2.5 节介绍。

3.2.1 认识 Java

本节简单介绍 Java 平台的 Web 开发技术总体特点。Java 是一种简单的、面向对象的、分布式的、解释型、支持多种操作系统、多线程的动态编程语言。1995 年，由大名鼎鼎的 Sun 公司推出，之后不久便成为主流的 Web 开发技术。

【Java 原名 OAK】

Java 语言最早诞生于 1991 年，起初被称为 OAK 语言，是 Sun 公司为一些消费性电子产品（类似现在使用的手机等）而设计的一个通用环境。他们最初的目的只是为了开发一种独立于平台的软件技术，而且在网络出现之前，OAK 可以说是默默无闻，甚至差点夭折，但是，网络的出现改变了 OAK 的命运。

在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那

些迷恋于 Web 浏览的人们来说简直不可容忍。他们迫切希望能在网络中看到一些交互式的内容。开发人员也极希望能够在 Web 上创建一类无须考虑硬件平台就可以执行的应用程序。当然这些程序还要有极大的安全保障。对于用户的这种要求，传统的编程语言显得无能为力。SUN 公司（2009 年已被 Oracle 公司收购）的工程师敏锐地察觉到了这一点，从 1994 年起，他们开始将 OAK 技术应用于 Web 上，并在第二年正式以 Java 这个本意为爪哇岛或者咖啡的名字来命名。

3.2.2 Java 语言的特点

Java 主要有平台无关性、面向对象、安全性、分布式等几个特点。其中，和 Web 开发有相对更紧密关系的是下面两个特点。

1. 平台无关性

平台无关性是指 Java 能运行于不同的平台。Java 引进虚拟机原理，并运行于虚拟机，实现不同平台的 Java 接口之间。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器无关，Java 虚拟机（Java Virtual Machine）是建立在硬件和操作系统之上，实现 Java 二进制代码的解释执行功能，提供于不同平台的接口。

电子商务要求程序代码具有基本的要求：安全、可靠，同时要求能与运行于不同平台的机器的全世界客户开展业务。Java 以其强安全性、平台无关性、硬件结构无关性、语言简洁同时面向对象，在网络编程语言中占据无可比拟的优势，成为了实现众多电子商务系统的首选语言。

如图 3-12 显示了某网站的首页。对于使用者来说，如果在浏览器的地址栏中出现 jsp 或者 servlets 等字样，就可以判断出当前系统采用了 Java 平台的 Web 开发技术。

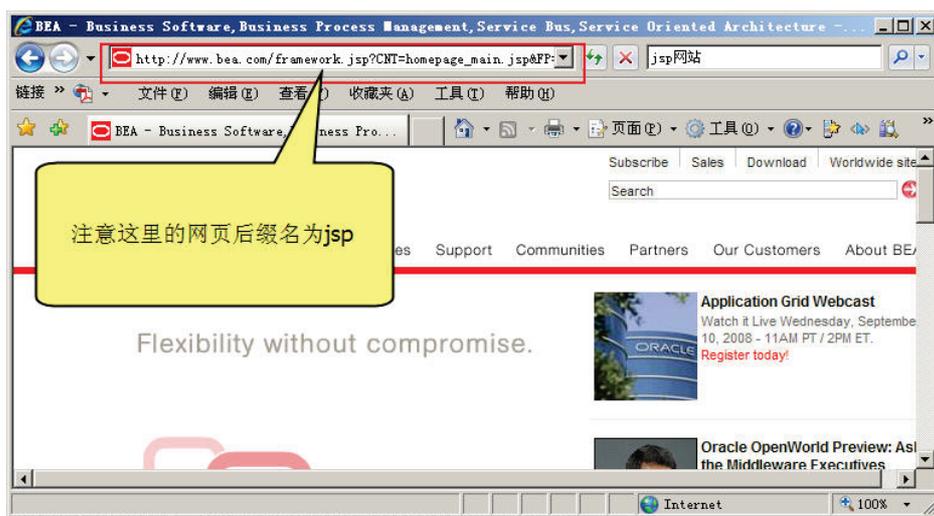


图 3-12 应用 Java、Jsp 技术的某网站首页

情况下只能在 Windows 平台下运行。.NET 平台与 Java 平台类似，在程序运行所在的机器上都要安装一个运行库，称为 .NET Framework。至笔者截稿为止，最新版本为 4.5。

在 Web 开发方面，.NET 平台提供了服务器端网页文件，其后缀名为 `aspx`。另外，在 .NET 平台推出之前，微软的 Web 开发解决方案主要采用后缀名为 `asp` 的服务器端脚本技术来实现，目前该项技术依然有一些使用者。如图 3-13 显示了 IE 8 中某网页的效果，从地址栏中的后缀名即可判断出来它采用 `aspx` 编写。代码 3-2 则是读取并显示某文件属性的 `aspx` 代码。



图 3-13 用 .NET 技术开发的一个网页

代码 3-2 读取并显示文件属性的 `aspx` 代码

```
<% @Page Language="C#" Debug="true"%>
<% @Import Namespace="System.IO" %> //引入 System.IO 命名空间
<html>
<head><title>File Info</title></head>
<body>
<%
string strFile2Show = Request.QueryString.Get("file"); //传入文件名作为参数
File thisOne = new File(strFile2Show); //取得该文件对象
%>
<table>
<tr><td>文件名: </td><td><%=thisOne.Name%></td></tr>
<tr><td>全名: </td><td><%=thisOne.FullName%></td></tr>
<tr><td>文件创建日期:</td><td><%=thisOne.CreationTime.ToString() %>
</td></tr>
</table>
</body>
</html>
```

【CodeFile 方式存放代码】

`aspx` 页面代码既可以存在于网页本身之中，也可以单独成为代码文件。前者即是动态网页的传统方式，与 JSP 等类似，用 `<%` 和 `%>` 标签将代码与 HTML 标签隔离开来；后者则是 CodeFile 方式，将代码保存在单独的 `aspx.cs` 文件中，在网页头部用 CodeFile 属性指向

这个代码文件。这样做的好处是页面表现与具体逻辑分开，另外代码文件被编译，比直接写在网页上的直译程序快，还可以防止程序代码外泄。

3.2.4 基于 PHP 的 Web 开发技术

PHP 和以上两种平台不同，它是开放源代码的。一般来说，在实际的开发和应用中，PHP 一般和开源数据库 MySQL 搭配使用。

如图 3-14 是 IE 8 中显示的利用 PHP 技术开发的某网页，代码 3-3 则显示了一个完成读取数据库功能的 PHP 页面代码。

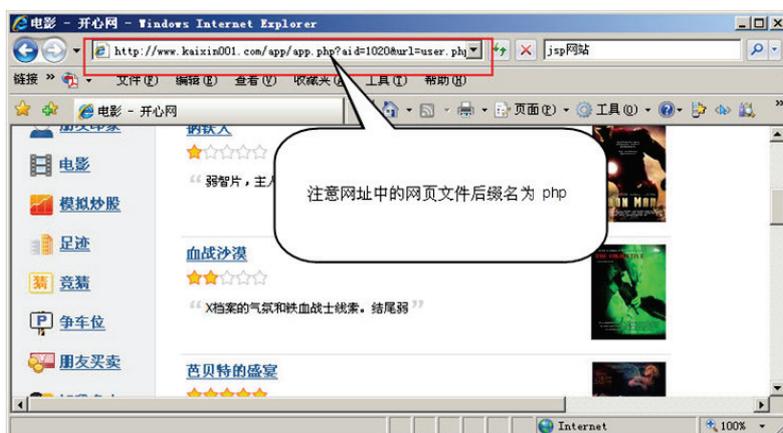


图 3-14 利用 PHP 技术实现的一个网页

代码 3-3 对数据库进行查询的 PHP 代码

```
<?php
#这是 PHP 的注释符号。下面一行是包含文件的写法
include("globalenv.php");
#判断传入参数是否存在
if(array_key_exists('user',$_GET)!=false){
#若参数存在，判断是否为空值
    if($_GET['user']!=NULL){
        #取得参数值
        $user=$_GET['user'];
        #数据库连接
        mysql_connect('localhost:'. $port, $username, $password);
        @mysql_select_db($database) or die("Error:Unable to select the data
        base");
        #获取当前时间
        $now=strftime('%Y-%m-%d');
        #计算出距今 7 天前的时间
        $lastWeek=date("Y-m-d", (strtotime("last week", time())));
        #查询 SQL 语句的拼接
        $query="SELECT * from play where user='".$user.'" and dateCreated
        between '".$lastWeek.'" and '".$now.'" order by dateCreated desc";
        #包含其他文件
        include("querydb.php");
    }
}
?>
```

注意其中 PHP 代码的分隔符是<?php 和?>, 当然, 也可以修改为<%与%>, 不过要额外设置。

【测试工程师的选择】

前面介绍的 3 种 Web 开发技术三足鼎立, 代表了目前网站开发的 3 种主流平台, 他们都各有千秋, 有自己固定的“粉丝”阵营。从以上几张图片及代码片段来看, 3 种平台技术实现的页面在外观、代码上并没有太多本质不同, 另外, Web 应用一般只会采用其中一种平台进行开发, 因此对于测试工程师来说, 如果主要进行黑盒测试, 并不需要对以上 3 种技术的具体编程技巧都非常了解, 只需要根据当前被测试 Web 应用的实际情况, 尽量多熟悉这种技术即可。

3.2.5 AJAX 开发技术

AJAX 开发技术与以上 3 种 Web 开发平台并不属于同类范畴, 它可以说是一种处理客户端与服务器交互的新方法, 因此 3 种平台都可以应用。

【AJAX 的设计思想】

AJAX 技术的主要思想就是在不用页面刷新的情况下, 在网页后台提交信息给服务器并返回结果。由于省去了页面刷新的过程, 所以 AJAX 页面只改变需要改变的内容部分, 其他网页元素不再重复装入浏览器, 省去了从服务器传输这些数据的时间和字节数, 从而使得用户感觉到的页面响应时间变短, 改善了用户体验。

如图 3-15 和图 3-16 分别显示了在利用 AJAX 技术的某网站中, 提交评论之前和提交过程中的界面, 可以发现页面并没有再次刷新。

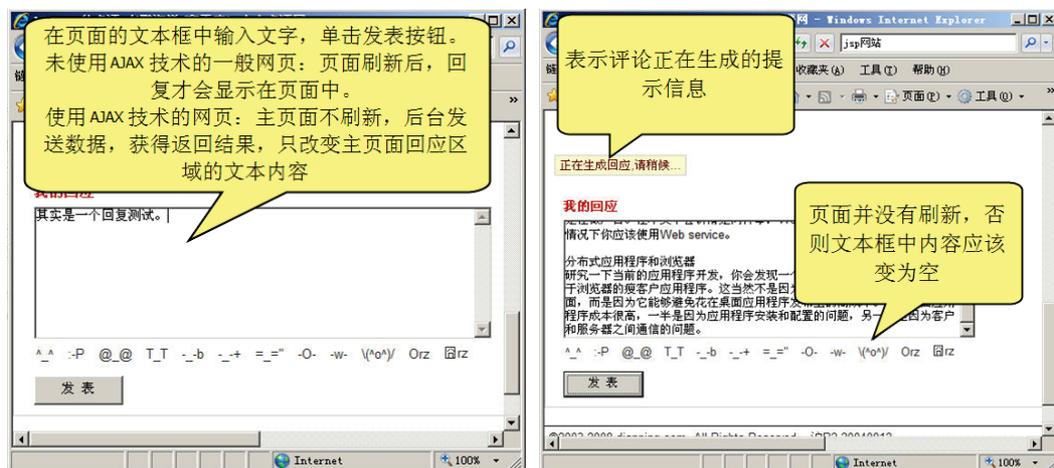


图 3-15 利用 AJAX 技术使得提交回复不刷新当前页面 图 3-16 回复提交过程中页面并没有刷新

在简单介绍了以上这几种技术之后, 相信读者已经对当今 Web 应用的主流技术有了粗浅的感性认识。因此, 3.3 节将回归到测试工作上来, 讲述 Web 测试的内容和功能测试的特点与方法。

3.3 Web 功能测试的特点与方法

3.2 节概述了 Web 开发的几种主流平台，还介绍了 AJAX 技术。对于具体技术有一定程度的了解是 Web 应用测试工程师应该具备的，因为它将非常有利于在工作中理解软件实现、开展白盒测试。不过，由于黑盒性能测试是本书的主要内容，笔者还是将被测试的网页作为黑盒来处理，不会过多考虑它背后使用了何种技术来实现。

本节是本章的重点，通过举例概要讲述 Web 测试的方法。在本节的开头，首先提出一个问题，请读者想一想，如果自己是小白，面对一个全新的网站，该如何着手去测试呢？

这个问题没有什么固定答案，但是，有一个答案可以说是很自然就能想到的：把自己所能看到的网页各个部分都“测试”，或者说单击一遍。之所以把测试这个词加上了引号，是因为目前这种操作方式更类似对网页的试用，不具备系统化、计划性的特点。

下面就从这个肤浅的出发点开始。

3.3.1 网页测试的组成部分

如图 3-17 是某网站一个典型的网页，首先观察一下网页中包含了哪些内容，一旦确定了这些内容，就可以对每一项分别展开测试。



图 3-17 一个典型的网页

在图 3-17 中，我们发现了文本框、下拉菜单、表单提交按钮（即查找按钮）、各类图片、文字链接和实现网页语言选择的链接（有 English、繁体中文等多个）等。因此，需要测试这些元素显示是否正常，功能是否正常，这就是网页的功能测试部分。

具体而言，网页的功能测试需要验证以下方面：

- ❑ 文字或者图片链接的有效性测试，简称链接测试。
- ❑ 表单提交测试，简称表单测试。
- ❑ 网页内容、用户界面和多语言测试。
- ❑ 浏览器交互测试，比如修改浏览器选项设置，以发现当前网页可能存在的问题。
- ❑ 其他网站特定功能的测试：比如上传功能等。

除了功能测试之外，网页测试还要考虑如下一些问题：

- ❑ 性能测试（包括负载/压力测试）：本书讲述的主要内容。
- ❑ 兼容性测试：对各个厂商的浏览器，及其不同版本之间网页显示是否正常的测试。
- ❑ 安全测试：Web 应用是否构成安全隐患，比如造成密码泄露、数据库被侵入等的风险。

由于性能测试是本书的重点，将在后面的章节陆续讲到，这里不再叙述。对于功能测试，由于内容相对其余的几种要多些，将在 3.3.2 节中做具体的讲解。至于兼容性测试和安全测试，将在本章的后面内容中再做介绍。

3.3.2 链接测试及其要点

本节讲述链接测试及过程中需要注意的问题。

1. 链接测试的目的

链接测试验证网页上的所有链接，目前包括文字、图片（静止图片和动画图片）和 Flash 动画 3 种，是否都指向正确的、真实存在的网页或者其他位置（比如下载的文件、网页的一个锚点）。需要注意的是，验证链接本身格式、字体大小等是否符合设计说明，属于后面要讲述的网页用户界面测试，将在随后讲到。

具体而言，做链接测试一般都要对表 3-1 中列出的 3 个问题进行验证。

表 3-1 链接测试需要关注的问题

问 题	出现错误时候的一般表现
链接指向是否正确？是否指向了正确的网页？	打开了其他的网页
指向的网页是否存在？	页面找不到的 404 错误
链接的打开方式符合设计说明吗？如果是利用 JavaScript 等脚本打开新网页，能否成功实现？ (链接的打开方式指的是单击链接后，新开页面还是在当前页面跳转)	错误的新网页打开方式。单击 JavaScript 后浏览器无反应 (注意：如果浏览器禁用了弹出窗口，浏览器会出现提示)

2. 不同链接的测试要点

对于文字链接，没有特别之处。对于图片链接，验证要复杂一些，要考虑以下的特殊情况：

- ❑ 有的图片链接是以图片地图的形式存在的，即一幅图片，不同的部位指向不同的目标网页，多用于导航条和网站地图等场合。如图 3-18 是 Yahoo 网站上各个国

家或者地区分站的入口。虽然是同一个世界地图的图片，但单击不同的蓝点会切换到不同的页面。

- ❑ 有的图片是动画 gif 格式或者直接就是 Flash 动画，这时候要注意验证动画的每一帧（即每一次图像内容变化）所链接的网页是否正确。容易被忽视的恰恰是它们第 2 帧之后的链接，因为测试工程师往往在网页显示完毕之后就测试各个链接，这时候动画图片后几帧尚未显示。不同帧对应不同链接的动画图片使用较少，但在首页的新闻提示栏或者各个友情链接被合并成一个动画图片的时候，还是能够碰到的。
- ❑ 对于以上这几种图片或者 Flash 动画，只能把它们当成多个链接来处理，一个一个地进行验证，确保测试完全。



图 3-18 Yahoo 的图片地图导航

3. 链接测试方式的选择

由于网页上的链接一般都非常多，手工地逐个验证是很容易遗漏和出现错误的。这时候可以考虑利用自动测试的方法：

- ❑ 利用某些链接验证工具软件进行网页的链接测试。
- ❑ 利用某些测试工具软件的脚本录制功能，将用户单击链接的行为录制下来进行验证。
- ❑ 利用 HTML 等知识，通过验证 DOM 里 <a> 标签的值进行测试。

三种方法都需要较多的测试准备工作，第二种方法的编程量可能会更多一些。因此，在实际使用中，要综合考虑进行方法的取舍。

【手工或自动链接测试的选择】

如果被测试的链接经常改动，无法预知，不妨直接采用手工测试链接的方式。反之，如果各个网页版本某链接都比较固定，可以将它的验证变为自动，这样可以省去大量的时间。

3.3.3 链接测试工具 Sleuth

在前文中，我们介绍了可以用某些链接验证工具软件达到测试的目的。这里简单介绍一种免费链接验证工具 Xenu's link sleuth，简称 Xenu 的使用。

所谓 Sleuth，英文本意是指动物的足迹，用来形容链接测试，发现哪里链接断了（相当于痕迹没有了），还是很贴切的。该软件可以在 <http://home.snafu.de/tilman/xenulink.html> 下载到，是一个压缩包，解压缩后软件不大，几兆而已。

压缩包中只有一个 Setup.exe，运行它，安装 Xenu 软件。安装完毕后，运行该软件，主界面如图 3-19 所示。

Xenu 软件的主要功能都是通过图 3-19 中高亮显示的两个菜单命令来完成的，只需要分别单击这两个命令，在其中输入待检测的网址，即可开始测试过程。测试结束后，根据提示确定即可生成链接检测的报告，如图 3-20 所示。

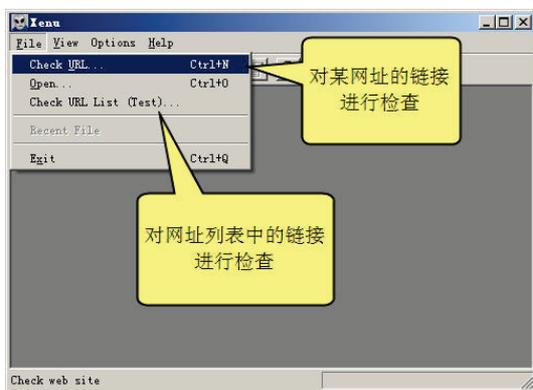


图 3-19 Xenu 的软件运行界面

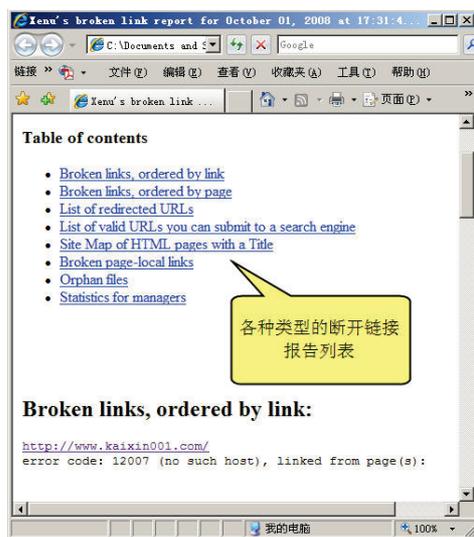


图 3-20 Xenu 针对某网址生成的链接验证报告

与另外的链接测试软件相比，Sleuth 有如下优点：

- ❑ 界面简单易用。
- ❑ 可以反复验证链接（对于偶然的网络错误导致链接不可访问是很有用的，增加了可靠性）。
- ❑ 可以发送邮件报告（可以让它晚上不上班的时候去测试，测试人员上班后查看邮件即可）。
- ❑ 支持安全套接层 SSL 的网站（也就是包含 https://开头那样的网页，用于安全性要

求比较高的场合，比如网上支付等）。

- 能够支持互相跳转网页上链接的验证。

【W3C 的官方工具】

此外，互联网标准组织 W3C 的官方网站上也有一个网页用于链接检查，网址是 <http://validator.w3.org/checklink>。它可以用于简单的链接不很多的网页验证，适于对网站的链接情况做一个大致的了解，因为并不需要软件的安装，使用快捷。

3.3.4 孤儿网页

和链接有关的还有一种所谓的孤儿网页，英文叫做 Orphan page。它是指在网站中的某个网页没有被其他任何网页上的链接联系到，像没有亲人的孤儿一样，故而得名。孤儿网页只有在用户完全知道页面的全部地址时才能被访问到。

【孤儿网页产生的原因】

孤儿网页产生的原因一般都是网站的页面不断修改、变化版本，导致之前开发后来被新页面替代的旧网页遗留在服务器上造成的。

既然没有其他链接能够指向孤儿网页，那么它留在服务器上也没有坏处吧？

回答虽然是没有很大的坏处，但容易产生 3 个方面的不利影响：

- 用户之前收藏了旧网页，并一直通过收藏夹直接访问该网页，造成用户获得信息得不到更新。这在获取网站的联系方式等页面上影响还是相当大的。
- 孤儿网页可能泄露一些网站的秘密信息，这样的孤儿网页一般是开发人员为了方便创建的，如果不在网站开发完成后及时删除，有可能泄露一些网站、服务器的安全信息，对网站造成潜在的风险。
- 孤儿网页增多会混淆网页功能，并占用服务器的空间。当然，现在硬盘和白菜一样便宜，损失几 KB、MB 硬盘空间并没有什么，但毕竟是一种浪费。另外，由网站旧版本文件演变形成的孤儿网页会干扰当前版本的网页，有时候会使得开发人员错误地修改了旧版网页，并将其替换为现有网页，导致网站功能出现异常。

因此，测试工程师有必要及时指出孤儿网页的存在及其位置。前文所讲述的 Xenu 软件也有发现孤儿网页的功能，如图 3-21 所示。

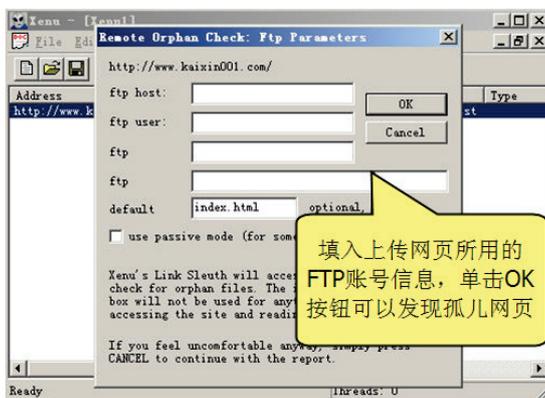


图 3-21 利用 Xenu 软件发现孤儿网页

其他不少的商业网站开发工具也都有这样的功能，比如业内常用的 Dreamweaver 等。

3.3.5 表单测试及其要点

表单是网页上用来接收用户输入的信息，并将结果反馈给用户的桥梁。表单由一系列文本框、下拉菜单、选择框和提交按钮组成的。表单测试的目的就是验证网页上的表单能否正确提交信息，正常工作。

根据表单使用的场合和一般网页的设计要求，表单正常工作指的是：

当使用表单进行网站用户的在线注册、订单生成、修改的操作，以及用户反馈信息、搜索关键字的收集时，要确保提交按钮单击后能够提交全部需要的信息至服务器，这些信息的部分或者全部能够保存到后台数据库或者其他形式的文件中。表单提交完成后，应该给予用户成功或者失败或者是用户需要的列表等有意义的信息。

表单测试其实包含了很多的内容，根据实际操作的过程，在实际的测试工作中主要分为 3 个步骤：

- (1) 客户端表单信息的验证、收集和提交。
- (2) 服务器端用户信息的保存过程。
- (3) 服务器端提示信息的返回。

下面将分别介绍这 3 个过程的测试要点。

3.3.6 客户端表单信息的验证、收集和提交

这个过程，对于用户来说，就是在各种各样的输入框中输入信息，然后确认无误后单击提交按钮的过程。由于所有的操作都是在用户的机器上通过浏览器打开网页来完成的，所以称为客户端输入信息过程。

一般情况下，我们必须验证用户输入信息的有效性。这么做主要有两个原因：

(1) 保证用户输入数据的有效，使得网站能够获取相对真实的用户信息。比如下面几种情况，都需要对数据进行验证。

- ❑ 对于网民来说，日期具备时效性，比如生日不可能是 1850 年 12 月 1 日。
- ❑ 对于网民来说，日期要符合规范，比如生日不可能是 1980 年 13 月 56 日。
- ❑ 对于目前的网上支付来说，信用卡号码必须符合发卡方的标准。
- ❑ 对于邮政编码、年龄等数据，根据需要，可以限制文本框中只能输入数字。
- ❑ 对于电子邮件来说，必须符合标准，简单地说，要具有@和.等符号。
- ❑ 对于备注、用户反馈类的信息，不可能让其填入很多的文字，要满足一定的字数限制。

(2) 保证服务器端程序的安全性。有一种“注入”（Inject）攻击就是通过用户输入特殊字符对服务器端数据库进行破坏。要在用户提交的信息中找到这些字符并过滤掉，这个过程可以在客户端进行，也可以在服务器端正式提交前进行。

【如何编写表单测试的用例】

总体说来，客户端表单信息的验证是比较重要的，Web 测试工程师需要利用前面章节介绍的划分等价类、边界值测试等进行充分的工作，保证不把不符合规范的数据提交到服

务器端。

3.3.7 服务器端用户信息的保存过程

这个过程，一般是将用户信息经过处理保存到数据库或者文件中。在实际测试当中，可以模拟一些客户端输入的数据，观察表单的行为。比如下面这些情况：

- ❑ 用户输入了合法的数据，但数据长度超过了数据库某字段的定义长度，也就是说存储数据的地方不够了，观察服务器端程序的行为和反馈。这样的情况，根据网站设计说明，一般有两种情况，即直接报错和截取用户输入的可保存最大长度。对于前者方案的测试，需要修改客户端验证的文本长度限制或者服务器端数据库字段的定义长度；对于后者，则需要验证服务器端程序这部分代码的功能能否正常实现。
- ❑ 用户输入了客户端无法验证的数据，但这些数据对于服务器端是不合法的。
- ❑ 一些特殊情况的容错，比如断网等连接中断下，服务器端程序和数据库事务的处理情况等。

在这个测试过程中，模拟客户端输入的数据就是我们前面章节所说的测试用例。通过将不同的测试用例提交到服务器端，查看程序的行为，是 Web 测试的一般工作过程。

3.3.8 服务器端提示信息的返回

在服务器端程序处理完毕之后，无论结果是正确还是错误，都要给予用户以可阅读的反馈，避免出现过于简单的提示或者只有工程师才能读懂的错误代码。它的要求如下：

- ❑ 无论正确还是错误，都能返回到一个或者相应的结果页面。
- ❑ 对于正确的结果，提示信息要为用户着想，尽量完整。按照网站设计说明，对用户下一步的操作给出建议，或者过一段时间跳转到初始界面。
- ❑ 对于错误的结果，提示信息要列出一些可能的原因，同时表明表单提交的状态，并按照网站设计说明对用户下一步的操作给出建议。

在这个过程中，主要需测试的是信息的完备性，对用户是否友好。

3.3.9 网页内容测试

这部分主要是为了保证用户体验，使得网站对用户友好。它分为两大类：

- ❑ 网页内容测试：针对内容信息是否正确的测试。
- ❑ 网页用户界面测试：网页除去内容部分，针对各元素功能、页面风格等方面的测试。

【网页内容测试要点】

网页内容测试根据网站内部各部门对于页面内容的要求，检验 Web 相应页面提供信息的正确性、准确性和时效性，它们也是网页内容测试的目的。

具体而言，这 3 个要求的详细内容如下：

(1) 所谓信息的正确性，是指网页上的信息是可靠的还是有谬误的。在前几年，曾经

有这样一个案例，某个很大的在线购物网站标错了商品的价格，在很短的一段时间，就有不少用户订购了该商品。由于价格远远低于实际销售价格，销售商拒绝发货，最终网站和用户只好通过司法途径解决问题。这个案例充分体现了信息正确性的重要性。

(2) 信息的准确性则主要针对专业术语，比如网站上的各项用户协议、在线支付的各种文本等都要符合相关规定和标准，要经过比较严格的检查，否则也会产生一些意想不到的后果。

(3) 信息的时效性主要保证信息的有效时间处于最近的时期内。这对于网站上，特别是首页的一些活动预告、广告宣传等是有用的。设想如果网站有一个嘉宾访谈类节目，但时间是去年某个时候举办的，当用户访问首页的时候，这个广告一直弹出来，很难设想用户看到后的反应。

有的人会问，这些都是编辑的工作，为什么要和 Web 测试工程师相关呢？测试工程师只测试和程序相关的内容，其实，这样理解是不对的。因为：

测试工程师负责测试整个网页。网页的内容也是网页的一部分，对用户体验有很大的影响。网页上的各个元素之所以存在，是为了用户更方便、更快捷、更舒服地浏览网页内容。

测试工程师有责任和义务发现问题。我们在前面的章节介绍过，错误的产生在所难免，比如，编辑在 Word 中写好的文字，可能由于其他人的感觉，觉得在网页中对齐不方便就增加几个字以保持美观。在这些情况下，测试工程师有责任、有义务尽早地发现问题，将修正 Bug 的成本降到比较低的水平。

3.3.10 网页用户界面测试

用户界面，英文名称叫做 User Interface，也简称 UI，是软件呈现给用户的外观部分。网页用户界面直接和用户体验相关，一般需要网站策划、网站内容负责人、网站美工、网页制作人员和网站测试工程师协同完成。测试工程师的工作一般是验证用户界面是否符合设计说明。

网页用户界面与表单测试不同，它是从大处着眼，一直到最小的页面元素，测试范围从总体的结果布局到具体的文字、图片、链接等网页元素。主要测试内容包括：

(1) 网页整体测试。这是指整个 Web 应用系统的页面结构设计，是给用户的一个整体感，主要考察页面的风格是否统一，易用性如何。测试时，可以考虑如下几个问题：

- 用户浏览网站的感觉是否舒适？
- 用户要查找的信息是否易于发现？
- 整个网站的设计风格是否一致？在切换栏目的时候是否有“跳出当前网站”的感觉？

(2) 页面导航测试。网页上的导航条或者菜单列举出用户在网站内切换的便捷方法，主要由文字或者图片链接、跳转按钮、单独的 Div 浮动层、下拉列表和网页窗口等组成。在测试时需要考虑下列问题：

- 网站现有导航是否直观？
- 网站现有导航是否以首页或者各栏目的首页为出发点，能够做到来去自如？
- 网站现有导航是否足够？是否需要单独的页面列出站点地图、提供搜索框或者专

门的帮助页面？

(3) 页面样式表测试。这部分主要针对网页上各个元素的显示进行测试，比如：

- 图片的替换文本是否具备？图片的大小是否限制？
- 文本、按钮等元素的字体、颜色、边框是否符合要求？同一个页面没有特殊要求的文字字体是否一致？
- 网页各应用了样式风格的元素是否符合设计说明中的样式表设定？样式表的层次关系是否与设计保持统一？

对于用户界面测试，一般都是采用人工测试的方法，但是，验证部分重要文字的内容（比如按钮文字、重要链接等）、验证页面元素位置等可以通过自动测试的方法：利用 JavaScript 获取 HTML 或网页相应标签的 InnerText 值、Top 值等，将其与标准的参照值进行比较确定正确显示与否等方法来实现。

3.3.11 浏览器交互测试

浏览器交互测试内容相对简单，主要考察网页与浏览器交互的部分功能是否正常，有如下几个问题需要考虑：

- 如果网页需要在浏览器状态栏中显示文字和一些特殊效果（比如部分网站采用的文字行进效果），确认它是否正常。
- 在浏览器中进行网页大小缩放比例的改变时，显示效果能够正常。
- 在浏览器中能否正确显示网页的题目。

以上3点的示意图如图 3-22 所示。

(1) 在某些网站，浏览内容时需要安装特别的插件，比如 Flash 播放插件、数字版权插件、播放音视频的插件。这时候，浏览器会弹出一些信息加以提示，在 IE 浏览器中是一个黄条，称为“信息条”（Information Bar）。要给予用户足够的信息对这些插件进行说明，保证用户的使用，如图 3-23 所示。



图 3-22 和环境和谐相处：网页与浏览器的交互

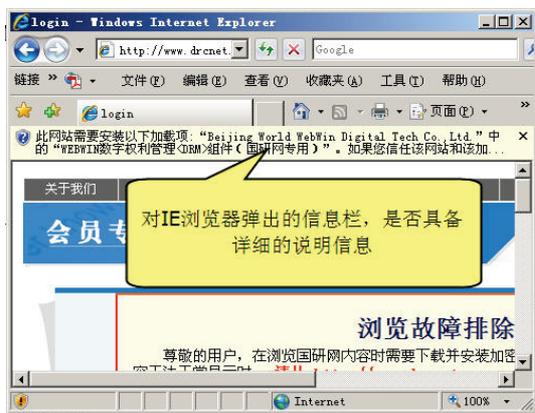


图 3-23 IE 浏览器中的信息条

(2) 在浏览器中修改字体的大小，网页中字体的大小显示要正常：

- 对于应用有样式表的文字，字体应该忠实于样式表的设置，大小不变。

□ 对于没有应用样式表的文字，字体应该服从浏览器的大小设置。

在 IE 中设置网页字体大小的菜单如图 3-24 所示。同样，对于其他浏览器也有类似设置。浏览器同时还可以禁用网页上应用的样式表，有时间的话也可以测试一下页面显示效果，以确保更多的用户能够正常浏览。



图 3-24 修改 IE 浏览器中网页显示的字体大小

【人工与自动的选择】

这一部分测试主要由人工来完成。自动进行浏览器交互测试也是可以实现的，但相比其重要性而言，实现的成本有一点不值得。这是因为正在使用的浏览器有很多种，针对每一种都需要开发自动测试平台，代码和维护的工作量较大，稳定性也不容易做的很好。另外，浏览器也是不断变化的，很可能有新的主流浏览器涌现。而且，对于 Web 应用来说，一般而言，不同浏览器下的表现大部分情况是一致的。

3.4 其他 Web 功能测试

针对 Web 应用进行的其他功能测试，包括前面没有提到的所有属于功能测试的部分，很难归为一大类。它主要包括 Cookie 测试、Web Service 测试和网站自定义功能的测试几部分。

3.4.1 Cookie 测试

Cookie 通常被网站用来存储用户信息和用户操作状态，以备下次访问网站或者访问其他网页时使用。Cookie 的生成过程大致如下（假设用户的浏览器打开了 Cookie 功能，这也是默认的设置）：

(1) 用户访问了某个应用了 Cookie 的网站。

(2) 该网站的 Web 服务器将像用户浏览器发送关于用户输入的信息等，浏览器将这些信息以特定格式的文本文件存储在本地计算机的特定目录中。

(3) 当用户再次访问该网站, 或者此时访问该网站的其他网页时, 网站可以根据 Cookie 中存储的信息发送不同内容的页面或者显示不同的内容。

Cookie 测试的内容就是检查 Cookies 是否能正常工作, 诸如下面的几个方面:

- Cookies 是否起作用, 客户端计算机硬盘中是否生成了类似文件。
- 由于 Cookie 有时间上的有效期, 确认 Cookie 的时效性是否设置正确。
- 网页读取、保存、修改用户信息到 Cookie 的功能是否正确地实现。
- 网页的刷新对 Cookie 的影响是否正常。
- 若在 Cookie 中保存注册信息, 确认该 Cookie 中的信息, 特别是设计到密码的部分已经加密。

可以通过一些工具软件来验证当前网页设置的 Cookie 内容, 比如 IE 下的 IE developer toolbar 和 Cookies Manager 等, 也可以通过编程来实现验证 Cookie 内容。要注意的是, 测试 Cookie 的代码要与网页中 Cookie 读取的代码实现方法不同, 否则就失去了测试的意义。

3.4.2 Web Service 测试

所谓 Web Service, 实际上就是一个网络应用程序, 它向使用者暴露出一个能够通过 Web 访问进行调用的接口 API。简单地说, 我们能够用在自己的网页中通过编程的方法访问 Web 来调用这个应用程序。比如当前网页上有一个在线支付的按钮, 单击后自动转向银行的支付页面, 用户输入信用卡信息后, 付款成功转回到当前网站。银行的支付页面, 就可以称为一种 Web Service。

对于 Web Service 的测试, 主要工作内容就是: 测试人员模拟一些用户数据, 人工或者自动地提交给 Web Service, 根据不同的返回值来验证功能和容错性。

【Web Service 测试的方法】

这种测试可以利用之前讲过的等价类、边界值等测试用例编写方法来模拟出用户的数据。特别需要注意的是 Web Service 的容错处理。Web Service 一般有一些参数, 可以通过改变传递到 Web Service 参数的数量及相应的数值范围来测试它的行为, 从中发现容错性的 Bug。

3.4.3 Web 功能测试的一般原则

每一个网站都可能会有自己独特的功能, 比如电子相册、买卖朋友、争车位等, 因此, 测试工程师不能满足于前文所讲述的基本 Web 测试方法, 需要对当前网站特定的功能需求进行思考, 找到更多可用于测试的验证方法。在这些各不相同的功能之中, 有一点是它们都具备的, 那就是: 所有的功能都需要用户来使用。

因此, 以用户的视角来观察、使用被测试的网页才是一个最基本的功能测试方法。尽量尝试各种各样背景用户所可能进行的所有操作, 遇到不顺畅、不明白的地方, 要记录下来, 查询网站设计说明, 和其他人员 (策划、美工、开发人员、用户) 多沟通, 这样才能把网站测试的工作做得更好。

3.5 兼容性测试与安全测试

除了很重要的功能测试之外，Web 测试还包括性能测试（本书的主要内容，将在第 4 章开始具体讲解）、兼容性测试和安全测试等。本节将介绍后两种测试。

3.5.1 兼容性测试

兼容性测试在 Web 测试方面可以说是一个比较复杂、令人头疼的事情。这是因为在用户群体的电脑中里存在各种各样的系统，各种各样的浏览器，各种各样的设置，要使得他们都能够正常浏览网站，还真不是一件很轻松的工作。根据软件的不同，兼容性测试可以分为以下几种。

- ❑ 平台测试：考察软件与操作系统之间的兼容性。
- ❑ 浏览器测试：考察软件与各浏览器之间的兼容性。
- ❑ 显示设置测试：考察软件与屏幕显示设置之间的兼容性。
- ❑ 打印效果测试：考察软件与打印设置之间的兼容性。
- ❑ 网络线路速度测试：考察软件与网络速度之间的兼容性。

以上这几个方面，归根结底都是考察被测试软件与外部环境的兼容性。为了“举一反三”，笔者在这里先解释一下平台测试的具体含义，通过它，读者对于其他的兼容性测试，也会很快理解。

【平台测试】

在用户上网的计算机当中，安装着不同的操作系统，最常见的是 Windows，不过 Windows 也有很多版本，XP sp3、Vista sp2 等，更不要说还有 Linux、Mac OS X 甚至 Solaris、HP-UX、AIX 等这些。验证网站是否支持现有的操作系统平台，即可称为平台测试。

下面将讲解上述几种兼容性测试的要点。

3.5.2 平台测试要点

由于种种原因，网站不太可能对于所有平台的最终用户都要满足，这通常是由于如果面面俱到的话，开发、测试成本较大造成的。究竟使用哪一种操作系统，取决于网站设计说明。一般来说，Windows、iOS、Mac、Android 和 Linux 的使用者占绝大多数，保证这 5 种平台用户的浏览正常已经很不错了。各种操作系统的市场份额如图 3-25 所示，这样的数据，包括后面要提到的浏览器市场份额数据，都可以在 <http://marketshare.hitslink.com/report.aspx?qprid=8> 通过选择不同的分类来获得。

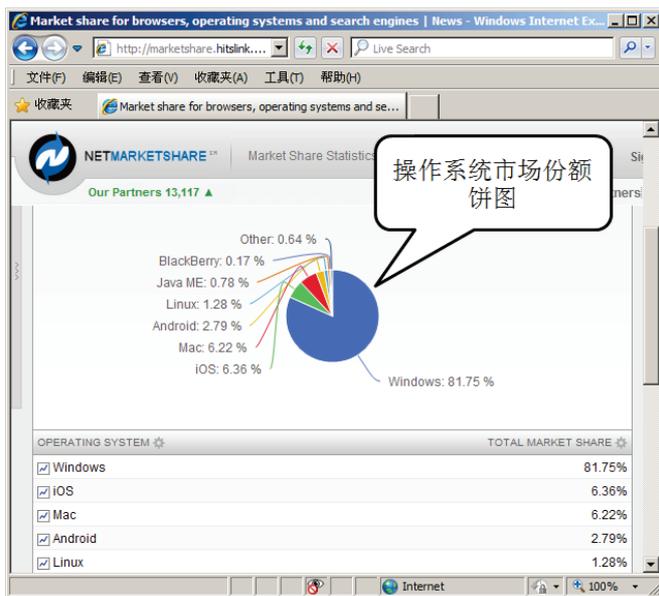


图 3-25 截止 2012 年 12 月的操作系统市场份额图

有人可能会问，我们测试的是网页，显示在浏览器中，应该只和浏览器相关，和操作系统有什么关系呢？实际上并非如此。举个例子来说，被测试网页上有在线播放音乐的功能，那么在不同操作系统上的表现就不一样了：

- ❑ 在 XP sp3 的系统上，系统默认带有 Windows Media Player 9，网页中可以控制 Media Player 插件进行播放。
- ❑ 在 Vista 系统上，系统默认带有 Windows Media Player 11，网页中也可以通过控制 Media Player 插件来进行播放，但是由于版本不同，会有一些差别。
- ❑ 在 Mac 系统上，系统默认播放器是 QuickTime，那么网页上如何放音乐呢？需要有特别的处理。

从上面的这个例子就可以看出，虽然我们测试的是网页，平台测试还是很重要的。

平台测试的目的就是验证网页在当前平台下能否正常工作，找到和平台相关的网页 Bug。

3.5.3 浏览器测试要点

对于网页来说，浏览器测试是非常关键的一项兼容性测试。浏览器是用户能够浏览网页最核心的软件，来自不同厂商的浏览器对 HTML 标签（比如表格间距、框架处理）、CSS 样式表（比如编写规范）、JavaScript（网页元素名称、方法名称等）、ActiveX 控件、浏览器插件 Plug-ins 和安全性等诸多方面都有着程度不同的支持。代码不做调整，基本上无法完全适应所有的浏览器。

浏览器测试的目的正是为了验证网页在特定版本的某浏览器下能够正常显示。目前主流的浏览器有 IE 7、IE 8、Firefox、Safari、Opera 等，IE 9 在本书完稿的时候也已经新鲜出炉，它们各自的市场份额可以通过 3.5.2 节中查询操作系统市场份额的网页，只不过要

选择浏览器份额的链接。

3.5.4 显示设置测试要点

显示设置主要有两部分内容。

(1) 分辨率测试：网页在桌面分辨率为 640×400 、 600×800 或 1024×768 乃至更大的模式下是否显示正常？字体是否太小以至于无法浏览？或者是太大？文本和图片是否对齐？并找到一个最合适的分辨率作为参考和推荐。

(2) DPI 测试：网页在系统处于不同的 DPI 设置下显示是否正常？DPI 是 Dot Per Inch 的缩写，代表每英寸多少点。Windows 系统的常用 DPI 为 96 和 120。数字越大，表明字体越大。在 XP 下修改 DPI 的界面如图 3-26 所示，通过在桌面空白处右击，在弹出的快捷菜单中选择“属性”选项，在弹出窗体中单击“设置”选项卡的“高级”按钮即可发现 DPI 设置的下拉列表框（在 Vista、Windows 7 中的设置方法可能有所不同，感兴趣的读者可自行寻找）。

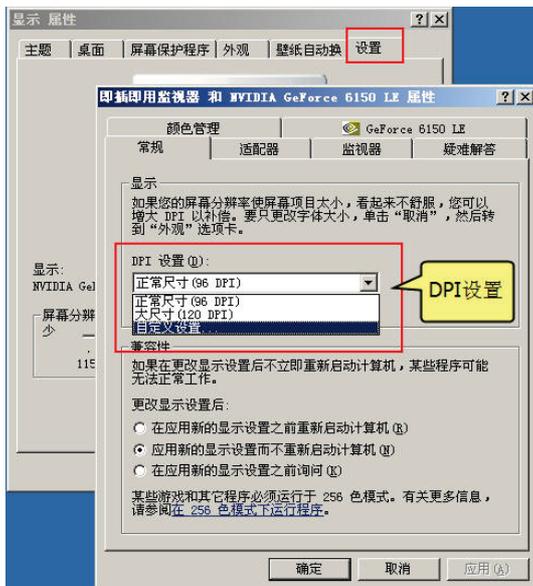


图 3-26 在 XP 下修改系统 DPI 设置

3.5.5 网络连接测试

网络连接测试主要是测试在不同连接速度下访问网页可能出现的情况。它可以是性能测试，也可以说是兼容性测试。不管分类如何，要具备这样的意识，在网站初具雏形的时候进行一下网络连接测试，了解大致情况，找到拖累显示速度的瓶颈，比如图片尺寸的大小、数据库连接的速度等。

目前来讲，主流的用户上网方式有 ADSL 宽带、小区宽带和局域网接入互联网等几种方式，它们的速度分别为每秒几十 K 到几百 K 不等。网络连接测试需要模拟这些用户的不同

同上网条件，这种速度上的限制可以通过专门的工具软件限制带宽来实现，感兴趣的读者可以通过搜索来自行获得。

3.5.6 打印测试

对于报名表格等重要的信息，用户一般会将网页打印下来。因此网站在设计和实现的时候就要考虑到打印的问题，采用包含打印风格的 CSS 样式表（注意样式表中的 Media 类型指定为 printer）。同时，为了验证这样的样式表是否真正起作用，有必要进行人工的测试，确保网页信息能够在页面内清晰、无损失地打印出来。

3.5.7 安全测试

目前，网络所面临的安全形势是比较严峻的，各种各样针对服务器的攻击，网络黑客因此进行安全性测试非常必要。网络安全测试的主要目的在于：

- ❑ 验证网站本身的安全性，发现安全性问题和风险。
- ❑ 确保网站保存或者传递的用户个人信息不被盗用。

网络安全是专门的一个学科，本书这里只是简单介绍基本的入门知识。针对常见的普通攻击方法，安全测试至少要进行如下的验证：

- ❑ 确认网站目录权限设置：在测试的时候，不输入具体的网页文件名称，只输入到目录一级，验证网站的目录浏览功能是否开启。如果开启，要在网站服务器中关闭，以防止信息泄露。
- ❑ 安全套接层 SSL（Socket Security Layer）的基本验证：很多网站使用 SSL 来进行数据的加密传输。当浏览器的地址栏中协议名称变为 `Https://` 的时候，就可以确认进入了一个支持 SSL 的站点。测试工程师需要验证进入、离开这样的页面时，无论成功还是失败时都会有较完备的提示信息。
- ❑ 用户登录信息的验证：很多网站采用登录用户才能查看或者进行更多操作的方式，以验证他们的身份。对于用户登录信息，特别是密码的保存，测试工程师要验证相关 Cookie 的设置、数据库对于密码的保存、网页 Session 超时的设置、文本框对于特殊字符的过滤、密码规则的有效性、登录失败次数的限制、登录 IP 地址的限制等功能的实现，作为网站功能测试的一部分，将测试的结果反馈给网络安全工程师等相关人员。
- ❑ 对于向 Web 服务器发送空 User-Agent、非法 Head 内容等被恶意篡改信息的情况，网站是否能正确处理。
- ❑ 还有其他的一些与安全相关的操作，由于测试工程师的网络安全知识有限，需要网络管理员、网络安全工程师等人员的多方参与。

以上这些属于基本的 Web 安全测试，如果网站在安全方面处理不好，也会带来性能问题，比如处于拒绝服务攻击（DDOS）下的情况。目前网络上有一些工具软件可供我们使用，比如微软的免费工具 WFetch，就可以用于 IIS 站点的某些安全性测试，感兴趣的读者可进一步阅读相关书籍。

由于网络安全是一个很复杂的领域，测试工程师可能无法掌握很多的专业知识，因此，在安全测试方面，测试工程师更需要做的是：按照各方（技术总监、项目经理、网络管理员、网络安全工程师、网站开发人员和测试工程师）所讨论制订出来的网站安全性说明书来进行测试，将结果反馈给各相关人员，以便及时调整测试目标，采取进一步的安全性措施，从而不断提高网站的可用性。

3.6 本章小结

本章对 Web 应用和 Web 应用的测试进行了简介，这样做的目的是为了明确一个前提：对 Web 应用进行性能测试需要首先保证它的功能是可用的。

也就是说，在完成大部分 Web 应用的功能开发，并进行了本章所介绍过的诸多 Web 测试之后，性能测试才能提上日程，同时其测试结果也才更有意义。在本章的开始，介绍了当前 Web 应用的几大技术平台，其中较主流的是如下 3 种：

- ❑ 微软的 .NET 平台；
- ❑ Sun 公司的 Java 平台；
- ❑ 开源的 PHP 平台。

另外，近年来为了提高 Web 应用与使用者的交互性，还兴起了 AJAX 技术。除了对 Web 应用的基本知识进行介绍之外，本章重点在于介绍 Web 功能测试的主要内容，其包括链接测试、表单测试、网页内容测试和用户界面测试等多种类别。

在注重功能测试的同时，兼容性测试（比如浏览器测试、平台测试），与安全性测试都是需要测试工程师在工作中特别留心的方面。对于 Web 测试，测试工程师需要建立一个从用户使用场景出发的“用户视角”方法，这样才能更好、更全面地发现问题。

有了前几章的铺垫，相信读者对于一般的 Web 应用测试已经有所了解。从第 4 章开始，将详细地讲解有关 Web 应用性能测试的知识。