**WEB网站测试**

[● 9WEB网站测试 3](#_Toc11515)

[● 9.1WEB网站概念 3](#_Toc6893)

[● 9.2WEB网站的特点 3](#_Toc6584)

[● 网络集约性 3](#_Toc10596)

[● 内容驱动性 3](#_Toc25702)

[● 持续演化性 3](#_Toc2748)

[● 即时性 4](#_Toc4437)

[● 安全性 4](#_Toc20905)

[● 美观性 4](#_Toc4484)

[● 9.3网页测试 4](#_Toc6110)

[● 功能测试 4](#_Toc6595)

[● 链接测试 4](#_Toc24877)

[● 表单测试 5](#_Toc16338)

[● 数据校验 5](#_Toc28838)

[● Cookies 测试 6](#_Toc30095)

[● 可用性测试 7](#_Toc26519)

[● 导航测试 7](#_Toc31339)

[● 图形测试 7](#_Toc1432)

[● 内容测试 8](#_Toc24005)

[● 整体界面测试 8](#_Toc23059)

[● 9.4功能测试 8](#_Toc16601)

[● 数据库测试 8](#_Toc31459)

[● WEB网站特定的功能需求测试 9](#_Toc15715)

[● 设计语言测试 9](#_Toc12536)

[● 9.5性能测试 9](#_Toc15915)

[● 压力测试 9](#_Toc21292)

[● Web 服务的特点： 9](#_Toc16654)

[● 压力测试的目的 10](#_Toc2217)

[● 压力测试的内容 10](#_Toc30279)

[● 连接速度测试 11](#_Toc14221)

[● 负载测试 11](#_Toc3844)

[● 9.6安全性测试 12](#_Toc18873)

[● 目录测试 12](#_Toc23938)

[● SSL套接字测试 12](#_Toc222)

[● 登录验证 12](#_Toc15186)

[● 日志文件 12](#_Toc7250)

[● 脚本语言 12](#_Toc3694)

[● 9.7兼容性测试 12](#_Toc26163)

[● 平台测试 12](#_Toc19527)

[● 浏览器测试 13](#_Toc10967)

[● 分辨率测试 13](#_Toc4595)

[● 连接速率测试； 13](#_Toc29861)

1. **9WEB网站测试**
   1. **9.1WEB网站概念**
      1. WEB网站的概念
         1. 大多数WEB网站都是采用B-S结构 。
         2. 能够交付一组复杂的内容和功能给大量的终端用户。
         3. WEB网站测试是用于测试高质量Web应用系统的过程，它借用了许多传统软件测试和系统测试的概念和原理。
   2. **9.2WEB网站的特点**
      1. **网络集约性**
         1. 就本质而言，一个Web网站是网络集约的。它可以驻留在网络上，并且服务 于变化多样的客户群的需要。例如时下流行的门户网站或者网络游戏。它们 都可以看成一个完善的大型Web应用系统，服务于各种客户群，但其本身只 需要一个服务器端，用各式各样的客户端满足不同要求的客户
      2. **内容驱动性**
         1. 一般来说，Web网站不是为了某个或某些特定用户量身定做的，它们一般都 拥有一个广大的服务群体，其服务的内容，往往由这些群体的要求所决定。 在大多数情况下，一个Web网站的主要功能是使用HTML（超文本标记语言） javascript等语言来表示文本、图形、音频、视频内容给终端用户
      3. **持续演化性**
         1. 不同于传统的、按一系列规律发布进行演化的应用软件（如微软每隔1-2年发 布新的Office办公软件），Web网站一般是采取持续演化的模式。对于某些 Web应用而言，按小时为单位进行更新都是司空见惯的
      4. **即时性**
         1. Web网站具有其他任何软件类型中都没有的即时性，或者称为快速性。对于 某些较大规模的Web网站，开发时间往往也只有几周或者几天，适度复杂的 Web页面可以仅在几小时内完成。这要求开发者必须十分熟练于开发Web应 用所需的压缩时间进度的规划、分析、实现以及测试方法
      5. **安全性**
         1. Web网站通过网络访问，为了提高系统效率，需要限制访问终端的用户的数 量。为了保护敏感内容，必须提供安全的数据传输模式。因此要求Web网站 必须有一定的安全性保障
      6. **美观性**
         1. 良好的观感会使一个Web网站锦上添花。在某种应用已经被市场广泛接受或 者定义为标准时，美观性可能和技术在同样程度上影响该应用的成功
   3. **9.3网页测试**
      1. **功能测试**
         1. **链接测试**
            1. 什么是链接？

链接是Web 网站的一个主要特征，它是在页面之间切换和引导用户去一些未 知地址页面的主要手段。

* + - * 1. 链接测试的内容：

测试所有链接是否按指示的那样确实链接到了应该链接的页面；

测试所链接的页面是否存在；

保证Web 网站上没有孤立的页面。所谓孤立页面是指没有链接指向该页面， 只有知道正确的URL 地址才能访问。

* + - * 1. 链接测试可以手动进行，也可以自动进行。
        2. 链接测试必须在集成测试阶段完成，也就是说，在整个Web 网站的所有页面开发 完成之后进行链接测试。
      1. **表单测试**
         1. 什么是表单？

表单就是一些需要在线显示和填写的表格。

表单有一些标准操作，如确认、保存、提交等。

* + - * 1. 例1：如果使用表单来进行在线注册，要确保提交按钮能正常工作，当注册完成后应 返回注册成功的消息。
        2. 例2： 如果使用表单收集配送信息，应确保系统能够正确处理这些数据，最后能让顾 客收到数据包。

需要验证服务器能正确保存这些数据；

后台运行的程序能正确解释和使用这些信息。

* + - * 1. 例3：当用户使用表单进行用户注册、登录、信息提交等操作时，必须测试提交操作的完整性。

用户填写的出生日期与职业是否恰当；

填写的所属省份与所在城市是否匹配；

如果使用了默认值，还要检验默认值的正确性；

如果表单某个字段只能接受指定的某些值，则对这个字段也要进行测试。

* + - 1. **数据校验**
         1. 目的：根据业务规则，需要对用户输入进行校验，则要保证这些校验功能正常工作。
         2. 例如：

省份的字段可以用一个有效列表进行校验。

需要验证列表完整性；

程序是否正确调用了该列表；

列表操作是否正确，如在列表中添加一个测试值。

数据校验测试和表单测试可能会有一些重复。

* + - 1. **Cookies 测试**
         1. 什么是Cookies？

Cookie是一个由网页服务器放在您硬盘上的非常小的文本文件. 它本质上就像您的身份证明 一样，并且不能像代码那样被执行或被用来散布病毒。它只能被您使用并且只能由提供的服务 器读取.

* + - * 1. 使用Cookie的目的

帮您节约时间。如果您自定义页面，或注册产品或服务。cookie记住您的身份.当下一次您再次访问的时候，将显示您需要的信息，将帮您填入任何您已经回答过的问题。

* + - * 1. Cookies 通常用来存储用户信息和用户在某些应用系统上的操作序列，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，并把该信息以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录等信息。
        2. 测试内容：

Cookies是否能正常工作；

Cookies是否按预定的时间进行保存；

刷新对Cookies 有什么影响等。

* + - * 1. 举例：

如果在cookies 中保存了注册信息，应确认该cookie 能够正常工作而且已对这些信息进行 加密。

如果使用cookie 来统计次数，需要验证次数累计正确。

* + 1. **可用性测试**
       1. **导航测试**
          1. 在不同的用户接口控制之间，例如按钮、对话框、列表和窗口等；
          2. 或在不同的连接页面之间，
          3. 导航描述了用户在一个页面内操作的方式。
          4. 导航测试的内容

导航是否直观？

Web 系统的主要部分是否可以通过主页访问？

Web系统是否需要站点地图、搜索引擎或其他的导航器帮助？

测试Web 系统的页面结构；

导航条、菜单、连接的风格是否一致？

各种提示是否准确，确保用户凭直觉就知道是否还有内容，内容在什么地方。

最好让最终用户参与导航测试，效果将更加明显。

* + - 1. **图形测试**
         1. 在Web 网站中，适当的图片和动画既能起到广告宣传的作用，又能起到美化页面 的功能。一个Web 网站的图形可以包括图片、动画、边框、颜色、字体、背景、 按钮等。
         2. 图形测试内容

要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免浪费传输时间。图片尺寸要尽量地小，并且要能清楚地说明某件事情。

验证所有页面字体的风格是否一致。

背景颜色应该与字体颜色和前景颜色相搭配。

图片的大小和质量也是一个很重要的因素，一般采用JPG 或GIF 压缩。

* + - 1. **内容测试**
         1. 内容测试的目的：

内容测试用来检验Web 网站提供信息的正确性、准确性和相关性。

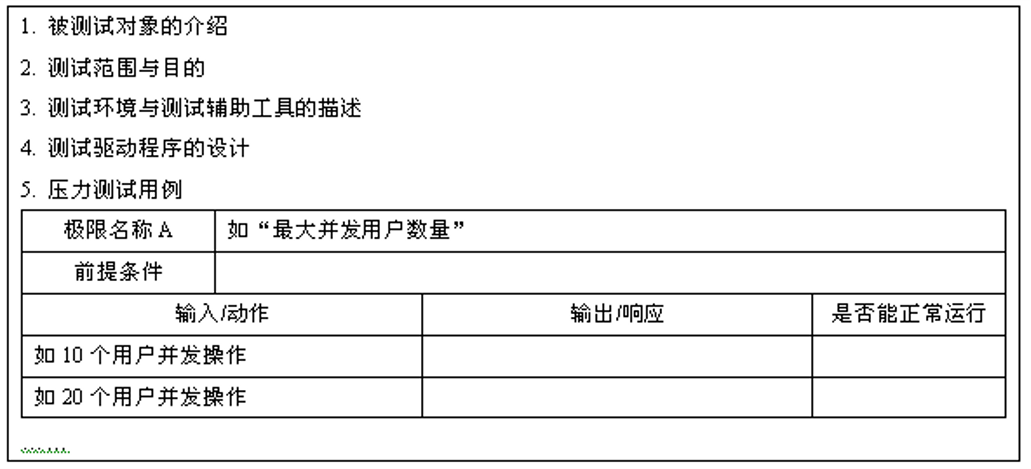
* + - 1. **整体界面测试**
         1. 整体界面测试是对整个Web 系统的页面结构设计的测试，是用户对系统的一个整体感受。
         2. 例如，当用户浏览Web 网站时，应考虑

是否感到舒适？

是否凭直觉就知道要找的信息在什么地方？

整个Web 应用系统的设计风格是否一致？

* 1. **9.4功能测试**
     1. 功能测试包括以下内容：
     2. **数据库测试**
        1. 数据库在Web 网站中的作用
           1. 数据库为Web 网站的管理、运行、查询和实现用户对数据存储的请求等提供空间。
           2. 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。
        2. 两种主要数据库错误
           1. 数据一致性错误：主要是由于用户提交的表单信息不正确而造成的。
           2. 输出错误：主要是由于网络传输速度或程序设计问题等引起的。
        3. 数据库测试就要针对这两种情况，分别进行测试。
     3. **WEB网站特定的功能需求测试**
        1. 测试人员需要对WEB网站特定的功能需求进行验证。
        2. 测试的依据：程序需求规格说明书
     4. **设计语言测试**
        1. 不同的Web 设计语言版本的差异可以引起客户端或服务器端严重的问题；
        2. 尤其在分布式环境中开发时，开发人员都不在一起，这个问题就显得尤为重要。
        3. 测试的语言，除了HTML 的版本问题外，不同的脚本语言，例如使用Java、JavaScript、ActiveX、VBScript或Perl 等开发的应用程序也要在不同的版本上进行验证。
  2. **9.5性能测试**
     1. **压力测试**
        1. **Web 服务的特点：**
           1. Web 服务处于分布式计算的核心位置，它们之间的交互通常很难测试。
           2. 分布式开发可能使 Web 服务的开发变得越来越容易隐藏错误
        2. 压力测试是检测这类代码错误的一种有效方法。
        3. **压力测试的目的**
           1. 压力测试目的是要弄清楚被测试的 Web 服务是不是不仅能做我们认为它能做的事，而且在被施加了某些高强度压力的情况下仍然继续正常运行。
        4. **压力测试的内容**
           1. 压力测试必须对 Web 服务应用以下四个基本条件进行有效的压力测 试。
           2. 重复（Repetition）：测试的重复就是一遍又一遍地执行某个操作或功能。比如重复调用一个 Web 服务，确定一个操作能否正常执行，并且能否继续在每次执行时都正常
           3. 并发（Concurrency）：并发是同时执行多个操作的行为。换句话说，就是在同 一时间执行多个测试，例如在同一个服务器上同时调用许多 Web 服务
           4. 量级（Magnitude）： 压力测试系统应该应用于产品的另一个条件，需要考虑每个操作中的负载量，即也要尽量给产品增加负担。例如，改变数据的大小、改变时间延迟的长度、资金数量的转移、输入速度以及输入的变化等
           5. 随机变化：任何压力系统都多多少少具有一些随机性。随机使用前面的压力原则 中介绍的无数变化形式，就能够在每次测试运行时应用许多不同的代码路径
        5. 压力测试用例参考模板（打印PPT）



* + 1. **连接速度测试**
       1. 用户连接方式的不同：
          1. 电话拨号上网；
          2. 宽带上网；
          3. 局域网；
          4. 有限电视网；
          5. 光纤网；
          6. 电力网。
       2. 不管用户使用那种方式的不同，系统都不能让用户可以等较长的时间。
       3. 连接速度测试的目的，就是要保证在许可的时间内响应用户的请求
       4. 举例：
          1. 如果访问一个页面Web 系统响应时间太长（例如超过5 秒钟），用户就会因失去 耐心而离开。
          2. 有些页面有超时的限制，如果响应速度太慢，用户可能还没来得及浏览内容，就需要重新登录了。
          3. 如果连接速度太慢，还可能引起数据丢失，使用户得不到真实的页面。
    2. **负载测试**
       1. 负载测试的目的：负载测试是为了测量Web 系统在某一负载级别上的性能，以保证Web 系统在需求范围内能正常工作。
       2. 负载测试内容：
          1. 某个时刻同时访问Web 系统的用户数量；
          2. 在线数据处理的数量。
       3. 例如：
          1. 系统最多能允许多少个用户同时在线？
          2. 如果超过了这个数量，会出现什么现象？
          3. 系统能否处理大量用户同时对同一个页面的请求？
  1. **9.6安全性测试**
     1. **目录测试**
     2. **SSL套接字测试**
     3. **登录验证**
     4. **日志文件**
     5. **脚本语言**
  2. **9.7兼容性测试**
     1. **平台测试**
        1. 平台测试就是要测试兼容性问题：
        2. 同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能 会运行失败。
        3. 因此，在Web 系统发布之前，需要在各种操作系统下对Web 系统进行兼容性测试。
     2. **浏览器测试**
        1. 来自不同厂商的浏览器对Java，、JavaScript、ActiveX、plug-ins 或不同的HTML 有不同的支持
        2. 框架和层次结构风格在不同的浏览器中也有不同的显示，甚至根本不能显示。 不同的浏览器对安全性和Java 的设置也不一样
     3. **分辨率测试**
        1. 页面版式在1280x720、1920x1080 或1366x768 的分辨率模式下是 否显示正常?
     4. **连接速率测试；**