

Web 测试中，各类 web 控件测试点总结

一、界面检查

进入一个页面测试，首先是检查 title，页面排版，字段等，而不是马上进入文本框校验

- 1、页面名称 title 是否正确
- 2、当前位置是否可见 您的位置：xxx>xxxx
- 3、文字格式统一性
- 4、排版是否整齐
- 5、列表项显示字段是否齐全,列表项字段名称是否跟表单统一
- 6、同一页面，是否出现 字段名称相同、值取不同的问题。
- 7、数据加载情况：除了文本框的值，还要注意：

复选框，是否保存打√，或者保存不打√

下拉框，是否保存选择的值

多文本框，值是否都被保存，空格，换行是否保存

二、单文本框 (type=text)

边界：字段长度

判空：是否可以为空

唯一性：是否唯一 (小归结：边界、判空、唯一性、特殊字符、正确性)

考虑语言，操作环境

特殊符号测试输入：

' or 1<>'1 ' or '1'='1 ' or '1'<>'2 '|?><

where a='xxx' 下划线是否允许 输入全部空格 输入 单引号

><script>alert("123");</script>>

特殊字段输入限定：

框内容是否合法 (tel, ip, url, email) 序号等，直接限制输入数字，其他过滤掉
输入金额文本框，整数首位为 0，过滤掉，小数点后面，一般保留两个有效数字。

正确性测试：(必不可少的步骤)

1)、(字段长度输入最大允许长度时)数据允许长度的测试：

- a、页面是否被挤出的测试(都输入长英文字符串，是否断行)；
- b、数据库是否允许最大字符(都输入汉字、都输入英文、混合……)；
- c、最短长度的正确流程，最大长度的正确流程覆盖。

2)、对于允许为空的字段，不填入，再次数据传递后，看是否报 500 错误。

3)、未规定字段长度 (或者数值大小)，不按死板输入，输入非常多字符 (或者非常大的数值) 时，做允许动作的正确性校验，看是否报错。(要达到的结果：不管有没有长度限制 (没有给最长、最大限制让你去测？)，最终页面不能抛数据库异常。)monkeytest

说明：通过不断输入长字符串，看是否有长度校验；

最终都会出现以下两种情况的一种：

- A、页面(前台)有校验长度、大小； 或者
- B、无校验，数据库报错。

所以：所有字段都要做长度、大小限制(不管需求有没有给出明确要求，不管测试颗粒度，都要限制长度，不允许报数据库错误，都要测!!!)。最大长度限制可限定方法：1、不允许再输入；2、自动截断处理，并且给用户提示。

关于长度概念：

- 1、数据库规定的字节长度 A
- 2、页面上可以输入的字符数 B

控制方法：

1)、页面上，不管输入什么字符（全角如汉字、半角如字母），统一规定不能超过 B 个字符，此种限制，

测试点：全部输入全角 B 个，测试（B*3 字节）会不会超过数据库字节长度

全部输入半角 B 个，测试（B*1 字节）会不会超过数据库字节长度

混合输入全角 X 半角 Y，测试（X*3+Y 字节）会不会超过数据库长度

2)、页面上，不以字符统计，以总的输入字节数统计，比如，全部输入全角字符，允许可以输入 A/3 个字符，全部输入半角字符，允许输入 A 个字符（民生网的设计）

测试点：全部输入全角，看是否允许输入 A/3 个字符

全部输入半角，看是否允许输入 A 个字符

混合输入全角 X，半角 Y，看是否允许 X*3+Y=A

(5 个：判空、唯一、边界值、特殊字符、正确流程(多种数据、多种分支))

+测试校验位置：ajax 鼠标事件校验、前台提交按钮 js 校验，服务器拿到数据后再次验证

证

三、多文本框（type=textarea）

1)、空格和换行的问题，看需求，是否需要做支持 HTML Encoding

输入全部空格时，是否判空处理？""空格，。

输入折行，是否也显示折行？

比如：列点说明原因，就需要支持。

2)、字母截断的问题

对于一串字母，开发人员往往会忘掉做截断，这样如果展示在我们的平台上的话，这一串字母就会把我们的 UI 撑开

3)、长度控制格式，您还可以输入***个字符

四、添加按钮

添加动作检查范围：

失败：是否提示

提示内容是否正确

失败时：保存用户已输入的内容，避免重新再输入

成功：对话框消失

记录是否可直接查看（还需要刷新？）

列表记录顺序

重复提交情况，点击一次后，是否变成 disable

上传附件的添加：

A. 文件名称：文件名称很长；文件名称字符多样化(汉字，英文，符号)；文件名称重复。

B. 判空？

C. 附件格式类型支持？

D. 附件个数？

E. 附件空间大小。

五、移除按钮

1.一般都要在前台先给出一个提示操作“确定移除该……”

2.相关联的东西，是否需要限制移除“该类型下存在应用，无法移除”有到后台比较

3.确定后，真正执行移除操作。

结果：

移除后，列表数据是否立即消失。

必须有确认删除的提示信息

六、列表

- 1)、列表记录顺序
- 2)、是否需要翻页、有没有翻页功能
- 3)、字段名称是否与表单一致

七、搜索-文本框

1、功能点、需求点考虑：

是否提供模糊查询、输入数值有种类有限定时，是否考虑换成下拉框搜索；

2、检查点：

文本框值是否消失(是否回填条件值)，再次点击“查询”可查看所有记录；

考虑搜索结果：是否存在分页，分页是否正常；是否有序；

注意：分页是否仍保存查询条件，检查后面的记录是否符合条件

3、查询数据多样性：

输入不存在的字段值测试、包括特殊字符查询测试例如：' or '1'='1；

输入类似程序语句的条件时是否执行查询，如：XXXX”、XXX and ；

4、操作类型：

- 1) 不输入的查询
- 2) 输入全部空格的查询
- 3) 模糊查询(输入部分字段，或者说，输入英文字母，查询到相关中文数据)
- 4) 输入不存在的查询
- 5) 输入存在的查询
- 6) 单个查询和多个条件复合查询。

八、搜索-下拉框

检查点：

- a) 搜索结果是否有序；
- b) 下拉框值是否齐全；(下拉框值本身也是一个动态查询的结果)
- c) 下拉框值是否自动消失，再次点击“查询”可查看所有记录(是否要回填条件值)；
- d) 分页时，是否保存搜索条件。

(从 UI、开发、业务逻辑、用户使用等角度测试)

PS:

以上总结的，是比较纯粹的从页面控件角度测试点出发，对于完整测试一个整体页面，需要各类测试有机结合起来：

1) UI 测试：

页面布局； 页面样式检查；控件长度是否够长；显示时，是否会被截断；支持的快捷键，Tab 键切换焦点顺序正确性等。

2) 功能测试：页面上各类控件的测试范围，测试点，可参考上方

结合控件的实际作用来补充检查点：比如，密码框是否*显示，输入是否做 trim 处理等

3) 安全测试：输入特殊字符，sql 注入，脚本注入测试

后台验证测试，对于较重要的表单，绕过 js 检验后台是否验证

数据传输是否加密处理，比如，直接请求转发，地址栏直接显示发送字符串？

数据库存储，特别密码等，是否加密形式存储

- 4) 兼容性测试
- 5) 性能测试

二.常见功能点测试思路

根据经验，总结常见的功能点的测试思路：

- 1. 新增 或 创建 (Add or Create)
 - .1 操作后的页面指向
 - .2 操作后所有绑定此数据源的控件数据更新，常见的排列顺序为栈 Stack 类型，后进先出
 - .3 取消操作是否成功
- 2.编辑 或 更新 (Edit or Update)
 - .1 操作后的页面指向
 - .2 操作后所有绑定此数据源的控件数据更新
 - .3 取消操作是否成功
 - .4 编辑界面是否读取正确、全部的数据源
 - .5 记录在工作流中的编辑功能可用性
 - .6 操作成功的生效时刻及生效范围
- 3.删除 或 移除 (Delete or Remove)
 - .1 操作后的页面指向
 - .2 操作后所有绑定此数据源的控件数据更新 (如下就是删除后，Tab 数据没有立即刷新的 bug)
- 3 取消操作是否成功
 - .4 记录在工作流中的编辑功能可用性
 - .5 操作成功的生效时刻及生效范围 (比如：购物网站，店家商品下架后，并没有同时删除买家的购买记录)
- 4.选中 或 全选 (Check or Check all)
 - .1 多页面中，全选对所有页面是否有效
 - .2 支持多页面的个别选中，且返回查看时保留选中状态
 - .3 界面上的按钮的操作范围是否均受选中功能控制
 - .4 前一页选中状态，在翻页后，应保留原来状态
 - .5 先全选-》移除某个单选-》全选按钮是否移除选中状态

谈谈性能测试分类

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

验收性能测试（狭义） 性能测试方法是通过模拟生产运行的业务压力量和使用场景组合，测试系统的性能是否满足生产性能要求。通俗地说，这种方法就是要在特定的运行

条件下验证系统的能力状态。

特点：1、这种方法的主要目的是验证系统是否有系统宣称具有的能力。2、这种方法要事先了解被测试系统经典场景，并具有确定的性能目标。3、这种方法要求在已经确定的环境下运行。也就是说，这种方法是对系统性能已经有了解的前提，并对需求有明确的目标，并在已经确定的环境下进行的。

负载测试(Load Test)通过在被测系统上不断加压，直到性能指标达到极限（例如“响应时间”）超过预定指标或都某种资源已经达到饱和状态。

特点：1、这种性能测试方法的主要目的是找到系统处理能力的极限。2、这种性能测试方法需要在给定的测试环境下进行，通常也需要考虑被测试系统的业务压力量和典型场景、使得测试结果具有业务上的意义。3、这种性能测试方法一般用来了解系统的性能容量，或是配合性能调优来使用。也就是说，这种方法是对一个系统持续不段的加压，看你在什么时候已经超出“我的要求”或系统崩溃。

压力测试(强度测试)(Stress Test)压力测试方法测试系统在一定饱和状态下，例如 cpu、内存存在饱和使用情况下，系统能够处理的会话能力，以及系统是否会出现错误

特点：1、这种性能测试方法的主要目的是检查系统处于压力性能下时应用的表现。2、这种性能测试一般通过模拟负载等方法，使得系统的资源使用达到较高的水平。3、这种性能测试方法一般用于测试系统的稳定性。也就是说，这种测试是让系统处在很大强度的压力之下，看系统是否稳定，哪里会出问题。

并发测试（Concurrency Testing）并发测试方法通过模拟用户并发访问，测试多用户并发访问同一个应用、同一个模块或者数据记录时是否存在死锁或其他性能问题。

特点：1、这种性能测试方法的主要目的是发现系统中可能隐藏的并发访问时的问题。2、这种性能测试方法主要关注系统可能存在的并发问题，例如系统中的内存泄漏、线程锁和资源争用方面的问题。3、这种性能测试方法可以在开发的各个阶段使用需要相关的测试工具的配合和支持。也就是说，这种测试关注点是多个用户同时（并发）对一个模块或操作进行加压。

配置测试（Configuration Testing）配置测试方法通过对被测系统的软\硬件环境的调整，了解各种不同对系统的性能影响的程度，从而找到系统各项资源的最优分配原则。

特点：1、这种性能测试方法的主要目的是了解各种不同因素对系统性能影响的程度，从而判断出最值得进行的调优操作。2、这种性能测试方法一般在对系统性能状况有初步了解后进行。3、这种性能测试方法一般用于性能调优和规划能力。也就是说，这种测试关注点是“微调”，通过对软硬件的不段调整，找出这他们的最佳状态，使系统达到一个最强的状态。

可靠性测试通过给系统加载一定业务压力（例如资源在 70%-90%的使用率），使系统运行一段时间，以此检测系统是否稳定运行。

特点：1、这种性能测试方法的主要目的是验证是否支持长期稳定的运行。2、这种性能测试方法需要在压力下持续一段时间的运行。（2~3 天）3、测试过程中需要关注系统的运行状况。如果测试过程中发现，随着时间的推移，响应时间有明显的变化，或是系统资源使用率有明显波动，都可能是系统不稳定的征兆。也就是说，这种测试的关注点是“稳定”，不需要给系统太大的压力，只要系统能够长期处于一个稳定的状态。

失效恢复测试如果系统局部发生故障，用户是否能够继续使用系统，以及如果这种情况发生，用户将受到多大程度的影响。

特点：1.这种性能测试方法的主要目的是验证在局部故障情况下，系统能否继续使用。

2.这种性能测试方法还需要指出，当问题发生时，“能支持多少用户访问”的结论和“采取何种应急措施”的方案。 3.一般来说，只有对系统持续运行指标有明确要求的系统才需要进行这种类型的测试。

大数据量测试针对某些系统存储、传输、统计查询等业务进行大数据量的测试。

疲劳强度测试主要特点是长时间对目标测试系统加压，目的是测试系统的稳定性，持续时间一般在 1 小时以上；感觉等同于可靠性测试。

注意：在做性能测试时请忘掉分类。例如，运行 8 个小时来测试系统是否可靠，而这个测试极有可能包含了可靠性能测、强度测试、并发测试、负载测试，等等。因此，在实施性能测试时决不能割裂它们的内部联系去进行，而应该分析它们之间的关系，以一种高效率的方式来设计性能测试。

Web 测试中的几个 case

一、页面上对引起 大量数据提交的 按钮/链接 点击一次后， disable

需求：

对于重要的表单、数量庞大/响应慢的系统，在做提交时，又有页面还在 loading 状态，此时连续做两次点击，经常引起各种报错，这种情况下，需要提出 对 按钮/链接 点击一次后，做 disable

测试：

1)、查看页面源代码是否有脚本控制，例如：

```
<a href="javascript: $('#next').val('true'); buttonDisable();headerFormSubmit();" type="submit" class="btn" id="nextButton"> Next </a>
function buttonDisable(){
$('#nextButton').attr("disabled", "disabled");
}
```

2)、对脚本进行调试，

可以借助 firebug 工具，在 Script Tab 上，在\$('#nextButton').attr("disabled", "disabled"); 这行脚本设置 disable，点击 nextButton，检查运行到断点处停止，按钮无法再次点击。运行断点后，disable 解除。

二、新增数据库字段测试需要考虑的几个点

1)、从数据库检查起，检查相关表：原表、历史表、与其同步库的表 有没有都添上该字段，并且注意在每个表中，字段类型是否统一

2)、校验：考虑字段本身类型，判空、边界、唯一性、特殊字符、正确性允许的 data 特别，在做判空时，若字段不允许为空时，考虑：需要提交脚本初始化历史数据 set default value

3)、流程覆盖：考虑该字段覆盖到哪几个相关页面，测试到整个流程，每个页面校验要一致；

三、查 log 测试的几个操作

一般情况下，项目都部署在 linux 环境上，测试时，有些需要查 log，或者有些服务需要自己去重启，此时就需要一些基本的 linux 操作命令：

1)、首先连接到 linux 系统的机器上，可以使用 putty 软件，要有 服务器地址+端口+协议 loginName+password，就可以登录

2)、cd 到脚本或者 log 放置的文件夹位置去重启服务或查看 log，还有一些常用的命令 less 文件名(W 向上翻页、F 向下翻页，Shift+F 自动翻页，Ctrl+C 停止自动翻页)；

grep "findString" 文件名；

执行脚本： ./脚本名 或者 sh./脚本名

web 常见安全问题以及测试方法

Web 安全是我们测试组一直以来作为和性能测试并驾齐驱的两个重点。开发的过程中还需要着重注意,该转义的地方转义 ;该屏蔽的地方屏蔽,该过滤的地方过滤等等。年底又到了,势必又有大批的发号抽奖之类的活动开发、上线,在这个过程中,安全问题是每个人应该紧绷的神经,对于我们测试人员来说,每个活动需要做到手动安全测试加自动化安全测试相结合。

常见的 web 安全问题有：

SQL 注入、跨站点脚本攻击、跨站点伪造请求、目录遍历、邮件表头注入、页面错误信息等。

对于手动安全测试来说,一般常用的有三点：

- 1、URL 有参数的,手动修改参数,看是否得到其他用户的信息和相关页面；
- 2、在登录输入框的地方输入 ' or 1=1-- 或 " or 1=1-- 等看是否有 SQL 注入；
- 3、在注重 SQL 注入的同时,一般在有输入框的地方输入

对于自动化安全测试来说：

测试组目前使用的安全测试工具为 IBM 的 AppScan (当然,是破解版,34 上已经放过该工具的安装包)

- 1、在使用之前务必确认自己绑定的 Host；
- 2、配置 URL、开发环境、错误显示类型；
- 3、结果保存后可根据提示的问题类型和解决建议进行分析。

Web 安全测试通常要考虑的测试点：

- 1、输入的数据没有进行有效的控制和验证
- 2、用户名和密码
- 3、直接输入需要权限的网页地址可以访问
- 4、认证和会话数据作为 GET 的一部分来发送
- 5、隐藏域与 CGI 参数
- 6、上传文件没有限制
- 7、把数据验证寄希望于客户端的验证
- 8、跨站脚本 (XSS)
- 9、注入式漏洞 (SQL 注入)
- 10、不恰当的异常处理
- 11、不安全的存储
- 12、不安全的配置管理
- 13、传输中的密码没有加密
- 14、弱密码,默认密码
- 15、缓冲区溢出
- 16、拒绝服务

SQL 注入：

所谓 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，比如先前的很多影视网站泄露 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。（

（select * from 表 where id=1 or 1

1 or 1 是输入框输入的

这样会导致满足 id=1 或 1 的数据都查出来

而所有的数据都满足 1

这样就查出来了很多不该被查出来的数据

这就是 sql 注入)