

# 针对 B/S、C/S 架构的 180 个简单测试案例

作者: tony, Saber

这是一个针对 web 应用和桌面应用程序的测试清单。

注释: 这篇文章有点长, 我的目标是能够分享其中一个已经启动但还没完成的综合测试清单。我将利用很多场景持续更新该清单。如果你现在没有时间阅读, 请随意的将它与你的朋友共享并放在书签中供以后阅读。

测试清单是测试用例编写过程中不可或缺的一部分。使用该清单你可以轻松地创建数以百计的测试用例来测试 web 或桌面应用程序。这些都是常用的测试用例, 基本上适用于所有的应用程序。在为你的项目编写测试用例时参考该清单, 我相信除了软件需求说明文档中的特定业务规则外, 你能覆盖到大多数测试类型。

尽管这是一个普通的清单, 但我还是建议根据你的特定需要和特定的测试利用下面的测试用例准备一个标准的测试清单。

测试过程中应用测试清单的重要性:

- 为你的应用程序维持一个标准的测试用例库将能保证更快地捕捉最常见的缺陷。
- 重用测试用例有助于节约编写重复用例花费的资源成本。
- 经常覆盖的重要测试用例不可能轻易忘记。
- 测试清单可以提供给开发人员查阅, 以保证在开发阶段就避免出现一些常见的问题。

几点说明:

- 1) 用不同的用户角色执行这些测试场景, 如: 管理用户, 来宾用户等。
- 2) 对于 web 应用, 这些场景应该在客户认可的多种浏览器的各个版本上进行测试, 如: IE, Firefox, Chrome, Safari 等。

- 3) 用不同的屏幕分辨率进行测试, 如: 1024x768, 1280x1024 等。
- 4) 应用程序应在多种显示器上进行测试, 如: LCD, CRT, 笔记本, 平板电脑, 智能手机等。
- 5) 在不同的系统平台上测试应用程序, 如: Windows, Mac, Linux 等系统。

针对 web 和桌面应用程序的综合测试清单

假设: 假定你的应用程序支持下列功能:

- 带有多种字段的表单
- 子窗口
- 与数据库交互
- 多种查询过滤规则和结果显示
- 图片上传
- 邮件发送

- 数据导出

一般测试场景

1. 所有必填字段都应校验并用星号“\*”标注
2. 验证错误提示信息应在正确的位置合理显示
3. 所有的错误信息都应用相同的 CSS 样式显示（如：红色）
4. 一般性的确认信息应该用错误消息意外的 CSS 样式显示（如：绿色）
5. 提示信息应是有意义的
6. 下拉字段的第一个条目应是空白或“请选择”之类的文本
7. 删除页面中的任何记录信息都应要求确认
8. 如果页面支持记录的添加/删除/更新功能，那么页面中应提供“全选”和“全不选”所有记录的选择项
9. 数量值应该显示正确的货币符号
10. 应提供默认页面排序
11. 重置按钮功能应将页面所有字段设置为默认值
12. 所有的数值都应正确地格式化
13. 输入字段应检查最大字段值，输入的字段值超过指定的最大值则不被接受或不被存储到数据库
14. 检查所有输入字段中输入特殊字符的情况
15. 使用标准的字段标签，如：一个接受用户姓名的字段标签可以被定义为“姓名”
16. 检查添加/编辑/删除操作后页面中信息记录的排序功能
17. 检查超时功能，超时的值应是可配置的，操作超时后检查应用程序的行为是否合理
18. 检查 Cookies 在应用程序中的使用
19. 检查可下载文件是否指向了正确的文件路径
20. 所有的资源键应该可以在配置文件或数据库中配置，而不是写死
21. 资源键的命名应始终遵循标准惯例
22. 验证所有的 web 页面标记（验证 HTML 和 CSS 的语法错误）以确保它符合标准
23. 应用程序崩溃或不可用页面应该重定向到错误页面
24. 在所有页面中检查文本的拼写和语法错误
25. 检查数字输入字段中输入字符的情况，应提示合适的校验信息
26. 如果字段允许输入数值，应该检查输入负数的情况
27. 检查数量字段值带有小数的情况
28. 检查页面中所有按钮的功能
29. 用户连续点击提交按钮时不能重复提交页面信息
30. 在任何计算中都应处理除以 0 的情况

31. 应正确处理输入数据前后的空格

### GUI 和可用性测试场景

1. 页面中的所有字段（如：文本框，单选选项，下拉列表）应该适当对齐

2. 除特殊指定外，数值一律靠右对齐

3. 在字段标签、列、行和错误提示等信息之间余留足够的空间

4. 只在必要时启用滚动条

5. 标题的字体大小、样式和颜色，描述文本，标签，字段内置数据和表格信息都应以软件需求说明中指定的为标准

6. 描述文本框应是多行文本框
7. 禁用字段应该灰色标记，用户不能对这些字段设置键盘关注
8. 鼠标点击任何输入文本的字段后，鼠标箭头应变为光标
9. 用户不能在下拉选择列表中输入信息
10. 当提交的页面中存在错误时，用户填写的信息应保持不变，用户更正错误信息后可以再次提交
11. 检查错误信息中提及的字段标签是否正确
12. 下拉字段值应以定义的顺序排列
13. Tab 键和 Shift + Tab 组合键功能正常
14. 默认单选选项在页面加载时是预先选中的
15. 特殊字段和页面级别的帮助信息应是可用的
16. 检查出现错误时是否正确高亮标记对应字段
17. 检查下拉列表中的选项是否易读，且不会因为字段长度截断显示
18. 页面中的所有按钮都能通过快捷键操作，用户可以通过键盘完成所有操作
19. 检查所有图片无法显示的页面
20. 检查所有链接失效页面
21. 所有页面都应有标题
22. 在执行任何更新或删除操作之前都应显示确认信息
23. 当应用程序忙时应该显示沙漏计时器
24. 页面文本应采用左对齐
25. 用户应能选择一个单选选项或多选的任何组合

### 过滤条件测试场景

1. 用户应能够使用页面中的所有参数过滤结果
2. 精确搜索功能应根据用户选择的所有搜索参数加载搜索页面

3. 当页面中至少需要一个过滤条件才能执行搜索操作时,必须保证用户没有设置任何过滤条件提交查询时能显示合适的错误提示信息

4. 当页面中至少有一个过滤条件是非强制的时,用户提交查询后那些非强制过滤条件使用默认搜索条件查询相关结果

5. 过滤条件设置为无效值时应显示合适的校验信息

### **结果表测试场景**

1. 当结果页面加载时长超过默认时长时,应该显示“页面加载中”之类的提示信息

2. 检查结果表中获取的数据是否满足所有的搜索条件

3. 结果总数都应在结果表中显示

4. 使用的搜索条件应该在结果表中显示

5. 结果表中的值应该按照默认列排序

6. 排序列应该显示排序的图标

7. 结果表中的结果正确且包含所有指定的列

8. 对支持排序的列,应能进行升序和降序排序操作

9. 结果表中的行列间距合理

10. 当结果多于每页默认显示的结果数时应正确分页

11. 检查上一页、下一页、首页和末页分页功能

12. 结果表中无重复的记录

13. 检查所有的列是否都可见,必要时启用水平滚动条

14. 检查数据动态列(列值由其他列计算得来的列)

15. 对于报表结果表,应检查行汇总和列汇总的值

16. 对于报表结果表,应检查有分页的情况下用户切换分页时的行汇总值

17. 检查显示列是否使用了正确的符号,如:%(百分号)应该显示在百分数计算结果中

18. 检查结果表中的数据是否启用了日期范围

### **窗口测试用例**

1. 检查默认窗口的大小是否正确

2. 检查子窗口的大小是否正确

3. 检查默认焦点是否放在了页面中的某个字段上(一般来说,焦点放在页面中第一个可输入的字段上)

4. 检查关闭父窗口或开着的窗口时是否会关闭子窗口

5. 当子窗口开着时,用户不能使用或更新父窗口或子窗口后面窗口的字段值

6. 检查窗口最小化、最大化和关闭功能

7. 检查窗口是否能重设大小

8. 检查父窗口和子窗口的滚动条的功能
9. 检查子窗口中的“取消”按钮的功能

### 软件测试 180 个综合案例 3

#### 数据库测试场景

1. 页面提交成功时检查数据是否正确地保存在数据库中
2. 检查不接受空值的列值
3. 数据应根据表设计被存储在单个或多个表中
4. 索引名称应按照标准如 IND\_ <表名> \_ <列名>
5. 表应该有主键
6. 应对表中的列给出相应的描述信息（除了诸如创建时间、创建人等审计列）
7. 应该为每个数据库的添加/更新操作添加日志
8. 应该为需要的表创建索引
9. 检查是否只有操作完全成功后才将数据提交到数据库中
10. 一旦事务失败数据应该回滚
11. 数据库名称应按照应用程序类型命名，即测试,UAT,沙箱,现场(尽管这不是一个标准，但对数据库维护是很有帮助的)
12. 数据库逻辑名称应根据数据库名称命名(这不是标准但又有利于数据库维护)
13. 存储过程不应该以前缀“sp\_”命名
14. 检查表审计列的值(如创建日期、创建人、更新日期、更新者、已删除、删除日期、删除者等等)填充正确
15. 检查输入数据保存时是否未被截断，在页面中显示的字段长度和数据库的字段长度应该是相同的
16. 检查包含最小、最大和浮点的数值字段
17. 检查数值字段含有负值(接受和拒绝两种情况)
18. 检查单选按钮和下拉列表正确地保存在数据库中
19. 检查数据库字段设计的数据类型和数据长度是否正确
20. 检查所有的表约束条件如主键、外键等是否正确实现
21. 测试存储过程和触发器的样本输入数据
22. 输入数据的首尾空格应在数据保存到数据库之前被自动隐去
23. 主键列不允许为 NULL 值

#### 图像上传功能的测试场景

(也适用于其他文件上传)

1. 检查图片上传路径

- 2.检查图像上传和修改功能
- 3.检查各种扩展图像文件的上传(例如 JPEG、PNG、BMP 等).
- 4.检查文件名中含有空格或其他可用特殊字符的图片的上传
- 5.检查重复名称图片上传
- 6.图片尺寸大于最大允许值，上传时应该显示适当的错误消息.
- 7.检查上传的图片文件类型外的其它文件时(例如 txt、doc、pdf、exe 等等)，应该显示适当的错误消息
- 8.检查如果上传的图片满足指定的高度和宽度(如果有定义的话)则可以成功上传，否则不能上传
- 9.上传大尺寸图片时应显示上传进度条
- 10.检查上传过程中的取消按钮是否有效
- 11.检查文件选择对话框中的文件列表是否只显示支持文件类型
- 12.检查上传多个图像的功能
- 13.上传后检查图像质量，图像质量不应该改变
- 14.检查用户是否能够使用/查看上传的图像

#### **发送电子邮件的测试场景**

(测试用例不包含撰写或验证电子邮件)

(在执行邮件相关测试之前务必使用假电子邮件地址)

- 1.所有电子邮件模板应该使用 CSS 标准
- 2.要验证电子邮件地址后再发送电子邮件
- 3.特殊字符在邮件正文模板应妥善处理
- 4.特定语言的字符(例如：俄文、中文或德文字符)应在电子邮件主体模板中妥善处理
- 5.电子邮件主题不能空
- 6.占位符字段中使用电子邮件模板应该替换为实际的值如{姓} {名}应该替换为所有收件人正确的名字和姓氏
- 7.如果报告有动态值包含在电子邮件的正文中,报告数据应正确计算
- 8.电子邮件发送者的名字不能为空
- 9.应该在不同的电子邮件客户端（如： Outlook,Gmail,Hotmail,Yahoo 邮件等）检查电子邮件
- 10.检查发送电子邮件功能使用 TO、CC 和 BCC 字段
- 11.检查纯文本邮件
- 12.检查 HTML 格式的电子邮件
- 13.查看邮件页眉和页脚相应的公司 LOGO,隐私政策和其他链接
- 14.检查带附件的电子邮件发送

15.检查给一个、多个或者联系人组发送电子邮件

16.检查回复电子邮件地址是否正确

17.检查发送大量的电子邮件

#### **Excel 导出功能测试场景**

1.文件输出时应该有适当的文件扩展名

2.导出 Excel 文件的文件名应该按照标准，例如：如果文件名使用时间命名，它应该在导出文件的时候妥善换成实际时间

3.当 Excel 文件包含日期列时需要检查导出的日期格式

4.检查数字格式的数值或货币值，格式应该和页面显示的相同

5.导出的文件应该有适当的列名称

6.默认页面排序应体现在导出文件中

7.Excel 文件数据应正确格式化包括页眉和页脚文本、日期、页码等所有页面的值

8.检查数据在页面上显示的文件与导出 Excel 文件是是否一样

9.检查使用分页时的导出功能

10.检查导出按钮图标是否根据导出的文件类型正确显示，如：导出的是.xls 文件，则显示 Excel 文件对应的图标

11.检查大文件的导出功能

12.检查页面包含特殊字符的导出功能，检查这些特殊字符是否正确地导出到 Excel 文件

#### **软件测试 180 个综合案例 4**

##### **性能测试的测试场景**

1.检查页面加载时间是否在可接受范围内

2.检查页面加载缓慢的链接

3.检查在轻度、正常、中度和重度负载环境下所有操作的响应时间

4.检查数据库存储过程和触发器的性能

5.检查数据库查询执行时间

6.检查应用程序的负载测试

7.检查应用程序的压力测试

8.在峰值负载条件下检查 CPU 和内存的使用情况

##### **安全性测试测试场景**

1.检查 SQL 注入攻击

2.安全页面应该使用 HTTPS 协议

3.崩溃页面中不应泄漏应用程序或服务器信息，只有错误页面才显示这些

- 4.转义特殊字符的输入
- 5.错误消息不应该透露任何敏感信息
- 6.所有凭证都应该通过一个加密传输通道
- 7.测试密码安全性和密码强制策略
- 8.检查应用程序的注销功能
- 9.检查暴力攻击
- 10.Cookie 信息只能以加密的格式存储
- 11.检查会话 cookie 持续时间和会话超时或注销后登录会话终止情况
- 12.会话标记应该通过安全通道传送
- 13.密码不应该存储在 cookie 中
- 14.对阻断服务攻击进行测试
- 15.检测内存泄漏
- 16.通过在浏览器地址栏中手动更改变量值访问未经授权的应用程序
- 17.验证对文件扩展名的处理方式以使得.exe 文件不能上传到服务器或在服务器上执行
- 18.如密码和信用卡信息等敏感领域不应该启用自动完成
- 19.对文件上传功能应使用文件类型限制和反病毒扫描上传的文件
- 20.检查目录是否可用
- 21.在输入密码和其他敏感字段时应该被伪装起来
- 22.检查忘记密码是否采用了密码保护功能，如：临时密码在指定的时间段后过期，更改密码或获取新密码有安全问题提问等
- 23.检查验证码功能
- 24.检查重要事件是否被记录在日志文件中
- 25.检查是否正确实现访问权限

渗透测试的测试用例——在本页我已经为渗透测试列出 41 个测试用例

我真诚的感谢 Devanshu Lavaniya(Sr.在 I-link Infosoft 工作的质量工程师)帮助我准备全面的测试清单.

我试图涵盖所有标准的 web 和桌面应用程序功能的测试场景，但我知道这不是一个完整的清单。测试人员在不同项目会根据他们的经验准备自己的测试清单。

请在下面的备注中添加更多的测试场景或负面测试用例，使之成为一个相对完整的清单.

我会很感激你会和你的朋友分享这个.