**流程图规范说明文档**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **概要** | **编写人** |
| V1.0 | 2017-3-20 | 建立文档 | 蔡美忠 |
| V1.1 | 2017-4-6 | 分享后文档整理 | 蔡美忠 |

# 一、基本规范

文档编写和流程图绘制的原则是：易懂、易读、易维护、美观。

1. 文档必须使用2345.com的文档模板
2. 文档统一使用“微软雅黑”
3. 设计文档必须要有“标题”、“修订记录”、“正文”，调研文档要有“标题”、“正文”
4. 正文部分，章节结构要有层次，标题字体尺寸依层减小
5. 标题、正文中的章节标题为粗体
6. 正文中重点内容用“粗体”、“红色字体”标出
7. 每次修改和提交要有修订记录

# 二、设计文档规范

为了达到易懂、易读、易维护的目的，设计文档应该至少包括：标题、修订记录、正文四个部分。其中，正文部分的描述应该遵守自上而下、先整体再局部原则。对设计文档的细化程度，终极要求是：别人要能按照设计文档进行编码开发。

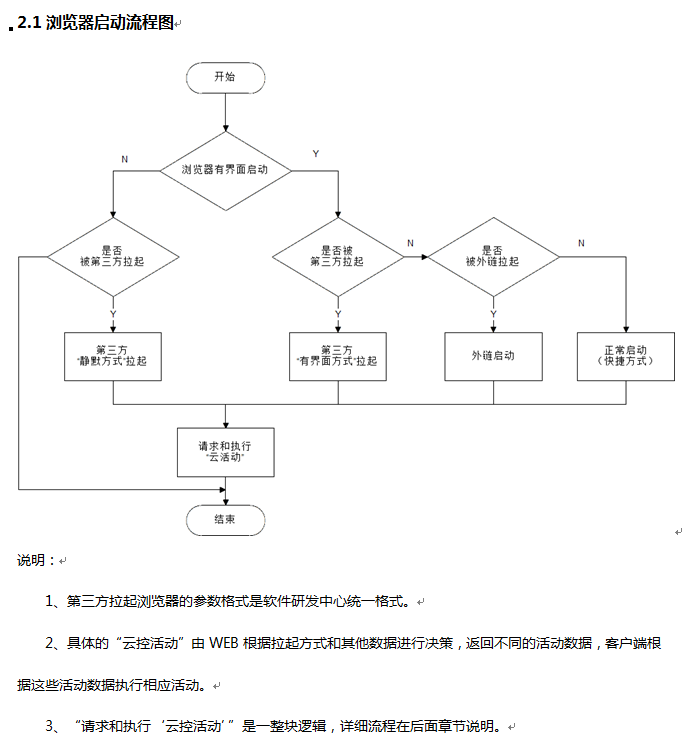
设计文档基本按照自上而下的原则写，从框架设计到业务模块设计，再到接口和数据结构设计，一层层细化。

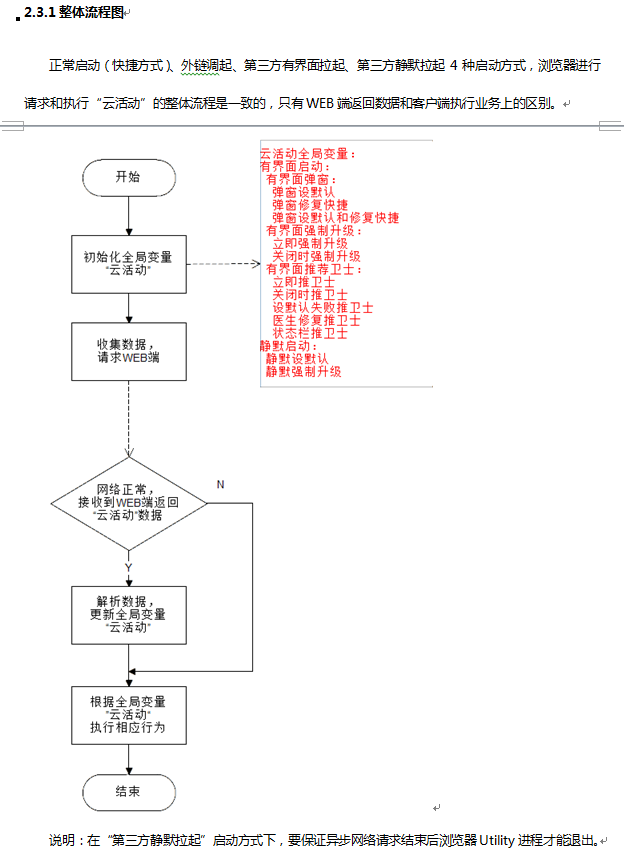
* 框架设计：方案整体设计、整体流程图、时序图、类图
* 业务模块设计：业务模块设计、流程图
* 数据结构设计：内存数据设计、配置数据设计、名单设计、通讯协议

按照业务复杂度，可以减少或者细化上述内容，只要能说清楚设计方案和技术要点，能让别人能看明白即可。

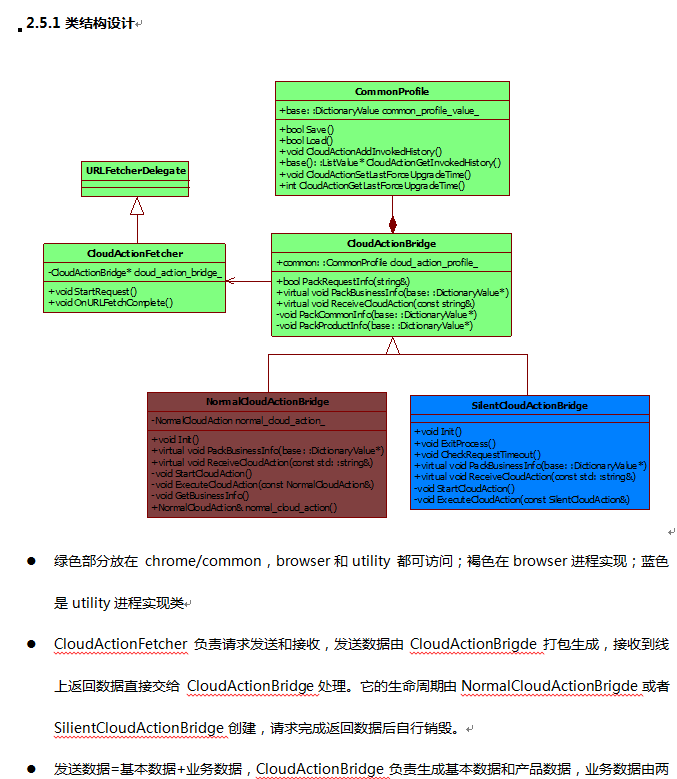
1. 首先要对所开发的功能做一个整体概要的说明，这部分可以包含：

（1）功能框架设计流程图，并附带必要的解释说明，例如：

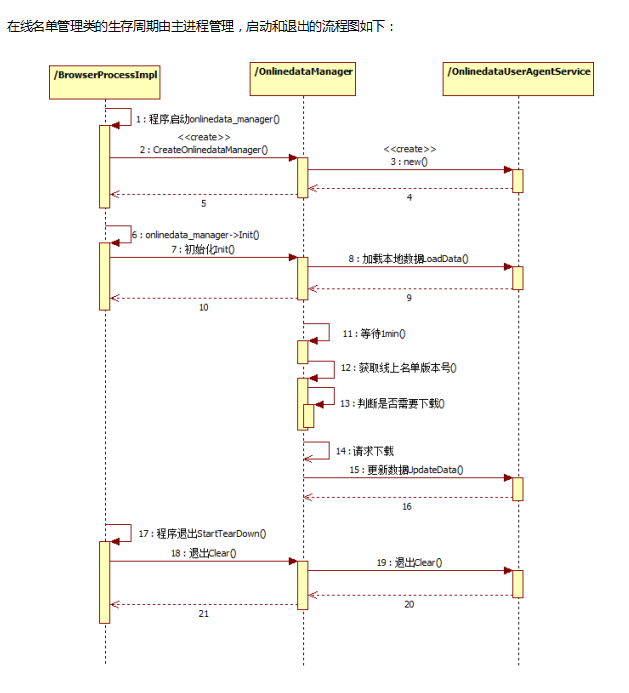




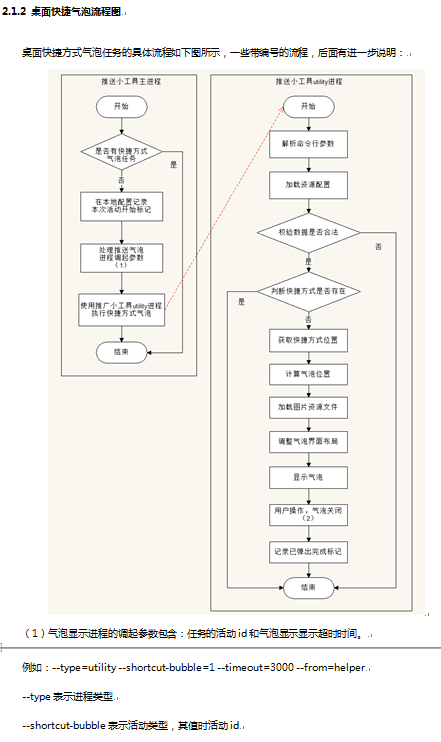
（2）主要功能类图，说明实现功能的主要类，及其之间的关系。



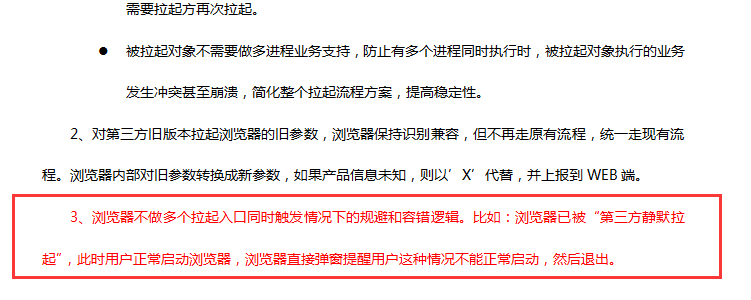
1. 如果有必要，还可以添加整体功能时序图。



1. 如果设计的功能比较复杂，需要分为多个业务模块，则需要分层次说明业务。例如桌面气泡推送业务，是推送工具的一个业务，整体流程图过于复杂，不便整体展示，则可以将这一个业务单独进行流程说明，如下图所示：

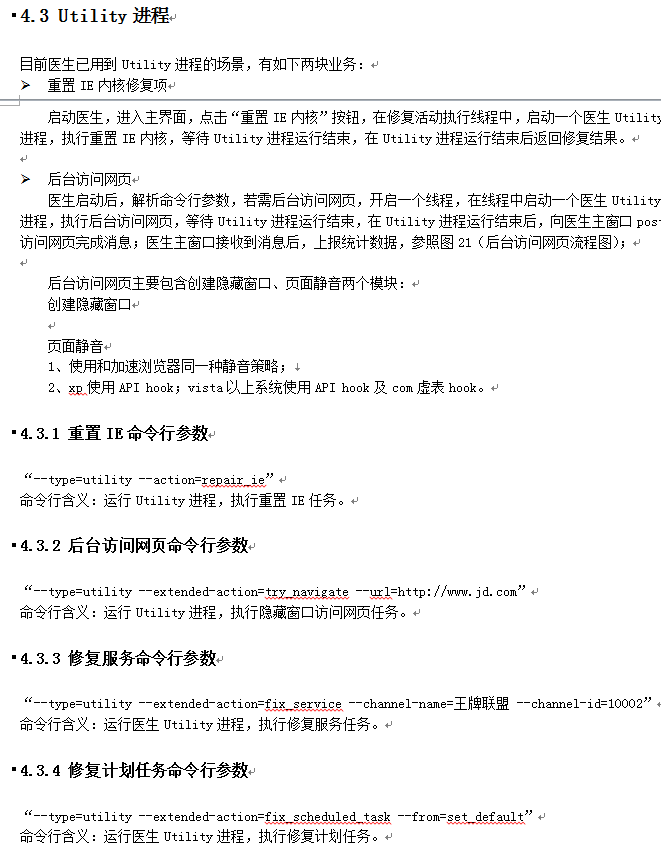


1. 流程图对各业务模块无法清除表达时，可以增加核心逻辑说明，以及相关注意事项，比如，标出设计方案中不支持的应用场景。例如：

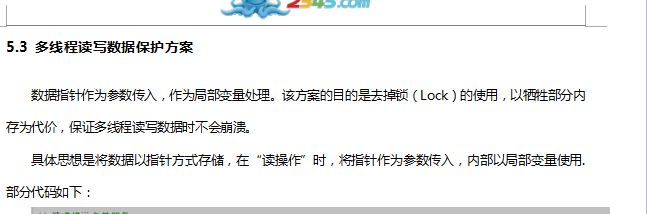


1. 如果业务有复杂的设计结构，比如有很多模块、业务有多进程、多线程相关，则需要说明技术要点，包括模块调用关系、启动参数、多进程、多线程关系等等。

例如，浏览器医生由于功能需要，支持多线程运行，则有如下说明：



例如，浏览器在线名单需要考虑多线程保护，则有如下说明：

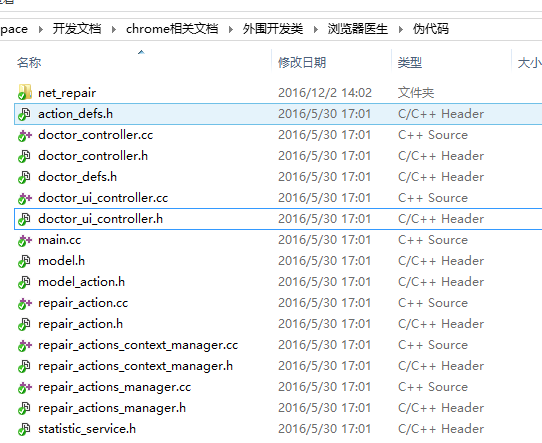


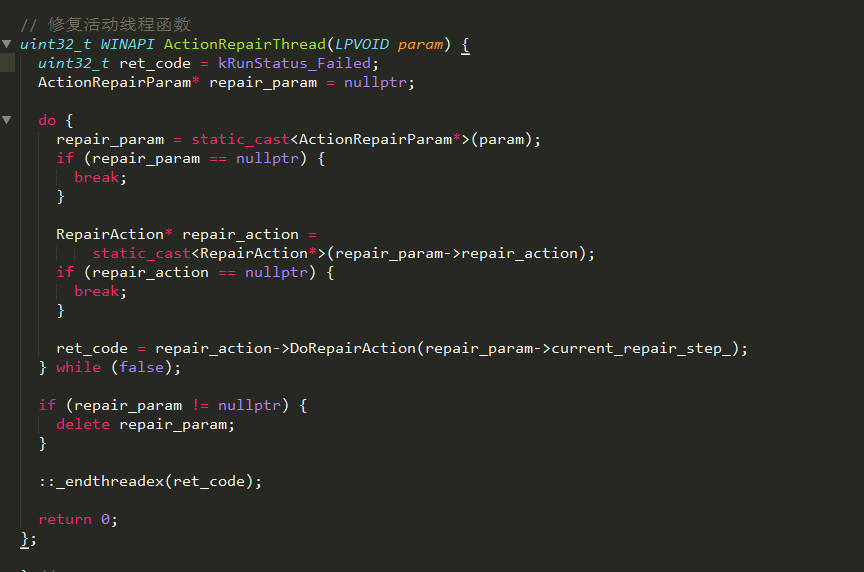
1. 设计方案需要用到复杂数据结构时，需要专门对数据结构进行说明，包括内存数据结构、配置数据结构等。本地配置可以说明其文件层次结构、格式规范、配置字段及其含义等。以服务推送小工具资源包为例：





1. 所开发的功能逻辑比较复杂时，可以通过伪代码来辅助理解逻辑。伪代码要求描述清楚业务实现核心类、接口、数据结构、基本逻辑。需要注意的是，命名要规范、正确、易懂。各个类和函数的命名保持对称性、一致性。例如：



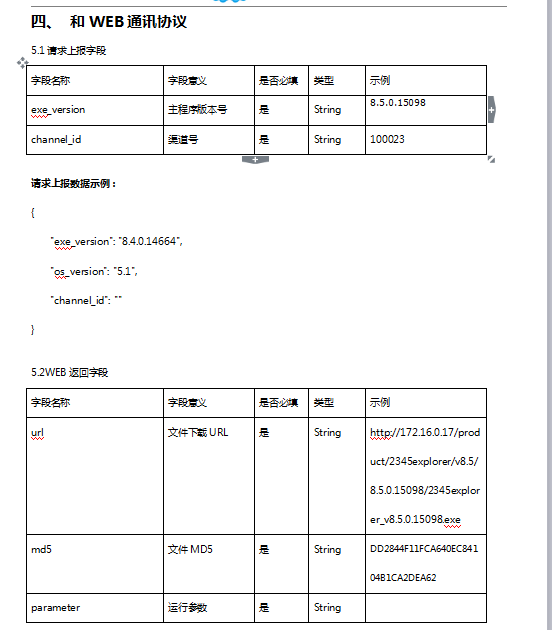


1. 开发的功能需要用到名单时，需求对名单进行单独说明，主要包括内存数据结构、名单数据结构、名单匹配算法等。

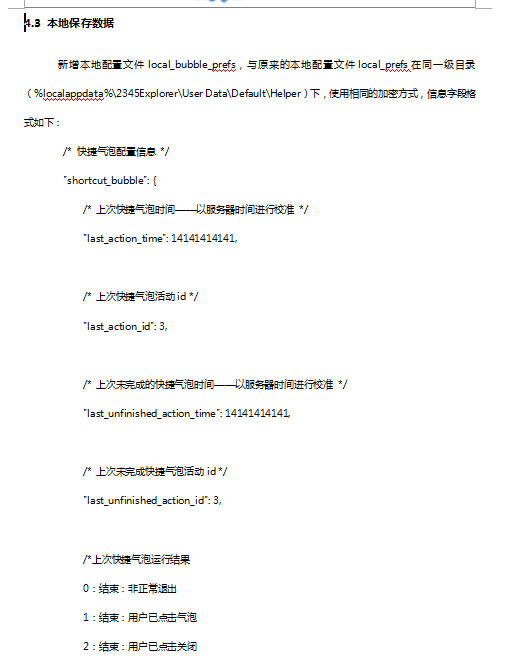
例如，特殊登录页面用小号窗口打开业务：



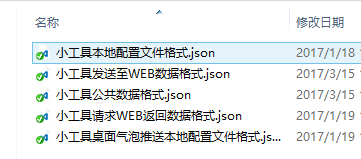
1. 所开发的功能如果与WEB有通信，则需要描述通讯协议。包括请求的php地址、请求方式、上报数据格式、返回数据格式、以及各个字段的含义。例如：



1. 所开发的功能有复杂或者独立的本地配置时，需要对配置文件进行说明。包括配置文件路径、文件名、是否加密、数据格式、配置字段。如服务推送小工具就使用了独立的配置文件：



1. 此外，如果数据结构、WEB通信协议内容很多，需要对外提供相关文档，方便测试和WEB对接，可以将这部分内容单独归档，以服务推送小工具为例：



1. “设计文档中一些看起来比较奇怪的方案”、“在技术评审时别人提出的疑问”、“结果来回修改过的方案”，类似于这些的“历史故事”，可以在文档中单独记录，方便以后更容易理解相关缘由，以免他人重复踩坑。以在线名单为例，部分核心逻辑经过了多次调整。

