# B3log Solo 插件机制

版本：0.1.0，Oct 29, 2011

作者：[88250](http://88250.b3log.org/)，[Vanessa](http://vanessa.b3log.org)

注意：该文档目前内容已经过时，如果继续了解设计细节，请咨询 B3log 开发团队。后续会修改该文档内容，以符合最新实现。

目录

[B3log Solo 插件机制 1](#_Toc353788387)

[概述 1](#_Toc353788388)

[插件定义 1](#_Toc353788389)

[适用场景 1](#_Toc353788390)

[插件机制 1](#_Toc353788391)

[设计原理 1](#_Toc353788392)

[关键点 1](#_Toc353788393)

[安全性 2](#_Toc353788394)

[详细设计 2](#_Toc353788395)

[服务端 2](#_Toc353788396)

[客户端 2](#_Toc353788397)

[资源构成 3](#_Toc353788398)

[附录 4](#_Toc353788399)

[术语表 4](#_Toc353788400)

## 概述

### 插件定义

运行在 [B3log Solo](http://b3log-solo.googlecode.com) 上完整实现了某独立功能的组件，包含视图（FreeMarker 模版）、控制器、模型（JSON、lang props）这三个基本要素。

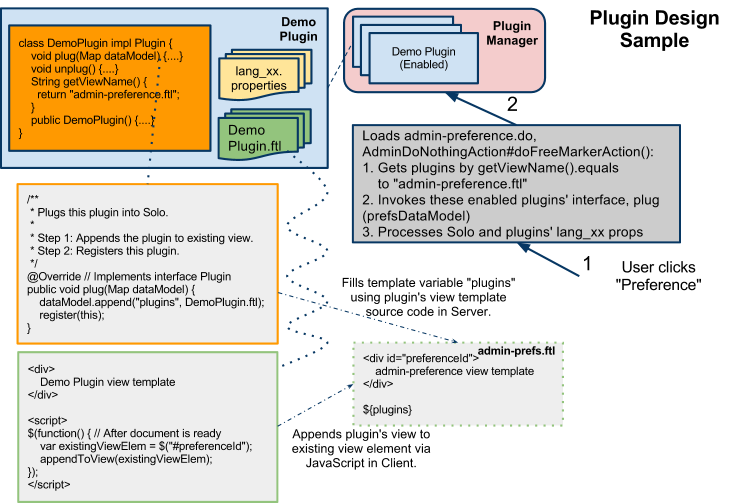
### 适用场景

插件主要用于 [B3log Solo](http://b3log-solo.googlecode.com) 功能特性实现。建议插件进行后台功能实现，因为前台展现部分可能会与特定皮肤耦合。

## 插件机制

### 设计原理

下图为“偏好设定”插件示例：



#### 服务端

服务端主要负责插件查找，然后调用模板引擎处理插件视图模版，并将处理结果添加到已有视图模版变量 ${plugins} 中。

#### 客户端

在服务端生成视图 HTML 返回至客户端后，插件视图中的 JavaScript（$(function(){...});）将在客户端完成插件 DOM 元素与已有元素的调整合并，及进行插件视图定位。

同一*寄主模版*可以存在多个插件，但它们之间不能出现相互的直接的依赖（JavaScript functions），因为插件模版在服务端处理时是无序的，在客户端也就不能确定某客户端插件 JS 函数一定在其他插件 JS 函数前。

### 关键点

* 插件视图模版内容读取、缓存
* 多语言
* 插件状态管理（启用/禁用），持久化
* DOM 元素查找（客户端 APIs）
* 如果插件涉及前台视图，则会依赖特定皮肤
* 与*事件机制*相结合
* GAE 多实例环境下插件状态的同步

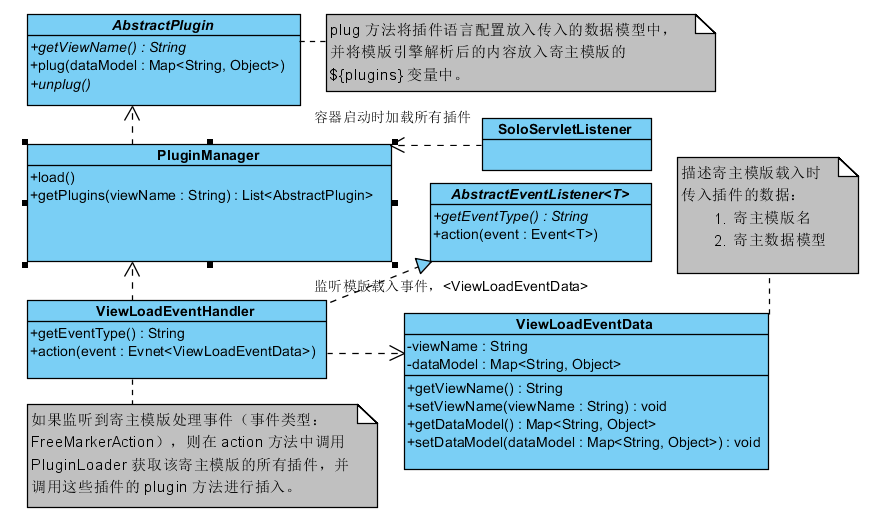
### 安全性

主要是针对插件实现的功能是否存存在不安全因素，例如删除、篡改用户数据。由于插件大多数由第三方开发，目前尚未找到有效解决方式应对潜在的安全问题。

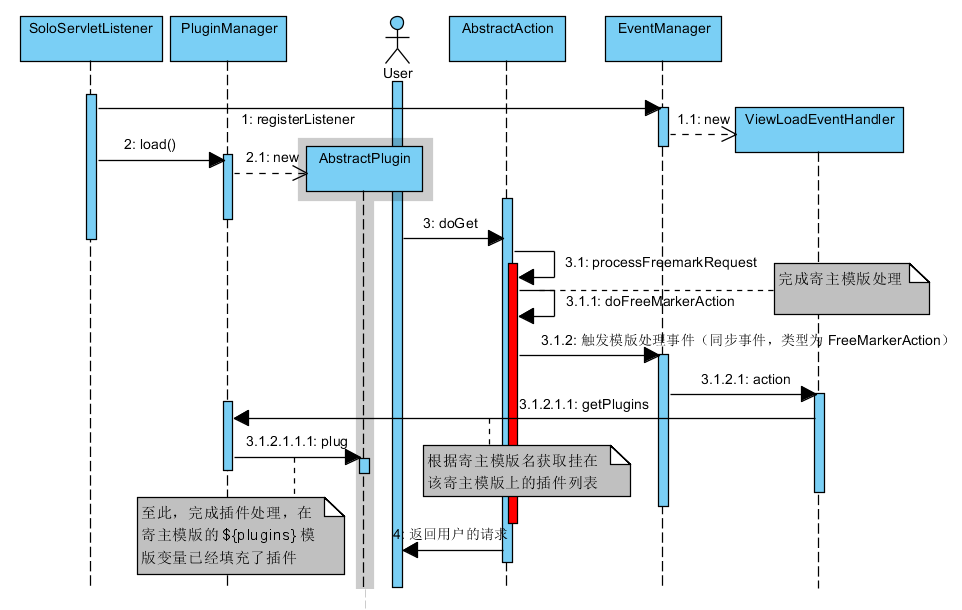
## 详细设计

### 服务端

#### 类图



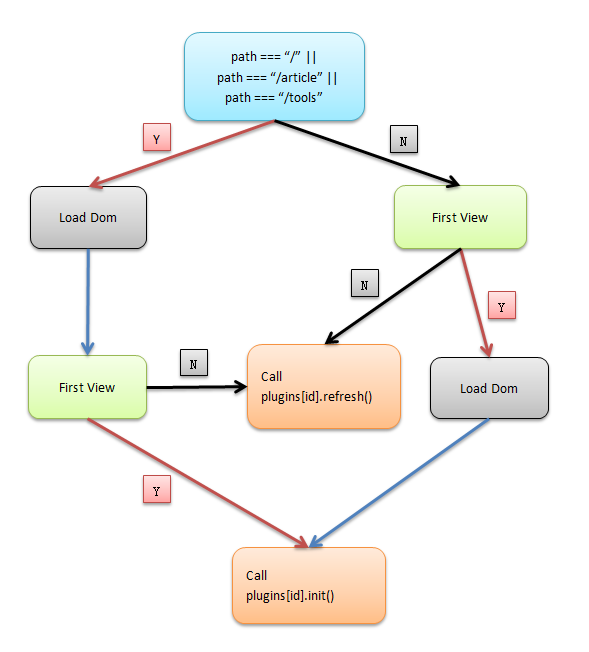
#### 时序图



### 客户端

客户端需要提供 JavaScript 函数及相应的机制用于客户端插的件定位和方法的回调：

**加载过程**



**方法**

admin.plugin.add({});

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| id | string | 插件唯一标识。  插件的变量与方法需放到 plugins[id] 命名空间下。  添加新页面时，做为 URL 后缀。 |
| path | string | 插件添加的位置。  添加新页面，有特定的 URL。地址为 “/admin-index.do#” + path.replace("/", "") + "/" + id。  /  /article  /tools  /tools/preference  添加到特定页面下。  /main/panel1  /article/article | article-list | draft-list  /comment-list  /tools/preference | file-list | page-list | link-list | user-list | plugin-list | others  /tools/preference/preferences | skins | sings | tencent |
| content | string | 添加内容。 |
| [text] | string | 添加新页面的标题。 |
| [index] | int | 添加新页面，标题所在位置的顺序，从1开始。 |

## 资源构成

将所有插件相关资源放到到一个目录中，其中应该包含如下文件：

* lang\_xx.properties  
  至少一份，这些文件为插件视图的多语言配置。
* \*.ftl  
  至少一份，这些文件为插件视图模版，其中 plugin.ftl 为*插件主视图*。
* \*.js/\*.css/图片等静态资源  
  零或多份，这些文件为插件资源。
* x/x/x/Xxx.class  
  至少一份，此文件是插件类实现类文件。在 plugin.properties 里的 classesDirPath 值指定类文件存放的位置。
* plugin.properties  
  一份，此文件是插件描述文件。其中的内容可参考官方插件。

使用时将此目录放置于 ${webRoot}/plugins/ 下，类文件放置于 plugin.properties 里的 classesDirPath 值指定的目录。

## 附录

### 术语表

|  |  |
| --- | --- |
| **术语** | **含义** |
| 寄主/寄主模版 | 插件主视图要插入的模版 |
| 事件/事件机制 | [B3log Latke](http://latke.googlecode.com) 事件处理框架，观察者模式 |
| 插件主视图 | 文件名为 plugin.ftl 的模版，插件模版处理将从该模版进行 |