E:\work\公司VI\公司logo(曲线）.wmf

**前端插件——Tabs**

**（v0.4）**

项目部/前端架构组

2010年 12月

版本

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 作者 | 评审人 | 修订内容 | 版本 |
| 2010-12-6 | 李丽媛 | 丁镇涛 | 无。 | 0.1 |
| 2010-12-8 | 李丽媛 | 丁镇涛 | 文档结构的调整；添加工作开展方式；添加周赟智的需求分析。 | 0.2 |
| 2010-12-10 | 李丽媛 | 丁镇涛 | API 的补充及修改。 | 0.3 |
| 2011-01-05 | 李丽媛 | 丁镇涛 | 添加 Change log。 | 0.4 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 需求分析 1](#_Toc279581257)

[1.1. 需求概要 1](#_Toc279581258)

[1.2. 功能需求 1](#_Toc279581259)

[2. 详细设计 2](#_Toc279581260)

[2.1. 功能描述 2](#_Toc279581261)

[2.1.1. 主要功能 2](#_Toc279581262)

[2.1.2. 扩展功能 2](#_Toc279581263)

[2.2. Init 2](#_Toc279581264)

[2.3. Options 3](#_Toc279581265)

[2.4. Methods 4](#_Toc279581266)

[2.5. Events 5](#_Toc279581267)

[2.6. HTML 结构 5](#_Toc279581268)

[2.7. CSS 命名 6](#_Toc279581269)

[2.7.1. Theme: default 6](#_Toc279581270)

[2.7.2. Theme: 皮肤名称 6](#_Toc279581271)

[3. 工作开展方式与进度计划 7](#_Toc279581272)

[3.1. 开展方式 7](#_Toc279581273)

[3.2. 进度计划 7](#_Toc279581274)

# 需求分析

## 需求概要

对于原系统耦合性强的缺陷，需要对系统进行解耦工作，目标是把没有必要的模块或组件之间的联系切断达到能够在任何情况下单独调用某个组件的能力。这样做是为了以后开发的便利以及系统维护和组件管理的可行性做准备。

解耦的第一个目标就是模块的主tab组件和左边栏的分离。设计预期计划：

* 重新或新作tab组件，该组件要求能在任何情况下独立使用，不与系统其它部分存在任何关联；
* 整合tab组件到模块首页，通过左边栏对其公共方法的调用实现页卡的添加，删除，和切换功能；
* 修改内部使用到的tab页卡，以便统一使用。

## 功能需求

Tab组件在显示上可以包括上边栏的Tab分页卡，和下边栏的页卡内容两个主要部分。

对于编程人员，该组件需要提供一些接口调用（方法，事件）。组件能够构造，在不需要页面元素的情况下也能构建，同时具有页面元素转化功能（Markup转化）。

Markup转化说明：Html页面上的标签具有tab特性时，可以进行转化，使它转化为tab组件。

# 详细设计

## 功能描述

### 主要功能

* 添加一个 tab
* 异步加载页面
* 添加 html 元素
* 删除一个 tab
* 限制 tab 数目
* 支持多皮肤
* 为每一个 tab header 绑定同一用户自定义事件
* 为每一个 tab header 单独绑定不同的用户自定义事件
* 更新 tab
* 设置某个 tab 不可关闭
* 设置 tab header 标题长度
* 设置 tab header 图标
* 自定义 tab header
* 根据特定的 html 构造出该 tabs 插件

### 扩展功能

* tab 拖拽排序
* tab 数过多可进行滚动
* 删除所有 tab
* 设置 tab 为 enable 或 disable 状态
* 键盘监听
* 鼠标监听

## Init

.tabs();

## Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 默认值 | 说明 | 示例 |
| theme | String | default | 皮肤样式 | “theme”: “black” |
| bind | Array | 无 | 为每个 tab header 绑定相同事件 | “bind”: [{  “type”: “mouseover”,  “action”: function (event) {}  }, {  “type”: “click”,  “action”: function (event) {}  }] |
| type | String | 无 | 绑定的事件类型 | “type”: “mouseover” |
| action | Function | 无 | 用户自定事件 | “action”: function (event) {  alert(“Hi, I’m Vanessa!”)  } |
| count | Int | 无 | 限制 tab 数目 | “count”: 5 |
| data | Array | 无 | tab header 设置 | “data”:[{  “title”: ”tab-1”,  “count”: 5,  “icon”: “tab1Icon.png”,  “close”: true,  “url”: “tab-1.html”  }, {  “titleHtml”: “表单<span class=‘simpleButton’>精简</span>”,  content: “<span>Hi, I’m content!</span>”  }] |
| title | String | 无 | tab header 标题 | “title”: “tab-1” |
| content | Int | 无 | tab header 标题长度 | “count”: 6 |
| close | Boolean | false | 该 tab 是否可关闭 | “close”: true |
| titleHtml | String | 无 | 用户自定义 tab header | “titileHtml”: “表单<span class=‘simpleButton’>精简</span>” |
| content | String | 无 | tab 内容 | “content”: “<span>Hi, I’m content!</span>” |
| url | String | 无 | 异步加载外部 html 作为 tab 内容 | “url”: “tab-1.html” |
| reload | Boolean | false | 异步加载外部 html 作为 tab 内容时，是否需要每次都重新进行加载 | “reload”: true |
| height | Int | 无 | tab 中内容部分的高度设置。 | “height”: 120 |
| event | String | “click” | 切换 tabs 时，需触发的事件名称。 | “event”: “mouseover” |
| closeSyn | Boolean | true | 关闭 tab 时，是否触发用户绑定的关闭事件。 | “closeSyn”: false |

## Methods

|  |  |
| --- | --- |
| 方法 | 说明 |
| .tabs(“option”, optionName) | 获取 options 值: height, count, theme |
| .tabs(“option”, optionName, value) | 设置 options 值: height, count, theme |
| .tabs(“destroy”) | 完全移除 tabs 功能 |
| .tabs(“length”) | 获取 tab 数目 |
| .tabs(“remove”, [index]) | 无 index 时，移除当前 tab  index 存在时  index 为 number 时移除第 index 个 tab, index 从零开始计数  index 为 string 时，移除 id 为 index 的tab |
| .tabs(“select”, index) | 设置第 index 个 tab 被选中  index 从零开始计数 |
| .tabs(“add”, {data options}, [index]) | 在末尾添加一个 tab  在第 index 个 tab 后添加一个 tab |
| .tabs(“bind”, [{  “type”: “mouseover”,  “action”: function (event) {}  }, {  “type”: “click”,  “action”: function (event) {}  }], index) | 为第 index 个tab header 绑定特定的用户事件 |
| .tabs(“unbind”, type, index) | 移除第 index 上绑定的类型为 type 的用户事件 |
| .tabs(“update”, {}, [index]) | * 无 index 时，更新当前 tab * index 存在时，更新第 index 个 tab   index 从零开始计数，目前只支持 title, content, url 更新   * index 为 “settings” 时，更新 tabs 的设置目前只支持 height   Demo:  $("#tabs").tabs("update", {  "title": "新建的远程收文文档"  } |

## Events

|  |  |
| --- | --- |
| 事件 | 说明 |
| .tabs({  load: function (event) {}  }) | 加载完毕后触发该事件 |
| .tabs({  add: function (event) {}  }) | 添加完一个 tab 时，触发该事件 |
| .tabs({  remove: function (event) {}  }) | 移除一个 tab 时，触发该事件 |

## HTML 结构

<div id="tabs">

<ul>

<li></li>

<li></li>

<li></li>

</ul>

</div>

<div id=”tabsContent”>

<div id="tabs-1">

<p>tabs-1 content.</p>

</div>

<div id="tabs-2">

<p>tabs-2 content.</p>

</div>

<div id="tabs-3">

<p>tabs-3 content.</p>

</div>

</div>

## CSS 命名

### Theme: default

‘.tabs-‘ + 需渲染的部位名称

例如： .tabs-header。

### Theme: 皮肤名称

所有默认的名称 +‘-’+ 皮肤名称。

例如: 皮肤名称为 red，.tabs-header 应相应的更改为 .tabs-header-red。

# 工作开展方式与进度计划

## 开展方式

* 不使用第三方插件，进行自主开发；
* 配合行政事务模块开发进度需求进行开展：
* 首先，实现表单中 tabs 的必须功能，使其可以正常使用;
* 其次，逐步完成该插件的主要功能;
* 最后，扩展功能放入项目完成后再进行开发。

## 进度计划

在没有其他任务分配的前提下，该插件开发进度如下：

2010-12-9 至 2010-12-13（3个工作日）

* 按照 tabs 插件的详细设计完成优先级较高的功能：行政事务模块表单中tabs 的基本功能；
* 参照 OA 发文模块表单中的 tabs 样式完成该插件的此款主题；
* 配合开发人员，使其能够正常运用在行政事务模块的表单中。

2010-12-14 至 2010-12-20（5个工作日）

* 按照 tabs 插件的详细设计完成主要功能；
* 参照 OA 发文模块中的 tabs 样式完成该插件的此款主题。

# Change log

## Version: 0.0.1.2

* 添加 event 事件，可自定义切换 tab 时所触发的事件
* remove 方法的改进：当参数无 index 时，默认移除当前 tab
* 添加 update 方法：当参数无 index 时，默认更新当前 tab。目前只支持 title, content 更新

## Version: 0.0.1.3

* 补充 update 方法：支持对 url 的更新
* 添加关闭 tab 时触发事件的调控：根据 closeSyn 的设置，调用 remove 方法时可触发或不触发用户绑定的关闭事件。
* 内嵌 frame 刷新时对 Firefox 的兼容
* 修改 remove 方法： index 为 string 时，按 id 进行删除；为 number 时，按 数组删除。