# **http://www.cnblogs.com/mipengine/p/mip\_cli\_3\_extension.html**

# **[MIP开发教程(三) 使用MIP-CLI工具调试组件](https://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html)**

* [一 . 在 mip-extensions 仓库中创建新的组件](http://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html" \l "no1)
* [二 . 预览调试组件](http://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html" \l "no2)
* [三 . 在 MIP 页中引用自己编写的 MIP 组件](http://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html" \l "no3)
* [四 . 组件提交到 GitHub 仓库时需要进行校验](http://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html" \l "no4)

站长开发的非通用组件，使用 [组件上线平台](http://www.cnblogs.com/mipengine/p/extension_platform.html) 提交，上线后代码位于 [GitHub/mip-extension-platform](https://github.com/mipengine/mip-extensions-platform) 仓库中。

## **一 . 在组件仓库中创建新的组件**

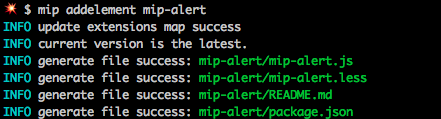
在创建组件之前，需要确认 mip-cli 工具已经安装。在命令行执行：

$ mip -V

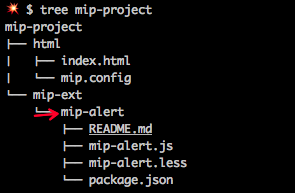
如果返回版本号，则安装成功，可以进行下一步。如果报错，则需要参考 [MIP 开发教程 (一) MIP-CLI 工具安装与环境部署](http://www.cnblogs.com/mipengine/p/mip_cli_1_install.html) 安装 MIP 依赖。

###### **1. 在mip-ext/src目录中创建组件：**

$ mip addelement mip-alert



此时目录结构如下：



###### **2. 开发组件**

mip-alert.js用于定义组件，可参考下方示例。

define(function (require) {

var customElement = require('customElement').create();

customElement.prototype.firstInviewCallback = function () {

var element = this.element;

var text = element.getAttribute('alert-text') || ' 默认 alert 内容 ';

element.addEventListener('click', function(){

alert(text);

});

};

return customElement;

});

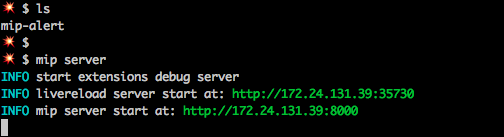
* mip-alert.less用于定义组件样式，可参考 [mip-fixed/mip-fixed.less](https://github.com/mipengine/mip-extensions/blob/master/src/mip-fixed/mip-fixed.less)。
* README.md用于说明组件用法，可参考 [mip-fixed/README.md](https://github.com/mipengine/mip-extensions/blob/master/src/mip-fixed/README.md)。

package.json用于记录组件版本及开发者信息，可参考 [mip-fixed/package.json](https://github.com/mipengine/mip-extensions/blob/master/src/mip-fixed/package.json)。

## **二 . 预览调试组件**

###### **1. 在mip-ext/src目录下启动mip server来预览组件，预览页面访问的是README.md文件中的示例。**

$ mip server



###### **2. 打开调试网页http://127.0.0.1:8000/会列出当前仓库中的组件，点击进入mip-alert组件预览。**

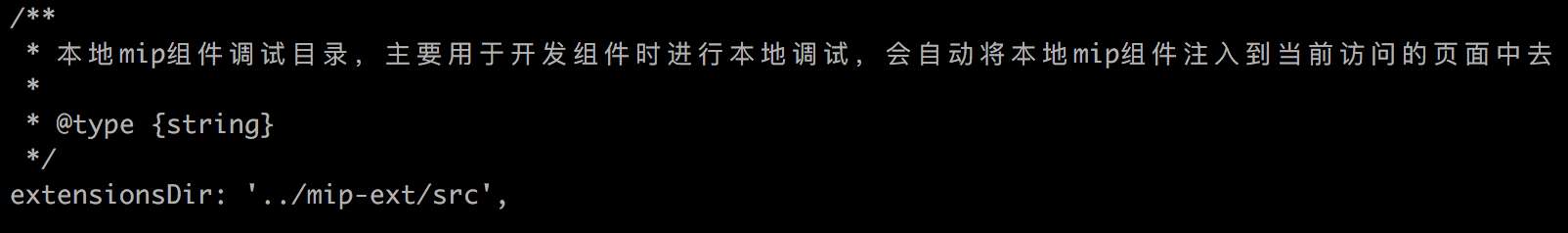
[页面不能预览如何解决？](http://www.cnblogs.com/mipengine/p/mip_cli_3_extension.html" \l "no5)

此时可以修改 mip-alert/mip-alert.js，mip-alert/mip-alert.less 实时预览效果。

## **三 . 在 MIP 页中引用自己编写的 MIP 组件**

###### **1. 修改mip.config**

进入mip-project/html文件夹下，如果没有mip.config文件则执行mip init命令创建此文件。如果已经存在，修改mip.config文件的字段extensionsDir为../mip-ext/src。



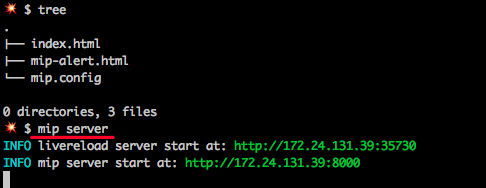
###### **2. 在html目录下创建mip-alert.html文件，并添加mip-alert组件**

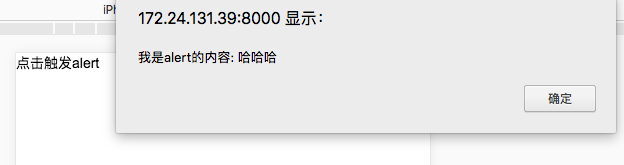
$ mip add mip-alert.html mip-alert

在 body 中引入

<mip-alert alert-text=" 我是 alert 的内容: 哈哈哈 "> 点击触发 alert</mip-alert>

###### **3. 在html目录下启动mip server**

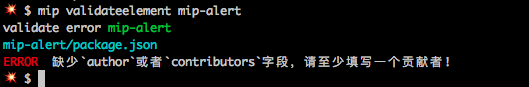


访问http://127.0.01:8000进入调试页面。进入mip-alert.html页面， 点击文字，可以看到组件效果。  


## **四 . 组件提交到 GitHub 仓库时需要进行校验**

在 mip-ext/src 文件夹下，使用如下命令校验：

$ mip validateelement mip-alert



根据报错修改对应文件，确认组件通过校验之后，就可以提交到 [MIP GitHub](https://github.com/mipengine/mip-extensions-platform) 或 MIP 组件平台了。MIP 组件平台的教程见 [MIP 开发教程 (四) MIP 组件平台使用说明](http://www.cnblogs.com/mipengine/p/extension_platform.html)

本系列共有四篇文章：

* [MIP 开发教程 (一) MIP-CLI 工具安装与环境部署](http://www.cnblogs.com/mipengine/p/mip_cli_1_install.html)
* [MIP 开发教程 (二) 使用 MIP-CLI 工具调试 MIP 网页](http://www.cnblogs.com/mipengine/p/mip_cli_2_page.html)
* MIP 开发教程 (三) 使用 MIP-CLI 工具调试组件
* [MIP 开发教程 (四) MIP 组件平台使用说明](http://www.cnblogs.com/mipengine/p/extension_platform.html)

## **附：常见问题解答**

页面不能预览如何解决？  
将mip-ext文件夹下的 mip.config文件删除。

mip-extension-platform 中的组件如何预览？  
如果代码已经提交到 mip-extension-platform 中，需要将组件文件夹复制到 mip-ext/src 中，然后使用 mip server 预览。