# **webDocument**

* ****web打包-------fis3****

一.fis3打包工具（[http://fex-team.github.io/fis3/docs/beginning/intro.html）](http://fex-team.github.io/fis3/docs/beginning/intro.html%EF%BC%89)

* FIS3 是面向前端的工程构建工具。解决前端工程中性能优化、资源加载（异步、同步、按需、预加载、依赖管理、合并、内嵌）、模块化开发、自动化工具、开发规范、代码部署等问题。 就项目而言，打包工具能够做到资源的定位、压缩，产生文件指纹，对于发布、更新等步骤起到帮助。

1.安装node和npm（[https://nodejs.org）](https://nodejs.org)/)

* 安装node和npm（新版本的node自带npm）
* Windows 用户安装完成后需要在 CMD 下确认是否能执行 node 和 npm
* 命令：node -v npm -v

2.安装fis3（cmd下执行）

* npm install -g fis3
* 安装完成后执行 fis3 -v 判断是否安装成功
* 升级fis3 npm update -g fis3
* 重装fis3 npm install -g fis3

3.命令使用

* 进入项目根目录，执行命令： fis3 release -d <path>, <path> 为任意目录，即打包后文件生成的目录

全局或本地安装插件  npm install [-g] fis3-hook-relative

启用插件  fis.hook('relative');

fis3打包工具能够针对png，jpg等文件产生文件指纹，并进行资源定位。 但该方式在js文件或 html中的script标签内的内容中出现的文件将不起作用，需另作处理，使用编译函数 \_\_uri(path) 来定位资源 例如：

document.getElementById('pic\_info').src = app.rel\_html\_imgpath(\_\_uri("../images/login.png"))

在css文件中或html中style标签内的内容，则采用url(path) 以及 src=path 字段来替换 而app.rel\_html\_imgpath()函数则是对编译后的指纹进行的一个处理，代码如下：

rel\_html\_imgpath:function(iconurl)

{

console.log(app.canonical\_uri(iconurl.replace(/.\*\/([^\/]+\/[^\/]+)$/, '$1')));

return app.canonical\_uri(iconurl.replace(/.\*\/([^\/]+\/[^\/]+)$/, '$1'));

}

canonical\_uri:function(src, base\_path)

{

var root\_page = /^[^?#]\*\//.exec(location.href)[0],

root\_domain = /^\w+\:\/\/\/?[^\/]+/.exec(root\_page)[0],

absolute\_regex = /^\w+\:\/\//;

// is `src` is protocol-relative (begins with // or ///), prepend protocol

if (/^\/\/\/?/.test(src))

{

src = location.protocol + src;

}

// is `src` page-relative? (not an absolute URL, and not a domain-relative path, beginning with /)

else if (!absolute\_regex.test(src) && src.charAt(0) != "/")

{

// prepend `base\_path`, if any

src = (base\_path || "") + src;

}

// make sure to return `src` as absolute

return absolute\_regex.test(src) ? src : ((src.charAt(0) == "/" ? root\_domain : root\_page) + src);

}

在完成页面开发之后，先自行在本地服务器上进行测试，代码上传到服务器上， 在电视上通过CordovaTest.apk进行测试。 在确认没有问题之后，发出邮件给运维人员进行正式环境([https://webapp.skysrt.com](https://webapp.skysrt.com/)) 的部署。