

# 移动端开发体系设计与实现

——开发规范、工具、框架设计与实践

张云龙

---

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体, 内容聚焦互联网、技术、科技创业等主题。为极客, 发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



# 大纲

- 为前端团队设计开发体系
- 前端开发技术体系关系图
- 前端开发体系设计步骤
- 体验新型开发体系——scrat
- 小结

# 为前端团队设计开发体系

- 要模块化开发，所以选了xxx模块化框架
- 要自动优化，所以用了xxx工具
- 要组件化开发，所以选了xxx组件框架
- 需要前端模板，所以选了xxx模板库
- let's go!?

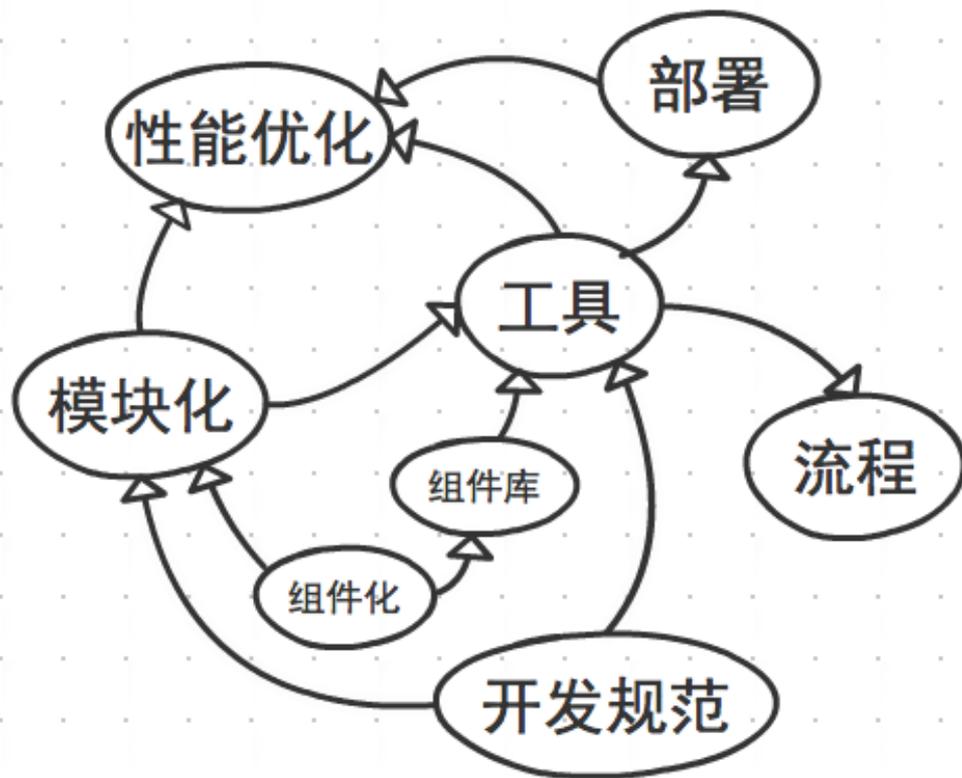
# 桥豆麻袋（稍等）

- 模块化框架只实现模块管理就可以了么？
  - 模块化框架肩负着模块管理和**模块加载**的双重任务
  - 涉及到资源加载就与网络请求优化相关，所选框架考虑到了这个**工程问题**么？
  - 模块化框架优化网络请求时依赖**combo服务**，你的团队打算支持么？
- 自动化工具只压缩就够了么？
  - 它的所有功能都能在各个平台尤其是**持续集成环境**下很好的工作么？
  - 它能很好的配合模块化框架的**目录规范**、**打包优化**和**预处理**么
  - 它能按照你们公司的部署要求进行**规范调整**么

# 大纲

- 为前端团队设计开发体系
- 前端开发技术体系关系图
- 前端开发体系设计步骤
- 体验新型开发体系——scrat
- 小结

# 前端各技术相互影响关系



拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。

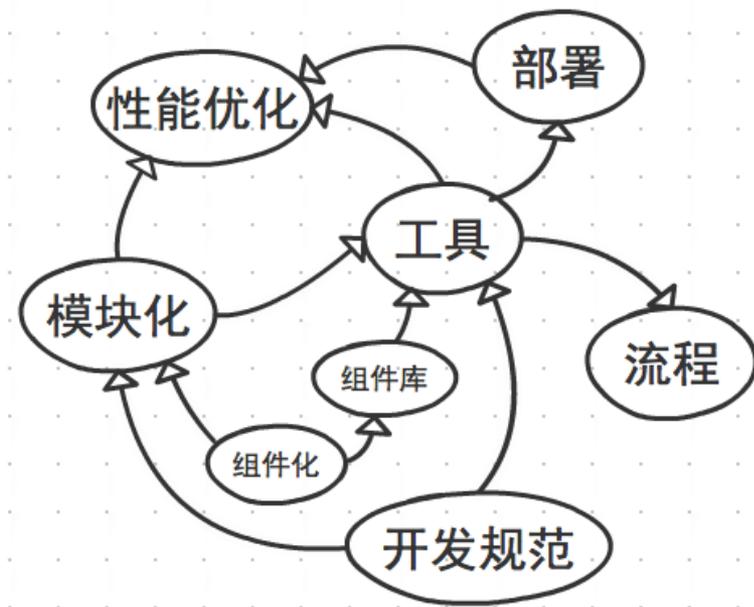


# 关系图解读

拔丝® 活动

Buzz, 连接极客的未来

- 模块化框架要求：
  - 基本的模块管理与加载
  - 考虑网络请求性能优化（cdn缓存、合并请求）
  - 提供舒服的编程规范
- 组件系统要求：
  - 在模块化的基础上设计组件化系统
  - 以js、css、模板为单位开发组件
  - js、css依赖管理
- 工具要求：
  - 支持单个资源优化
  - 提供舒服的目录规范
  - 与模块化框架无缝配合
  - 获取、管理组件
  - 与团队现有的流程整合
  - 提供方便获取、分享组件的功能
  - 符合团队资源部署要求
  - 非覆盖式发布、灰度发布



拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。

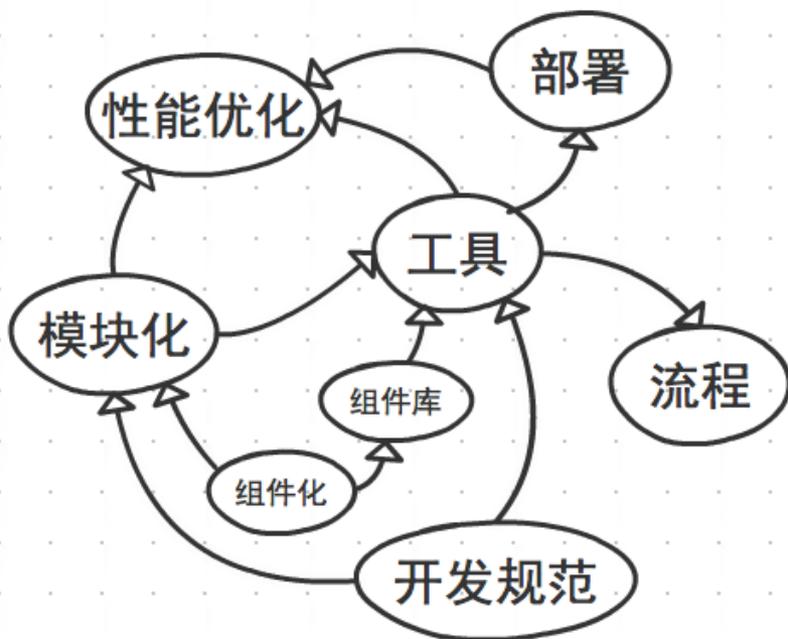


火龙果•整理  
uml.org.cn

# 大纲

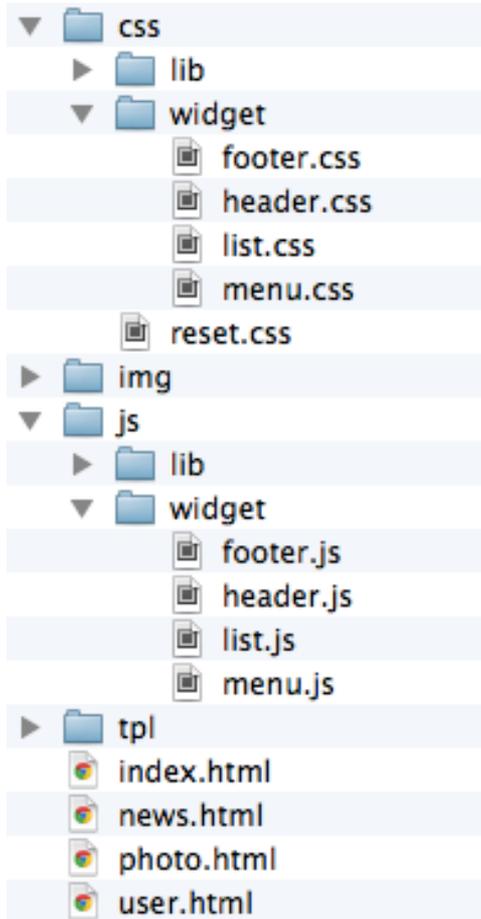
- 为前端团队设计开发体系
- 前端开发技术体系关系图
- 前端开发体系设计步骤
- 体验新型开发体系——scrat
- 小结

# 前端开发体系设计步骤



- 定制规范
  - 开发目录规范：哪里放页面，哪里放组件，哪里放模板
  - 部署规范：静态资源部署、后端模板部署
- 选型框架
  - 模块化框架
  - 组件化框架
  - 基础库/组件库
- 系统划分
  - 业务划分、组件划分
- 流程打通
  - 搭建持续集成平台
  - 本地调试与开发环境
- 脚手架
  - 组件脚手架、业务项目脚手架
- 工具开发
  - 把规范、框架、流程、脚手架需求串联起来
- let's go

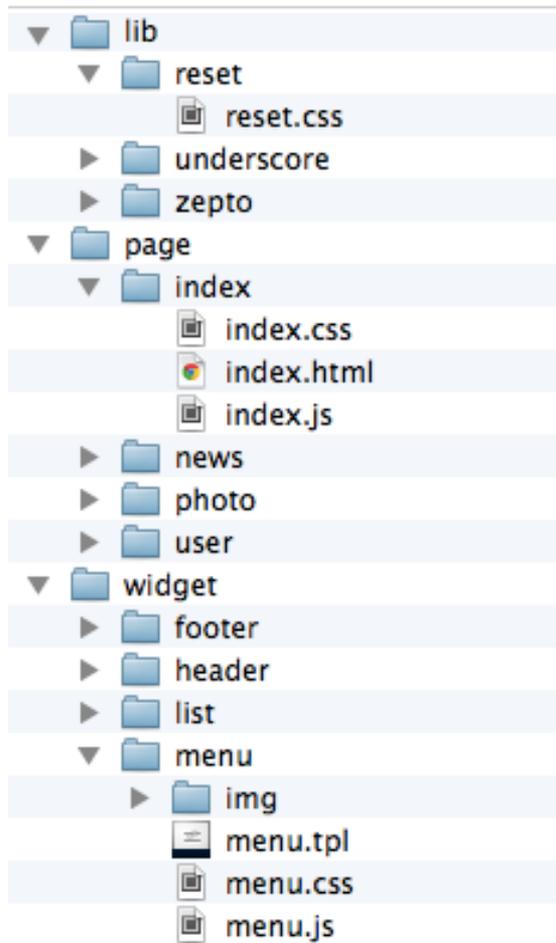
# 目录规范——这样好么？



一直以来，这是前端开发的典型目录结构，这样做合理么？

- 要升级menu组件功能
  - 去js/widget/menu.js修改逻辑
  - 去css/widget/menu.css 修改样式
  - 去tpl/widget/menu.tpl 修改模板
- img目录的图片不知道都被那些组件使用，组件删除对应资源通常不会删除，img中的文件有增无减
- 把目录命名为“css”、“js”、“tpl”和把css的class命名为“red”，“green”，“blue”一样缺乏逻辑性

# 目录规范——这样如何？



按功能逻辑命名划分目录，资源依赖就近维护

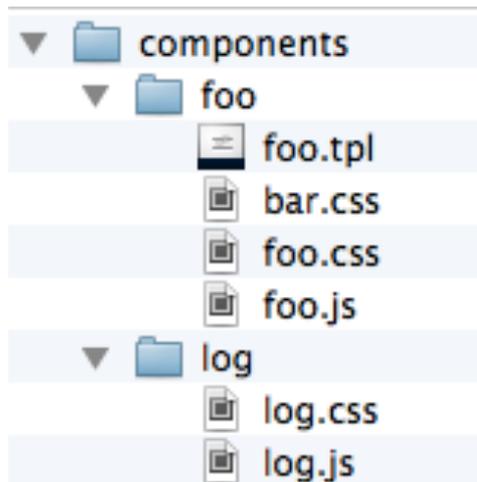
- 按功能逻辑划分目录“lib”，“widget”，“page”
- 就近维护依赖的资源，节省开发中文件定位成本
- 以目录为单位删除组件及其资源

引入的问题：如果文件要部署在不同的服务器上怎么办

# 组件开发

拔丝® 活动

Buzz, 连接极客的未来



```
1 var log = require('log');
2
3 exports.start = function(msg){
4   ... log.notice(msg);
5 };
6
7 exports.stop = function(msg){
8   ... log.error(msg);
9 };
```

```
1 /**
2  * @require ./bar.css
3  */
4
5 .foo {
6   ... width: 300px;
7   ... height: 100px;
8 }
```

- 以目录为单位，将js、css、模板放在一起维护
- js、css中均可以使用依赖声明
- 像写nodejs一样编写js模块

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



火龙果•整理  
uml.org.cn

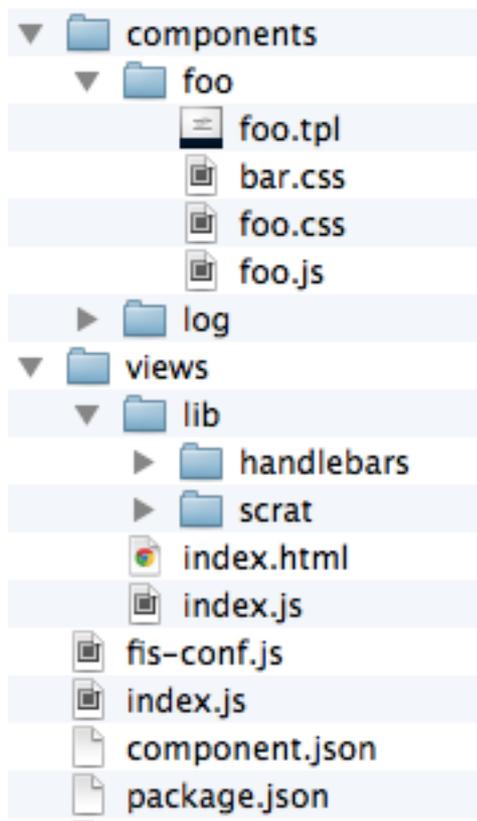
# 大纲

- 为前端团队设计开发体系
- 前端开发技术体系关系图
- 前端开发体系设计步骤
- 体验新型开发体系——**scrat**
- 小结

# 什么是scrat

- 专为移动项目打造的前端开发体系
  - 模块化框架: **scrat.js**
    - 模块管理预加载
    - 网络请求优化 (combo)
  - 开发工具: **scrat**
    - 压缩(js\css\png)
    - 模块依赖分析
    - 资源内嵌(图片、字体、声音)
    - 本地调试
    - 多终端浏览器自动刷新
    - 组件获取与管理
  - 组件仓库: **GitHub/GitLib**

# 目录规范



- **component\_modules**: 外部组件目录
  - 使用工具从GitHub或Gitlab安装到项目中
- **components**: 组件目录
  - 一个目录一个组件
  - 组件内维护自己的js、css、模板、图片
- **views**: 页面及非组件化文件目录
- **fis-conf.js**: **scrat**工具配置文件
- **index.js**: 调试服务器
- **package.json**: 调试服务器

# 模块化框架 `scrat.js`

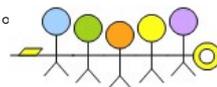
- GitHub: <https://github.com/scrat-team/scrat.js>
- 接口:
  - `require.async(ids, callback);`
  - `require.config(options);`

# 模块化框架 `scrat.js`

- 模块代码

```
1  /**
2   * @require bar.css
3   */
4
5  var log = require('log');
6
7  exports.start = function(msg){
8    ... log.notice(msg);
9  };
10
11 exports.stop = function(msg){
12   ... log.error(msg);
13  };
```

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



# 模块化框架 `scrat.js`

- 页面启动器

```
1  <!doctype html>
2  <html>
3  <head>
4  ... <title>hello world</title>
5  </head>
6  <body>
7  ... <h1>it works</h1>
8  ... <script type="text/javascript" src="lib/scrat/scrat.js"></script>
9  ... <script>
10 ...     require.config( __FRAMEWORK_CONFIG__ );
11 ...     require.async( 'foo', function( foo ){
12 ...         foo.start( 'it works' );
13 ...         foo.stop( 'it works' );
14 ...     } );
15 ... </script>
16 </body>
17 </html>
```

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



# 模块化框架 `scrat.js`

- 模块管理，包括对css的依赖管理

---

	127.0.0.1
	scrat.js /proj/0.0.1/lib/scrat
	bar.css /c/proj/0.0.1/foo
	log.css /c/proj/0.0.1/log
	foo.css /c/proj/0.0.1/foo
	log.js /c/proj/0.0.1/log
	foo.js /c/proj/0.0.1/foo

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



# 模块化框架 `scrat.js`

- 按需加载，请求合并，以`require.async`的调用为单位，根据工具生成的依赖表，按需发起js、css的合并请求



127.0.0.1



`scrat.js`  
`/proj/0.0.1/lib/scrat`



`??proj/0.0.1/foo/bar.css,proj/0.0.1/log/log.css,proj/0.0.1/foo/foo.css`



`??proj/0.0.1/log/log.js,proj/0.0.1/foo/foo.js`

# 工具 - scrat

- Github: <https://github.com/scrat-team/scrat>
- 安装: `npm install -g scrat`
- 三条命令, 解决所有开发问题
  - `scrat release`: 发布项目(压缩|优化|打包|浏览器刷新|文件监听)
  - `scrat server`: 本地服务器
  - `scrat install`: 从GitHub/GitLab安装组件

# 工具 - scrat

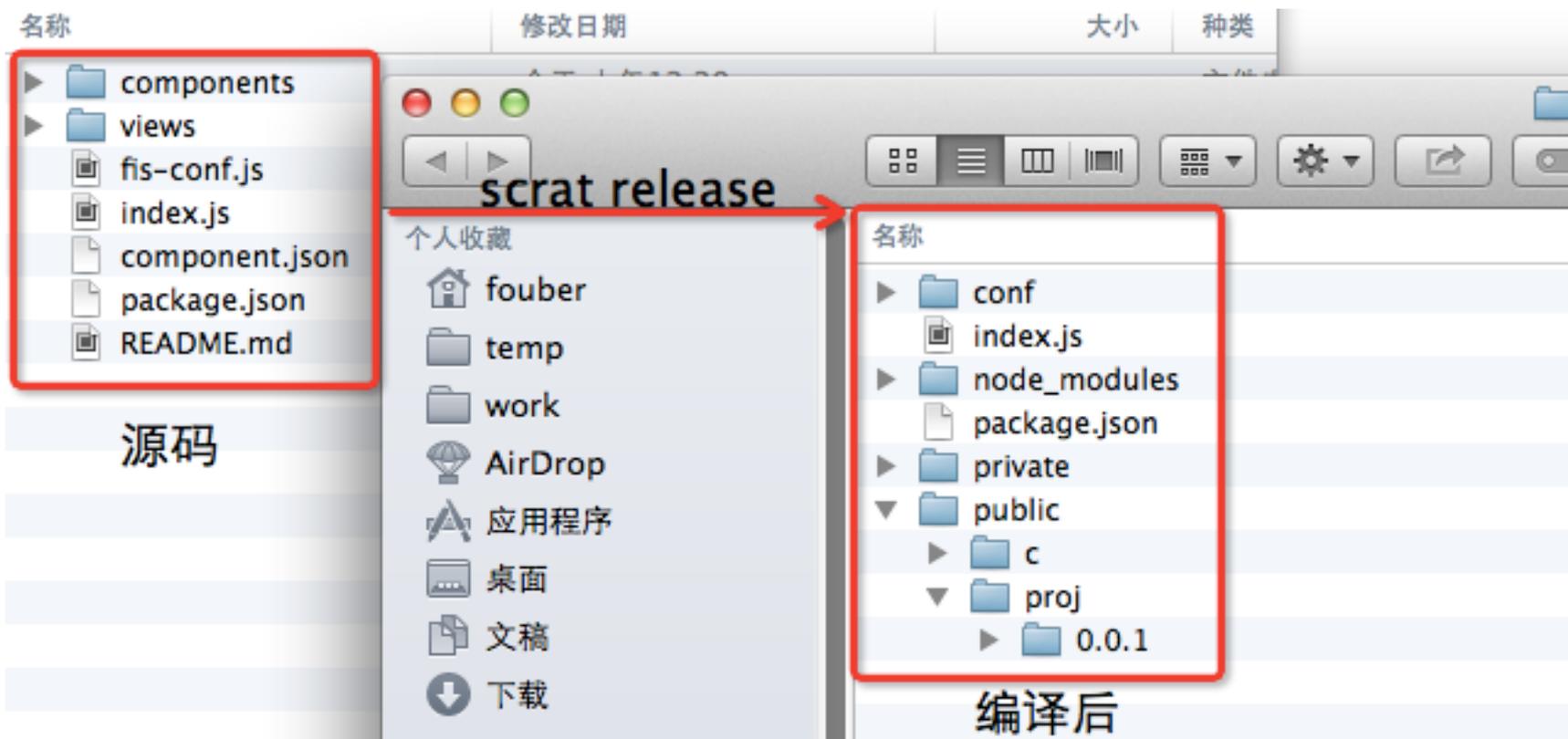
- 开发命令: `scrat release`
  - `scrat release` 发布项目到调试服务器
  - `scrat release -d <path>` 发布项目到指定目录与远程服务器
  - `scrat release -o` 压缩(js\css\png)发布
  - `scrat release -p` 开启combo
  - `scrat release -w` 开启文件监听
  - `scrat release -L` 开启浏览器自动刷新
  - `scrat release -wopL` 开启文件监听、压缩、**combo**、浏览器自动刷新
- 通过命令行参数任意切换组合 压缩/非压缩、打包/非打包等开发状态

# 工具 - scrat

拔丝® 活动

Buzz, 连接极客的未来

- 构建结果发布到调试服务器，不污染源码环境



拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



火龙果•整理  
uml.org.cn

# 工具 - scrat

- scrat release -wL



- 文件监听
- 所有终端自动刷新

拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



# 工具 - scrat

- 插件可扩展
  - 编译插件: less、coffee、jada、marked、sass、ejs、handlebars、react、...
  - 校验插件: jshint、csslint、...
  - 优化插件: uglify-js、clean-css、...
  - 测试插件: phantomjs

# 发布到远程服务器

```
3  fis.config.set('deploy', {
4  ... test1: {
5  ... receiver: 'http://10.12.9.123:8088/receiver.php',
6  ... to: '/home/work/static'
7  ... },
8  ... test2: {
9  ... receiver: 'http://24.13.0.574:8018/receiver.php',
10 ... to: '/home/work/static'
11 ... }
12 });
```

- `scrat release -wLd test1,test2`

将一次编译的结果同时发布到两个地方，并且文件监听+浏览器自动刷新

# 工具 - scrat

```
<script>
  require.config({
    "version": "0.0.1",
    "name": "proj",
    "combo": false,
    "urlPattern": "/c/%s",
    "comboPattern": "/??%s",
    "alias": {
      "foo": "proj/0.0.1/foo/foo.js",
      "log": "proj/0.0.1/log/log.js"
    },
    "deps": {
      "proj/0.0.1/foo/foo.css": [
        "proj/0.0.1/foo/bar.css"
      ],
      "proj/0.0.1/foo/foo.js": [
        "proj/0.0.1/foo/bar.css",
        "log",
        "proj/0.0.1/foo/foo.css"
      ],
      "proj/0.0.1/log/log.js": [
        "proj/0.0.1/log/log.css"
      ]
    }
  });
  require.async('foo', function(foo){
    foo.start('it works');
    foo.stop('it works');
  });
</script>
```

- 分析依赖，生成scrat.js所需配置
- **scrat.js**根据生成的依赖表可以提生成完备**combo**请求，无需运行时分析依赖

# 工具 - scrat

- 资源内嵌
  - 源码:

```
var img = __inline('mask.png');  
var css = __inline('bar.css');  
var tpl = __inline('foo.tpl');
```

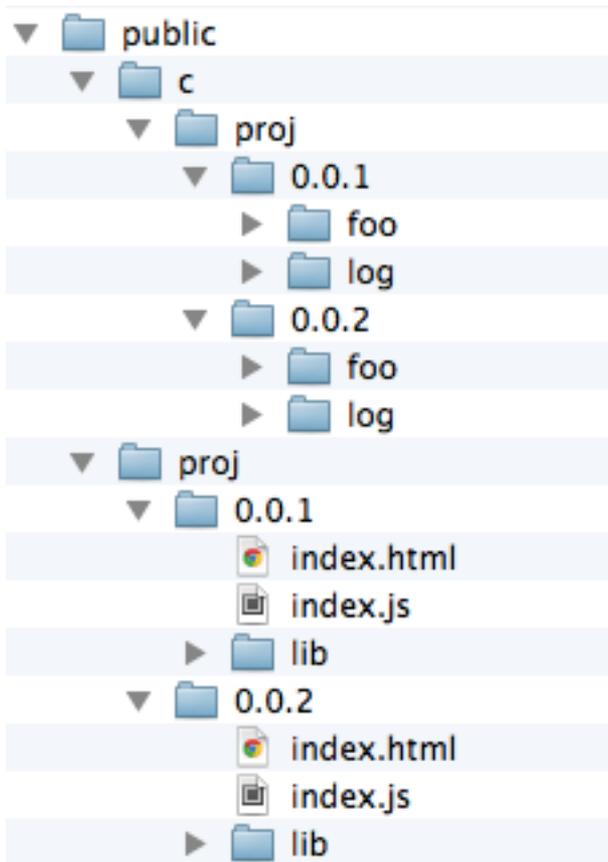
- scrat release 后:

```
var img = 'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAA  
var css = ".bar {\n···color: red;\n}";  
var tpl = "<h1>foo</h1>";
```

# 工具 - scrat

- 资源内嵌
  - css中内嵌css、图片(base64)、字体(base64)
  - js中内嵌js、css(文本)、模板(文本或预编译函数)、图片(base64)、声音(base64)
  - html中内嵌html、js、css、图片(base64)、声音(base64)、视频(base64)
- 支持把所有资源打包成一个html文件发布。

# 工具 - scrat



- 部署配置

```
fis.config.set('name', 'proj');  
fis.config.set('version', '0.0.2');
```

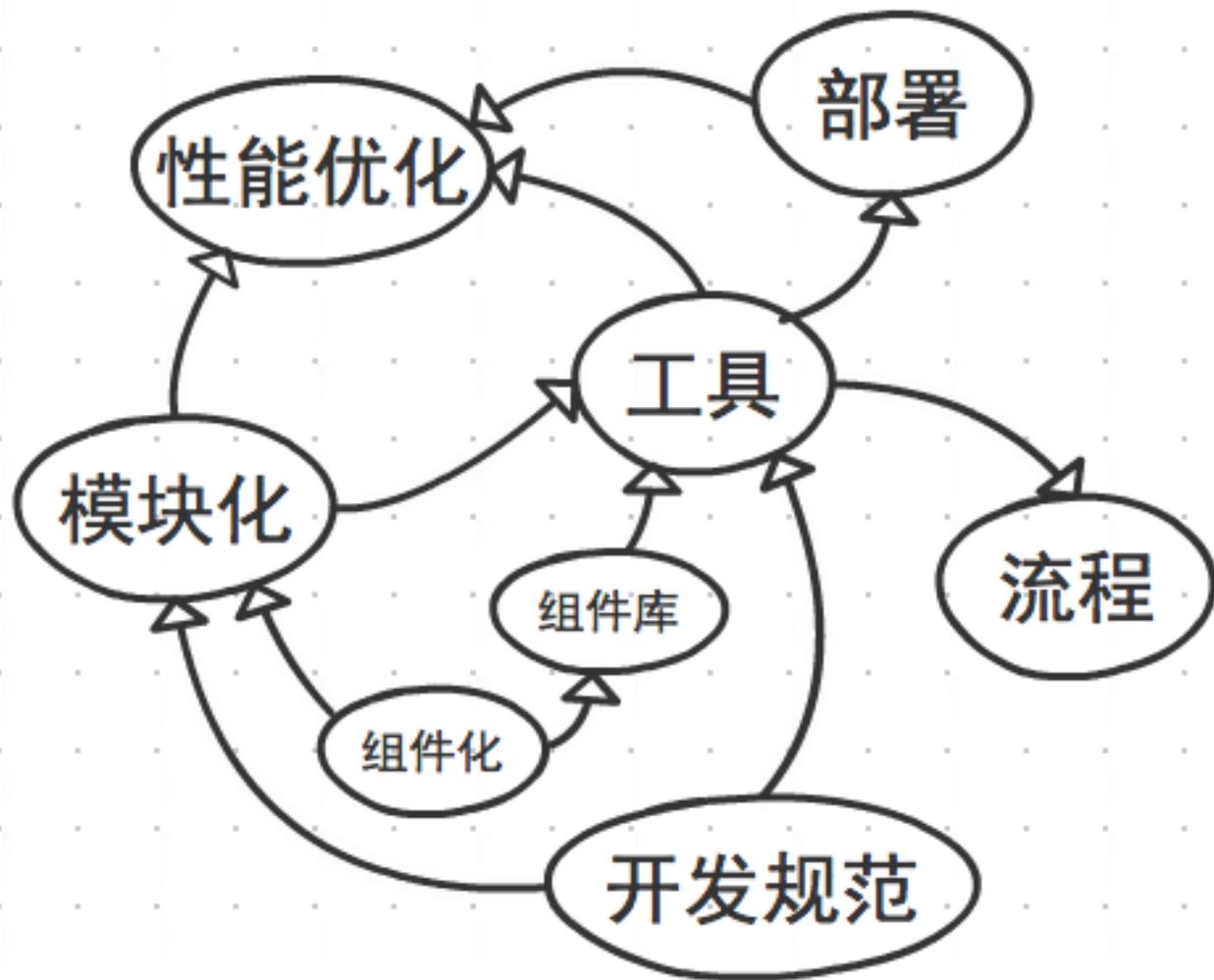
- 部署结果

- 非覆盖式发布
- CDN永久缓存
- 组件共享
- 灰度发布

# 小结

拔丝® 活动

Buzz, 连接极客的未来



拔丝活动 (Buzz.cn) 是一个聚合专业活动、沙龙的新媒体，内容聚焦互联网、技术、科技创业等主题。为极客，发烧友提供专业活动的热门推荐、图文直播、资料下载、评论互动、职业社交等一站式服务。



火龙果•整理

uml.org.cn