

# 京东前端工业化实践之路

刘威

2014-10



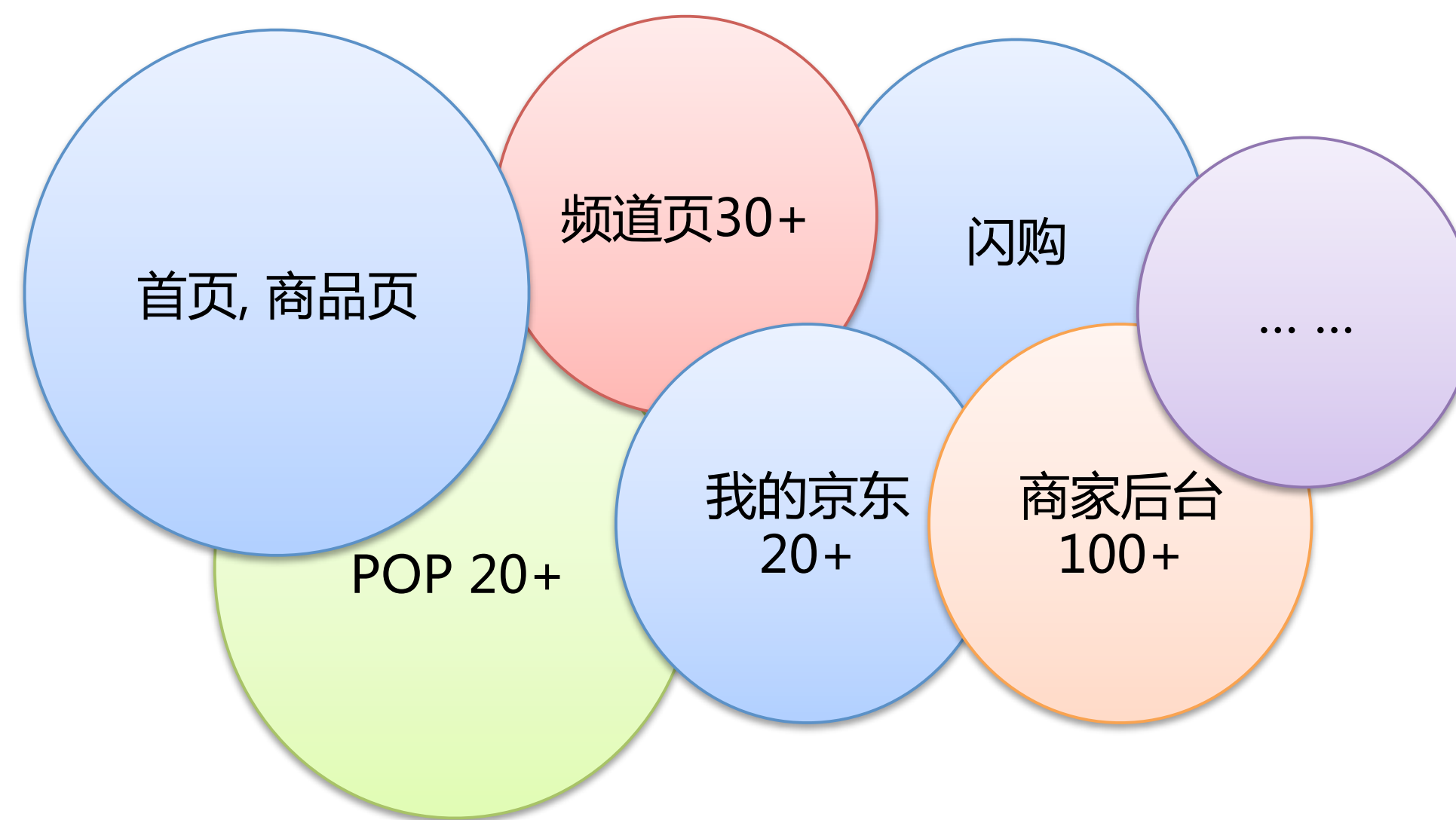
# 关于我

- 刘威 (@putaoshu: [微博](#), [微信](#), [github](#))
- 现就职于京东用户体验部 (JDC), 负责前端架构组
- 曾工作于百度-百科、新浪-云计算

# 一、业务

# 业务

- 支持的业务线众多
- 电商：交付速度要快
- 人力资源往往有限



# 业务

- 业务要求我们
  - 项目代码可复用，易扩展
  - 开发流程要统一化，标准化，工业化

# 工业化

“工业化特点：机器化大生产的生产方式越来越广泛使用，取代手工生产方式”

百度百科 <http://baike.baidu.com/view/143403.htm>

# 工业化

前端框架

前端规范

组件demo

开发流程

模块云

命令行工具

.....

组件api

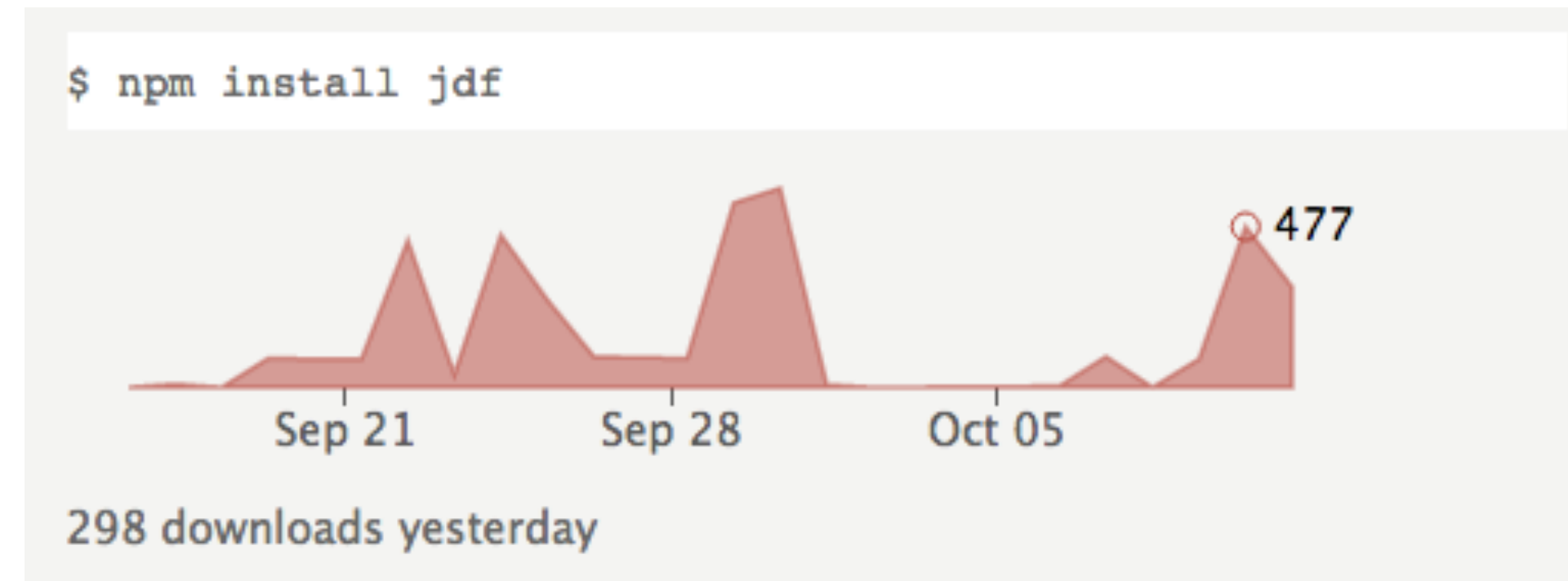
任务系统

## 二、前端开发流程



# 工具

- JDF
- nodejs命令行工具
- 涉及到开发流程为构建、编译、输出、联调、上线、辅助等开发流程，以及模块的预览，安装，下载，发布



# 构建

- 生成标准化的项目目录文件夹

```
jdf install init
```



```
.
├── config.json
├── css
│   └── i
├── html
│   └── index.html
├── js
└── widget
```

# 模块

- 定义
- 目录结构

```
1 |—— ui-product-list
2 |   |—— component.json 配置文件
3 |   |—— ui-product-list.json vm的数据源文件
4 |   |—— ui-product-list.vm 模板文件
5 |   |—— ui-product-list.css css文件
6 |   |—— ui-product-list.js js文件
7 |   |—— i/ 等待上线的图片文件夹
8 |   |—— images/ 素材图片文件夹
```

- 引用

```
16 {%widget name="ui-product-list"%}
```

# 模块

- 配置文件
- js
- SCSS
- vm
- 数据源

```
1  {  
2      "name": "ui-product-list",  
3      "version": "1.0.0",  
4      "description": "商品列表",  
5      "dependencies": {  
6          "ui-lib": "1.0.0"  
7      }  
8  }
```

- Component规范

# 模块

- 配置文件
- js
- SCSS
- vm
- 数据源

```
1  define(function(require,exports,module){
2      var jdLogin = require('jdf/1.0.
          0/unit/login/1.0.0/login.js');
3      var getPrice = require('
          /widget/getPrice/getPrice.js');
4
5      function init(el){
6          //todo
7      }
8      exports.init = init;
9  });
```

- CMD规范
- 依赖使用相对路径，开发路径和发布路径分离

# 模块

- 配置文件
- js
- SCSS
- vm
- 数据源

```
1  @import "../ui-lib/ui-lib.scss";
2
3  $color: #333;
4  .ui-product-list{
5      @include border-radius(2px);
6      color: $colorcolor;
7      background:url(i/ui-product-list.png) no-
          repeat;
8  }
```

- Scss语法
- 依赖和图片，使用相对路径，方便移植
- 以模块名开头，嵌套子元素，避免多人协作命名冲突

# 模块

- 配置文件
- js
- SCSS
- vm
- 数据源

```
1  <div class="ui-product-list">
2  <ul>
3      #foreach( $product in $allProducts )
4      <li>
5          <div class="p-name">$product.name</div>
6          >
7          #if($product.price)
8          <div class="p-price">$product.price</div>
9          #else
10         <div class="p-price">暂无价格</div>
11         #end
12     </li>
13 #end
14 </ul>
15 </div>
```

- Velocity (java模板引擎) 语法
- 前后端共用
- 命令行工具提供前端编译环境



# 模块

- 配置文件
- js
- SCSS
- vm
- 数据源

```
1  {
2      "allProducts": [
3          {
4              "name": "苹果6",
5              "price": 5000
6          },
7          {
8              "name": "鞋",
9              "price": 100
10         },
11         {
12             "name": "上衣"
13         }
14     ]
15 }
```

- 来源
  - 1) 模块名.json文件
  - 2) 引用时传参  
{%widget name="a" data='{ "name": "a" }'%}



# 模块

- 模块安装：解决可复用的问题，不用每次都新建，可以在模块云上查找合适的模块
- 模块新建
- 发布模块：解决了共享和沉淀的问题，项目结项，可以把可复用的模块提交至模块云
- 模块预览：用于调试单个模块，同时解决了团队协作问题，项目分成多个独立模块，各负其责
- 模块列表

```
jdf widget -install ui-product-list
```

```
jdf widget -create ui-product-list
```

```
jdf widget -publish ui-product-list
```

```
jdf widget -preview ui-product-list
```

```
jdf widget -list
```

# 模块

- 模块云平台
  - 搭建成本极低，只需一台FTP服务器
  - 通过FTP服务器储存，下载，以及用户权限分配
  - 版本管理依赖于componet.json中的version
  - 使用方便，可以使用命令行工具进行发布，下载



# 编译

- vm编译
- SCSS编译
- 模块编译

```
1  <div class="ui-product-list">
2    <ul>
3      <li>
4        <div class="p-name">苹果6</div>
5        <div class="p-price">5000</div>
6      </li>
7      <li>
8        <div class="p-name">鞋</div>
9        <div class="p-price">100</div>
10     </li>
11     <li>
12       <div class="p-name">上衣</div>
13       <div class="p-price">暂无价格</div>
14     </li>
15   </ul>
16 </div>
```

# 编译

- vm编译
- SCSS编译
- 模块编译

```
1  .ui-product-list {  
2    -webkit-border-radius: 2px;  
3    -moz-border-radius: 2px;  
4    border-radius: 2px;  
5    color: #333;  
6    background: url(i/ui-product-list.png) no-  
        repeat;  
7  }
```

# 编译

- vm编译
- SCSS编译
- 模块编译

```
16 {%widget name="ui-product-list"%}
```

```
1  <link type="text/css" rel="stylesheet" href="
  /widget/ui-product-list/ui-product-list.css"
  source="widget"/>
2  <script type="text/javascript" src="/widget/ui
  -product-list/ui-product-list.js" source="
  widget"></script>
3
4  <div class="ui-product-list">
5      <ul>
6          <li>
7              <div class="p-name">苹果6</div>
8              <div class="p-price">5000</div>
9          </li>
10         <li>
11             <div class="p-name">鞋</div>
12             <div class="p-price">100</div>
13         </li>
14         <li>
15             <div class="p-name">上衣</div>
16             <div class="p-price">暂无价格</div>
17         </li>
18     </ul>
19 </div>
```



■

- 静态资源加CDN前缀
- 非覆盖式发布，静态资源采用版本号管理（递增）
- CMD模块自动提取文件id和dependencies
- 字符编码统一化

```
1  /* product-1.0.0 ui-product-list.css Date:2014
   -10-10 21:04:56 */
2  .ui-product-list {
3      -webkit-border-radius: 2px;
4      -moz-border-radius: 2px;
5      border-radius: 2px;
6      color: #333;
7      background: url(http://misc.360buyimg.
                       com/product/1.0.0/widget/ui-product-
                       list/i/ui-product-list.png) no-repeat; }
```

```
13 <link type="text/css" rel="stylesheet" href=
    "http://misc.360buyimg.com/product/1.0.
    0??/widget/ui-product-list/ui-product-list.
    css,/widget/header/header.css" source="widget
    "/>
14 <script type="text/javascript" src="
    http://misc.360buyimg.com/product/1.0.
    0??/widget/ui-product-list/ui-product-list.
    js,/widget/header/header.js"></script>
```

# 输出

- 性能优化
  - 压缩
    - js: Uglify-js
    - css: Clean-css
    - png: pngquant
  - 合并
    - js/css
    - css sprite
  - combo
    - <http://misc.360buyimg.com/??a.js,b.js>

# 联调

- 一键上传js/css/widget/images至联调服务器
- html上传至联调/预览服务器



# 上线

- 开发的静态资源提交至svn
- 编译系统从svn中取出代码进行编译
- 发布至CDN服务器

# 辅助

- 本地serve
- 错误提示
- 自动刷新
- 自动监听
- 代码质量检查
  - jslint 、 csslint、 htmlhint
- 根据项目配置文件，按选项统一编译、输出
- 字符编码统一化

# 工具配置

```
48
49   "build":{
50     "jsPlace": "insertBody", //调试时js文件位置 insertHead|insertBody
51     "widgetIncludeComment":true,//widget引用带注释
52     "livereload":true, //是否开启livereload
53     "sass":true,//是否开启sass编译
54     "less":true,//是否开启less编译
55     "csslint":false//是否开启csslint
56   },
57
58   "output":{
59     "concat":{},//文件合并
60
61     "cssImagesUrlReplace": true,//css中图片url加cdn替换
62     "jsPlace": "insertBody", //编译后js文件位置 insertHead|insertBody
63     "cssCombo": true, //css进行combo
64     "jsCombo": true, //js进行combo todo
65
66     "combineWidgetCss":false,//合并所有引用的widget中的css
67     "combineWidgetJs":false,//合并所有引用的widget中的js
68
69     "hasBanner": true, //压缩后的js和css是否有banner前缀
70     "vm":true, //是否开启vm编译
71     "compressJs":true,//是否开启压缩js文件
72     "compressCss":true,//是否开启压缩css文件
73     "compressPng":true,//是否开启压缩png图片
74
75     "cssSprite":true, //是否开启css sprite功能
76     "cssSpriteMargin":10, //css sprite图片之间的间距
77     "imagesSuffix":false,//图片后缀
78
79     "jsRemove":[]//移除js中函数或者方法,比如console,y.log即配置为['console','y.log']
80   },
```

# 工具命令

jdf init

jdf build

jdf release

jdf upload

jdf output

jdf csslint

.....

jdf widget

jdf compress

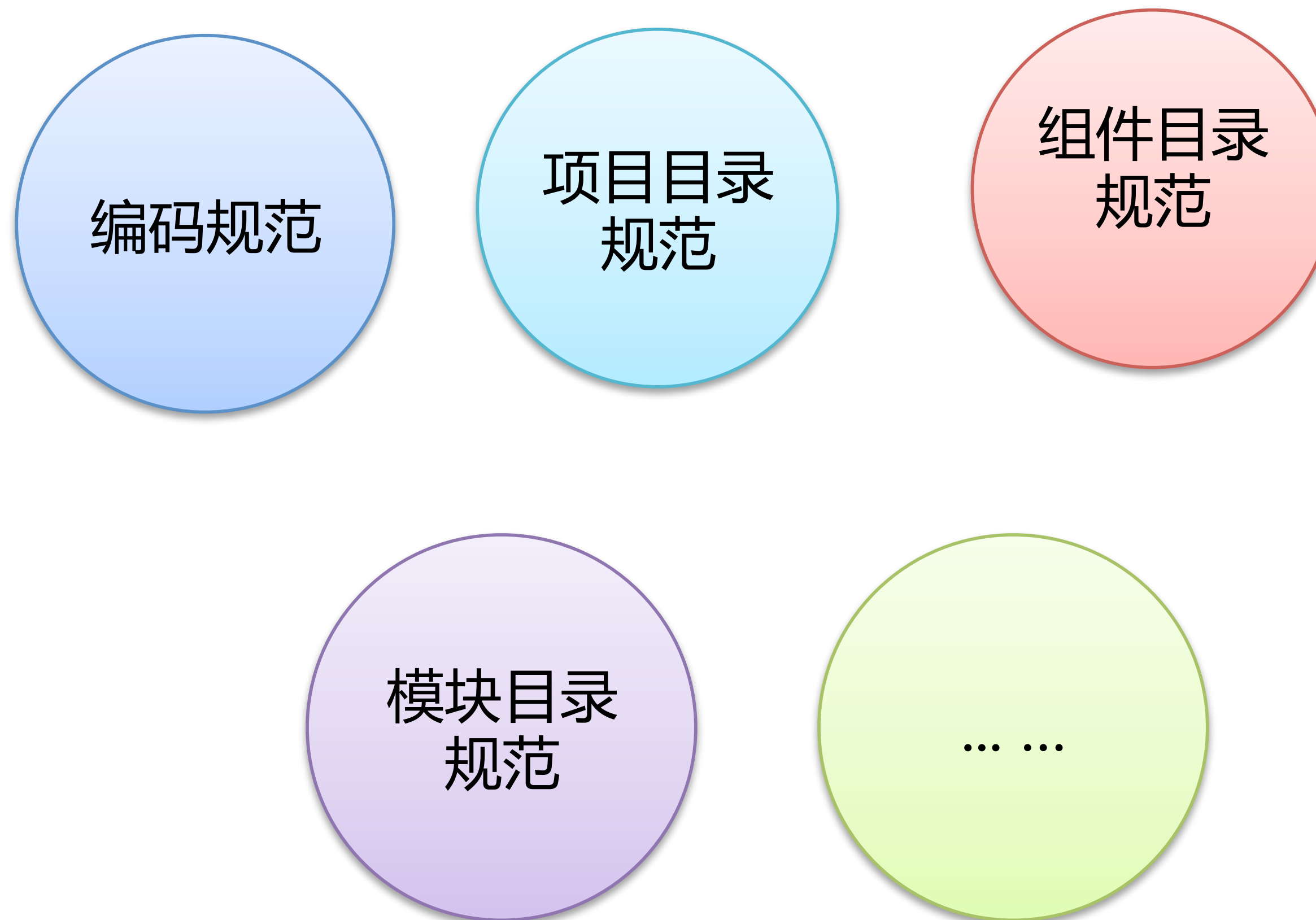
jdf htmlint

# 三、前端基础设施

# 前端框架

- 通用库
    - js: jQuery/seajs
    - css: reset/scss基类
  - 组件库
    - js: ui组件、业务组件
    - css: 元件、静态组件
  - 模块库
    - js/css/vm/images/json
    - 来源于通用和组件库
    - 分为本地和云端
- .....
- 模块化
    - js: seajs
    - CSS: SCSS

# 前端规范





# 前端文档





# 组件demo

[Home](#)[demo](#)[doc](#)[changelog](#)[api](#)


## demo

手风琴	动画	日历	旋转木马	吸顶灯	倒计时	计时器	对话框	拖拽
下拉菜单	位置固定	焦点图	美化表单	返回顶部	图片切换	懒惰加载	跑马灯	数字滚动
分页	placeholder	滑块	模拟滚动条	分享到	slider	老虎机	suggestion	switchable
tab	tips	abtest	category	商品对比	event	关注商品	脚印	login
myjd	到货通知	search	购物车					

### 动画

- 动画
- animate css3
- animate css3 工具
- 自定义批量动画
- 缓动函数实例
- 缓动函数工具
- 翻动动画

#### 自定义动画



[新窗口打开](#)  
[API](#)

# 组件API

**>>Home**

**>>demo**

**accordion**

**anim**

**calendar**

- Date.isLeapYear
- Date.getDaysInMonth
- Date.getDaysInMonthArray
- Date.getMonthName
- Date.getDayName
- Date.addMonths
- Date.reduceMonths
- Date.addYears
- Date.reduceYears
- Date.format
- Date.getMonthFirstDay
- Date.getDaysInPrevMonth
- Date.getMonthArray
- Date.getNextMonth
- Date.getPrevMonth
- Date.getCurrentMonth
- Date.getFestival
- rander
- show
- hide

## jdf组件库API

version: 1.0.0 By@jd fe



jdf组件库包括UI组件和Unit组件，UI组件是基于jQuery的轻量级UI组件库，符合jquery ui组件规范组件提供了一些必要的公共方法，同时兼容IE6-IE10，chrome，firefox，safari等常见浏览器

Created time: 2014-10-13 15:54:28

## accordion

### accordion

#### Demo

[accordion](#)

#### 参数

- `itemClass` {String} 'ui-accordion-item' item的className

# 任务系统

前端任务系统

首页 | 团队成员 | 列表视图 | 团队视图 | +新增项目 |

团队视图

经理

网站导航

用户交易

POP业务

新产品支持

运营支持

前端架构

序号	类型	项目	前端负责人	产品经理	开始时间	上线时间	预计工时(人天)	进度	状态	备注
6	新项目	jdf组件-地区	巫耀恒	刘威	2014-09-29	2014-10-09	8	80%	进行中	

前端任务系统@2014 v1.0.1014



前端框架

前端规范

组件demo

开发流程

模块云

命令行工具

.....

组件api

任务系统

## 四、未来展望

# 未来展望

- 丰富模块云平台上模块数量，形成前端页面构建平台
- 命令行工具（JDF）对Mobile，WebApp个性化支持
- 完善前端基础设施
- ...

QA

谢谢

