

十月份规划前端开发流程草稿

——By Hgwn 2018/9/13

Why 目标：

打造一个适合/实用/半自动化前端开发流程，以期提高整个 DL 前端团队开发效率，节省项目开发周期和降低开发人员成本，同时希望整个项目实施阶段的开发节奏可控，可快速响应公司各个项目紧、快，小步快步的项目需求。

里程碑：

- 1、10.1~10.30 规范整个 DL 前端开发流程/规范 （开发规范/开发流程）
- 2、争取 11.1~11.30 落地试用推行 （针对问题逐步完善阶段）
- 3、12.1 项目组落地执行

前端工程化构建解决方案：

根据公司项目实际情况，量身制作一套，约定开发规范和流程，利用工具提高开发效率，降级后期项目维护成本，提升用户产品满意度（web 性能）。

什么是前端工程化

推荐阅读：

<http://www.cnblogs.com/ihardcoder/p/5378290.html>

上图：

2. 前端工程化面临的问题

要解决前端工程化的问题，可以从两个角度入手：开发和部署。

从开发角度，要解决的问题包括：

1. 提高开发生产效率；
2. 降低维护难度。

这两个问题的解决方案有两点：

1. 制定开发规范，提高团队协作能力；
2. 分治。软件工程中有个很重要的概念叫做模块化开发其中心思想就是分治。

从部署角度，要解决的问题主要是资源管理，包括：

1. 代码审查；
2. 压缩打包；
3. 增量更新；
4. 单元测试；

要解决上述问题，需要引入构建/编译阶段。

2.2 模块/组件化开发

2.2.1 模块还是组件?

很多人会混淆模块化开发和组件化开发。但是严格来讲，组件（component）和模块（module）应该是两个不同的概念。两者的区别主要在**颗粒度**方面。《Documenting Software Architectures》一书中对于component和module的解释如下：

“

A module tends to refer first and foremost to a design-time entity. ... information hiding as the criterion for allocating responsibility to a module.

A component tends to refer to a runtime entity. ... The emphasis is clearly on the finished product and not on the design considerations that went into it.

“

In short, a module suggests encapsulation properties, with less emphasis on the delivery medium and what goes on at runtime. Not so with components. A delivered binary maintains its "separateness" throughout execution. A component suggests independently deployed units of software with no visibility into the development process.

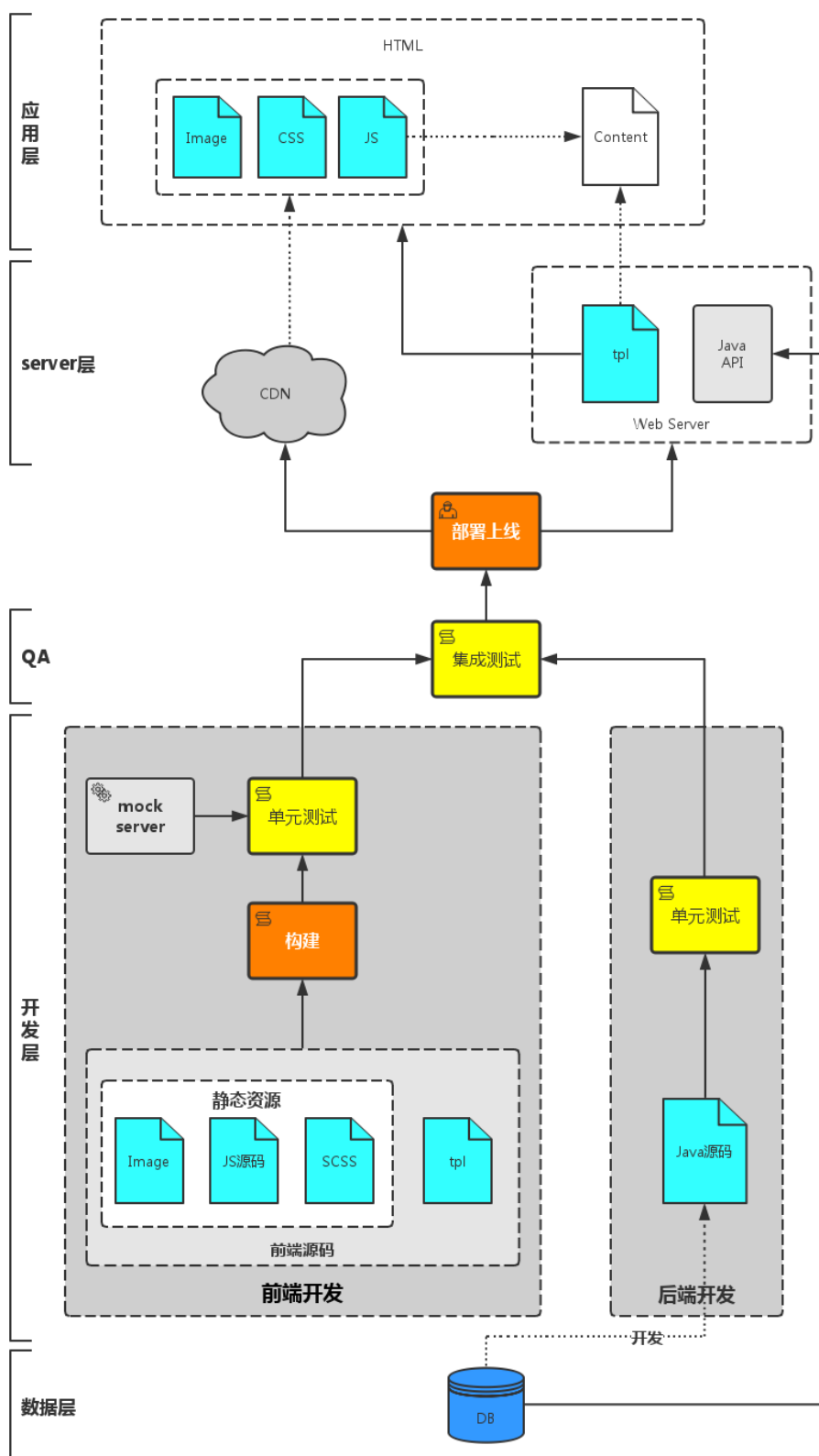
简单讲，module侧重的是对属性的封装，重心是在设计和开发阶段，不关注runtime的逻辑。module是一个白盒；而component是一个可以独立部署的软件单元，面向的是runtime，侧重于产品的功能性。component是一个黑盒，内部的逻辑是不可见的。

用通俗的话讲，模块可以理解为零件，比如轮胎上的螺丝钉；而组件则是轮胎，是具备某项完整功能的一个整体。具体到前端领域，一个button是一个模块，一个包括多个button的nav是一个组件。

3.1 构建在前端工程中的角色

在讨论具体如何组织构建任务之前，我们首先探讨一下在整个前端工程系统中，构建阶段扮演的是什么角色。

首先，我们看看目前这个时间点（2016年），一个典型的web前后端协作模式是什么样的。请看下图：



开发流程：

构建工具

Webpack/Node.js

前端代码管理工具：Git

前端共享代码库：GitHub / 码云 ~待定

前端远程办公工具：TeamViewer 13

第三方工具或平台：

字体图标：Iconfont-阿里巴巴矢量图标库 <http://www.iconfont.cn/>

项目文档汇总：蓝湖

<https://lanhuapp.com/web/#/user/login>

性能测试：OneAPM

<https://www.oneapm.com/index.html>

接口测试：postman 工具 或 研发一款 H5 web Api

前端代码规范：

以参考帝隆供应链金融项目为例。

举个例子

例如：

http://192.168.16.16:86/#/loanDetail_bl/10000071220147

020-3891 1638
9:00 - 18:00

首页我要借款我的借款关于我们

1342****799退出登录

申请时间 2018-09-11

我的信贷

我的保理

个人信息

银行卡信息

实名认证

修改密码

借款金额

Loan amount

¥10086

审核状态

Audit state

审批中

客户经理

电子合同

借款主体	保理费率	起息日	借款期限	银行账户
李鸿文	%	2018-09-06	3%月	工商银行(6000)
证件号	保理服务费率	结息日	还款方式	买方名称
4408*****4932	%	2018-12-05	一次性到期还本...	阳光城集团

还款计划

提前还款

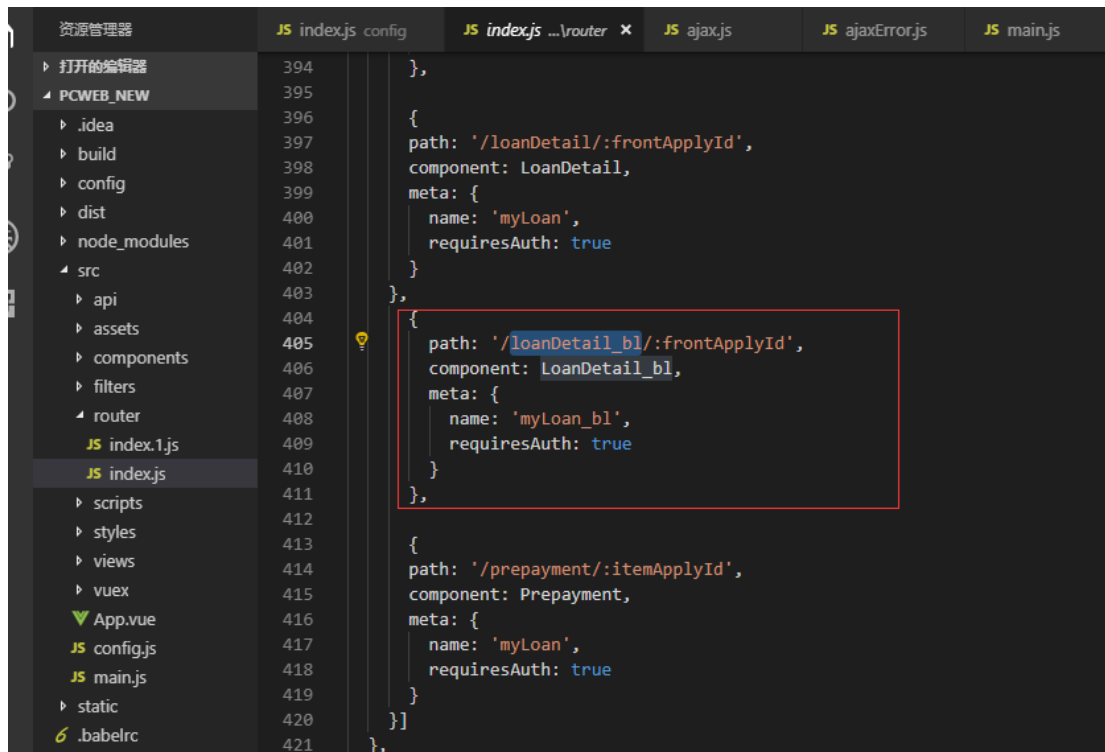
期数	本金	保理费	保理服务费	计划还款日期	计划还款金额	未还金额	状态
暂无数据							

凭证列表

买方	卖方	发票合同金额	应收账款金额	状态	应收账款结束日期
阳光城集团	李鸿文	¥6,000.00	¥5,000.00	已融资	2018-10-11

< 1 >

© 2017 广州帝隆科技股份有限公司 All rights reserved | 粤ICP备18000346号



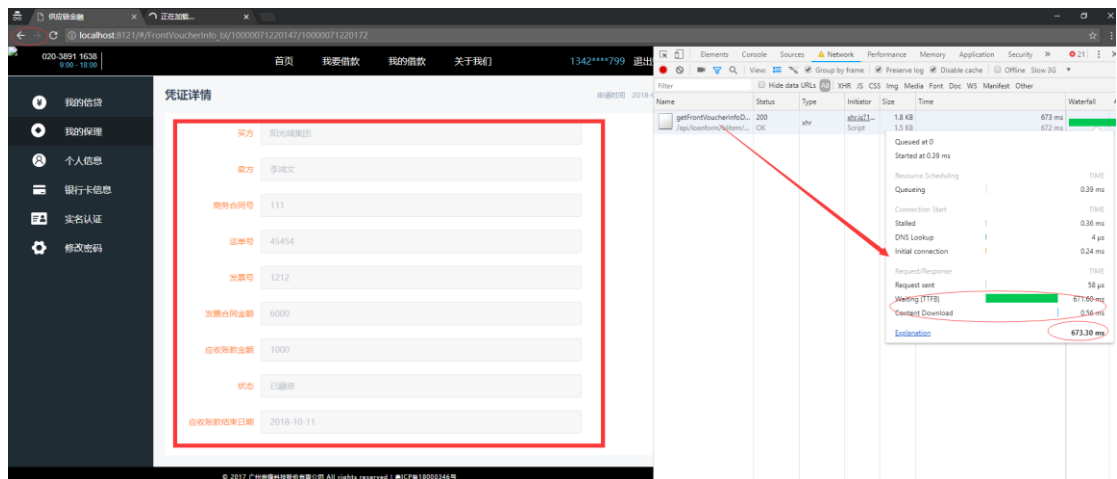
处理 ajax 跨域问题

1. webpack 下 config\index.js

E:\myjob_new\9m_zsgyl_baoli\code\PcWeb_new\config\index.js

```
index.js ...\router  JS index.js config x  <> index.html  JS ajax.js  JS ajaxError.js
22     bundleAnalyzerReport: process.env.npm_config_report
23   },
24   dev: {
25     env: require('./dev.env'),
26     port: 8121,
27     autoOpenBrowser: true,
28     assetsSubDirectory: 'static',
29     assetsPublicPath: '/',
30     proxyTable: {
31       '/api': {
32         //target: 'http://192.168.16.35:8191', // 电票35测试--818
33         //target: 'http://192.168.16.35:8292', // 保理测试环境 --8/27
34         target: 'http://192.168.16.16:8292', // 保理开发环境 --9/13
35         //target: 'http://192.168.16.62:8080/gyljr', //huarong --9/12
36         //target: 'http://192.168.16.191:8088/dl_gyljr', //yongjin --9/13
37         changeOrigin: true,
38         pathRewrite: {
39           '^/api': '/'
40         }
41       }
42     },
43     // CSS Sourcemaps off by default because relative paths are "buggy"
44     // with this option, according to the CSS-Loader README
45     // (https://github.com/webpack/css-loader#sourcemaps)
46     // In our experience, they generally work as expected,
47     // just be aware of this issue when enabling this option.
48     cssSourceMap: true
49   }
```

举个栗子：



前端代码：

1. 路由

```

65 //我的保理
66 import MyLoan_b1 from '../components/personalCenter/MyLoan_b1.vue' //我的保理
67 import LoanLists_b1 from '../components/personalCenter/LoanLists_b1.vue' //我的保理列表
68 import LoanDetail_b1 from '../components/personalCenter/LoanDetail_b1.vue' //保理详情
69 import RepaymentInfo_b1 from '../components/personalCenter/RepaymentInfo_b1.vue' //还款计划详情
70 import FrontVoucherInfo_b1 from '../components/personalCenter/FrontVoucherInfo_b1.vue' //凭证列表--凭证详情接口

```

```

378 },
379 {
380   path: '/repaymentInfo_b1/:repayPeriod/:financePayId', // 我的保理列表--还款计划--还款详情
381   component: RepaymentInfo_b1,
382   meta: {
383     name: 'myLoan_b1',
384     requiresAuth: true
385   }
386 },
387 {
388   path: '/FrontVoucherInfo_b1/:frontApplyId/:voucherInfoId', // 我的保理列表--凭证详情列表----凭证详情
389   component: FrontVoucherInfo_b1,
390   meta: {
391     name: 'myLoan_b1',
392     requiresAuth: true
393   }
394 },

```

2. 组件

JS index.jsFrontVoucherInfo_bl.vueLoanDetail_blvue ×JS ajax.js

```
1 <template>
2   <section> ...
147 </section>
148 </template>
149 <script>
150   import { mapGetters, mapActions, mapMutations } from 'vuex'
151   import * as types from '../vuex/types'
152   import loanApi from '../api/myloan_bl'
153   export default {
154     data () {
155       return { ...
200     },
201     },
202     computed: { ...
230   },
231     methods: { ...
419   },
420     mounted () {
421       //this.getData()
422       this.getInfo();
423       this.getRepayList();
424       this.getFrontVoucherInfoList();
425     }
426   }
427 </script>
428 <style lang="scss"> ...
777
778
```

```

101  },
102  getInfo(){
103      loanApi.getFrontVoucherInfoDetails({ //详情
104          frontApplyId: this.$route.params.frontApplyId,
105          tokenId: localStorage.getItem("dl_tokenId_b1"),
106          voucherInfoId: this.$route.params.voucherInfoId
107      }).then((res)=>{
108          if(res.rspcode ==1){
109              this.infoData = res.data.voucherInfo;
110              if(res.data.applyDate){
111                  this.applyDate = res.data.applyDate;
112              }
113              if(res.data.voucherInfo.accountsReceivableEndTime){
114                  this.infoData.accountsReceivableEndTime = this.formTime(res.data.voucherInfo.accountsReceivableEndTime)
115              }
116          }
117      }).catch((error)=>{
118      })
119  })
120  },
121  },
122  formTime(value){
123      var time=new Date(value);
124      var year =time.getFullYear();
125      var month=time.getMonth()+1;
126      var day=time.getDate();
127      var h = time.getHours();
128      var mm = time.getMinutes();
129      if(month<=9){
130          month="0"+month;
131      }
132      if(day<=9){
133          day="0"+day;
134      }
135      if(h<=9){
136          h="0"+h;
137      }
138      if(mm<=9){
139          mm="0"+mm;
140      }
141      return year+"-"+month+"-"+day;
142  }
143  },
144  mounted () {
145      this.getInfo()
146  }
147  }
148  </script>
149  <style lang="scss">
150  #frontVoucherInfoBox .el-input{ width: 80%;}

```

优化: tokenId全局处理, 放在header

3. Api

```

JS index.js  FrontVoucherInfo_bl.vue JS myloan_bl.js x JS ajax.js JS ajaxError.js JS main.js
1  /*
2  Notes:我的保理Api接口
3  PS:
4  Author:Hgwn
5  Date:2018/9/4
6  */
7  import ajax from './ajax';
8
9  export default {
10     getMyBlItemInfoCount:function(params){ //获取用户看板数据
11         return ajax.get('/loanform/blitem/blLoanItemFrontApply/getMyBlItemInfoCountNew.ht',params); // 9/12
12     },
13     getItemApplyList:function(params){
14         return ajax.get('/loanform/blitem/blLoanItemFrontApply/getItemApplyListNew.ht',params); // 9/12
15     },
16
17     delFrontItem:function(params){
18         return ajax.postForm('/loanform/blitem/blLoanItemFrontApply/delFrontItem.ht',params);
19     },
20
21     getItemDetails:function (params) { //项目详情接口
22         return ajax.postForm('/loanform/blitem/blLoanItemFrontApply/getItemDetails.ht',params);
23     },
24     getRepayList:function (params) { //项目详情还款计划接口
25         return ajax.postForm('/loanform/blfinance/blLoanItemFinanceRepay/getRepayList.ht',params);
26     },
27     getRepayDetails:function (parmas) { //还款计划详情
28         return ajax.postForm('/loanform/blfinance/blLoanItemFinanceRepay/getRepayDetails.ht',parmas)
29     },
30     getFrontVoucherInfoList:function (params) { //凭证列表接口
31         return ajax.postForm('/loanform/blitem/blLoanAccountVoucherInfo/getFrontVoucherInfoList.ht',params);
32     },
33     getFrontVoucherInfoDetails:function (params) { //凭证详情
34         return ajax.postForm('/loanform/blitem/blLoanAccountVoucherInfo/getFrontVoucherInfoDetails.ht',params);
35     },
36     //项目删除接口
37     delFrontItem:function(params){
38         return ajax.postForm('/loanform/blitem/blLoanItemFrontApply/delFrontItem.ht',params);
39     },
40     showContract:function (params) { // 下载电子合同
41         return ajax.postForm('/mobile/item/pdfSignatureController/blReadContract.ht',params)
42     }
43 };
44

```

4. 全局 Ajax.js, 基于 vue axios 进行二次封装

```

index.js  FrontVoucherInfo_bl.vue  JS myloan_bl.js  JS ajax.js  JS ajaxError.js  JS main.js
1  /*
2  Notes:二次封装axios常用get、post、put、delete请求
3  PS: 注意区分postForm 表示 Ajax Form提交
4  Date:2018/8/30 by Hgwn
5  */
6
7  import axios from 'axios'; // 引入axios ajax http请求插件
8  import config from '../config'; // 引用全局js配置, 如 ajaxBaseUrl
9
10 axios.defaults.baseURL = config.ajaxBaseUrl;
11 axios.defaults.timeout = 15000;
12
13 // 全局响应处理放这里
14 axios.interceptors.response.use((res) => {
15   if (Number(res.data.code) === 999) {
16     //debugger;
17     // store.dispatch(types.SET_FAIL_MESSAGE, {
18     //   show: true,
19     //   content: res.data.result
20     // })
21     // store.commit(types.SET_USER, {})
22     // router.push('/login')
23   }
24   return res
25 })
26
27
28 var getCookie=function(key){
29   var getCookie = document.cookie.replace(/[/g,""); //获取cookie, 并且将获得的cookie格式化, 去掉空格字符
30   var arrCookie = getCookie.split(";"); //将获得的cookie以"分号"为标识 将cookie保存到arrCookie的数组中
31   var tips; //声明变量tips
32   for(var i=0;i<arrCookie.length;i++){ //使用for循环查找cookie中的tips变量
33     var arr=arrCookie[i].split("="); //将单条cookie用"等号"为标识, 将单条cookie保存为arr数组
34     if(key==arr[0]){ //匹配变量名称, 其中arr[0]是指的cookie名称, 如果该条变量为tips则执行判断语句中的赋值操作
35       tips=arr[1]; //将cookie的值赋给变量tips
36       break; //终止for循环遍历
37     }
38   }
39   return tips;
40 };
41
42 var ajaxToken = function(){
43   var token = getCookie("UserLoginToken");
44   if(token){
45     axios.defaults.headers.common['Token'] = token; //自定义全局请求头 Token
46   }
47   // var user = JSON.parse(sessionStorage.getItem("user")); //判断是否需要请求Token
48   // if(user){
49   //   axios.defaults.headers.common['Token'] = user.token; //自定义全局请求头 Token
50   // }

```

```

5 index.js      FrontVoucherInfo_bl.vue  JS myloan_bl.js  JS ajax.js  X JS ajaxError.js  JS main.js
76 // });
77
78 export default {
79   get:function(url,params){
80     ajaxToken()
81     return axios.get(url,{params: params}).then(res => res.data);
82   },
83   post:function(url,params){
84     ajaxToken()
85     return axios.post(`${config.ajaxBaseUrl}`+url, params).then(res => res.data);
86   },
87   postForm:function(url,params){ // postForm 表示 Ajax Form提交
88     ajaxToken()
89     return axios({url: url, method: 'post', data: params,
90       transformRequest: [function (data) {
91         let ret = ''
92         for (let it in data) {
93           ret += encodeURIComponent(it) + '=' + encodeURIComponent(data[it]) + '&'
94         }
95         return ret
96       }],
97       headers: {
98         'Content-Type': 'application/x-www-form-urlencoded'
99       }
100     }).then(res => res.data);
101   },
102   put:function(url,params){
103     ajaxToken()
104     return axios.put(url, params).then(res => res.data);
105   },
106   putForm:function(url,params){
107     ajaxToken()
108     return axios({url: url, method: 'put', data: params,
109       transformRequest: [function (data) {
110         let ret = ''
111         for (let it in data) {
112           ret += encodeURIComponent(it) + '=' + encodeURIComponent(data[it]) + '&'
113         }
114         return ret
115       }],
116       headers: {
117         'Content-Type': 'application/x-www-form-urlencoded'
118       }
119     }).then(res => res.data);
120   },
121   delete:function(url,params){
122     ajaxToken()
123     return axios.delete(`${config.ajaxBaseUrl}`+url, params).then(res => res.data)
124   },
125   deleteForm:function(url,params){

```