

一、埋点: fe-labs/luker

1. pv

willmount 发送埋点请求使用 view.use 的中间件

2. 页面点击行为

重写 `createElement` 劫持 `onclick checked` 事件

3. Request send receive 添加drucation 埋点, 为了查用户请求

Request error 请求埋点

页面js异常埋点, promsie 异常未catch时的捕获

2, 3 使用全局的方式

use 全局即可使用

所有埋点放到plugin 中

二、

pexs 中的bind, 解析bind="a.b.c" 采用正则处理

Vue 中的 v-if 表达式解析使用with 或new Function实现

`new Function ([arg1[, arg2[, ...argN]],] functionBody)`

`dangerouslySetInnerHTML` 会导致xss注入

React 在渲染的时候, 会将用户的输入的内容转化后渲染的, 不会导致xss注入

三、

Js 页面导出pdf:

1. html2canvas (可以通过纯JS对浏览器端进行截屏)

2. Jspdf (将canvas 插入到pdf里)

四、

开发 ES6/7 新特性的库:

`babel-plugin-transform-runtime` // 可用`async await` `generator`

`babel-runtime`

五、

删除xcode 之后 git 会出错

```
sudo xcode-select -r
```

```
xcode-select -p
```

`xcode-select`命令来重置系统默认的`CommandLineTools`路径

```
sudo xcode-select --switch /Library/Developer/CommandLineTools
```

六、正则表达式：

\1 取捕获的结果，从正则中取

[] 匹配一个

() 分组

/(\w)\1*/g 或 /(.)\1+/g 可将 'abcaakjbb' =>['aa', 'bb'] 'abcaakjbb'.match(/(.)\1+/g)

七、input 原生事件

compositionstart:

compositionupdate: a

compositionupdate: a k

compositionupdate: 爱看

compositionend: 爱看

处理输入中文，在compositionend时处罚事件

八、Immer proxy

Immutable

Elm

Electron 桌面(pc)应用

ssr 框架：

beidou 国内经验过的高并发

react-component-caching

react native 使用console.log 卡顿

1.dev 环境 使用的浏览器的console

2.beta prod native自己实现console

QUIC 协议

幕布 脑图， MECE 归纳演绎 结论 理由 事实

主动学习 - - - - - 实践 不要放过不会的东西

技术成长：

言语和框架

如何写好代码

打开 electorn控制台，这个设置是在btalk中设置的：
open dev tools: Command + Option + Shift + I

项目中RN包所在目录：node_modules/react-native/Libraries/RNComponents
node_modules/react-native/Libraries/RNComponents/libs/ImageManager/
RNImageManager.js

RN 上打开一个文档：

```
import { NativeModules } from 'react-native'  
Const { ReadDocumentModule } = NativeModules  
ReadDocumentModule.readDocument(url);
```

React Native 原理：

- 1.使用javascript编写代码； native内置js引擎，解析js代码
install native function; c++ 跟js的桥梁
Evaluate Script 执行一段js代码
Call as function c++ 可以调用js代码 js 和native 通讯关键点：
callFunctionReturnFlushedQuene
js 引擎可以有多个实例
- 2.提供操作Native View的接口；
createView
updateView
setChildren
onBatchComplete native view的测量，最后render （1s 渲染60帧）
RootView 对应到客户端一个view

1. invoke addView()
 - 2.onBatchComplete
 - 3.FlexBox
 - 4.Vsync Signal 同步帧信号
 - 5.Render
- 3.提供扩展功能（异步&队列 setTimeout的实现 View渲染的实现 触摸事件的实现）
Js 引擎 与 Native 通过 NativeMoudle JavascriptModule 通过序列化及反序列号来调用方法

RNX启动过程：

- 1.实例化 Native Module 、 Native Module
- 2.初始化 JS引擎 挂载bridge function module spec
- 3.在app内部加载平台包
- 4.从离线包加载业务包
- 4.实例化ReactRootView
- 6渲染含有<Navigtor>的router

离线包更新机制：

- 1.每5分钟检查一次更新
- 2.如果app 冷启动，替换新版本
- 3.如果离线包未使用，替换新版本
- 4.如果在后端超过5m，替换新版本

React native & Scheme

VC含有：

iOS: view controll

Android: Activity

native 下 VC 会出现多个，
react native 只有一个VC

1.componentDidMount 中不能写异步， onDidFoucs 页面切换，有切换动画，是通过setTimeout实现，会出现卡
在ready， actived 生命周期中执行异步

2.PView, PComponent 包装redux属性

3.router.js 配置全局属性，如果页面切换记录日志，关闭之前页面的弹框

4.书写style

直接使用<Flex>包裹

style文件 有已经书写好的

5. hitSlop 撑开热点区

6. 元素不能超过外容器范围， 不能使用负值 top: -30px

7. A 打开B, A 中有带上个页面的参数请求，从B 回到A，这个问题如何解决

8.IOS 打开文件时， url 需要使用 encodeURIComponent(url) 转化

```
import { NativeModules } from 'react-native';
```

```
const { ReadDocumentModule } = NativeModules;
```

```
export default class DocReader {
```

```
  static openFile = url =>
```

```
  ReadDocumentModule.readDocument( encodeURIComponent(url));
```

```
}
```

URL Scheme

1. 概念：URL Scheme 是一种内部跳转协议，通过定义自己的scheme协议，可非常方便跳转app中的各个页面；通过scheme协议，服务器可以定制化告诉App

跳转那个页面，可以通过通知栏消息定制化跳转页面，可以通过H5页面跳转页面等

2. 应用场景：

- a. 服务器下发跳转路径，客户端根据服务器下发跳转路径跳转相应的页面
- b. H5页面点击锚点，根据锚点具体跳转路径APP端跳转具体的页面
- c. APP端收到服务器端下发的PUSH通知栏消息，根据消息的点击跳转路径跳转相关页面
- d. APP根据URL跳转到另外一个APP指定页面

3. URL Scheme 协议格式

`opengs://tasks:8088/tasksDetail?taskId=102`

1. opens 代表该Scheme协议名称
2. tasks代表Scheme作用于哪个地址域
3. taskDetail 代表Scheme指定的页面
4. taskId代表传递的参数
5. 8088代表该路径的端口号

RxJS

参考文档 <https://rxjs-cn.github.io/learn-rxjs/operators/operators/transformation/mergemap.html>

1. concatMap 内部 observables 要求发送和订阅的顺序
2. mergeMap 当想要打平内部 observable 并手动控制内部订阅数量 (mergeMap 允许同一时间存在多个活动的内部订阅, 故常用到 不会被取消的请求)
3. switchMp 如果同一时间应该只有一个内部 subscription 是有效的
4. forkJoin 相当于Promsie.all 功能

脚手架：

nrm 管理安装依赖源 可是taobao源

优化点：

1. 可选择框架版本
2. rbac 作为插件使用

静态模版处理

替换package.json , config 中{{ }} 表示的属性：

一、metalsmith 库，静态网站生成器 作用：

读取目标目录所有文件

调用一系列操作文件的插件

将结果回写到目标目录

=== 利用他实例的use插件方法

二、consolidate：支持各种模板引擎的渲染； 可直接用 handlebars 或mustache 模版库

1. consolidate.handlebars.render // 替换属性并重新渲染 默认占位符是 {{projectName}} 不可替换
2. consolidate.mustache.render // 替换属性并重新渲染 默认占位符是 {{projectName}}, 可自定义占位符，如：const Mustache = require('mustache'); Mustache.tags = ['<%', '%>']

三、multimatch （可以支持多个条件的匹配。）

!/usr/bin/env node 调用系统的node解析脚本

```
git clone --depth 1 git@git.corp.bianlifeng.com:fe/device-center.git -b device-center-plugin device-center
// clone 指定分支到某个目录，且最近一个commit，不包含整个history
```

```
// 从某个commit创建分支
git checkout -b [branch name] [commitId]
```

直接 git clone #{branch} release 分支，到临时文件
os.tmpdir() (node 本身自带)

用户输入：

sed shell 脚本替换 项目名字 package.json 和 config 项目名称

Shell 脚本或node 拷贝文件到当前项目

Shell替换文件，但需要回写

```
#!/bin/bash
set -x
project_name=$1
if [ ! $project_name ]; then
    # 删除原来代码树下的所有文件
    rm -rf "$project_name/.gitignore"
    # 替换项目中的Project_Name 为当前用户输入的project name
    project_path="$project_name/config/base.js"
    # sed "s/{{PROJECT_NAME}}/$project_name/g" $project_name
    sed "s/{{PROJECT_NAME}}/$project_name/g" $project_path
fi
```

蜂利器

1.新版本 **rn lineHeight** 是**float**， 以前是**int**， **transPxToDp** 会生成小数点，会导致**andriod** 壳子崩溃

2.字体库下载时，选择不增加版本，如果使用字体增加版本，字体注册时找不到，线上字体不会出来

beta离线包发布时打qp包是拉取的线上静态资源，但beta壳使用时，并没有用此qp包，而是使用秘密花园设置的静态资源

- 3. lineHeight 在IOS下不会达到预期效果，不建议使用
- 4. ListView 父元素需用使用flex:1, 否则 ListView 不能滚动

1.blibee-affair 可参考 MaterialCycle 目录下代码，引入mobx, rx

结构化思想：

David述职 下周一下午2点：

之前做过的事件 梳理逻辑 当场能回答提问 数据支持
选项目？？？

准备两个项目：

1.流程

能画出整个流程图

节点 流转实现及各规则（后端）

存着什么问题：

流程图不能在线展示： bpmn 发散

1.app 创建任务存在性能问题

2.扩展性强不太好控制，不好用

3.组件库不够强，不能支持 table —— 可讲现在做这个

2.资产收货/或其他项目

配置化 wpage

FD 基础组件支撑

设备

评估需求收益

数据埋点 （只要发埋点请求即可）

验证case

需要明确需求具体是要干什么的，要了解这个功能涉及到的业务

提测前自己需要review下

发现问题 - - - -> 改进问题 (怎么才能做的更好，才会有思考，才会有技术提升)

nginx配置文件直接下载下来，而不是直接打开

```
location / {  
    if ($request_filename ~* ^*?\. (txt|pdf|doc|xls)$){  
        add_header Content-Disposition attachment;  
    }  
}
```

appleId:

kyy010@163.com

kyyYU1234

安全问题：

少年朋友：彭元琴

理想工作：老师

父母在哪里认识：老家

