

# 禧云前端监控

## 一、需求背景

目前公司的前端项目均没有接入任何的监控系统，前端线上出现问题时，只能是由客户反馈-->技术支持上报-->前端研发修改，这样一个流程，线上产品出现问题了第一个发现的却不是研发，显得很被动。如果有一套前端监控系统支持的话，如果线上出现问题了，监控系统就会记录错误，然后直接报警给相关研发人员，发现问题-->报告问题-->解决问题的时间将会大大缩短，进一步提高了线上产品的稳定性。因此有必要为前端项目接入一套前端监控系统。

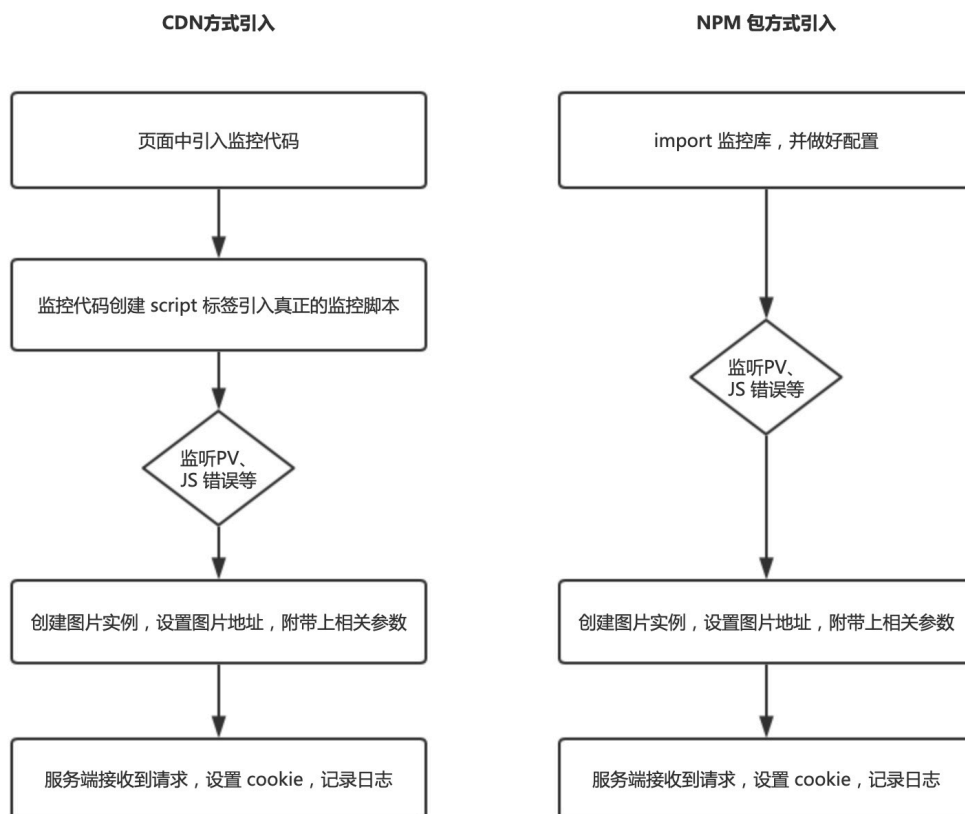
## 二、架构设计



前端监控首要工作就是获取数据源，大致可以从稳定性、用户体验和业务扩展来区分数据源。

## 三、技术调研

### 1、前端监控脚本流程



### 2、前端如何采集 JS 错误

JS 错误采集大致可以分为可以使用 `addEventListener` 采集的错误和不能使用它来采集的错误这两类。

#### 2.1 可以使用 `addEventListener` 采集的错误

##### 2.1.1 error 事件

一般可以触发 `error` 事件的错误有通用的 `Error`（包含自定义错误）以及其它 `InternalError`、

ReferenceError 等 7 种错误，这些都是 JS 本身运行时发生的错误。

还有一类是 Web API 中的一些 API，其中的一些异常也会触发 error 事件，比如使用 DOM API 在错误操作 DOM 元素时，就会产生一个 DOMException 异常，它会触发 error 事件。

最后是在网页中引入的一些资源错误，比如：img 标签的图片地址请求错误，script 资源错误，都可以触发 error 事件。但是要注意的是，监听代码必须要放到这些请求的前面。同时也需要做一下浏览器对跨域脚本的错误处理。

window.onerror 的用处，只要 js 出现错误，在浏览器控制台都会打印出错误信息，如果不想把这些错误信息打印出来，可以使用：

```
window.onerror = function () {return true}
```

就可以屏蔽掉错误输出。这是 addEventListener('error') 无法做到的。

## 2.1.2 unhandledrejection 事件

在使用 Promise 的时候，如果没有添加 catch，则 Promise 的异常会被抛出。只能使用 addEventListener('unhandledrejection') 才能捕获到该异常。

## 2.2 需要做入侵捕获的错误

这一类错误是无法用 addEventListener 来捕获到的，所以必须做入侵捕获。

### 2.2.1 劫持 console.error

有的框架会自己捕获错误和异常，然后交由 console.error 打印出来，这时我们通过 addEventListener 是无法收集到错误的。只能重写 console.error 加入监控逻辑，这样在使用它的时候就会采集到了。

### 2.2.2 劫持 XMLHttpRequest

XMLHttpRequest 出现请求错误的时候是属于用户侧需要处理的问题，所以不会被当成错误抛出，而是通过指定的方法输出这些错误信息，所以需要做入侵捕获。

使用 XMLHttpRequest 做请求的时候，有以下这几种情况需要收集：

- 1、接口访问失败，包括业务请求中因参数、权限等原因请求失败，400+错误和500+错误；
- 2、接口超时；
- 3、手动上报的跟业务相关的错误。

### 粗略收集

收集这些错误信息可以监听万能的 `readystatechange` 事件，只要请求状态发生变化，这个事件就会执行，一般来讲，如果接口发生错误，那么其 `readyState = 4` 且 `status = 0`，可以通过这对组合来粗略地判断接口是否有错。

### 精细收集

在 `XMLHttpRequest` 中还有 `error` 和 `timeout` 这两个事件，如果接口出错则会触发 `error` 事件，如果接口超时（前提是设定了超时时间）则会触发 `timeout` 事件。

当然，也可以只监听 `readystatechange` 事件，记录请求时间等必要信息，最后也可以判断出是什么类型的错误或者是超时。

## 2.2.3 劫持框架中的错误处理

有的前端框架中自己封装了错误处理函数，使用框架时产生的错误都会被错误函数捕获到，再由它统一输出错误信息。这样一来就无法使用 `addEventListener` 监听事件的方式来收集到错误信息了。必须重写它的错误处理函数才可以。

### 2.2.3.1 劫持 Vue 的错误处理函数

Vue 框架提供了一个错误处理函数，叫 `errorHandler`，默认情况下，它会由框架来接管，我们可以重写它，在函数里面加上监控代码：

```
Vue.config.errorHandler = function(error, vm, info) {  
  // ...前端监控代码  
};
```

### 2.2.3.2 Angular 中自定义错误处理

Angular 提供了一个 ErrorHandler 接口，可以按照如下方式自定义一个错误处理函数来收集错误信息：

```
class MyErrorHandler implements ErrorHandler {
  handleError(error) {
    // do something with the exception
  }
}

@NgModule({
  providers: [{provide: ErrorHandler, useClass: MyErrorHandler}]
})
class MyModule {}
```

经过调研，发现对 Angular 的错误处理进行劫持很复杂，需要对它非常熟悉才能做好。

## 2.3 能力边界

虽然我们可以采集到大部分页面中的错误，但也有一些是无法捕获到的。

- 1、用户使用 try...catch 时，自行吞咽错误却未抛出。
- 2、<link>，经测试，页面中引入样式文件，但是这个文件又不存在时，不会抛出任何错误。