创建 .cfg 配置文件bug：

1. Error parsing YAML config file: yaml-cpp: error at line 2, column 13: illegal map value

2. Unrecognized option: destination

解决方案：在所有的键值对的冒号之后加一个空格，例如 a: b；

并且不得使用tab空格键。有的话就删除，使用一个空格代替。

使用npm install 时的bug：

1. Bower : ENOGIT git is not installed or not in the PATH

解决方案：配置你的Git到path，也就是把你git的安装目录的bin以及cmd文件路

径添加到环境变量的PATH中。

1. npm ERR! errno -4048，Error: EPERM

解决方案：是缓存的问题，清理下缓存就行。

npm cache clean --force（cmd命令清除）

1. Npm err :npm ERR! code ELIFECYCLE

解决方案：在安装模块之前，运行一下npm run clean

vue使用sass时的bug：

1. <style lang="scss" scoped type="text/css"> 会报出 “{“ expected 。

解决方案：删除type=”text/css”。

1. Padding：20px auto; 不起作用。

解决方案：删除auto或者将auto改为0。

1. Vue修改data数据的方法。

解决方案：直接this.index = index。

在使用mint-UI的轮播组件时的bug：

1. Swipe不显示。

解决方案：给swipe加一个父元素，并设置这个父元素的高度。

使用git的bug：

1. Git 推送版本到远程仓库，副分支不能直接推送到主分支上。
2. Git stash 把当前所有新修改的部分都保存到另一个分支上（虚拟），还原未修改之 前的状态，可以通过git stash pop 将其拿回来。

使用axios的bug：

1. 返回的时候进入了then和catch。

主要是因为在then里面你又触发了axios的请求。

使用mintui遇到的bug：

1. 如果你的路径页面跳转不放在弹框执行完毕之后再执行的话，那么弹框会弹出不一 样的信息。

使用MongoDB的bug：

1. 很久没有使用MongoDB服务，导致服务停止运行，再次启动服务的时候提示路径不 正确；使用net start MongoDB启动服务报错。

解决方案：把根目录下的data文件夹清空，新建db和log文件夹；然后在MongoDB的安装目录下的bin目录下运行 mongod.exe --config “\*\*\*.cfg路径” --install --serviceName “MongoDB”；然后在运行 net start MongoDB。

在使用cross-env 改变环境的时候，没有改变成功：

1. 这是因为你没有安装 cross-env 。安装成功就好了。

在render 的return里面使用for循环组件：

1. 直接循环会报错，因为{}内不允许直接使用循环。

解决方案： 把for循环写在箭头函数里面，或者使用map()、forEach()。

注意，如果是for循环的话，需要返回一个数组（成员是节点）。

在使用任何的工具之前，我们最好仔细看看说明书。

在使用promise的时候，reject or resolve之后的语句还是会执行的。

在使用mysql的时候，query是异步函数。

关闭node.js ： process.exit();

TRUNCATE TABLE mine\_day\_statis 清空表数据并将index重置为0

在clone项目的时候，不要随意的移动clone的项目。

写vue的时候，代理不起作用：

1. 代理是 ‘/v1’，写成了 ‘v1’,解决方案： 写成‘/v1’。

ln -s 命令的坑：

ln -s Setuna.exe /usr/bin/ 报错失败：

ln: failed to create symbolic link '/usr/bin/Setuna.exe': No such file or directory。

ln -s Setuna.exe /usr/bin/Setuna.exe 报错失败：

ln: failed to create symbolic link '/usr/bin/Setuna.exe': No such file or directory

ln -s /d/is/setuna/Setuna.exe /usr/bin/ 成功。

解决方案：ln -s 绝对路径 绝对路径

MonthPicker 设置默认值时 月份显示是英文：

解决方案：moment.locale('zh-cn');

Pm2 开启服务之后，网络访问空白的bug：

解决方案：自定义启动文件的 name属性唯一。

Nginx Bug：

1. nginx: [emerg] CreateFile() "D:\installPackage\nginx/pid

ginx.pid" failed (123: The filename, directory name, or volume label syntax is incorrect)。

解决方法： 把配置文件里面的所有的路径斜线 \ 换成 /。

利用vue框架开发页面时，引用图片时，路径用相对对于html页面文件的路径，这样会造成部署上线之后找不到图片，原因是代码被打包后，图片不会再对于html文件的路径了。

解决方法：找到dist里面，index.html和static的关系，相对于他们的路径关系来引用图片。

手机h5网页在苹果手机上打开会出现，手指滑动时会出现整体页面不固定的bug。

解决方法： document.addListener(‘touchmove’,function(e){e.preventDefault();});

Event.preventDefault(); 阻止触发默认事件

Event.stopPropagation(); 捕获和冒泡阶段中当前事件的进一步传播。

event.stopImmediatePropagation(); 截断该节点绑定的多个相同类型事件的事件监听函数的顺序执行。

在使用文件配置文件配置同一个域名不同路径的时候的bug：

1. 静态资源找不到。

解决方法：将路径location /pepb {...} 改为 location /pepb/ {...}。

区别，你加一个 / 代表以这个之前的路径不会把pepb加入 静态资源的路径中。

代码：

location /pepb/ {

access\_log /home/\*\*\*/logs/access.log;

error\_log /home/\*\*\*/logs/error.log;

root /home/\*\*\*/dist/;

...

}

如果不加 / ，内部文件寻找静态资源的路径就是 /pepb/dist/static/\*\*\*;

加 / ，内部文件寻找静态资源的路径就是 相对路径了。

前端页面处理proto文件：

<https://hk.saowen.com/a/8c89fd74e0faee017447945b2780d9744a3aa4fcb928a915920f0997f631b38c>

命令：protoc.exe -I=static/img/proto --js\_out=import\_style=commonjs,binary:. \*\*\*.proto

Webpack 压缩文件：

webpack -p xxx.js -o path/yyy.js

-p: 压缩

xxx.js: 需要压缩的文件名

path: 压缩之后的文件存放地址

yyy.js: 压缩之后的文件名

Crontab 命令的使用：

命令总是找不到的bug：

例子：/usr/bin/node /usr/bin/pm2 pm2 restart 4

原因： pm2 命令重复， /usr/bin/pm2 就已经拿到了pm2了

正确写法：/usr/bin/node /usr/bin/pm2 restart 4

使用pm2的遇到的问题：

描述：sudo 下的node版本和 全局下的node版本不一致导致...扩展运算符报错。

实例： {

"apps":[

{

"name":"node-gm",

"script":"app.js",

**"node\_args":"--harmony",**

"watch":false,

"log\_date\_format":"YYYY - MM - DD HH: mm Z",

"exec\_mode":'cluster',

"instances":1,

"env":{

"NODE\_ENV":"development"

}

},

{

"name":"node-gm-timerTask",

**"node\_args":"--harmony",**

"script":"timerApp.js",

"watch":false,

"log\_date\_format":"YYYY - MM - DD HH: mm Z",

}

]

}

解析： 红色的参数就是解决node版本不一致导致的错误的根本。

Async-await:

1. Await 没有等待；

解决方案： await后面必须是一个Promise 对象。

注意： async 不能将一个方法变成 Promise对象；它只是使用await的前提。

