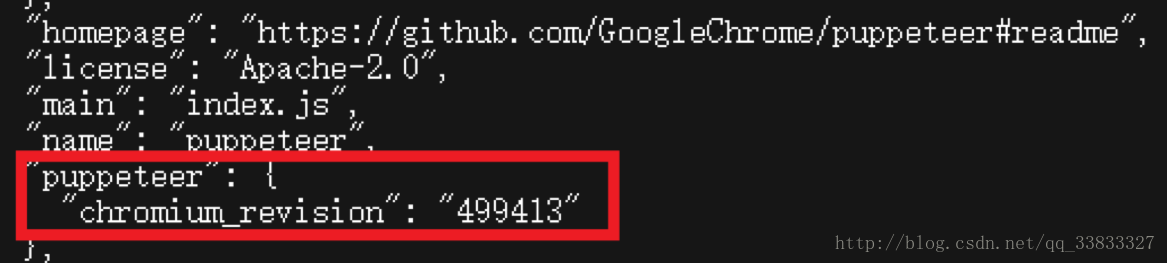
**Chrome Headless环境要求：**

1. Puppeteer要求node版本不低于v6.4.0，但是async/await只在Node v7.6.0或更高的版本支持。
2. 需要最近版本的Chromium浏览器

Puppeteer与其绑定的版本的Chromium配合最好，版本不对应不保证正确运行

node\_modules\puppeteer下面的 package.json 中有绑定的版本



* 由于封网，直接下载 Chromium 会失败，可以先阻止下载 Chromium 然后再手动下载它

**安装问题记录**

### 问题一

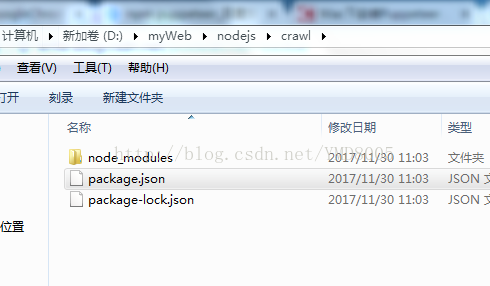
其实是因为下载完puppeteer包之后有执行node install.js去下载chromium（实验版chrome），但是由于网络的原因，下载失败了，我们可以避免缺省的下载 chromium，通过设置环境变量来阻止set PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD=1  
然后手动去下载chromium，上面有给出链接https://storage.googleapis.co...

解决方法：

1.创建项目crawl

2. npm

init



3.cmd

set PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD=1

http://img.blog.csdn.net/20171130111053983?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvWU1EODAwNQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

4.npm install

5. npm i --save puppeteer

puppeteer下载完成

第二种：

是因为在执行安装的过程中需要执行install.js，这里会下载Chromium,官网建议是进行跳过，我们可以执行 —ignore-scripts 忽略这个js执行。也可以通过设置环境变量set PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD=1阻止下载 Chromium （因为封网，直接下载会失败）

npm i --save puppeteer --ignore-scripts

### 问题二

需手动下载Chromium

1. 下载地址：

官网地址：

http://www.softpedia.com/dyn-postdownload.php/9b06c15ea58a97346e7bb5b6decc0dc5/5a6ad047/1a74b/4/1

* 更换国内Chromium源

PUPPETEER\_DOWNLOAD\_HOST=https://storage.googleapis.com.cnpmjs.org

npm i puppeteer

* 或者用 cnpm 安装

npm install -g cnpm --registry=https://registry.npm.taobao.org

cnpm i puppeteer

2.把下载刚刚下载的文件解压出来会有chrome-win32文件夹，把里面的文件拷贝到项目新建的chromium文件夹中；

3.因为chromium默认的下载路径是在node\_modules/puppeteer/.local-chromium/目录，这时候我们的chromium是在项目根目录，所以需要配置指定路径，修改index.js文件()：

const puppeteer = require('puppeteer');

(async () => {

const browser = await puppeteer.launch({//用指定选项启动一个Chromium浏览器实例。

executablePath: './chromium/chrome.exe',

headless: false

});

const page = await browser.newPage();//创建一个页面.

await page.goto('http://music.163.com/'); //到指定页面的网址.

await page.screenshot({path: 'music.png'}); //截图并保存到当前路径，名称为example.png.

browser.close();//关闭已打开的页面，browser不能再使用。

})();

puppeteer launch参数说明：

executablePath： 运行Chromium或Chrome可执行文件的路径

headless： 是否运行在浏览器headless模式，true为不打开浏览器执行，默认为true

timeout： 等待浏览器实例启动的最长时间（以毫秒为单位）。默认为30000（30秒）。通过0禁用超时

args： 传递给浏览器实例的其他参数

###问题三

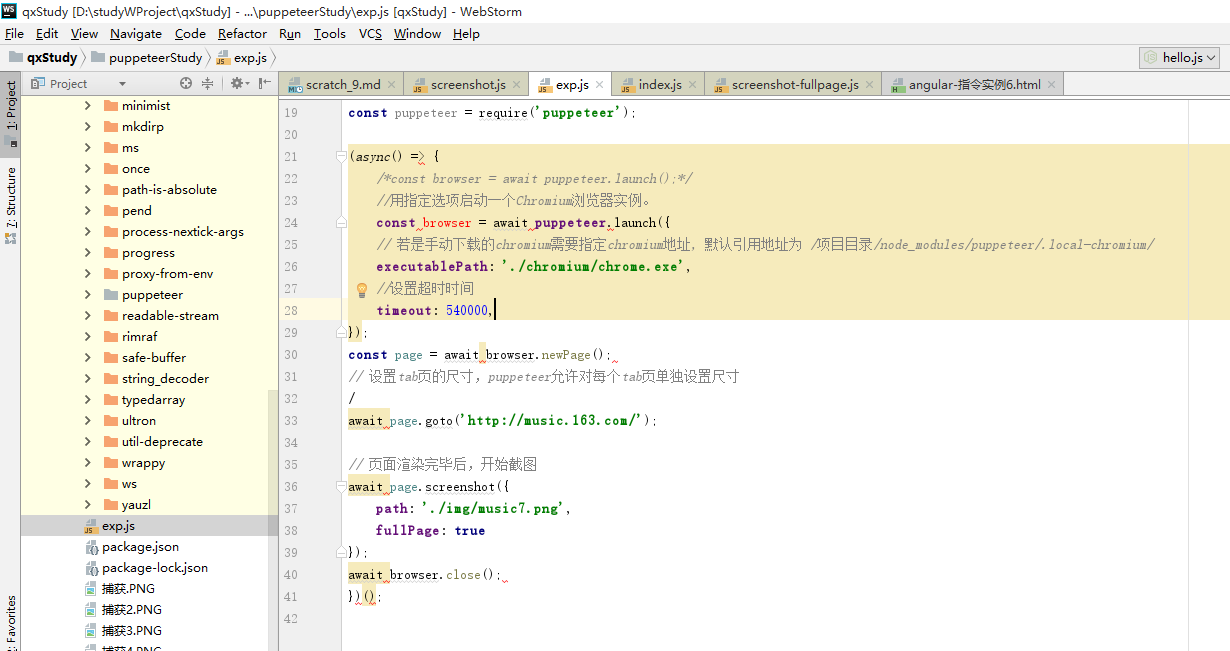
可能是chromium的版本存在差异，然后重新在chromium官网下载最新版本解压到项目（要注意下相应系统chromium）

执行index.js， 脚本运行chromium浏览器跳转到界面，截图保存到项目中，这样就成功了...

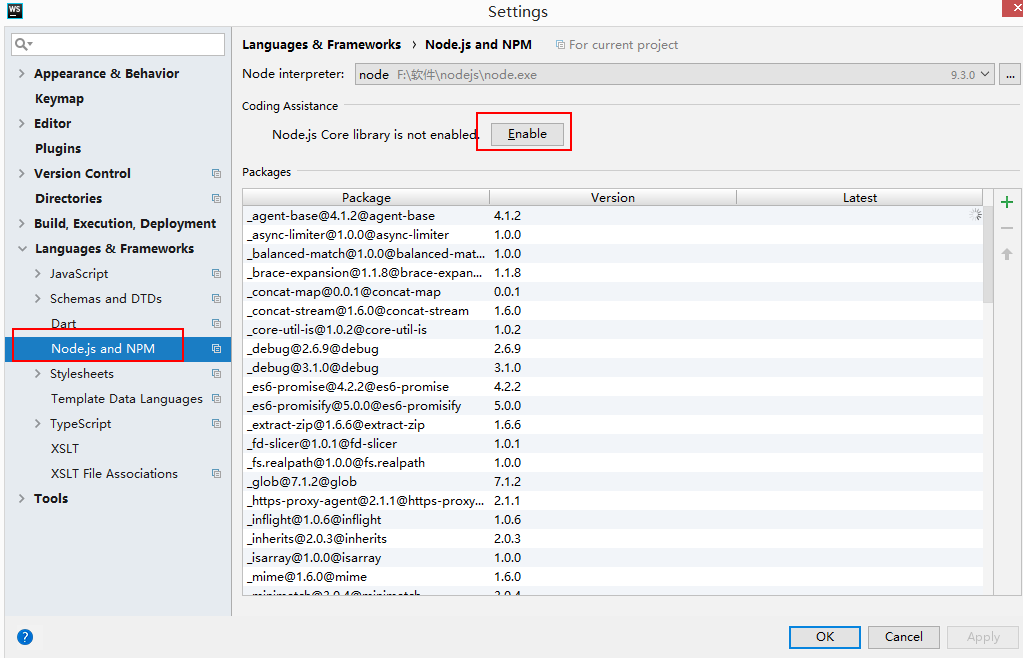
###问题四

###问题五

**代码出不来“asnyc”标识以及有红色波浪线解决办法：**

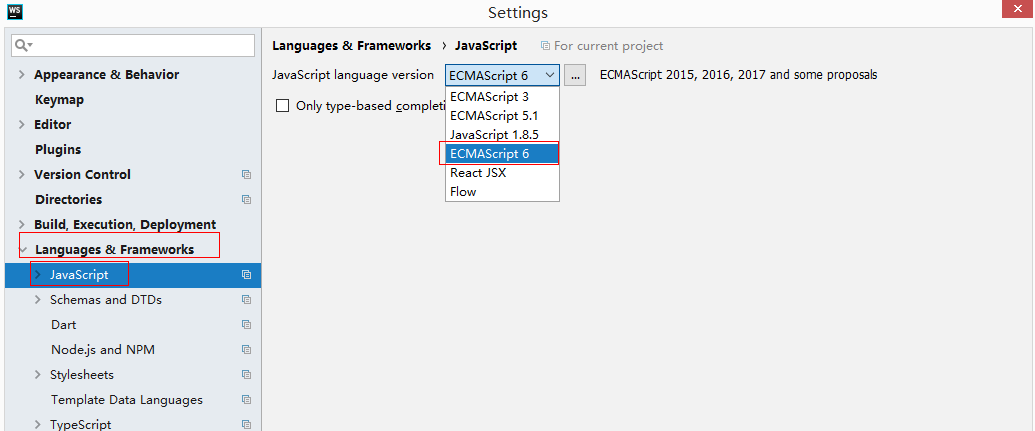


**解决步骤一：**



**选择“Enable”就可以解决出不来“asnyc”标识问题了**

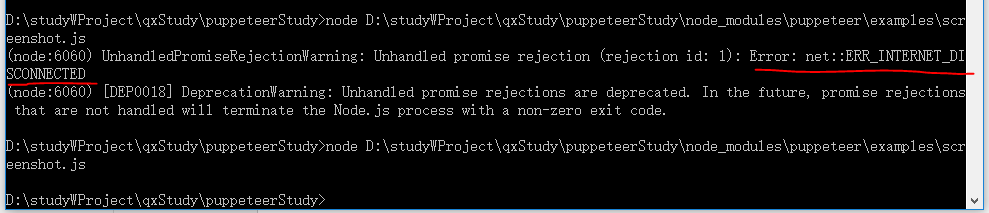
**解决步骤二：**

****

**选择“ECMAScript 6”就可以解决红色波浪线问题了**

**异常捕获：**

1. **未联网**

****

**API**

因为是一个开源一个多月的项目，现在项目很活跃，所以使用时自行查找[api](https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md)才能保证参数、使用方法不会错。

* page.waitForSelector 意思是等待指定元素出现在网页中，如果已经出现了，则立即继续执行下去, 后面跟的参数为 selector 选择器，与我们常用的 document.querySelector 接收的参数一致
* page.waitFor 后面可以传入 selector 选择器、function 函数或 timeout 毫秒时间，如 page.waitFor(2000) 指等待2秒再继续执行，例子中用这个函数暂停操作主要是为了演示
* page.$(selector) 与我们常用的 document.querySelector(selector) 一致，返回的是一个 ElementHandle 元素句柄
* page.$$(selector) 与我们常用的 document.querySelectorAll(selector) 一致，返回的是一个数组

// 地址栏输入网页地址

await page.goto('https://google.com/', {

// 配置项

// waitUntil: 'networkidle', // 等待网络状态为空闲的时候才继续执行

});

在D:\studyWProject\qxStudy\puppeteerStudy\node\_modules\puppeteer\ DeviceDescriptors.js文件中：

定义的是移动设备的各个机型的尺寸和配置。

**现存bug**

1. 这里有个Puppeteer的bug一直没有解决，发现截图的高度最大只能是16384px， 超出部分被截掉了。

