

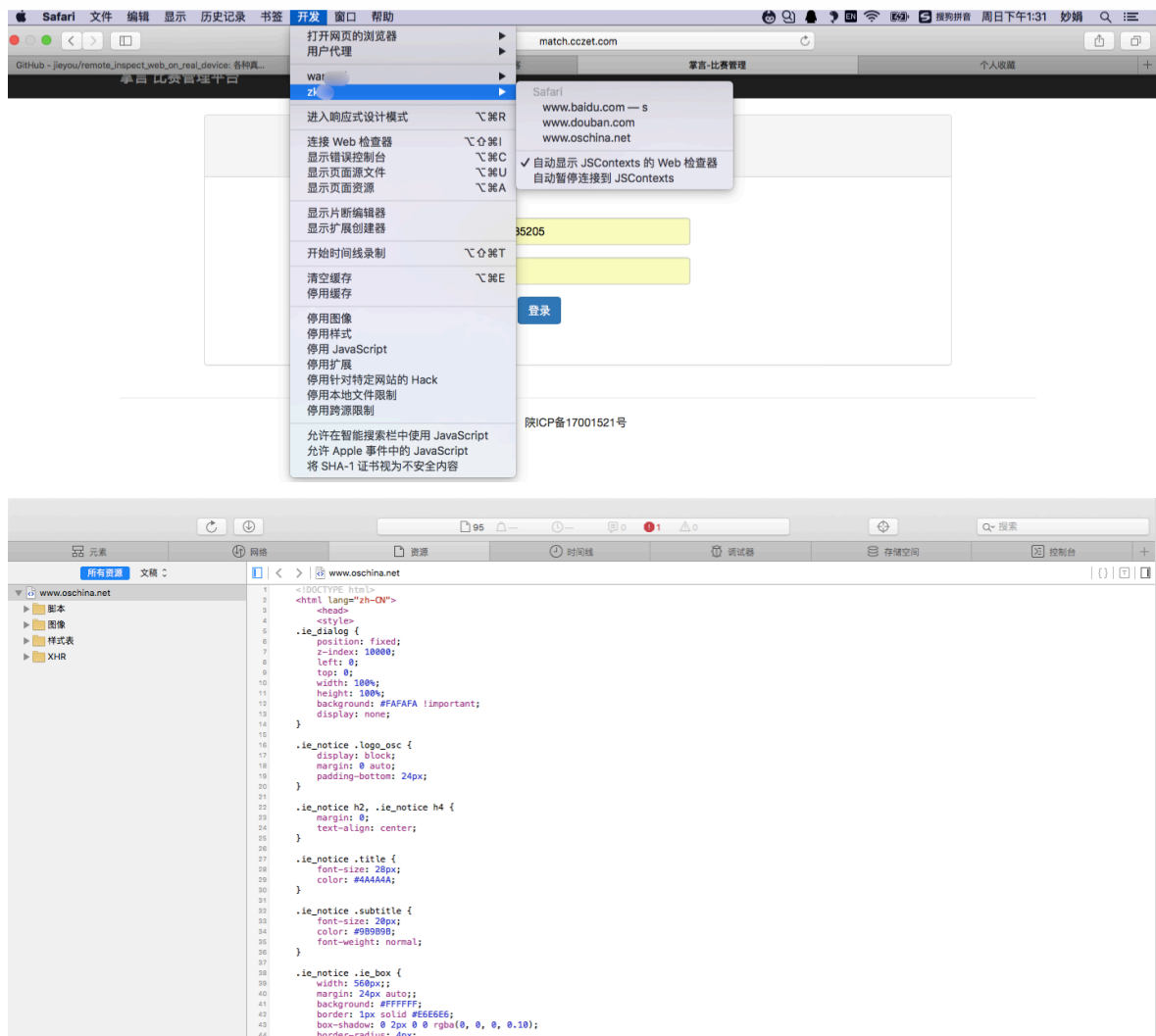
方式概览：

方法	必备条件	说明	局限
Safari 开发者工具			
使用Weinra			
使用 chrome			

调试方式说明

1. 用Safari调试iOS下页面

- 必备条件
 - **PC端和手机端均安装Safari**
- 操作步骤
 1. 首先开启PC端Safari开发菜单。操作步骤：“偏好设置” – “高级” – “在菜单栏中显示开发选项”。
 2. 开启ios端Safari浏览器的调试功能。操作步骤：“设置” – “Safari” – “高级” – “Web 检查器”。
 3. 通过数据线连接iOS和OS X
 4. 在iOS Safari 浏览器中打开要调试的网页，然后在OS X Safari 浏览器的开发菜单中可以看到调试的页面。
- 调试效果展示



2. 使用 Weinre 调试

- 必备条件

安装node.js环境，安装weinre

- 操作步骤

1. 安装node.js环境，并且安装npm。node下载链

接：<https://nodejs.org/en/download/>

2. 安装weinre。执行命令：`npm -g install weinre` 回车等待安装 weinre。

3. 运行weinre。

键入命令：`weinre -httpPort 8081 -boundHost 192.168.1.105` 启动weinre。打开浏览器，地址栏输入 <http://192.168.1.105:8081>。即可看到如下界面：

weinre - web inspector remote

Access Points

debug client user interface: <http://192.168.1.105:8081/client/#anonymous>
documentation: <http://192.168.1.105:8081/doc/>

点击此处进入调试界面

Target Demos

The following links point to an already instrumented sample application, run in a couple of different environmental configurations. First open a new browser window for the debug client user interface, as specified above. Then open another new browser window and connect and result in an active debug connection between the client and the target demo.

- [the non-minified demo](#)
- [the minified demo](#)
- [the non-minified strict demo](#)

Target Script

被调试界面需要引入的js

You can use this script to inject the weinre target code into your web page.

<http://192.168.1.105:8081/target/target-script-min.js#anonymous>

Example:

```
<script src="http://192.168.1.105:8081/target/target-script-min.js#anonymous"></script>
```

说明:

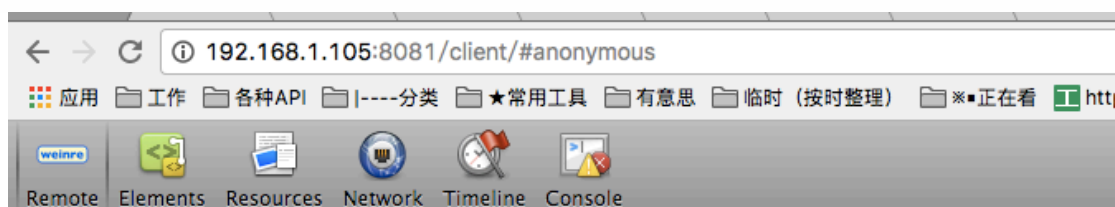
1. weinre的默认访问链接为localhost:8080。可以通过 `-httpPort` 修改端口, 通过 `-boundHost -all-` 使得同一局域网内的电脑均可查看weinre主页。

4. 在要被调试的页面加上weinre的js配置。

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>掌言</title>
  <base href="/">
  <script src="https://res.wx.qq.com/open/js/jweixin-1.2.0.js"></script>
  <script src="http://192.168.1.105:8081/target/target-script-min.js#anonymous"></script>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="format-detection" content="telephone=no">
  <meta name="viewport" content="width=device-width,minimum-scale=1.0,maximum-scale=1.0,user-scalable=no"/>
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body class="container-fluid">
<app-root></app-root>
</body>
</html>
```

5. 手机端访问页面, PC端查看debug。

手机和电脑处在同一局域网下。手机访问需要调试的界面, 在电脑上点击 `Access Points->debug client user interface`。可以看到如下界面:



Targets

- [192.168.1.103 \[channel: t-3 id: anonymous\] - http://192.168.1.105:4200/](http://192.168.1.103:4200/)
- [192.168.1.105 \[channel: t-2 id: anonymous\] - http://192.168.1.105:4200/](http://192.168.1.105:4200/)

Clients

- [192.168.1.105 \[channel: c-1 id: anonymous\]](#) targets会自动检测被调试页面的访问。选择可选中一个查看。绿色表示选中

Server Properties

```
boundHost: 192.168.1.105
deathTimeout: 15
debug: false
httpPort: 8081
readTimeout: 5
staticWebDir: /usr/local/lib/node_modules/weinre/web
verbose: false
version: 2.0.0-pre-I0Z7U9OV
```

调试效果展示:

Name	Method	Status	Type	Size	Time	Timeline
Path		Text		Transfer	Latency	
<input type="checkbox"/> banners	GET	200 OK	application/json	0B	1.16s	
<input type="checkbox"/> 192.168.1.105				46B	1.16s	
<input type="checkbox"/> login	POST	200 OK	application/json	0B	2.60s	
<input type="checkbox"/> 192.168.1.105				46B	2.60s	
<input type="checkbox"/> init-config	GET	200 OK	application/json	0B	934ms	
<input type="checkbox"/> 192.168.1.105/wxs				46B	929ms	
<input type="checkbox"/> info	GET	200 OK	application/json	0B	197ms	
<input type="checkbox"/> 192.168.1.105/sockjs-node				242B	130ms	
<input type="checkbox"/> home	GET	200 OK	application/json	0B	676ms	
<input type="checkbox"/> 192.168.1.105				46B	674ms	
<input type="checkbox"/> banners	GET	200 OK	application/json	0B	369ms	
<input type="checkbox"/> 192.168.1.105				46B	354ms	

3.chrome开发者工具