

Web 服务器的调试方法

先了解 web 服务器编程：

Web 服务器创建一个侦听套接字并开始循环接受新连接。客户端启动 TCP 连接，并在成功建立 TCP 连接之后，客户端向服务器发送 HTTP 请求，服务器以显示给用户的 HTTP 响应进行响应。为了建立 TCP 连接，客户端和服务端都使用套接字。

(<https://ruslanspivak.com/lbaws-part1/>)

常规调试方法：

Web 开发中遇到的问题分为：前端实现问题，接口调用问题，接口实现问题提。

1. 前端实现问题：

前端实现问题是指，前端在调用接口前或者调用接口后，业务逻辑错误或者程序错误造成的系统用户交互异常。

Pc 站与微网站可以使用 chrome（推荐）或者 firefox 的开发者工具进行调试。

2. 接口调用问题：

接口调用问题，多为前端调用接口错误或者接口本身存在 bug 引起的，无法获得前端人员预期的返回结果。

Pc 站可以通过 pc 浏览器进行故障分析（抓包）。

如果没有接口调用的数据包，则前端程序存在问题；如果接口的请求参数与请求头信息存在错误，则前端调用接口使用错误；如果接口的响应参数与响应头信息存在错误，则接口实现存在问题；如果接口调用正常，则前端程

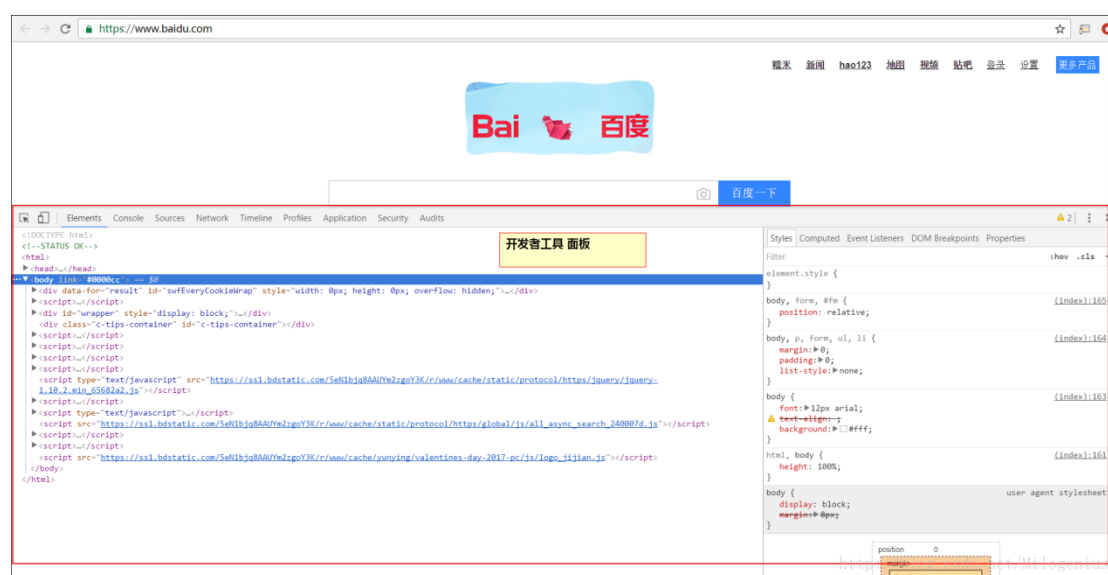
序对数据结果处理存在问题。

3. 接口实现问题:

接口实现问题, 多为后端开发人员没有根据正确的请求参数返回正确的响应结果。多由逻辑错误, 数据持久化处理错误与系统级错误引起的。

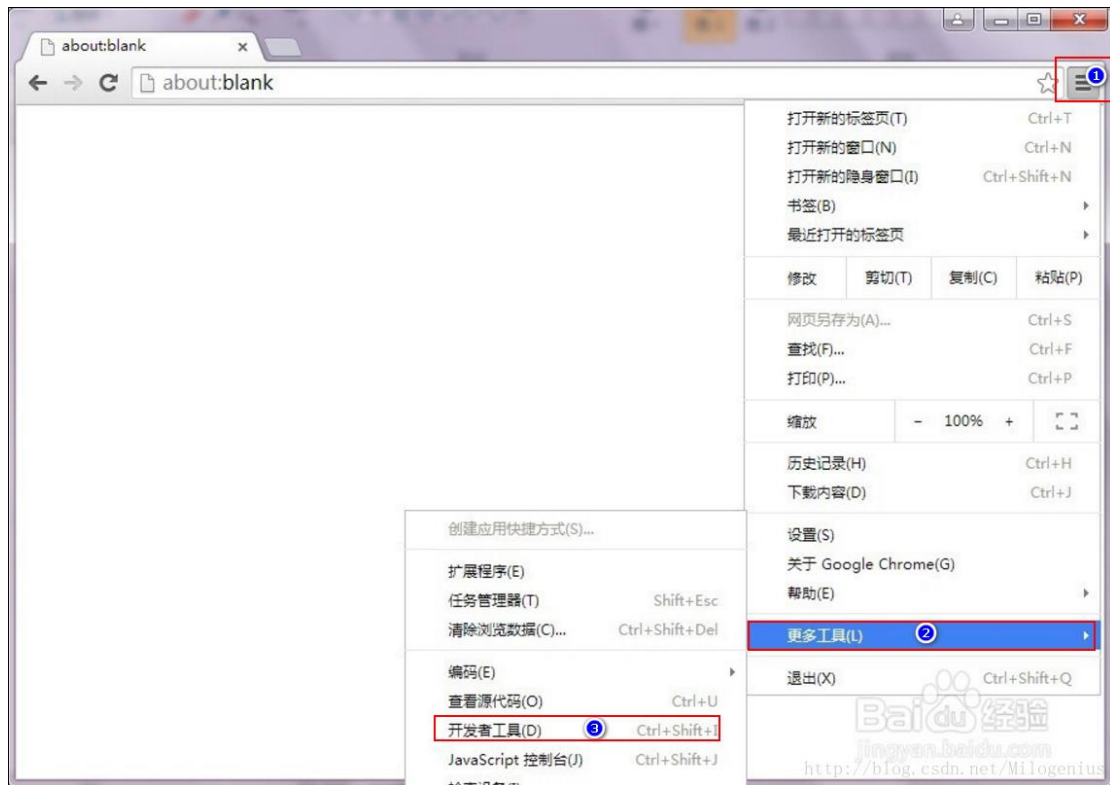
在开发时可以通过断点跟踪进行分析。生成环境需要通过分析系统日志或者程序的日志进行分析。

Chrome 浏览器的调试方法:



1. Chrome 浏览器开发者工具的打开方式:

- 1) 功能键 F12 或者 Ctrl+Shift+j 或者 Ctrl+Shift+j 打开;
- 2) 打开浏览器右上角用户设置下拉菜单, 选择更多工具, 再选择开发者工具即可打开开发者工具。



2. 开发者工具的工具栏中常用面板介绍



Elements:查找网页源代码 HTML 中的任一元素,手动修改任一元素的属性和样式且能实时在浏览器里面得到反馈。

Console:记录开发者开发过程中的日志信息,且可以作为与 JS 进行交互的命令行 Shell。

Sources: Sources 功能面板是资源面板,他主要分为四个部分,四个部分并不是独立的,他们互相关联,互动共同实现一个重要的功能:监控 js 在执行期的活动。简单来说就是断点啦。

Network:从发起网页页面请求 Request 后分析 HTTP 请求后得到的各个请求资源信息（包括状态、资源类型、大小、所用时间等），可以根据这个进行网络性能优化。

Timeline:记录并分析在网站的生命周期内所发生的各类事件，以此可以提高网页的运行时间的性能。

Profiles:如果你需要 Timeline 所能提供的更多信息时，可以尝试一下 Profiles,比如记录 JS CPU 执行时间细节、显示 JS 对象和相关的 DOM 节点的内存消耗、记录内存的分配细节。

Application:记录网站加载的所有资源信息，包括存储数据 (Local Storage、Session Storage、IndexedDB、Web SQL、Cookies)、缓存数据、字体、图片、脚本、样式表等。

Security:判断当前网页是否安全。

Audits:对当前网页进行网络利用情况、网页性能方面的诊断，并给出一些优化建议。比如列出所有没有用到的 CSS 文件等。

(十分详细的图文教程见链接 <https://blog.csdn.net/milogenius/article/details/78897745>，在此略去。参考链接 <https://www.cnblogs.com/alexzhang92/p/9359141.html>)

web 前端 Chrome 调试常见操作

dom 元素调试：

1. 在打开的调试工具中的最上面一栏，选择 Element，则可以看到当前页的 dom 结构
2. 在 Element 中，划到哪个元素就可以在页面对应位置看到相关选中效果；对应的一个操作，点击调试工具的左上角的箭头，然后在页面中点击，则可以

查看页面点击元素在 Element 中的位置

3. 在 Element 中，点击选中某个元素，直接键盘按 backspace 或者 delete 就可以直接删除，在排除某个样式影响错误的时候，可以这样快速定位到错误
4. 在 Element 中，双击标签名称，可以改变标签名；双击属性，可以选中修改属性，复制；双击内容可以修改内容；选中整个标签元素，右键——“Edit as Html”或者直接 F2 可以修改整句标签，包括标签名、属性、内容等，一般测试滚动样式是否正常时，直接选中
5. 标签，复制粘贴达到迅速获得较多内容出现滚动

css 样式调试：

1. 选中 dom 元素时，在 Styles 栏可以看到该元素的所有样式设置，包括继承得到的样式，覆盖和被覆盖的样式情况；在 Styles 栏旁边有个 Computed 按钮，点击可以看到该元素经过计算后，真正生效的样式列表
2. 在 Styles 栏中，双击可以修改样式；点击大括号{}中最后一个样式的下面可以新添加样式；选中某个样式名字或者值，删除为空，可以删除样式
3. 调整属性中的数字时，选中，按向上或者向下可以实现+1 或-1 的调整
4. 双击选中某个属性后，回车可以切换不断向下切换选中的样式
5. 点击颜色显示的那个标识可以打开颜色的修改设置

Console 中调试

1. Console 里面可以进行编程序，提示功能十分强大，常用于测试一些拿不准的判断或者功能函数，也可以查询 API
2. 当 Console 里面有报错时，可以直接点击定位错误，这是自动跳转到 Sources 栏，这一栏可以断点调试，快捷键主要使用 F8（从当前断点向下继续执行直

到下一个断点)、F10 (从当前中断地向下执行一步)、F11 (进入某个函数中一步一步执行)。使用 webpack 压缩打包时, 开启 sourcemap 可以映射到未压缩前的代码中调试。

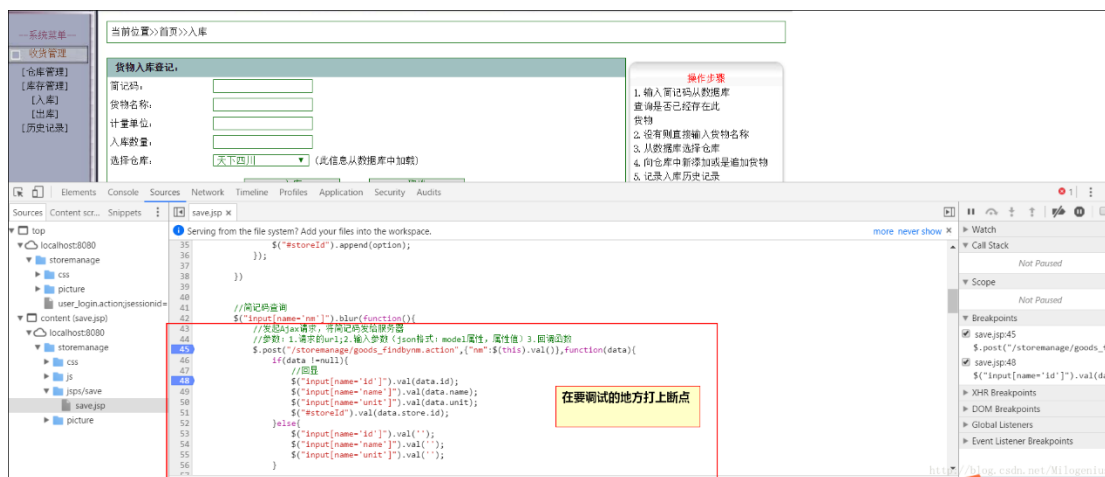
NetWork 中调试

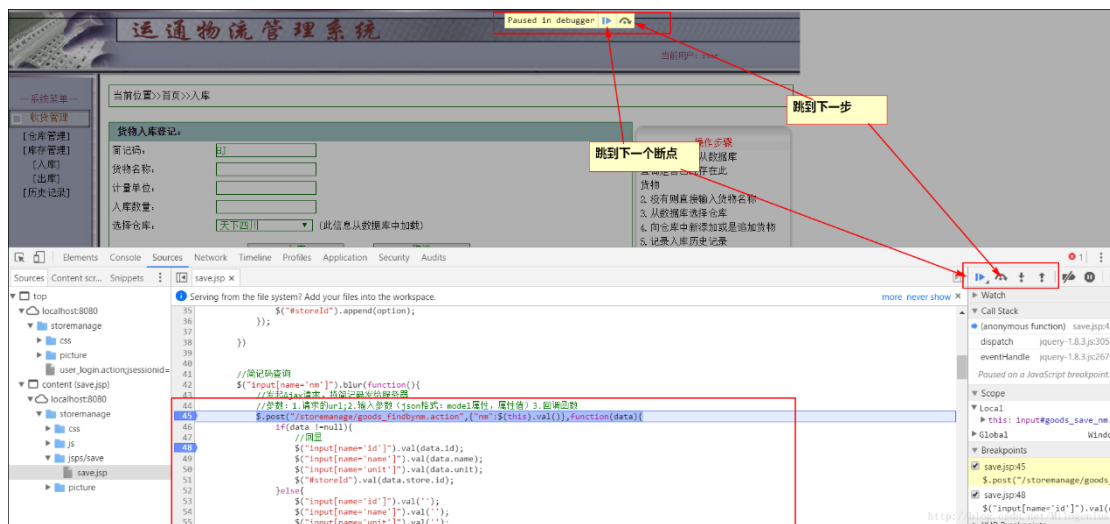
1. 这里在刚打开时, 可能什么都没有, 需要重新刷新页面, 得到加载资源的顺序、体积大小和时间消耗等信息。
2. 在上面有一栏加载资源的分类, 有 All, XHR、JS、CSS、Img... 。其中, 常用的是 XHR, 因为要查看 ajax 请求, 点击后, 在 Header 栏可以查看此次请求的首部信息, 在 Response 栏可以看到服务器返回信息

Application 中查看保存信息

1. 在该栏可以查看 cookie、localStorage 等信息

断点调试





(同样在前述链接 <https://blog.csdn.net/milogenius/article/details/78897745> 中)