

跨域

同源策略

- 同源策略是浏览器的一种安全策略，所谓同源是指域名，协议，端口完全相同
- 在同源策略下，只有同源的地址才可以相互通过 AJAX 的方式请求。
- 同源或者不同源说的是两个地址之间的关系，不同源地址之间请求我们称之为跨域请求

同源

- http://www.example.com/detail.html 与以下地址对比

对比地址	是否同源	原因
http://api.example.com/detail.html	不同源	域名不同
https://www.example.com/detail.html	不同源	协议不同
http://www.example.com:8080/detail.html	不同源	端口不同
http://api.example.com:8080/detail.html	不同源	域名、端口不同
https://api.example.com/detail.html	不同源	协议、域名不同
https://www.example.com:8080/detail.html	不同源	端口、协议不同
http://www.example.com/other.html	同源	只是目录不同

在同源策略下，浏览器不允许 Ajax 跨域获取服务器数据

现代化的 Web 应用中肯定会有不同源的现象

必然要解决不同源问题，从而实现跨域请求。

跨域解决方案

JSONP

JSONP 原理

- JSON with Padding, 是一种借助于 script 标签发送跨域请求的技巧。
- 原理就是在客户端借助 script 标签请求服务端的一个地址
- 地址返回一段带有某个全局函数调用的 JavaScript 脚本
- 在调用函数中, 原本需要返回给客户端的数据通过参数传递给这个函数
- 这样客户端的函数中就可以通过参数得到原本服务端想要返回的数据

演示服务端

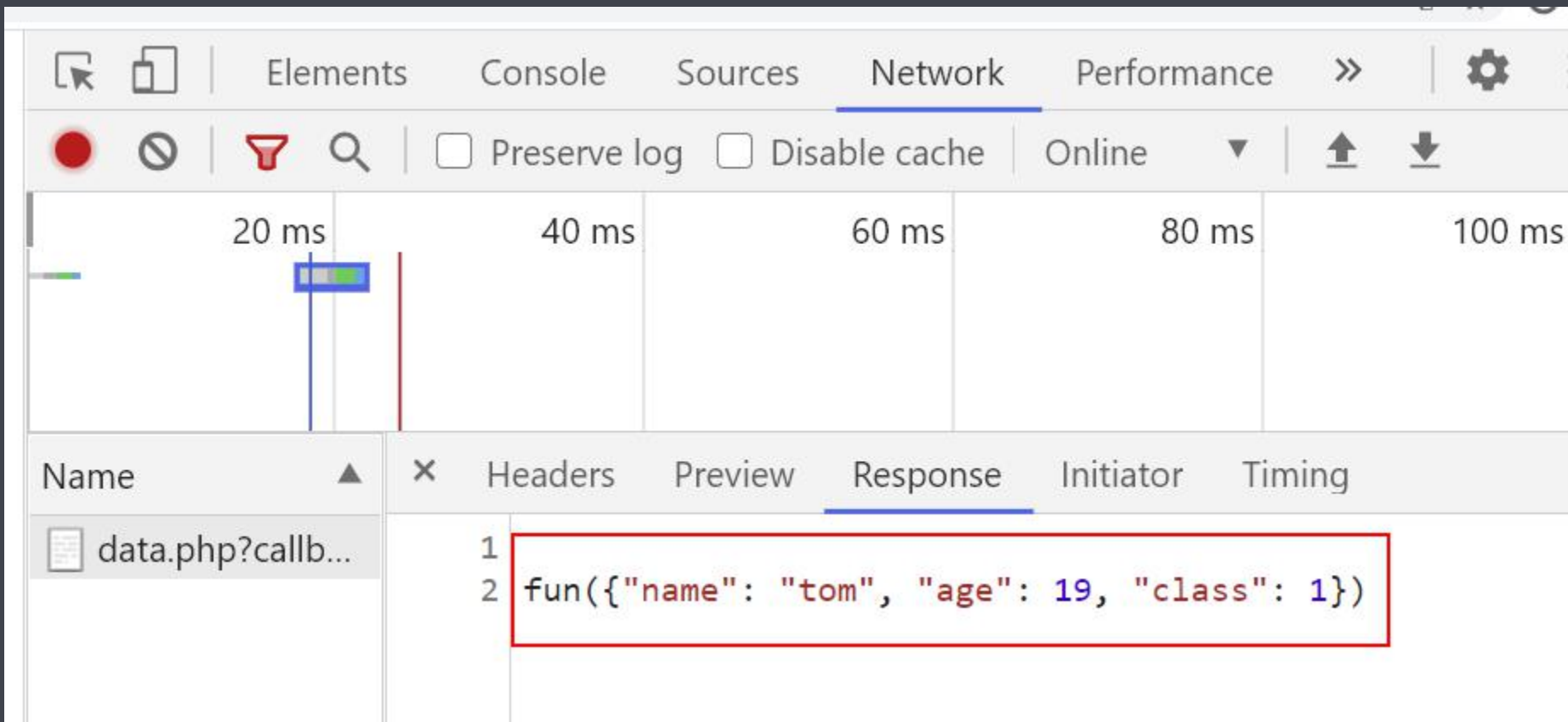
```
<?php
    // 为了让后台传输的数据中函数名与前台保持一致, 需要动态获取回调函数的函数名
    $callback = $_GET["callback"];
    echo $callback.'({"name": "tom", "age": 19, "class": 1})';
?>
```

演示客户端

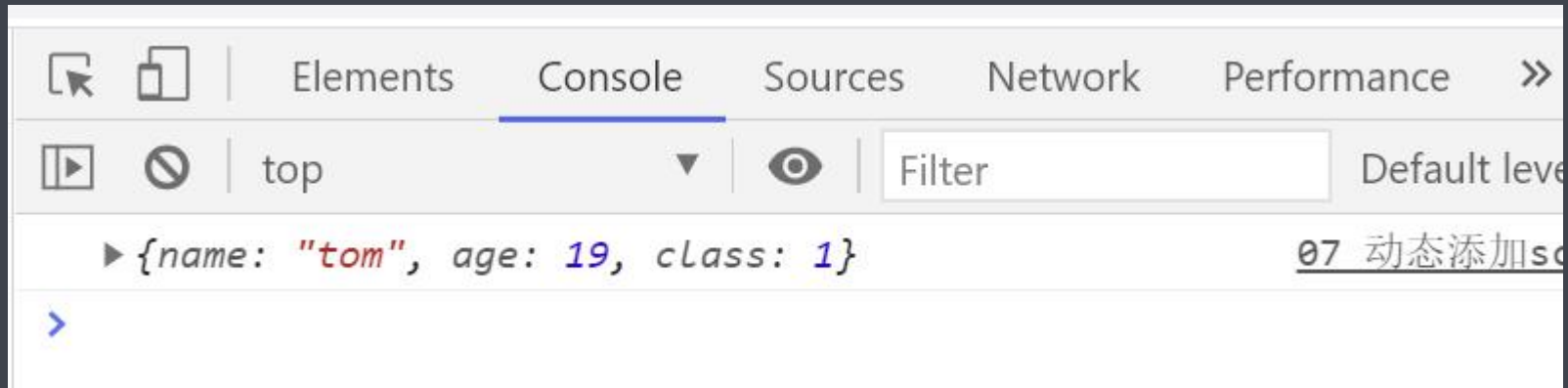
```
// 动态生成 script 标签
var script = document.createElement("script");
script.src = "http://new.com/data.php?callback=fun";
// 添加到head标签内部
var head = document.getElementsByTagName("head")[0];
head.appendChild(script);

// 客户端定义的回调函数, 内部 data 参数接收服务端传来的数据
function fun(data){
    // 通过函数的调用执行, 可以处理后台得到的数据
    console.log(data);
}
```

演示响应结果



演示控制台结果



JSONP 只能发送 GET 请求

JSONP 用的是 script 标签，
与 AJAX 提供的 XMLHttpRequest 没有任何关系

jQuery 中对 JSONP 的支持

jQuery 中的 jsonp

- jQuery基本使用 \$.ajax()
- jQuery 中使用 JSONP 就是将 dataType 设置为 jsonp

CORS

cors 跨域

- Cross Origin Resource Share, 跨域资源共享
- 这种方案无需客户端作出任何变化（客户端不用改代码），只是在被请求的服务端响应的时候添加一个 Access-Control-Allow-Origin 的响应头，表示这个资源是否允许指定域请求。
- Access-Control-Allow-Origin 的值：

*

表示允许任意源访问，不安全

http://foo.com

允许指定的源访问

服务端演示

```
<?php
    // 通过设置响应头, 允许其他的网站服务器跨域访问这个文件
    header("Access-Control-Allow-Origin: http://foo.example");
    header("Access-Control-Allow-Origin: *");
?>
```

案例应用

案例

- 模仿百度搜索案例
- 接口：`https://www.baidu.com/sugrec?pre=1&p=3&ie=utf-8&json=1&prod=pc&from=pc_web`

模板引擎

模板引擎作用

- 减少字符串拼接
- 在模板里面解析json，然后跟 html 内容拼接，性能会更好

artTemplate

- art-template 是一个简约、超快的模板引擎。
- 网址: <https://github.com/aui/artTemplate>
- 中文使用文档: <https://aui.github.io/art-template/zh-cn/docs/>
- 常用语法:
 - `<% 与 %>` 符号包裹起来的语句则为模板的逻辑表达式
 - `<%= content %>` 为输出表达式

artTemplate 使用步骤

- 引入模板文件
- 创建模板
- 将数据跟模板进行绑定
- 在模板里面编写代码解析数据
- 绑定数据和模板之后得到内容
- 将数据内容写到页面上面。

留言板案例

拉勾教育

— 互联网人实战大学 —



下载「拉勾教育App」
获取更多内容



扫码联系老师

技能评估、福利资料、课程优惠

Made with ❤️ by LagouFed