Atitti 提升开发效率--ajax跨域解决方案

[1.1. 跨域的优点，完全的前后端分离。可以使用file协议访问http端的内容。。 1](#_Toc17930)

[1.2. Atitit.js跨域解决方案attilax大总结 后台java php c#.net的](#_Toc9212)[CORS支持目录 1](#_Toc9212)

[1.3. jsonp 需要服务端支持调制，为返回jsonp格式。 1](#_Toc1062)

[1.4. Attilax总结的最佳策略 cors 2](#_Toc4697)

## 跨域的优点，完全的前后端分离。可以使用file协议访问http端的内容。。

## Atitit.js跨域解决方案attilax大总结 后台java php c#.net的CORS支持目录

1. 设置 document.domain为一致 推荐 1

2. Apache 反向代理 推荐 1

3. jsonp , 2

4. CORS , 2

5. iframe+post 3

6. 回调解决方案>>服务端推送dwr 反向ajax 3

7. 服务器中转 3

8. Js跨 3

9. 从所有的浏览器都支持来看，CORS将成为未来跨域访问的标准解决方案 3

10. 参考 4

## jsonp 需要服务端支持调制，为返回jsonp格式。

## Attilax总结的最佳策略 cors

服务器端对于CORS的支持，主要就是通过设置Access-Control-Allow-Origin来进行的。如果浏览器检测到相应的设置，就可以允许Ajax进行跨域的访问。

/AtiPlatf\_ee/src/aaaPKg/CrossDomainFilter.java

@Override

public void doFilter(ServletRequest req, ServletResponse response, FilterChain chain) throws IOException, ServletException {

HttpServletResponse res=(HttpServletResponse) response;

res.setHeader("Access-Control-Allow-Origin", "\*");

//"Access-Control-Allow-Methods:POST,GET"

chain.doFilter(req, response);

System.out.println("--CrossDomainFilter exeed");

}

参考

Atitit.js跨域解决方案attilax大总结 后台java php c#.net的CORS支持