# 同源策略

# 方案分析

## Cookie不能传递的解决方案

1. 使用原生XHR

var xhr = new XMLHttpRequest();

xhr.open("POST", "http://xxxx.com/demo/b/index.php", true);

xhr.withCredentials = true; //支持跨域发送cookies

xhr.send();

1. 使用jQuery的Ajax方法来处理Post请求

$.ajax({

type: "POST",

url: "http://xxx.com/api/test",

dataType: 'jsonp',

xhrFields: {

withCredentials: true

},

crossDomain: true,

success:function(){},

error:function(){}

})

1. 服务器端设置：

header("Access-Control-Allow-Credentials: true");

header("Access-Control-Allow-Origin: http://www.xxx.com");

## OPTIONS请求方法的两个用途

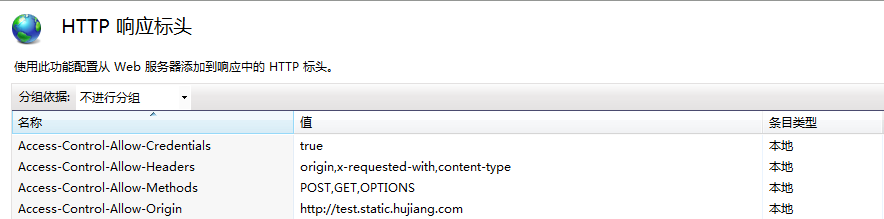
1. 获取服务器支持的HTTP请求方法；也是黑客经常使用的方法。
2. 用来检查服务器的性能。例如：AJAX进行跨域请求时的预检，需要向另外一个域名的资源发送一个HTTP OPTIONS请求头，用以判断实际发送的请求是否安全。

# 方案一：Jsonp

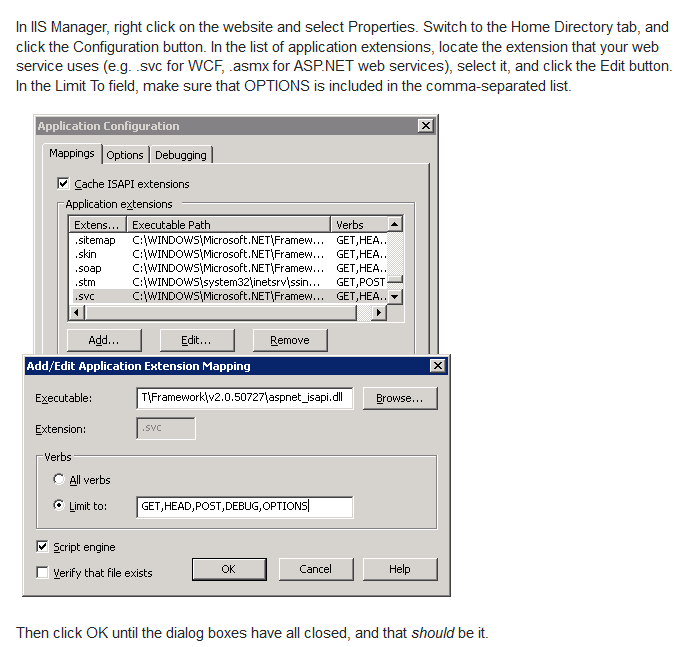
1. 优势
2. 劣势
   1. Jsonp只支持Get

# 方案二：IIS配置

1. IIS7.0及以上



1. IIS6.0要补充以下配置



# jQuery的独特之处

1. 使用jQuery进行跨域Post处理时，不需要先进行Options处理,只进行Post处理;