

فرض کنید مثال دریافت اطلاعات API فیدبرنر را بخواهیم با استفاده از jQuery پیاده سازی کنیم، یعنی امکان برنامه نویسی سمت سرور را نداریم و می‌خواهیم با استفاده از جاوا اسکریپت، تعداد مشترکین فید یک سایت را نمایش دهیم.

برای مثال آدرس دریافت اطلاعات مربوط به API فیدبرنر وبلاگ جاری به صورت زیر است:

<https://feedburner.google.com/api/awareness/1.0/GetFeedData?id=fhphjt61bueu08k93ehujspu234>

و در حالت کلی :

`http://api.feedburner.com/awareness/1.0/GetFeedData?uri=<feeduri>`

که حاصل آن برای مثال یک فایل XML با فرمت زیر می‌باشد:

```
<rsp stat="ok">
  <feed id="fhphjt61bueu08k93ehujspu234" uri="vahidnasiri">
    <entry date="2009-01-23" circulation="153" hits="276" reach="10"/>
  </feed>
</rsp>
```

همانطور که مطلع هستید چند روزی است که jQuery 1.3.1 [ارائه شده](#) است. جهت استفاده از آخرین نگارش موجود آن تا این زمان، می‌توان از گوگل به عنوان هاست این کتابخانه به صورت زیر استفاده کرد:

```
<script src='http://ajax.googleapis.com/ajax/libs/jquery/1.3.1/jquery.min.js'
type='text/javascript'></script>
```

نحوه خواندن

مقدار circulation فایل xml ذخیره شده بر روی کامپیوتر:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>FeedBurner API</title>
    <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.3.1/jquery.min.js'
type='text/javascript'>
    </script>
    <script type="text/javascript">
      function parseXml(xml){
        //find every entry and print the circulation
        $(xml).find("entry").each(function(){
          $("#output").append($(this).attr("circulation"));
        });
      }

      $(document).ready(function(){
        $.ajax({
          type: "GET",
          url: "GetFeedData_local.xml",
          dataType: "xml",
          success: parseXml
        });
      });
    </script>
```

```
</head>
<body>
  <div dir="rtl" style="font-family:tahoma; font-size:12px;">
    تعداد مشترکین تغذیه خبری سایت:
    <div id="output">
    </div>
  </div>
</body>
</html>
```

با استفاده از قابلیت Ajax کتابخانه jQuery ، اطلاعات فایل محلی GetFeedData_local.xml دریافت شده و محتوای آن به تابع parseXml پاس می‌شود (توسط قسمت success). سپس در این تابع تمام تگ‌های entry یافت شده و مقدار circulation آن‌ها به یک div با ID معادل output اضافه می‌شود.

مثال فوق در مورد خواندن اطلاعات از یک فایل xml می‌تواند برای مثال این کاربرد را در یک سایت داشته باشد: نمایش اتفاقی سخن روز یا سخن بزرگان و امثال آن بدون برنامه نویسی سمت سرور جهت انجام این کار از یک فایل xml تهیه شده، بدون نیاز به استفاده از دیتابیس خاصی.

تا اینجای کار مشکلی نیست. اما همانطور که در مطلب مقابله با حملات [CSRF](#) نیز ذکر شد، مرورگرهای جدید امکان ارسال یا دریافت اطلاعات به صورت Ajax را بین سایت‌ها [ممنوع کرده‌اند](#) (ماجرا هم از آنجا شروع شد که یکبار جی‌میل این باگ امنیتی را داشته است). بنابراین اگر شما بجای url قسمت Ajax فوق، آدرس سایت فید برنر را قرار دهید با خطای زیر متوقف خواهید شد:

Access to restricted URI denied

تمام موارد دیگری هم که در jQuery برای دریافت اطلاعات از یک فایل یا url موجود است (مثلا تابع load یا get و امثال آن) فقط به سایت جاری و دومین جاری باید ختم شوند در غیر اینصورت توسط مرورگرهای جدید متوقف خواهند شد.

نظرات خوانندگان

نویسنده: MicroDump
تاریخ: ۱۷:۴۰:۰۰ ۱۳۸۷/۱۱/۰۹

پس نمی‌شه!

نویسنده: وحید نصیری
تاریخ: ۰۰:۲۳:۰۰ ۱۳۸۷/۱۱/۱۰

با استفاده از jsonp از امکان cross site Ajax request در jquery مهیا است اما زیاد به صورت مساله ما نمی‌خوره!

نویسنده: علی یگانه مقدم
تاریخ: ۱۹:۵۷ ۱۳۹۳/۱۱/۱۸

ممنون از مطلب شما
من مشکل CORS را بر روی سیستم هم دارم، رفع این مورد به چه صورت هست؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۱۴ ۱۳۹۳/۱۱/۱۸

لطفا مثالی را که کار نمی‌کند برای بررسی بیشتر پیوست کنید.

نویسنده: علی یگانه مقدم
تاریخ: ۲۰:۴۷ ۱۳۹۳/۱۱/۱۸

```
function ReadFlagsOnXml(divControlName, XMLFile) {
    var div = $("#" + divControlName);
    console.log($(div).text());

    $.ajax({
        type: "GET",
        url: XMLFile,
        dataType: "xml",
        success: function(xml) {
            var xmlDoc = $.parseXML(xml),
                $xml = $(xmlDoc);
            console.log(xml);

            var countries = $(xml).find('Countries');
            $(countries).find('Country').each(function() {
                var countryName = $(this).find('Name').text();
                var flagImage = $(this).find('Image').text();
                $("", {
                    src: flagImage,
                    title: countryName
                }).appendTo($(div));
            });
        }
    });
}
```

این کد روی IIS به خوبی اجرا میشه ولی داخل یک دایرکتوری، روی پروتکل file خیر

نویسنده: وحید نصیری
تاریخ: ۲۱:۰۶ ۱۳۹۳/۱۱/۱۸

بله. در نگارش‌های اخیر مرورگرها، این مساله‌ی خاص دسترسی به فایل‌های محلی، ممنوع شده‌است. کروم این خطا را نمایش می‌دهد:

Cross origin requests are only supported for protocol schemes: http, data, chrome-extension, https, chrome-extension-resource.

و فایرفاکس هم این خطا را:

Access to restricted URI denied.

علت هم اینجا است که چون جاوا اسکریپت بر روی مرورگر کاربر اجرا می‌شود، وجود این دسترسی می‌تواند به دریافت فایل‌های محلی کاربر بدون اجازه‌ی او ختم شود. به همین جهت این مورد خاص کار با پروتکل فایل راه حلی ندارد. کلاً ممنوع شده‌است. سیلورلایت هم به همین شکل عمل می‌کند. تنها راه حل موجود، هاست کردن فایل بر روی وب سرور و سپس کار با سرور است (به صورت معمول و از طریق پروتکل http).