

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

درخواست یک فایل از دامنه ی دیگر با استفاده از JSONP

مدرس : مهندس افشین رفوآ

دوره آموزشی JQuery

درخواست یک فایل از دامنه ی دیگر با استفاده از JSONP

در مبحث قبلی، درباره ی سیاست امنیتی منشا مشترک (**same origin policy**) که مانع از درخواست **AJAX** دادن به دامنه یا زیردامنه ی متفاوت، برای بارگذاری اطلاعات از آن ها در صفحه ی متعلق به سایت/دامنه ی جاری می شد، توضیحاتی را ارائه دادیم. **JSONP** راه حل بسیار کارآمدی را برای غلبه بر این مانع فراهم می کند.

در مثال های زیر، سعی می کنیم اسکریپت های **PHP** را بر روی سرورس دهنده جاری، اما از یک **subdomain** دیگر فراخوانی کنیم. این اسکریپت آرایه ای از دو کاربر (متشکل از دو شی حاوی اطلاعات مربوط به اسم و سن هریک) را در فرمت **JSON** به نمایش می گذارد. این خروجی با استاندارد **JSONP** کاملاً سازگار خواهد بود، زیرا اطلاعات مورد نظر توسط پارامتر ارسال شده به اسکریپت و پیرامتر باز و بسته محصور شده است.

```
<?php
$users = array
(
    array("name" => "John Doe", "age" => 42),
    array("name" => "Jane Doe", "age" => 39)
);
echo $_REQUEST['callback'] . "(" . json_encode($users) . ")";
?>
```

برای مشاهده ی داده های بازگشتی، این آدرس **URL** را در نوار آدرس مرورگر خود وارد کنید:

<http://tests.jquery-tutorial.net/json.php?callback=test>

نتیجه به این صورت خواهد بود:

```
test([{"name":"John Doe","age":42},{ "name":"Jane Doe","age":39}])
```

اگر پارامتر **callback** را بر روی مقدار دیگر تنظیم کنید (مقدار دیگری را بجای **callback** به عنوان پارامتر پاس دهید)، در آن صورت تغییر را در خروجی مشاهده خواهید کرد. این نوع نشانه گذاری (**notation**)، دقیقاً همان چیزی است که **JSON** عادی را از **JSONP** جدا ساخته و تمایز می بخشد. حال، زمانی که داده های **JSON** به **jQuery** بازگردانده می شوند، آن را به اشیایی **parse** (تجزیه) می کند که به راحتی مانند یک شی جاوا اسکریپت، قابل دسترسی و استفاده می باشد. برای مثال، خروجی بالا به دو شی مختلف تقسیم می شود که هر یک دارای خاصیت (**property**) **age** و **name** مجزا می باشد.

اکنون، صفحه ی مورد نظر را با استفاده از جی کوئری درخواست می کنیم و سپس اطلاعات بازگشتی آن را مورد استفاده قرار می دهیم. توجه داشته باشید که این صفحه را بر روی دامنه ی دیگر (**tests.jquery-tutorial.net**) و غیر از دامنه ی جاری که صفحه را در حال حاضر اجرا می کند، صدا می زنیم.

```
<ul id="ulUsers"></ul>
<script type="text/javascript">
$(function()
{
$.get
(
"http://tests.jquery-tutorial.net/json.php?callback=?",
function(data, textStatus)
{
$.each(data, function(index, user)
{
$("#ulUsers").append($("#<li></li>").text(user.name + " is " + user.age + " years old"));
});
},
"json"
);
});
</script>
```

اگر مباحث مربوط به متدهای **get()** و **post()** را مطالعه کنید، خواهید دید که تنها دو تفاوت عمده بین آن ها وجود دارد: پارامتر **callback** در **URL** و دیگری آرگومان اضافی که مشخص می کند داده ی بازگشتی باید از نوع **"json"** باشد. پارامتر **callback** بر روی مقدار علامت سوال "?" تنظیم شده که سبب می شود جی کوئری یک مقدار به صورت تصادفی ایجاد کند. همان طور که مشاهده می کنید، مقدار این پارامتر، در اسکریپتی که **call** را دریافت می کند، مورد استفاده قرار می گیرد.

پس از دریافت داده، آن را داخل متد **each()** می ریزیم، که داخل داده های بازگشتی حلقه می زند. در هر بار اجرای حلقه یک متد بی نام (**anonymous**) صدا زده می شود و طی آن مجموعه داده های ذخیره شده در

متغیر "user" مورد دسترسی قرار می گیرد. سپس با مقدار کلیدهای **name** و **age** یک رشته ایجاد کرده و آن را داخل یک لیست (به صورت یک تگ **li**) به المان **ul** اضافه می کنیم. در نتیجه، یک لیست **HTML** حاوی کاربران دریافت می کنیم که توسط اسکریپت بازیابی شده است.

