

بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

لغو و بی نتیجه گذاشتن یک درخواست به طور ناگهانی

مدرس : مهندس افشین رفوآ

دوره آموزشی JQuery

### لغو و بی نتیجه گذاشتن یک درخواست به طور ناگهانی (abort request)

گاهی شاید لازم باشد یک درخواست **AJAX** در حال اجرا و فعال را قبل از اینکه کاملاً اجرا شود، لغو کنیم. این وضعیت معمولاً در شرایطی رخ می دهد که کاربر عملی را انجام می دهد که منجر به ارسال چندین درخواست **AJAX** در بازه زمانی کوتاه، به صورت پشت سرهم می شود. به عنوان مثال می توان به قابلیت **auto complete** کادر جستجو (**search box**) اشاره نمود که در آن سعی می کنید مبنی بر **input** (داده های ورودی) جاری کاربر، با پیدا و منطبق کردن عبارت جستجوی مربوطه با آنچه کاربر وارد می کند، در پیدا کردن آیتم مورد نظر به کاربر کمک کنید. این کار با ارسال یک درخواست **ajax** به ازای هر کلیدی که توسط کاربر فشرده شده (و حرفی داخل فیلد جستجو تایپ می شود)، صورت می پذیرد. در سناریوی مزبور، این احتمال وجود دارد که کاربر سریع تر از مدت زمان لازم برای اجرای هر درخواست **ajax**، حروف را تایپ کند. در نتیجه مجبور می شوید تمامی درخواست های ناتمام را پیش از شروع درخواست بعدی، لغو نمایید. مثال:

```
<input type="button" name="btnDoRequest" value="Start" onclick="PerformSimpleCalculation();" />
<script type="text/javascript">
function PerformSimpleCalculation()
{
    $.get("/tests/calc.php", function(data, textStatus)
    {
        alert(data);
    });
}
</script>
```

این مثال یک اسکریپت PHP درخواست کرده که در سمت سرور یک محاسبه ی بسیار پیچیده را انجام می دهد، بدین معنا که محاسبه و بازگردانی نتیجه ی آن 3~ ثانیه طول می کشد. حال دکمه ی مورد نظر را بارها، پشت سرهم فشار دهید. همان محاسبه چندین بار اجرا شده و نتیجه ی نیز چند بار (هر بار با 3 ثانیه تاخیر) اجرا می شود.

خوشبختانه، فراخوانی متد `get()` یا هر تابع **AJAX** دیگری در جی کوئری، یک شی برمی گرداند که علاوه بر امکانات دیگر، دربرگیرنده ی متد `abort()` نیز می باشد. می توانیم این ارجاع را ذخیره کنیم و سپس متد `abort()` را در صورت نیاز برای آن صدا بزنیم. نمونه ی زیر نسخه ی ویرایش شده ی مثال قبلی می باشد:

```
<input type="button" name="btnDoRequest" value="Start" onclick="PerformAbortableCalculation();" />
<script type="text/javascript">
var calculationRequest = null;

function PerformAbortableCalculation()
{
    if(calculationRequest != null)
        calculationRequest.abort();
    calculationRequest = $.get("/tests/calc.php", function(data, textStatus)
    {
        alert(data);
    });
}
</script>
```

مثال را با تعریف یک متغیر معمولی برای ذخیره ی ارجاع `request` تعریف می کنیم. در تابع `PerformAbortableCalculation()`، مقدار بازگشتی متد `get()` را به این متغیر تخصیص می دهیم. اما قبل از اینکار بررسی می کنیم آیا مقدار متغیر مزبور برابر `null` است یا خیر (متد هنوز مورد استفاده قرار نگرفته است). اگر مقدار متغیر `null` نبود، متد `abort()` را برای آن فراخوانی می کنیم. اکنون اگر در مثال بالا، بارها هم بر روی دکمه ی **Start** کلیک کنید، محاسبه تنها یکبار صورت می پذیرد.