

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش متدهای get و post

مدرس : مهندس افشین رفوآ

آموزش متدهای get و post

آموزش برنامه نویسی : آموزش برنامه نویسی PHP

دو شیوه ای که مرورگر کاربر با کمک آن ها اطلاعات را به سرویس دهنده ی وب می فرستد به شرح زیر است.

متد GET {دریافتن}

متد POST {فرستادن}

پیش از این که مرورگر اطلاعات را ارسال کند، ابتدا آن ها را با کمک الگویی بنام کد گذاری URL، رمزگذاری می کند. در این الگو جفت اسم / مقدار {value/name} با علامت تساوی به هم وصل شده، ولی جفت های متفاوت توسط این علامت & که به آن ampersand می گویند از هم جدا می شوند.

name1=value1&name2=value2&name3=value3

جای خالی حذف شده و یا توسط کاراکتر + جایگزین می شود و هر کاراکتر غیر حرفی دیگری که وجود دارد نیز توسط مقدار هگزا دسیمایل جایگزین می گردد. اطلاعات پس از این که رمز گذاری می شوند به server ارسال می شوند.

متد GET

متد GET اطلاعات کد گذاری شده ی کاربر را که به صفحه ی درخواست {page request} ضمیمه شده به server یا سرویس دهنده تحویل می دهد.

آدرس صفحه و اطلاعات کد گذاری شده ی کاربر توسط کاراکتر ؟ از هم تفکیک می شوند.

http://www.test.com/index.htm?name1=value1&name2=value2

متد **GET** رشته طولانی می سازد که در لاگهای سرور، در کادر آدرس **{location box}** مرورگر ظاهر می شود.

متد **GET** تنها می تواند تا 1024 کاراکتر را بفرستد.

زمانی که شما اطلاعات حساس و رمز عبور دارید که باید به **server** فرستاده شود از متد **GET** استفاده نکنید.

GET نمی تواند اطلاعات باینری مثل تصاویر و فایل **word** را به سرور بفرستد.

اطلاعات ارسال شده با متد **GET** با استفاده از متغیر محیطی **QUERY-STRING** قابل دسترسی است.

PHP برای دسترسی به اطلاعات فرستاده شده بوسیله متد **GET**، آرایه شرکت پذیری **\$_GET** را در اختیار شما قرار می دهد.

مثال زیر را با قرار دادن متن برنامه **source code** در **test.php script** انجام دهید.

```
<?php
if( $_GET["name"] || $_GET["age"] )
{
    echo "Welcome ". $_GET['name']. "<br />";
    echo "You are ". $_GET['age']. " years old.";
    exit();
}
?>
<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="GET">
        Name: <input type="text" name="name" />
        Age: <input type="text" name="age" />
        <input type="submit" />
    </form>
</body>
</html>
```

متد POST

متد **POST** اطلاعات را از طریق هدر **HTTP** انتقال می دهد. اطلاعات ابتدا همان گونه که در مورد متد **GET** توضیح داده شد، کد گذاری شده و سپس در داخل هدری که **QUERY-STRING** نام دارد قرار داده می شود.

متد **POST** هیچ گونه محدودیتی در خصوص اندازه اطلاعاتی که قرار است ارسال شود ندارد.

متد **POST** قادر است هم اطلاعات باینری و هم اسکی را ارسال کند.

اطلاعات فرستاده شده با **POST** از هدر **HTTP** گذر می کند، پس امنیت وابسته به پروتکل **HTTP** است. شما با استفاده از **HTTPs** می توانید مطمئن باشید که اطلاعات شما کاملاً ایمن هستند.

PHP برای دستیابی به اطلاعات ارسال شده، از طریق متد **POST** آرایه ی شرکت پذیر **\$_post** را برای کاربر فراهم می کند.

مثال زیر را با قرار دادن متن برنامه {source code} در **test.php script** امتحان کنید.

```
<?php
if( $_POST["name"] || $_POST["age"] )
{
    echo "Welcome ". $_POST['name']. "<br />";
    echo "You are ". $_POST['age']. " years old.";
    exit();
}
?>
<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="POST">

        Name: <input type="text" name="name" />
        Age: <input type="text" name="age" />

        <input type="submit" />
    </form>
</body>
</html>
```

متغیر \$_REQUEST

در php متغیر \$_REQUEST محتوای هر دو متغیر \$_GET، \$_POST و \$_COOKIE را دربردارد. هنگامی که به تشریح مبحث متغیر \$_COOKIE برسیم، درباره ی خود cookie ها نیز مفصل بحث می کنیم. اگر بخواهیم نتیجه ی بدست آمده از اطلاعات فرستاده شده توسط هر دو متد POST و GET را بازیابی و دریافت کنیم باید متغیر \$_REQUEST مورد استفاده قرار دهیم.

مثال زیر را با قرار دادن متن برنامه {source code} در test.php script آزمایش کنید.

```
<?php
if( $_REQUEST["name"] || $_REQUEST["age"] )
{
    echo "Welcome ". $_REQUEST['name']. "<br />";
    echo "You are ". $_REQUEST['age']. " years old.";
    exit();
}
?>
<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="POST">

        Name: <input type="text" name="name" />
        Age: <input type="text" name="age" />

        <input type="submit" />
    </form>
</body>
</html>
```

در اینجا متغیر \$_PHP_SELF اسم خود اسکریپت که فرا خوانده می شود را دربر دارد.