

Formlar

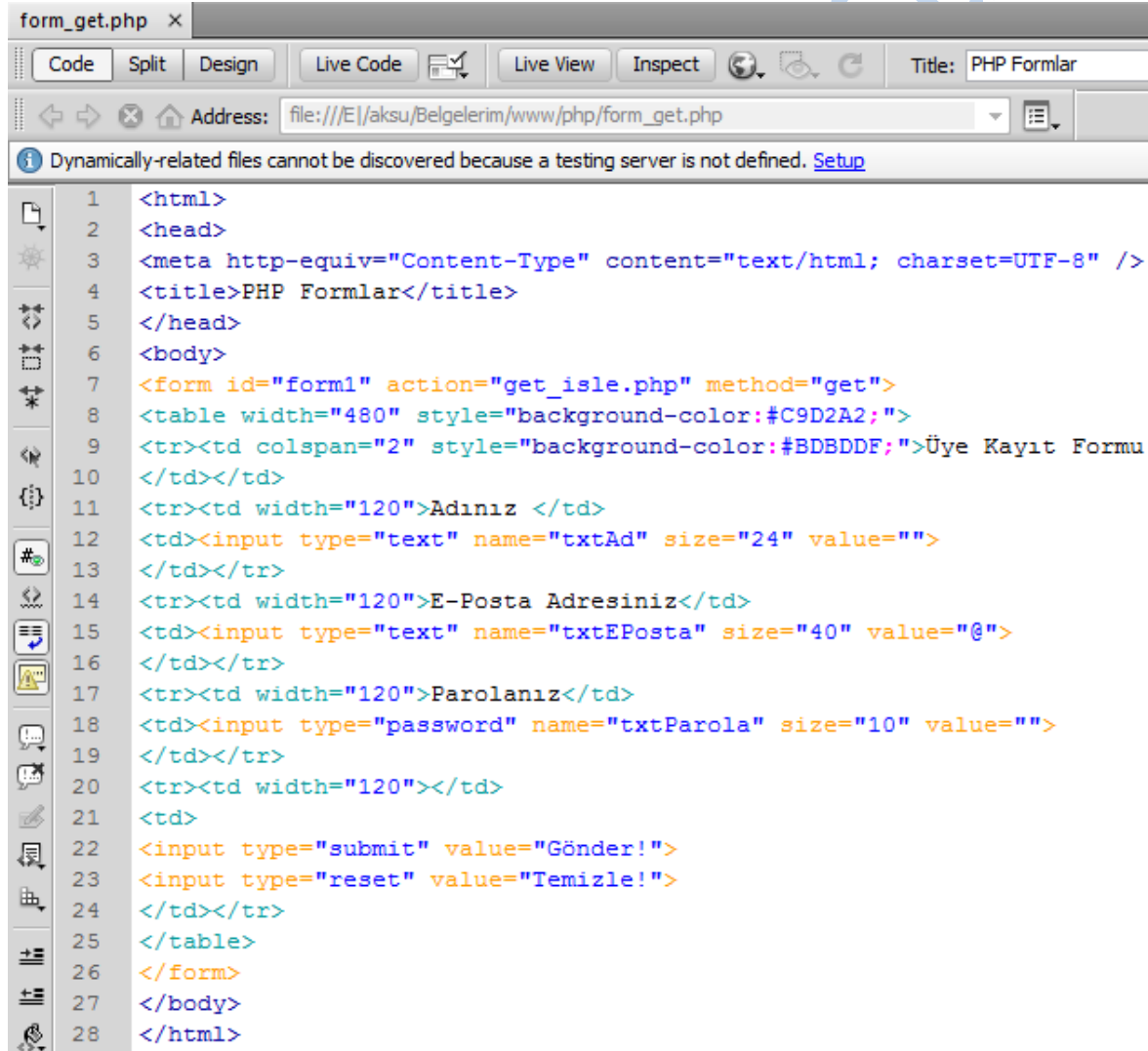
Formlar ile web sayfaları ziyaretçilerden veri alır ve bunları sunucuya ulaştırır. Formlar ile etkileşimli web sayfaları gerçekleştirmek mümkündür.

Web programcısı ve Web tasarımcısı olarak bu bilgileri kullanarak sunucu üzerinde değişik görevleri yerine getiren kodlar yazarız. Ziyaretçiden alınan bilgilere göre sunucudaki veritabanı üzerinde sorgulama ve güncelleme gibi işlemler gerçekleştirilebilir.

Form'dan GET Metoduyla Bilgi Alma

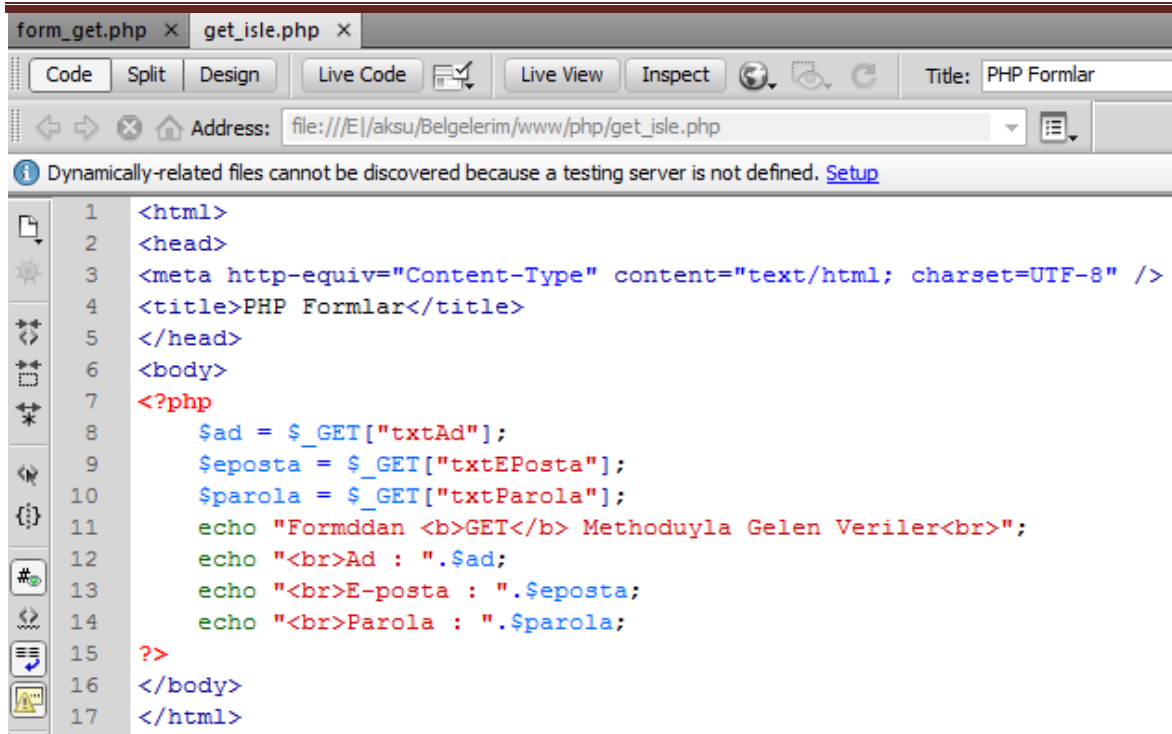
Formdan **GET** metodu ile alınan veriler **\$_GET** dizisine kaydolur.

Basit bir HTML Form'u tasarlayalım. Aşağıdaki HTML sayfasını **form_get.php** adıyla kaydedelim.



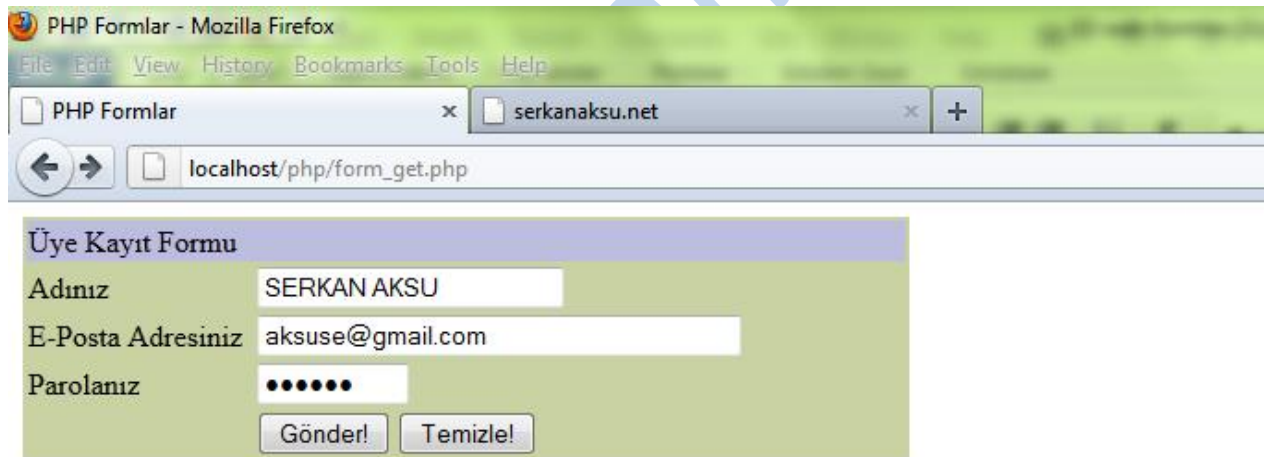
```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <title>PHP Formlar</title>
5 </head>
6 <body>
7 <form id="form1" action="get_isle.php" method="get">
8 <table width="480" style="background-color:#C9D2A2;">
9 <tr><td colspan="2" style="background-color:#BDBDDF;">Üye Kayıt Formu
10 </td></tr>
11 <tr><td width="120">Adınız </td>
12 <td><input type="text" name="txtAd" size="24" value="">
13 </td></tr>
14 <tr><td width="120">E-Posta Adresiniz</td>
15 <td><input type="text" name="txtEPosta" size="40" value="@">
16 </td></tr>
17 <tr><td width="120">Parolanız</td>
18 <td><input type="password" name="txtParola" size="10" value="">
19 </td></tr>
20 <tr><td width="120"></td>
21 <td>
22 <input type="submit" value="Gönder!">
23 <input type="reset" value="Temizle!">
24 </td></tr>
25 </table>
26 </form>
27 </body>
28 </html>
```

Veri girişinin yapıldığı form sayfasının kodları



```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <title>PHP Formlar</title>
5 </head>
6 <body>
7 <?php
8     $ad = $_GET["txtAd"];
9     $eposta = $_GET["txtEPosta"];
10    $parola = $_GET["txtParola"];
11    echo "Formddan <b>GET</b> Methoduyla Gelen Veriler<br>";
12    echo "<br>Ad : ".$ad;
13    echo "<br>E-posta : ".$eposta;
14    echo "<br>Parola : ".$parola;
15    ?>
16 </body>
17 </html>
```

Formdan GET yöntemiyle gönderilen verileri işleyen PHP kodları



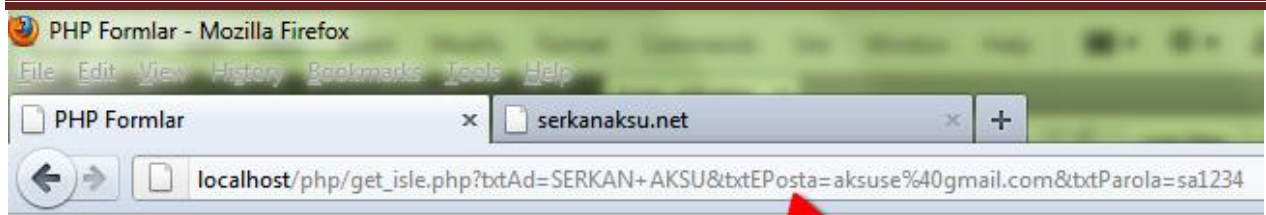
Üye Kayıt Formu

Adınız

E-Posta Adresiniz

Parolanız

Kullanıcıların bilgi girişi yaptığı form sayfası verileri GET metoduyla gönderecek



Formddan **GET** Methoduyla Gelen Veriler

Ad : SERKAN AKSU
E-posta : aksuse@gmail.com
Parola : sa1234

Formdan gönderilen veriler [get_isle.php](#) adlı sayfada işlenerek sayfaya yazdırılır

Browser, **GET** yoluyla veri göndereceği zaman, Form'daki bütün verileri **URL-Encoding** denen sistemle kodlar; Form'un alan adlarına o alanlara ziyaretçinin yazdığı bilgileri eşittir işaretiyle ekler; bu tür **alan=girdi** çiftlerinin arasına & (ve işareti) koyar ve gönderir. Web sunucu, bu bilgileri alınca, önce kendi oluşturduğu bazı değişkenlere (hem **QUERY_STRING**, hem de **\$_GET** dizisine) yazar ve sonra URL hanesinde adı yazılı olan programa (sayfaya) verir.

http://localhost/php/get_isle.php?txtAd=SERKAN+AKSU&txtEPosta=aksuse%40gmail.com&txtParola=sa1234

GET metodu ile gönderilen verilerin **URL**'de (adres çubuğunda) açık olarak gitmesi önemli bir dezavantajdır. Kullanıcının girmiş olduğu kredi kartı numarası ve parola gibi bilgiler bu şekilde gönderilirse istenmeyen kişilerin eline geçme olasılığı çok yüksektir.

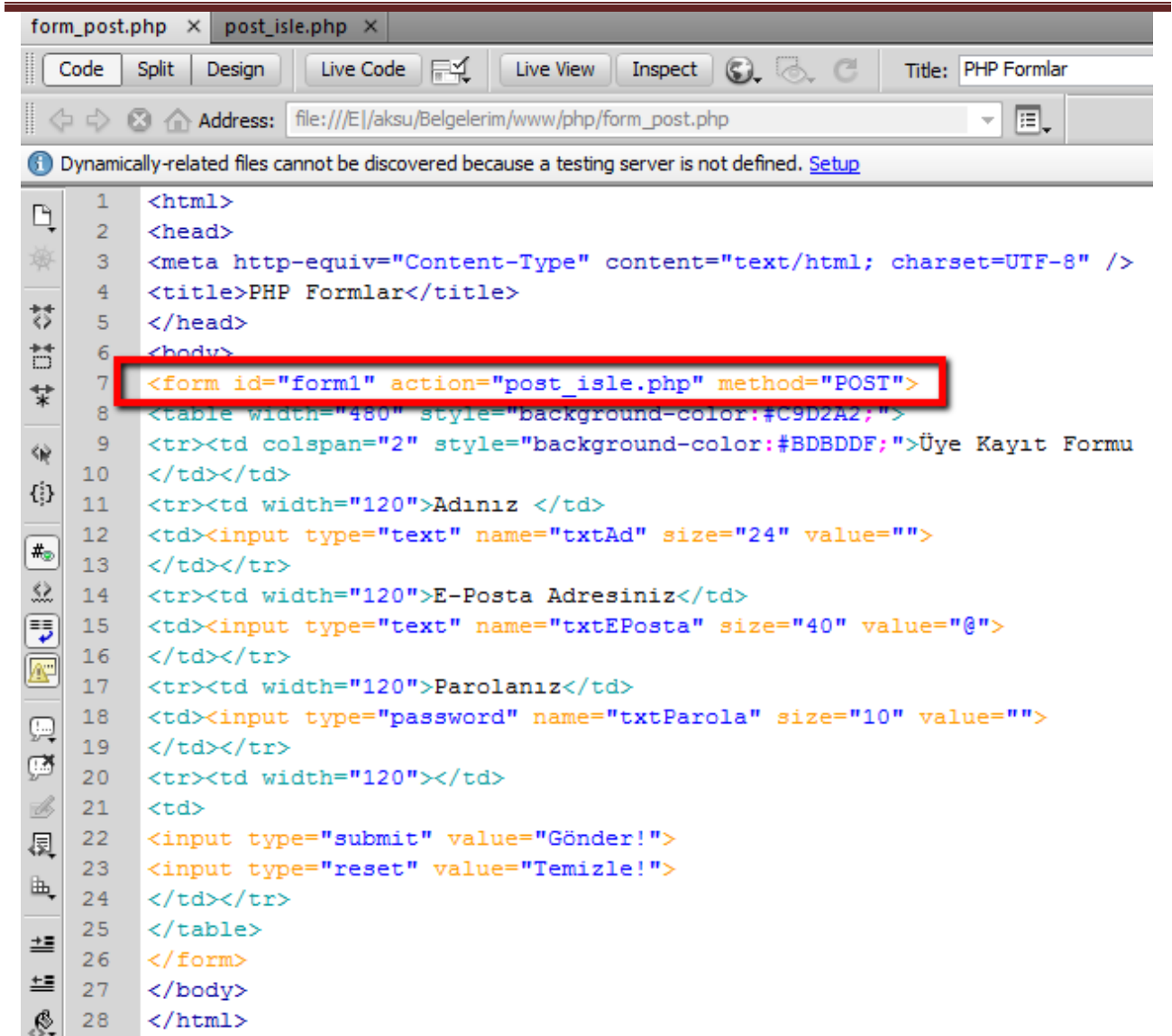
Bunun gibi gizlilik gerektiren form işlemlerinde POST metodu kullanılır.

Form'dan POST Metoduyla Bilgi Alma

POST metodu ile alınan veriler ise **\$_POST** dizisine kaydolur.

Basit bir HTML Form'u tasarlayalım. Aşağıdaki HTML sayfasını **form_post.php** adıyla kaydedelim.

Browser, **POST** yoluyla veri göndereceği zaman, Form'daki bütün verileri sunucuya gizli olarak gönderir. Web sunucu, bu bilgileri alınca **\$_POST** dizisine yazar ve sonra **ACTION** ile belirtilen **PHP** sayfasına gönderir. POST metoduyla sunucuya gönderilen veriler adres satırında görüntülenmez. Böylece verilerin gizliliği sağlanmış olur.

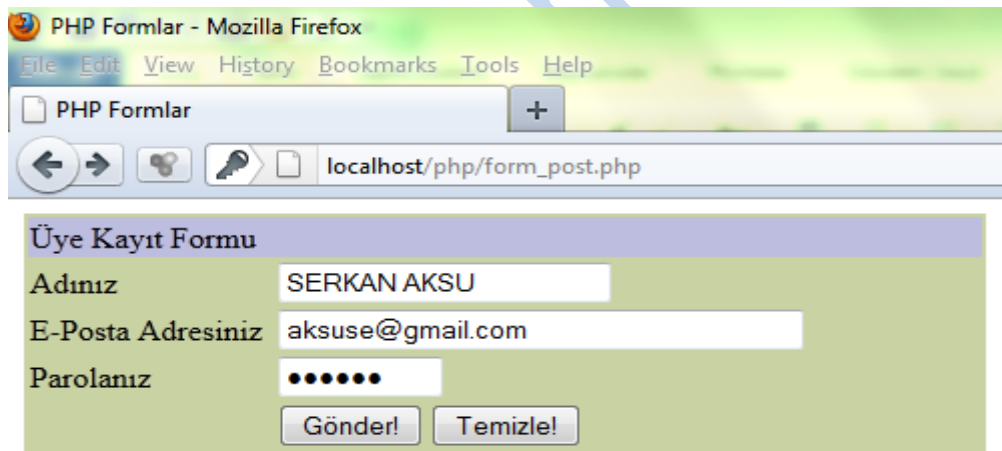


```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <title>PHP Formlar</title>
5 </head>
6 <body>
7 <form id="form1" action="post_isle.php" method="POST">
8 <table width="480" style="background-color:#C9D2A2;">
9 <tr><td colspan="2" style="background-color:#BDBDDE;">Üye Kayıt Formu
10 </td></td>
11 <tr><td width="120">Adınız </td>
12 <td><input type="text" name="txtAd" size="24" value="">
13 </td></tr>
14 <tr><td width="120">E-Posta Adresiniz</td>
15 <td><input type="text" name="txtEPosta" size="40" value="@">
16 </td></tr>
17 <tr><td width="120">Parolanız</td>
18 <td><input type="password" name="txtParola" size="10" value="">
19 </td></tr>
20 <tr><td width="120"></td>
21 <td>
22 <input type="submit" value="Gönder!">
23 <input type="reset" value="Temizle!">
24 </td></tr>
25 </table>
26 </form>
27 </body>
28 </html>
```

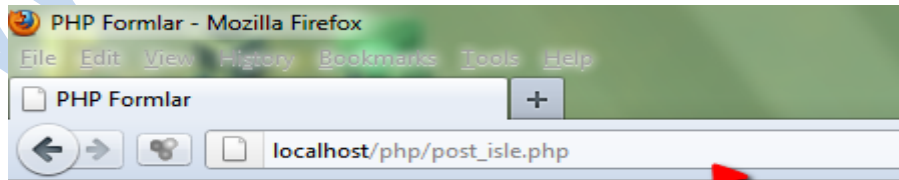
Veri girişinin yapıldığı form sayfasının kodları

```
form_post.php x post_isle.php x
Code Split Design Live Code Live View Inspect Title: PHP Formlar
Address: file:///E:/aksu/Belgelerim/www/php/post_isle.php
Dynamically-related files cannot be discovered because a testing server is not defined. Setup
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <title>PHP Formlar</title>
5 </head>
6 <body>
7 <?php
8     $ad = $_POST["txtAd"];
9     $eposta = $_POST["txtEPosta"];
10    $parola = $_POST["txtParola"];
11    echo "Formddan <b>POST</b> Metoduyla Gelen Veriler<br>";
12    echo "<br>Ad : ".$ad;
13    echo "<br>E-posta : ".$eposta;
14    echo "<br>Parola : ".$parola;
15    ?>
16 </body>
17 </html>
```

Formdan POST yöntemiyle gönderilen verileri işleyen PHP kodları



Kullanıcıların bilgi girişi yaptığı form sayfası verileri POST metoduyla gönderecek



Formddan **POST** Metoduyla Gelen Veriler

Ad : SERKAN AKSU
E-posta : aksuse@gmail.com
Parola : sa1234

Formdan gönderilen veriler **post_isle.php** adlı sayfada işlenerek sayfaya yazdırılır