

L1_2.3 GET und POST

Betrachten wir uns folgendes Formular, welches den Besucher auffordert, eine PIN einzugeben. Nach Analyse des Programmcodes erkennen wir, dass die Daten an das Zieldokument „/tools/ftp.php“ gesendet werden. Da es sich bei den einzugebenden Daten um sensible Informationen handelt, nutzen wir ein Feld zur verdeckten Eingabe (type=“password“). Die weiteren Attribute „size“ und „maxlength“ wirken sich lediglich auf die Darstellung und auf die Anzahl der zulässigen Eingaben aus.

Zieldokument

PIN
 Geben Sie Ihre PIN ein:

```

<form action="/tools/ft.php">
  <fieldset>
    <legend>PIN</legend>

    <label for="p">Geben Sie Ihre PIN ein:</label><br>
    <input id="p" name="pin" type="password" size="4" maxlength="4"><br>

    <input type="submit" value="Prüfen" class="btn">
  </fieldset>
</form>

```

Name des Feldes

Lösen wir das Formular aus, indem wir auf die Schaltfläche „Prüfen“ klicken, werden die Daten vom Browser eingesammelt, in der URL codiert und an das Zieldokument gesendet. Betrachtet man sich die Adresszeile des Browsers, dann sieht diese wie folgt aus:

localhost/tools/ft.php?pin=1234

Zieldokument
Query-Komponente

Wie erwartet, werden die Daten in der Query-Komponente der URL codiert. Dazu wird der Name des Eingabefeldes als Parametername und die Benutzereingabe als Parameterwert verwendet. Die Adresszeile ist von jedem einsehbar. Diese Art der Codierung der Daten ist in diesem Fall sehr ungünstig. Die Eingabe erfolgt zwar verdeckt, nach dem Auslösen des Formulars ist diese jedoch im Klartext in der Adresszeile zu erkennen.

J2	BPE 7: Dynamische Webseiten Informationsmaterial	Wirtschaftsinformatik
----	--	-----------------------

Über ein zusätzliches Attribut namens „method“ im form-Element können wir Einfluss auf die Art der Codierung der eingegebenen Daten nehmen. Als Attributwert stehen zwei Alternativen zur Verfügung:

GET	<p>Die zu versendenden Daten werden in der URL in Form einer Query-Komponente codiert und sind damit in der Adresszeile sichtbar.</p> <p>Vorteil:</p> <p>Beim Anlegen eines Lesezeichens werden die Daten ebenfalls gespeichert. Die Seite lässt sich somit in gleicher Form erneut aufrufen.</p> <p>Beispiel: Barcodes können beispielsweise über die Google Chart-API erstellt werden. Alle Informationen zum Barcode werden in der URL codiert. Speichert man diese URL, dann kann diese jederzeit wieder aufgerufen und damit der Barcode erzeugt werden.</p> <p>Nachteil:</p> <p>Wie im Beispiel gesehen, werden die Daten im Klartext in der Adresszeile dargestellt. Dies betrifft auch angelegte Lesezeichen oder der im Browser gespeicherte Verlauf der besuchten Seiten. Werden sensible Daten übertragen, ist dies nicht gewünscht. Ein weiterer Nachteil ist die beschränkte Länge einer URL. Diese ist browserabhängig, darf aber in der Regel eine maximale Länge von 2048 Zeichen nicht überschreiten. Die GET-Methode eignet sich also nicht zur Übertragung von großen Datenmengen.</p>
POST	<p>Bei dieser Methode werden die Daten in der Anfrage intern gespeichert. Sie sind damit für den Benutzer nicht sichtbar.</p> <p>Vorteil:</p> <p>Die Daten sind für den Benutzer nicht sichtbar und die Länge bzw. die Größe der Daten unterliegt keinerlei Beschränkungen. Die Daten werden nicht im Cache des Browsers gespeichert und erscheinen auch nicht im Browserverlauf.</p> <p>Nachteil:</p> <p>Aktualisiert der Besucher die Seite manuell, so müssen die Daten erneut eingegeben werden – eine entsprechende Meldung haben Sie sicherlich schon gesehen.</p> <div data-bbox="622 1675 1133 1877" data-label="Image"> </div> <p>Dabei besteht die Gefahr, dass die Daten versehentlich mehrmals gesendet werden, was etwa bei einem Bestellformular unerwünschte Doppelaufträge auslösen kann.</p>

J2	BPE 7: Dynamische Webseiten Informationsmaterial	Wirtschaftsinformatik
----	--	-----------------------

In unserem Fall eignet sich somit die Übertragungsmethode „post“. Fehlt die Angabe einer Methode, findet standardmäßig die Methode „get“ Verwendung.

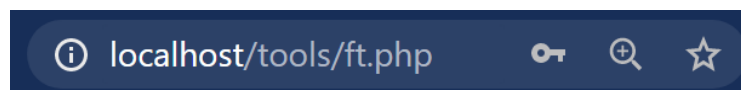
Verbesserte Variante:

```
<form action="/tools/ft.php" method="post">
  <fieldset>
    <legend>PIN</legend>

    <label for="p">Geben Sie Ihre PIN ein:</label><br>
    <input id="p" name="pin" type="password" size="4" maxlength="4"><br>

    <input type="submit" value="Prüfen" class="btn">
  </fieldset>
</form>
```

Die Adresszeile sieht dem Auslösen der Anfrage nun wie folgt aus, die Benutzereingaben sind nicht mehr sichtbar:



Hinweis:

Die Übertragung per „post“ ist nur vermeintlich sicher. Werden die Daten unverschlüsselt versendet, landen die Benutzereingaben im Klartext im Datenstrom und können von dort abgegriffen werden.

Weiter beinhalten alle modernen Browser Werkzeuge (→ Entwicklertools), um Übertragungen zu analysieren. Einen solchen Auszug finden Sie auf der folgenden Seite, auch hier ist das Kennwort, neben zahlreichen anderen Informationen, im Klartext zu erkennen.

J2	BPE 7: Dynamische Webseiten Informationsmaterial	Wirtschaftsinformatik
----	--	-----------------------

▼ General

Request URL: http://localhost/tools/ft.php
Request Method: POST
Status Code: 200 OK
Remote Address: [::1]:80
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers [view source](#)

Connection: Keep-Alive
Content-Length: 7836
Content-Type: text/html; charset=UTF-8
Date: Tue, 07 Jul 2020 08:01:23 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.28 (Win32) OpenSSL/1.0.2l PHP/7.1.10
X-Powered-By: PHP/7.1.10

▼ Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE;de;q=0.9,en-US;q=0.8,en;q=0.7,da;q=0.6,fr;q=0.5,nb;q=0.4,la;q=0.3
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 8
Content-Type: application/x-www-form-urlencoded
Host: localhost
Origin: http://localhost
Referer: http://localhost/formular/login/
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36

▼ Form Data [view source](#) [view URL encoded](#)

pin: 1234