Ajax und JSON

Die JavaScript Object Notation, kurz JSON, stellt eine Alternative zu XML als universelles Datenaustauschformat dar.
Hier dient es zum Transport zwischen PHP- und JavaScript-Anwendungen.
PHP hat einige eingebaute Funktionen um JSON zu behandeln.

Eine allgemeine Nutzung von JSON ist, Daten zu / von einem Web-Server auszutauschen.

Wenn Daten von einem Web-Server empfangen werden, sind die Daten immer ein String, d.h. sie werden als String vom Server zum Webbrowser übertragen. Wenn der String den Browser erreicht, muss er durch das Skript in ein JavaScript-Objekt umgewandelt werden. Dieser Vorgang wird als Deserialisierung bezeichnet. Dazu dient die Methode

parse()

des integrierten JSON-Objekts. Da es sich dabei um ein globales Objekt handelt, kann man es verwenden, ohne zuerst eine Instanz davon zu erstellen. D.h. die Daten werden mit JSON.parse () geparst und dann zu einem JavaScript-Objekt gemacht.
Wenn der String analysiert ist, kann das Skript auf die Daten in dem Objekt zugreifen und die HTML-Elemente erstellen, die auf der Seite angezeigt werden sollen-
Diese HTML-Elemente werden mithilfe der Eigenschaft

innerHTML

zu der Seite hinzugefügt.

Das JSON-Objekt verfügt auch über die Methode

stringify(),

die Objekte in Strings mit JSON-Schreibweise umwandelt, sodass sie vom Browser zurück an den Server gesendet werden können. Dieser Vorgang wird Serialisierung des Objekts genannt.

Wenn ein Benutzer durch seine Aktionen auf der Seite die Daten in dem JavaScript-Objekt geändert hat, z.B. durch Ausfüllen eines Formulars, kann man mit dieser Methode die auf dem Server gespeicherten Informationen aktualisieren.

Daten mitsenden
Kann man mit GET oder POST machen.
Wenn Sie GET als Methode verwendet haben, können Sie die Parameter direkt in die URL schreiben. Im folgenden Codeausschnitt werden einem Skript ein paar Parameter übergeben:

…

```
xmlhttp.open("GET","test.php?name=wolf&plz=89000",true);
```

```
xmlhttp.send();
```

…

Bei POST hingegen können Sie die Daten bei der Methode send() desXMLHttpRequest-Objekts mitangeben. Zusätzlich müssen Sie bei POST noch einen speziellen HTTP-Header mitsenden. Dies machen Sie mit der Methode setRequestHeader(). Der erforderliche HTTP-Header lautet Content-Type, und der entsprechende Wert ist“application/x-www-form-urlencoded”. Folgendermaßen können Sie also mit POST dem Skript die Daten übergeben:

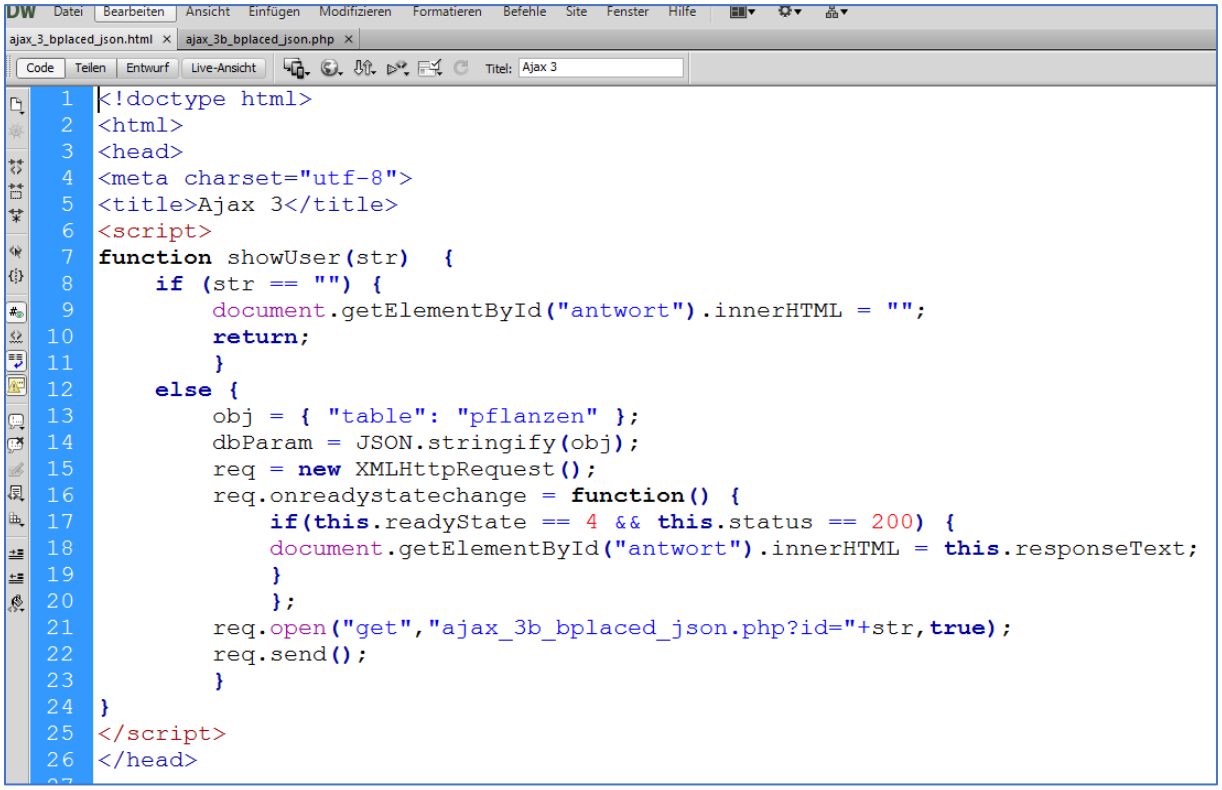
…

```
xmlhttp.open("POST","test.php", true);
xmlhttp.setRequestHeader("Content-type, application/x-www-form-urlencoded");
xmlhttp.send("name=wolf&plz=89000");
```

…

Übung aufbauend auf „7.ajax.docx“

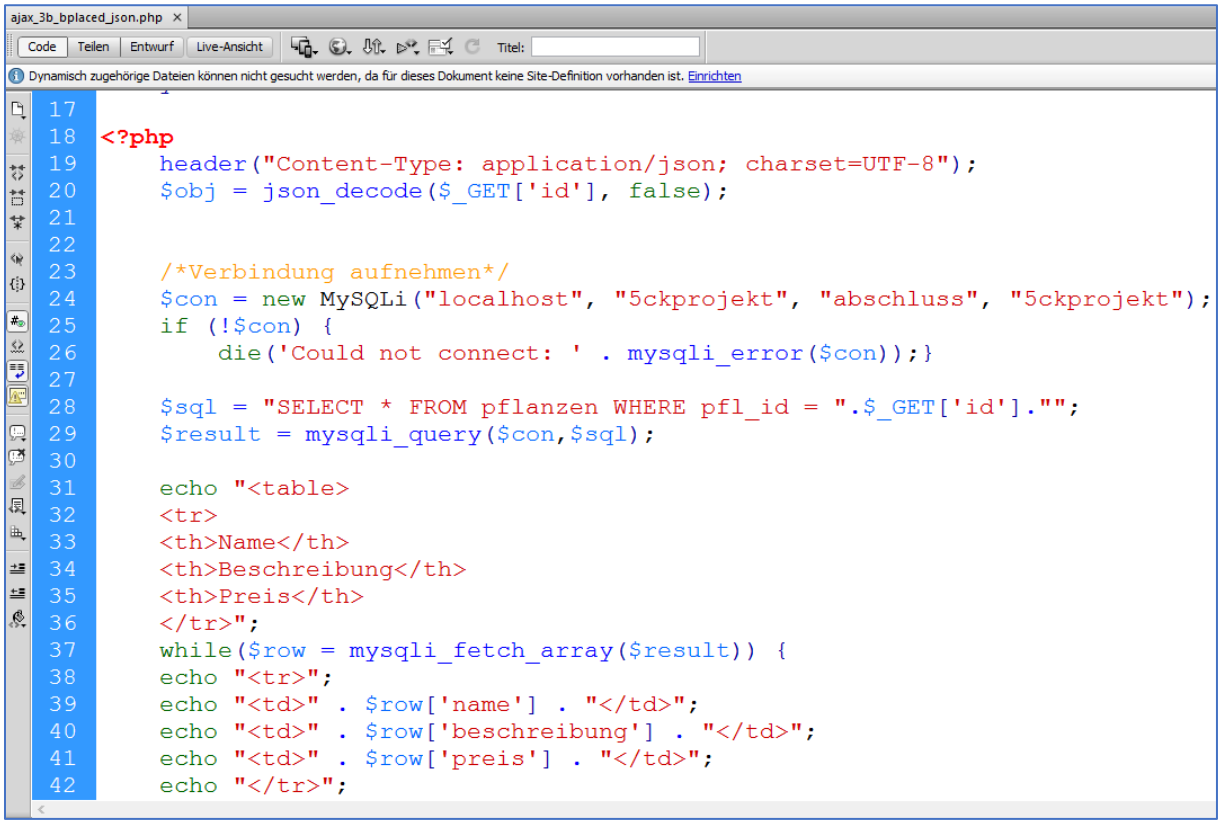
ajax_3_bplaced_json.html



```
27
28  <body>
29  <form>
30  <select name="users" onchange="showUser(this.value)">
31    <option value="">Wähle eine Pflanze:</option>
32    <option value="1">Feldahorn</option>
33    <option value="2">Warzenbirke</option>
34    <option value="3">Pfeifenwinde</option>
35    <option value="4">Filigranfarn</option>
36    <option value="5">Kahle Apfelbeere</option>
37    </select>
38  </form>
39  <br>
40  <div id="antwort"><b>Hier wird die Info angezeigt...</b></div>
41
42  </body>
43  </html>
```

Änderungen in der PHP-Datei:

ajax_3b_bplaced_json.php



Ergebnis mit Fehlermeldung: