

Hands On: Catching Data at the Server

In this lesson, we will catch data at the server.

WE'LL COVER THE FOLLOWING

- **EXERCISE 4-4:** Visualizing form data
 - Step 1:
 - Step 2:
 - Step 3:
 - Step 4:
 - Step 5:
 - Step 6:
 - Step 7:

It is good to know that the server receives the form data, but it would be even better if it could visualize it. In the next exercise, you will change the `formprocessor.html` file so that it will display the data received by the server page.

EXERCISE 4-4: Visualizing form data

In this exercise, you will use a prepared project that can be found in the `Exercise-04-04` folder below.

Although this exercise uses client-side JavaScript to extract the form parameters, it mimics the operation the server-side can carry out.

```
/*! jQuery v3.4.1 | (c) JS Foundation and other contributors | jquery.org/license */  
!function(e,t){  
  "use strict";  
  "object"==typeof module&&"object"==typeof module.exports?module.e
```

To display form data, follow these steps:

Step 1:

The extraction of form parameters is handled by the `getFormParameters()` JavaScript method.

Do not feel intimidated, if you do not understand how the code works, just accept that it does what its name says. A short jQuery code snippet turns the form parameter key-value pairs into an HTML table.

formprocessor.html

```
...
<body>
  <h1>Form Data</h1>
  <table>
    <thead>
      <tr><th>Property</th><th>Value</th></tr>
    </thead>
    <tbody>
      <!-- Filled up by JavaScript -->
    </tbody>
  </table>
  <script src="scripts/jquery.min.js"></script>
  <script>
    $(function () {
      var params = getFormParameters();
      for (var parName in params) {
        var row = $("<tr>")
          .append($("<td>").html(parName))
          .append($("<td>").html(params[parName]));
        $("tbody").append(row);
      }
    });

    function getFormParameters() {
      var uri = window.location.search;
      var result = {};
      if (uri.indexOf("?") === -1) {
        return {};
      }
      var query = uri.slice(1);
      var params = query.split("&");
      var i = 0;
      while (i < params.length) {
        var parameter = params[i].split("=");
        result[parameter[0]] = parameter[1];
        i++;
      }
      return result;
    }
  </script>
</body>
...
```

Step 2:

Step 2:

Start the app. When the registration form is displayed, simply click Register. The form is posted to the server and the browser will display the form data, as shown in the image below:

Form Data

Property	Value
salutation	Mr.
fname	
lname	
email	
membership	gold
login	
pwd	
pwd2	
dev	on
comments	

Form data—default values

As you can see above, the form contains empty strings for all text fields, “Mr.” for the salutation field, “gold” for the membership field, and signals that the dev checkbox is turned on.

Step 3:

Use the Back button of the browser to go back to the **index.html** page. In the code editor above, change the option values of the salutation field, and remove the selected attribute from the second option:

HTML index.html

```
<select id="salutation" name="salutation">
  <option label="" value="" />
  <option label="Mr." value="mrval" />
  <option label="Mrs." value="mrsval" />
  <option label="Ms." value="Ms." />
  <option label="Dr." value="Dr." />
</select>
```

Step 4:

Remove the checked attribute from the “gold” membership option, and from the “dev” checkbox.

Step 5:

Turn back to the browser and click Register again. Now, you see the form data as shown in the image below. The salutation field is empty, and there is no membership value.

Form Data

Property	Value
salutation	
fname	
lname	
email	
login	
pwd	
pwd2	
comments	

Form data—No salutation and membership option specified

Step 6:

Use the Back button of the browser to go back to the form. Fill each text field, specify a membership option, and check at least two checkboxes in the Sessions section. Click Registration and take a look at the form data.

The image below shows the data we have specified. Now, you can see a selected membership option (“silver”), the dev, and the csharp fields with the value set to “on” (I checked the “Development” and “C#” options in Sessions).

Form Data

Property	Value
salutation	mrval
fname	Istvan
lname	Novak
email	myemail@mycompany.hu
membership	silver
login	istvan
pwd	secret
pwd2	secret
dev	on
csharp	on
comments	Surprise me!

Form Data—all fields have values

Step 7:

Close the browser.

In the *next lesson*, we will understand the workings behind the above exercise.