Seminarium 4

Applikationer för internet, grundkurs, ID1354

Maher Jabbar | maherj@kth.se

Okt 28, 2016

Innehåll

- **1 Introduction** **3**
- **2 Literature study** **4**
- **3 Method** **5**
- **4 Result** **6**
- **5 Discussion** **12**
- **6 Comments about the course** **12**

1 Introduktion

The task for this seminar was to choose between mandatory and optional task 1. For this, seminar i chosed mandatory task. then i should use JavaScript and AJAX for at least read and write and delete recipe comments. To navigate between the index, recipes and calendar pages, and to log in / log out, can still be done by downloading the entire HTML document. It is also permissible to store all the data, such as recipes comments in HTML elements in the DOM tree instead of implementing a ViewModel layer. Using jQuery is voluntary and easier but not required as saying.
Objectives for this workshop:
• Learns JavaScript.
• Learn how to use jQuery to facilitate event handling and DOM updates.
• Learn how to use a frame, for example, Knockout improving the architecture and design of JavaScript client.

2 Literature Study

It was the first time i coded in JavaScript so i had to take help from the lectures 12-13 when they got to see how to use JavaScript syntax and each quite similar to PHP code. Also took the help of assistants in the tutorial when you got lots of great help from them and that you googled around pretty much on stack overflow and w3schools.

3 Method

Netbeans has been used to solve this seminar task and has also assumed to use
code from seminar 3 for the task. Firstly i create an exact javascript file with various features that would download and read comments and write and also delete comments without reloading the page the reason for that is that you should get a very smoother web site through the use of javascript.

4 Resultat

```
9
10          <script src="jquery-3.1.1.js"></script>
11          <script src="comments.js"></script>
12      </head>
13      <body onload="fetchComments('<?=$dish?>')">
```

FIGURE 1. FILES AND FUNCTIONS

First, as a JavaScript file retrieved from jQuery side to facilitate all the encoding without needed to write lots of code. Since then created a separate JavaScript file, using the <script src = ""> so it points to the file that you created o it is included in the file that pointed away from almost as require_once (). With onload executes a JavaScript that is now in our case the function fetchComments, it retrieves and reads dish that's when the comments in the page for example. meatballs. This is done immediately after a page has loaded.

```
function addComment(data) {

        var username = document.createTextNode(data.username);
        var msg = document.createTextNode(data.comment);
        var buttontxt = document.createTextNode("Delete");

        var author = document.createElement("p");
        var message = document.createElement("div");
        var button = document.createElement("button");
        var comment = document.createElement("div");

        comment.setAttribute("data-id", data.id);
        button.className = "delete";
        author.className = "name";
        message.className = "comment";

        button.appendChild(buttontxt);
        author.appendChild(username);
        message.appendChild(msg);
        comment.appendChild(author);
        comment.appendChild(message);
        comment.appendChild(button);
        document.getElementById("comments").appendChild(comment)

}
```

FIGUR 2. THE FUNCTION ADD COMMENT

createTextNode create a text node with the specified text. Here you create text nodes data.username and data.comment and that's when the name of the user and the comment is written. createElement creates an element node with a specified name as one can see in the picture. setAttribute () method adds the specified attribute of an element and give it a value. It makes it appendChild add something between tags that you can see on the picture. Last, so put everything into one document and get an ID "comments". The recipe pages so are the a with id = "comments" it is away addComments function we have above, that is what will also appear in the DOM tree. A picture of the DOM is in the discussion section showing how the comments are stored in the browser

```
        <div class = "comments">
            <form id="publish" onsubmit="postComment();
```

FIGURE 3. FUNCTION POSTCOMMENT ON RECIPE PAGE

With onsubmit will run a JavaScript function, and in our case it is Postcomment as a form sent.

```
27  function postComment(){
28
29          var formdata = $('#publish').serialize();
30
31          var insert = $.post('makeComment.php', formdata);
32
33          insert.done(function (data) {
34
35                  addComment(data);
36          });
37  }
38
```

FIGURE 4. THE FUNCTION POSTCOMMENT

The comments posts by helping of this function on the recepts side. first will be created a variable that has a method $ ('# publish'). Serialize (); #publish is an ID in a form available in the recipe pages where Postcomment will be executed and the Serialize () creates a URL encoded string by serializing the form values. Then it will create a variable with a ajax method when complete so posted comments on the recipe page to comment.

```
function fetchComments(dish){

        var fetch = $.get('getComments.php?dish=' + dish);
        fetch.done(function (data){

                for (var i = 0, le = data.length; i < le; i++) {
                        addComment(data[i]);
                }
        })
}
```

FIGURE 5. THE FUNCTION FETCH COMMENTS

With the help of this method is retrieved and read the comments on the recipe pages with Javascript instead of PHP as it did before. First, it will create a variable that has an AJAX get method retrieves comments from PHP file getComments. When its completed then, then it run the function and the function we have a for loop in which it adds to every comment, we have written in recipe pages.

```
$(document).on('click', '.delete', function() {

        var fd = new FormData();
        var comment = $(this).parent();
        fd.set('data-id', comment.attr('data-id'));

        $.ajax({
          url: 'deleteComment.php',
          data: fd,
          processData: false,
          contentType: false,
          type: 'POST',
          success: function(data){
                comment.remove();
          }
        });
});
```

FIGURE 6. THE FUNCTION TO DELETE A COMMENT

Here we have the operation to remove a comment when a user wants to delete a comment, click on the Delete button that will trigger the code. Each comment has a data id using AJAX method will delete comments by deleteComment.php included and runs every time you click the Delete button. And then also runs comment.remove function () and the comment will be deleted.

```php
$data = $controller->storeComment($name, $comment, $dish);

header('Content-Type: application/json');
echo json_encode($data);
```

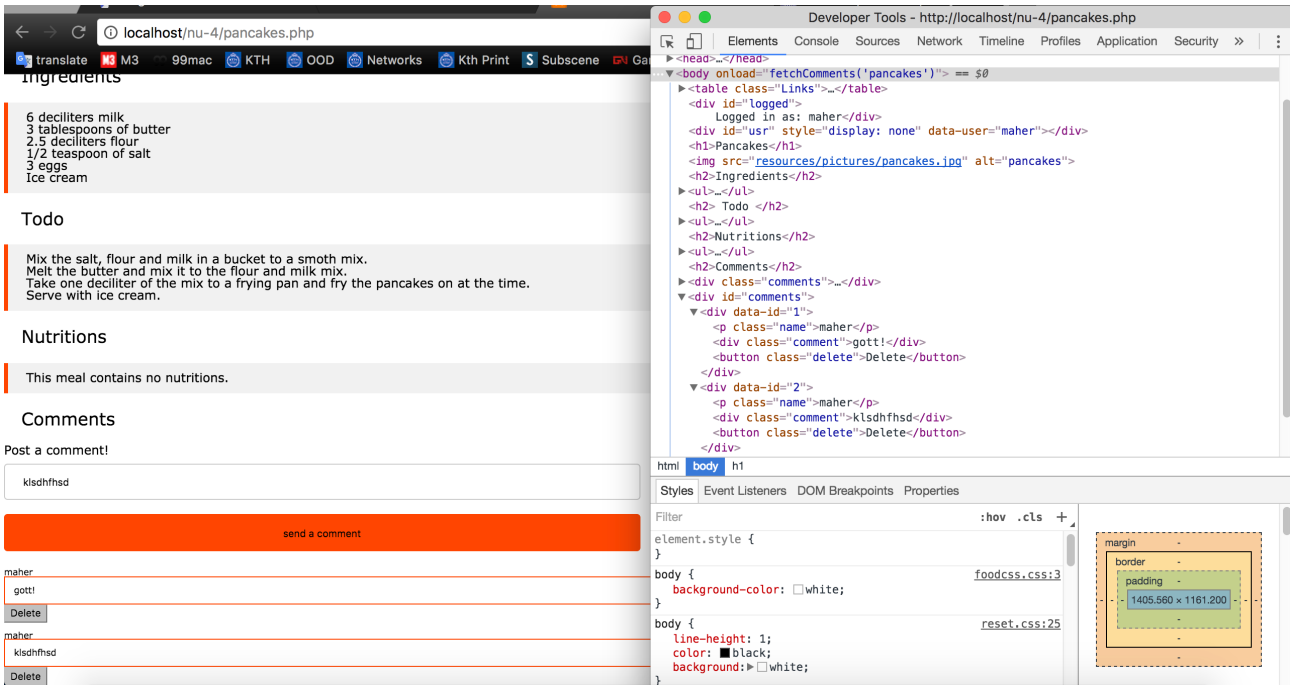
FIGURE 7. THE USE OF JSON

JSON is a format for storing and transporting data. JSON is often used when data is sent from a server to a web page. json_encode returned JSON representation of a value such that JavaScript can read the value. In our case it is $ data JavaScript to read then JSON is a tool to do it.

5 Diskussion

this was the first time I used JavaScript, so I had to check a lot of videos on JavaScript, and that I have been had to be check on the slidsen from the lectures as soon as I did it, it was easy to get in the seminar and everything work without problem . Because this seminar task did not have as much as the last one since JavaScript is quite similar to PHP, but that some syntaxes differ.

The report will show how the recipe comment is stored in the client (browser)



Recipes comments stores all data recipe for HTML elements in the DOM tree as you can see in the picture to the right in the DOM tree, and it is done with the help of the features that I had created. That's when I used "document.getElementById (" comments "). AppendChild (comment)" located in addComments function in Figure 2. The "comments" is an ID that is added to a div in the code of recipe pages. Then it is stored even in the DOM tree.

I've gotten a little better grip on the use of Javascript after i checked a lot of videos and slidsen etc. As well as to the use of jQuery facilitated event management and updating of the DOM. This seminar was real fun to get involved with then instead of the sides would be reloaded when you commented and removed the comments as they would with Javascript and AJAX code a bit to get the go a lot smoother.

6 Kommentarer om kursen

I spent about 15 hours to keep on with the task and write the report, it was not as hard as the last seminar task. I think this course is well structured.