

# Labb rapport 4

Internet Applications, ID1354

Evan Saboo  
saboo@kth.se

2015-10-15

## 1 Introduktion

Målet med fjärde labbet är att lära sig om JavaScript och använda jQuery för att kunna underlätta händelsehantering och uppdateringar av DOM. Man ska också lära sig använda en JavaScript ram t.ex. Knockout för att förbättra arkitekturen och designen av JavaScript klienten.

Labbens uppgift går ut på att implementera Javascript kod som använder AJAX (Asynchronous JavaScript) för att hämta data från servern och uppdatera klienten (webbläsaren). JavaScript koden måste också använda jQuery.

Jag samarbetade med David Nartey ([dnartey@kth.se](mailto:dnartey@kth.se)) med uppgiften.

## 2 Litteratur Studie

Uppgiften utfördes med hjälp av dessa källor:

\* Jag gick igenom alla tre föreläsningarna om JavaScript och dess funktioner.

\* I boken 'Programming the WWW 6<sup>th</sup> av Sebesta Robert W' gick jag igenom några delar i kapitel 4 och kapitel 5:

- 4.1 Overview of JavaScript
- 4.4 Primitives, Operations, and Expressions
- 4.6 Control Statements
- 4.8 Arrays
- 4.9 Functions
- 5.2 The Document Object Model

\* Jag kollade också i Google och Youtube guider för att kunna utföra JavaScript funktionen genom AJAX via jQuery.

## 3 Metod

Vi valde att implementera en JavaScript funktion som kollar om användarnamnet finns i databasen när en ny användare registrerar sig. PHP funktionen kontrollerar om det angivna användarnamnet är redan registrerat i databasen. Resultatet av kontrolleringen visas i klienten utan att ladda om registreringssidan med hjälp av AJAX via JQuery. Vi använde funktioner från tidigare uppgifter för att koppla till databasen och kontrollera om användarnamnet finns i databasen. Vi använde oss av föreläsningens jQuery CDN för att undvika använda jQuery lokalt.

## 4 Resultat

### Uppgift 1. JavaScript klient

		id	username
		1	
a	⊖	Radera	2 hello2
a	⊖	Radera	3 hello1
a	⊖	Radera	12 alexs
a	⊖	Radera	13 ashley
a	⊖	Radera	14 Dale
a	⊖	Radera	15 happy123
a	⊖	Radera	17 magnus
a	⊖	Radera	18 eirik
a	⊖	Radera	19 narthey

Username is available

Username

Choose a password

Enter your password again

Enter your name

Register

Username is not available

Username

Choose a password

Enter your password again

Enter your name

Register

Figur 1. En bild på de registrerade användarna i databasen

Figur 2. Bilder på registreringsformuläret vid två olika tillfällen.

```

1 registerForm.php x
2 source History
3 <!-- A simple login form used in login page-->
4 <?php
5     require_once $_SERVER['DOCUMENT_ROOT'] . '/Config.php';
6     require_once URL . '/data/core/init.php';
7
8     use Util\Input;
9     use Util\Token;
10    ?>
11    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
12
13    <script type="text/javascript">
14
15        $(document).ready(function() {
16            $('#feedback').load('check.php').show();
17            $('#username').keyup(function(){
18
19                $.post('check.php', {username:form.username.value },
20                    function(result) {
21                        |
22                        $('#feedback').html(result).show();
23                    });
24            });
25        });
26    });
27
28    </script>
29    <div id="loginDesign">
30    <form action="/data/register.php" method="post" name="form">
31    <div class="field">
32        <label for="username">Username</label>
33        <input type="text" name="username" id="username" value="<?php echo escape(Input::get('username'));?>" autocomplete="off">
34    </div>
35

```

Figur 3. Bild på koden av registreringsformuläret.



The screenshot shows a code editor with two tabs: 'registerForm.php' and 'check.php'. The 'check.php' tab is active, displaying a PHP script. The script includes a shebang line, two require\_once statements for 'Config.php' and 'init.php', and several use statements for 'Model\User' and 'Util\Input'. It then creates a new 'User' object and retrieves the 'username' from the input. A series of nested if-else statements check for an empty username, its length (less than 2 or greater than 51), and its availability. Depending on these checks, it echoes messages like 'Too Short!', 'Too Long!', 'Username is not available', 'Username is available', or 'Choose a username'. The script ends with a closing PHP tag. A vertical line of icons on the left side of the code editor indicates the structure of the code, with expandable sections for the if-else blocks.

```
<?php
2
3 require_once $ _SERVER['DOCUMENT_ROOT'] . '/Config.php';
4 require_once URL . '/data/core/init.php';
5
6 use Model\User;
7 use Util\Input;
8 $user=new User;
9 $username= Input::get('username');
10 if(!$username==''){
11     if(strlen($username) < 2)
12         echo 'Too Short!';
13     else if(strlen($username) > 51)
14         echo 'Too Long!';
15     else{
16         $check = $user->available($username);
17         if($check == true)
18             echo 'Username is not available';
19         else
20             echo 'Username is available';
21     }
22 }
23
24 else
25     echo "Choose a username";
26
27 ?>
```

Figur 4. En funktion för att meddela till användaren om det angivna användarnamnet är tillgänglig eller inte.

```
/*@available checks if username exists in the database
 *
 */
public function available($user = null){
    if($user){
        $data = $this->_db->get('users', array('username', '=', $user));

        if($data->count()){
            $this->_data = $data->first();
            return true;
        }
    }
    return false;
}
```

Figur 5. En funktion i User klassen som kollar om det angivna användarnamnet finns i databasen.

## 5 Diskussion

### Uppgift 1, JavaScript klient

Denna uppgift krävde att vi implementerade JavaScript kod som använde AJAX för att hämta data från servern och uppdatera klienten (webbläsaren). Det var tillräckligt för att hämta ett enda värde, till exempel användarnamnet på den inloggade användaren, och visa den i webbläsaren. Målet med uppgiften var att vi skulle visa att AJAX och jQuery användes i klienten (webbläsaren).

Uppgiften utfördes framgångsrikt då vi använde JavaScript för att implementera de funktioner där det behövdes.

## 6 Kommentarer om kursen

Vi har utfört uppgiften på 1 timme och har gått igenom JavaScript guider i 6 timmar inklusive föreläsningarna, totalt 7 timmar.