Software Engineering 2

Lösungen der Übung JavaScript, Formularprüfung

Dr. F.-K. Koschnick, Sybit GmbH

Aufgabe (Story): Prüfen der Eingabe beim Anlegen eines neuen User mit JavaScript

Als Administrator möchte ich beim Anlegen eines Users vor dem Submit eine Prüfung meiner Eingaben haben, damit ich korrekte Angaben zum User mache.

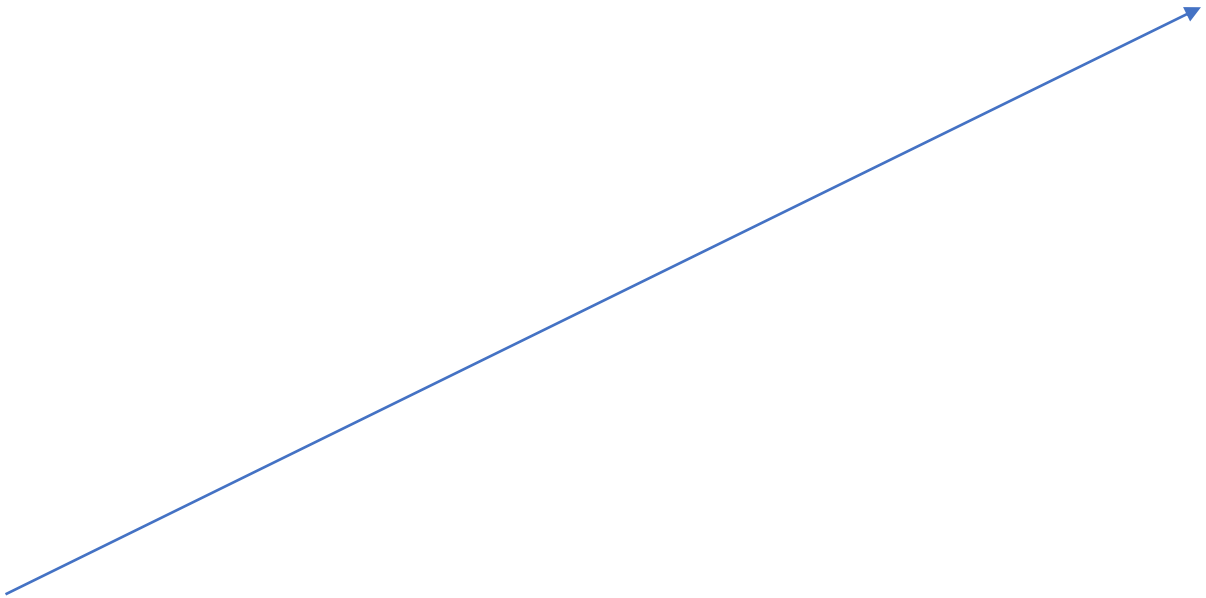
Verfiziere, dass

- das Eingabefeld für den Namen nicht leer ist
- das Eingabefeld für die Email auf korrekte Eingabe clientseitig geprüft wird. Dabei wird die Email-Eingabe auf folgendes Format geprüft: [Buchstaben, Zahlen, Unterstrich, Bindestrich, Punkt]@[domain].[xyz] (xyz bedeutet 2 oder 3 Buchstaben)
- das Eingabefeld für das Password auf korrekte Eingabe clientseitig geprüft wird. Dabei wird bei der Password-Eingabe Folgendes geprüft: Minimum 10 Zeichen, Maximum 20 Zeichen, nur Buchstaben und Zahlen, das Password muss mindestens einen Klein-, einen Groß-Buchstaben und mindestens eine Zahl enthalten.
- die clientseitige Prüfung mit JavaScript in einer getrennten JavaScript-Datei erfolgt (nicht mit dem Attribut pattern im HTML)
- das Formular nur abgeschickt wird, wenn alle Prüfungen erfolgreich sind

Tipp: Nutze reguläre Ausdrucke (regular expressions) für die Prüfung der Email und des Passwortes.

Vorgegebenes HTML

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>User Administration | RetroWeb</title>
    </head>
    <body>
        <form name="formNewUser" onsubmit="return checkInputs()">
            <label>Name</label>
            <input type="text" name="name"/><br/>
            <label>Password</label>
            <span title="Allowed are lowercase, uppercase letters and numbers. The length must be between 10 and 20 characters. A lowe
                <input type="password" name="password"/>
            </span><br/>
            <label>Email</label>
            <input type="email" name="email"/><br/>
            <label>Administrator</label>
            <input type="checkbox" name="admin"/><br/><br/>
            <input type="submit" value="add user"/>
        </form>
        <script src="scriptUser.js"></script>
    </body>
</html>
```



```
A lowercase letter, an uppercase letter and a number must be present.">
```

Lösung: JavaScript (scriptUser.js)

```
function checkInputs()
{
    var form = document.formNewUser;
    if(form.name.value == ''){
        alert("Please enter a name!");
        return false;
    }//lookahead (?=regex)
    if(/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{10,20}$/.test(form.password.value) == false){
        alert("Please enter a password that complies with the password policy!");
        return false;
    }
    if(/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/.test(form.email.value) == false){
        alert("Please enter a valid email address!");
        return false;
    }
    return true;
}
```

Formular-Element mit Namen formNewUser in Variable form speichern.

Inputfeld „name“ auf Eingabe prüfen, wenn Feld leer, Benutzer über Alert-Box informieren und false zurückgeben.

Inputfeld „email“ auf korrekte Eingabe prüfen, wenn Email-Adresse nicht korrektes Format, Benutzer über Alert-Box informieren und false zurückgeben.

Inputfeld „password“ auf korrekte Eingabe prüfen, wenn Password-Policy nicht eingehalten, Benutzer über Alert-Box informieren und false zurückgeben.

Regular Expression

Password: */^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{10,20}$/*

(?= ist der Anfang einer Lookahead-Gruppe, d.h. die ganze Regex wird untersucht. Dahinter steht .*[a-z], was bedeutet, dass beliebige Zeichen vor einem Kleinbuchstaben stehen können Die Klammer zu schließt die Lookahead-Gruppe.
Mit den Lookahead-Gruppen erreicht man, dass Kleinbuchstaben, Großbuchstaben und Zahlen (\d bedeutet Digit) nicht in einer bestimmten Reihenfolge vorkommen müssen, aber vorhanden sein müssen. Der hintere Teil der RegEx stellt sicher, dass nur Kleinbuchstaben, Großbuchstaben und Zahlen im Passwort erlaubt sind und dass das Passwort mindestens 10 und maximal 20 Zeichen hat (Quantor {10,20}).

Email: */^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/*

\w bedeutet ein Word-Character ([a-zA-Z_0-9], aber auch nicht lateinische Zeichen), ? bedeutet einmal oder kein Mal vorkommen (wie {0,1}), + bedeutet mindestens einmal vorkommen (wie {1,}) und * bedeutet beliebig oder kein Mal vorkommen (wie {0,}).

Bemerkung: Bei RegEx gibt es oft mehrere Möglichkeiten, die zum Ziel führen. Außerdem gibt es insbesondere bei Email Möglichkeiten, die besser oder schlechter sind. Man muss einen Kompromiss zwischen Komplexität und Güte der RegEx finden.