

Ajax
Werkcollege

Web Development Advanced

Docent: Frauke Vanderzijpen

Dig-X

Design & Technologie

2016 - 2017

1. Gegevens versturen via GET

Maak een nieuwe HTML pagina met een formulier. Dit formulier heeft een inputveld voor een naam en een submit button.

Wanneer het formulier wordt verzonden, blijft de pagina gewoon openstaan. Via Ajax wordt de naam verstuurd naar het serverscript (via GET). De naam heeft als key "name", de waarde wordt uit het formulier gehaald.

De server antwoordt met JSON code. Werd er geen naam verstuurd naar de server, bevat het antwoord een foutmelding, met volgende structuur:

```
{"error": "Geen naam verstuurd via GET"}
```

Indien er wel correct een naam werd verzonden ziet het JSON antwoord er als volgt uit:

```
{"success": "Welkom naam"}
```

Verwijder het formulier en toon de boodschap aan de gebruiker.

2. Gegevens versturen via POST

Kopieer vorige opgave. Wanneer het formulier wordt verzonden, blijft de pagina gewoon openstaan. Via Ajax wordt de naam verstuurd naar de server (via POST).

De server antwoordt met JSON code. Werd er geen naam verstuurd naar de server, bevat het antwoord een foutmelding, met volgende structuur:

```
{"error": "Geen naam verstuurd via POST"}
```

Indien er wel correct een naam werd verzonden ziet het JSON antwoord er als volgt uit:

```
{"success": "Welkom naam"}
```

Verwijder het formulier en toon de boodschap aan de gebruiker.

3. Chatbox

Open de opgavebestanden.

Voor deze oefening bouw je een chatbox. Pak de bestanden uit en plaats ze op je webserver. De bestanden zijn:

Logon.html

Deze pagina bevat een login-formulier. Het formulier heeft een veld waar een naam kan ingevuld worden, en een submit-knop om het formulier te verzenden.

Chat.html

Deze pagina bevat een chat-box. De div #chat bevat de berichten die werden verstuurd. De div #post bevat een invoerveld waarin de gebruiker een boodschap kan typen.

Save.php

Dit bestand bevat een php-script. Er staat reeds een functie in geplaatst die een boodschap wegschrijft naar een XML-bestand.

log

log is een folder waarin de chatsessie automatisch zal worden opgeslagen. Het wordt opgeslagen als een bestand met naam xml.

Pas de bestanden volgens onderstaande instructies aan:

logon.html

Vul deze pagina aan met jQuery zodat aan volgende vereisten wordt voldaan:

- Als de gebruiker niets invult, gebeurt er niets.
- Als de gebruiker wél iets invult (zijn naam), wordt zijn naam opgeslagen in een Cookie. Meer info over JS cookies: https://www.w3schools.com/js/js_cookies.asp
- Nadat zijn naam werd opgeslagen, wordt de pagina "chat.html" automatisch geladen. Hiervoor gebruik je location.href.

chat.html

Voeg jQuery toe aan de pagina, en voeg ook een JavaScript bestand toe dat ervoor zorgt dat de chatbox werkt. Gebruik volgende richtlijnen.

- Haal de naam op uit de cookie.
- Indien de naam niet bestaat, OF als de naam een lege string bevat, laad je de pagina "logon.html" automatisch in. Hiervoor gebruik je location.href.
- Controleer als het invoerveld actief is of de return-toets wordt ingedrukt. Gebruik hiervoor het keypress-event.
- Wanneer de return-toets werd ingedrukt, stuur je een ajax-request naar het script php. Dit request moet volgende gegevens doorsturen via POST:
 - name met als waarde de naam van de gebruiker.
 - time hierin plaats je de huidige tijd.
 - message heeft als waarde de inhoud van het invoerveld van de chatbox.
- Schrijf een functie die elke seconde wordt uitgevoerd. Hiervoor kan je de setInterval-functie van JavaScript gebruiken.

```
setInterval(loadMessages,1000);
```

- De functie die elke seconde wordt uitgevoerd, stuurt een ajax-request naar het bestand xml.
- Wanneer de XML is geladen, zet je de gegevens om in HTML, die je vervolgens toevoegt aan de div#chat.
 - Elk chatbericht plaats je in een <article>-element.
 - Plaats de naam van het bericht in een -element.

- o Plaats de tijd van het bericht in een `<time>`-element.
 - o Plaats de boodschap van het bericht in een `<p>`-element.
- Je kan ervoor zorgen dat binnen de `div#chat` automatisch naar onder wordt gescrolled door gebruik te maken van volgende code:

```
$chat.scrollTop($chat[0].scrollHeight);
```

4. Spellenlijst

Open de opgave.

Het XML-bestand in de data folder bevat een reeks spellen. Elk spel heeft een naam; een minimum en een maximum aantal spelers; een beschrijving; een afbeelding (bevat het pad/naam naar een afbeelding, bv. roborally.png).

Voeg jQuery toe aan het HTML bestand en maak een nieuw JavaScript bestand aan dat je koppelt aan het HTML bestand. In dit JavaScript bestand lees je de XML in met jQuery. Loop over het herhalende element met de `each()` functie.

Zorg ervoor dat de resultaten in een HTML-tabel worden weergegeven (`<table>`-element). Elke tabelrij bevat een spel. De eerste kolom toont de afbeelding, de tweede bevat een `<h2>`-element met de titel, de derde het minimum aantal spelers, de vierde bevat het maximum aantal spelers.

5. Spellenlijst - data in objecten en arrays

Kopieer de HTML van de vorige oefening waar je een XML-bestand met spelletjes inleest en toont.

Deze oefening ga je nu uitbreiden. Voorzie in de pagina boven de tabel drie links. De eerste link toont de tekst sorteer op naam (naam), de tweede link heet sorteer op minimum aantal spelers (min) en de derde link sorteer op maximum aantal spelers (max).

Om effectief te kunnen sorteren, moet je ervoor zorgen dat de data in het geheugen bewaard wordt. Maak in JavaScript een globale variabele aan. Deze variabele zal de spellen bevatten. Het deeltje van de code uit de vorige oefening waar de spellen uit XML wordt omgezet naar HTML, mag je verwijderen (of in commentaar plaatsen). Er is nu een extra tussenstap nodig.

```
var spellen = []; //aanmaken array
```

Bij het uitlezen van de XML-data, maak je per spel een tijdelijk JavaScript object aan. Je kan je baseren op volgende code:

```
var spel = {  
    "naam" : "naam",  
    "min" = "3",  
    //....
```

```
};
```

Na het aanmaken van het object, voeg je het toe aan de Array.

Maak een functie die de Array omzet naar een HTML-tabel en deze tabel toevoegt aan de pagina. Om dit te doen loop je de Array, per element in de Array haal je er de gegevens uit om de HTML-tabel aan te vullen.

6. Spellenlijst - sorteren

Vervolgens implementeer je het sorteren. Een array sorteren kan je doen aan de hand van de `sort()`-functie:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.sort();
```

De array bevat na bovenstaande code Apple, Banana, Mango, Orange.

Nu moet er in deze oefening geen array van strings gesorteerd worden, maar een array van objecten, op basis van één van de property values van de objecten. Hiervoor moet je een sorteerfunctie schrijven - deze functie heeft twee objecten als argument, en gaat een property van beide objecten vergelijken. Staan ze in de juiste volgorde, retournt de functie 0, staan ze fout, retournt de functie -1. Onderstaande code is hier een voorbeeld van: een array met objecten die telkens een `fname` en `lname` bevatten wordt gesorteerd op `lname`.

```
var objs = [
    { fname: 'Piet', lname: 'Pieters' },
    { fname: 'Jan',   lname: 'Janssens' },
    { fname: 'Zulma', lname: 'O' }
];

function compare(a,b) {
    if (a.lname > b.lname) return 0;
    else return -1;
}

objs.sort(compare);
```

Indien je bovenstaande code gebruikt, zal je drie sorteerfuncties moeten schrijven: een om te sorteren op naam, een om te sorteren op minimum aantal spelers, en een om te sorteren op maximum aantal spelers.

Je kan ook een sorteerfunctie schrijven waaraan je de property waarop gesorteerd moet worden meegeeft.

```
function compare(property) {  
    return function (a, b) {  
        return (a[property] < b[property]) ? -1 : (a[property] > b[property])  
        ? 1 : 0;  
    }  
}  
  
objs.sort(compare("lname")); //sorteert op lname  
objs.sort(compare("fname")); //sorteert op fname
```

Zorg ervoor dat de spellen gesorteerd worden nadat één van de links wordt aangeklikt. Na het sorteren zet je de array opnieuw om naar een HTML-table, gebruik makende van de functie uit de vorige oefening.

7. Spellenlijst – JSON

Hermaak nu de oefening van de spellenlijst maar zet het XML bestand om naar een JSON bestand.