

CMISDT011T
Smart Devices en Technologie in de zorg

van

Paul Sohier (0806122)



CMI-Opleiding *Technische Informatica* – Hogeschool Rotterdam

17 november 2011

Eerste docent *ir J.F. Blankert*
Tweede docent

Inhoudsopgave

Week 1	2
Internet Explorer 6	2
Java vs PHP	2
Blueprint	2
Conclusie	3
Week 2	4
Wijzigingen	4
Mobile device	4
Conclusie	5
Week 4	6
Android en iOS	6
Werking standaard phonegap applicatie	6
Pure Android applicatie	7
Emulator problemen	7
Conclusie	7
Week 5	8
Onderdrukken van geolocation melding	8
Nauwkeurigheid van GPS	8
Bereken van afstand	8
LocalStorage	9
Conclusie	9
Week 6	10
websockets	10
localStorage	10
Conclusie	10

Week 1

Internet Explorer 6

Internet explorer 6 is uitgekomen in 2001 bij de release van Windows XP[1]. Microsoft heeft in 2011 besloten dat het tijd wordt, ondanks dat IE6 officieel nog ondersteund wordt door Microsoft, dat IE6 niet langer gebruikt moet worden voor websites. In de wereld van Web Development is het algemeen bekend dat IE6 gebruikers zwaar achterlopen, ondanks dat updaten voor hun, of voor het bedrijf waarbij ze werken, niet voor veel problemen hoeft te zorgen. Het gebruiken van hacks om problemen in IE6 te voorkomen is binnen de webdevelopment wereld af te raden en wordt door mede ontwikkelaars niet gewaardeerd. De enige correcte oplossing om problemen met IE6 op te lossen is het updaten naar een nieuwe versie van Internet Explorer, of het installeren van een modernere browser. Voor web developers is de enige juiste oplossing het weergeven van de countdown banner (<http://www.ie6countdown.com/join-us.aspx>) van Microsoft zodat gebruikers gewaarschuwd worden dat ze IE6 niet moeten gebruiken.

Java vs PHP

Bij zowel Java (met behulp van een applet draaiend op de webserver) of met PHP (Direct draaiend op de webserver) kan relatief eenvoudig een verbinding gelegd worden met javascript via AJAX. Beide zijn eenvoudig aan te passen aan de wensen van de programmeur en kunnen de data afhandelen die de website opvraagt. De meeste programmeurs vinden het werken met php eenvoudiger aangezien de taal makkelijker is opgezegd en eenvoudiger te leren voor de meesten mensen. Hierdoor kan je sneller een eindresultaat opbouwen wat vaak minder werk is als in java. In beide talen zal je in principe op dezelfde manier werken en hetzelfde resultaat krijgen als eindresultaat. De keuze van de taal is afhankelijk van de kennis van de ontwikkelaar en de wensen van de klant. Voor het javascript deel zal door de betere ontwikkelaar gebruik gemaakt worden van een standaard framework zoals bijvoorbeeld jQuery. Er wordt in de huidige webdevelopment wereld bijna geen gebruik meer gemaakt van plain javascript, zonder framework. Het voordeel van het gebruik van een framework is dat deze erop gebouwd is dat hij dit framework in principe in alle gevallen met alle browsers daadwerkelijk werkt. Om deze opdrachten te maken heb ik er dan ook voor gekozen om geen gebruik te maken van losse functies, maar direct van jQuery.

Blueprint

Blueprint wordt enkel en alleen gebruik gemaakt voor de opmaak van de webpagina, en niet voor de afhandeling van de gegevens en verwerking hiervan. Dit heeft geen invloed op het eindresultaat

met betrekking tot de werking, enkel met betrekking tot het uiterlijk.

Conclusie

Bij deze opdracht wordt in principe niet de echte kracht gebruikt van de mogelijkheden van AJAX en javascript. Door gebruik te maken van standaard plain javascript kan je niet het maximale uit eruit halen. Ook is hetgeen wat uitgevoerd moet worden sneller te doen zonder gebruik te maken van AJAX en kost dit ook minder CPU kracht en dataverkeer, wat erg belangrijk is bij grote websites. Door gebruik te maken van een standaard framework kan een beter en sneller resultaat gehaald worden met minder code, wat voor een betere ontwikkeling van het geheel zorgt.

Week 2

Wijzigingen

Doordat ik bij week 1 al gebruik gemaakt heb van jQuery in plaats van plain javascript zijn de wijzigingen welke ik nodig heb om het te laten werken met JSON minimaal. De volgende wijzigingen zijn nodig:

- Vervangen XML in java en php deel door JSON. In php kan eenvoudig gebruik gemaakt worden van `json_encode` met als parameter de array om die moet opgeslagen worden, in java moet de JSON zelf samengesteld worden.
- In het javascript deel moet het deel dat XML opvroeg vervangen worden door de `JSON.Parse()` functie (Welke een extra jQuery plugin vereist. Door gebruik te maken van `JSON.Parse` kan er geen extra javascript worden ingevoerd door een gebruiker, wat met het gebruik van `eval` wel mogelijk is. `Eval` moet zo min mogelijk gebruikt worden aangezien het een zeer onveilige functie is. Dit is niet alleen in javascript zo, maar in principe in alle script en programmeertalen. Door het gebruik van `eval` kan vaak code welke uit user input is verkregen uitgevoerd worden waardoor de eigenaar van de website, of de webontwikkelaar, niet langer de volledige controle heeft over de site wat betekent dat een kwaadwillend persoon code kan uitvoeren welke hij niet hoort uit te kunnen voeren.). In plaats van de jQuery functie aanroep in de XML moet er nu de waarde opgevraagd worden uit de Array welke teruggegeven wordt vanuit `JSON.Parse`.

Wanneer ik gebruik gemaakt zou hebben van plain javascript waren de wijzigingen voor het gebruik van JSON niet veel groter. Het php/java deel heeft op dat moment dezelfde wijziging nodig. Voor het opvragen van XML in plain javascript is meer code nodig als wanneer gebruik gemaakt wordt van jQuery. Deze code kan vervangen worden door `JSON.Parse()`, welke niet alleen als jQuery plugin beschikbaar is, maar ook gewoon zonder jQuery werkt. Je kan ook gebruik maken van `eval`, echter is dit mogelijk onveilig, wanneer een gebruiker toegang heeft tot de php/java files, of het netwerk. Ook hier moet de rest van de code dan de Array met gegevens worden gebruikt.

Omdat in de eerste week al gebruik gemaakt was van `blueimp` was het niet nodig om hieraan wijzigingen aan te brengen.

Mobile device

Tegenwoordig hebben alle mobile devices een browser met hierop javascript ondersteuning. Doordat deze browsers vaak gebaseerd zijn op bijvoorbeeld de webkit engine (Waarop ook Google

Chrome en Apple Safari zijn gebaseerd) is de javascript ondersteuning niet veel anders als op browsers direct op een PC. Voor jQuery is er tevens een aparte plugin beschikbaar welke een GUI kan maken specifiek voor mobile devices. Web ontwikkelaars hoeven niet dus tegenwoordig niet specifieke javascript code meer te schrijven voor mobile devices. Uiteraard moeten ze wel nog steeds testen of de website, en dus het javascript deel, correct werkt erop.

Conclusie

Zoals ik al eerder vermeldde zijn de wijzingen ten opzichten van de opdracht uit de eerste week. Doordat de code met jQuery geschreven is zijn er maar zeer minimale wijzingen benodigd. Wat wel een groot voordeel is van het gebruik van JSON is dat het minder dataverkeer verbruikt doordat het kleiner is omschreven en dat er in php functies zijn die standaard aan staan welke het verzenden van JSON vergemakelijken. Tevens is JSON een standaard in javascript voor de notatie van Arrays, en verstuur je dus eigenlijk gewoon een javascript array welke je gebruikt.

Week 4

Android en iOS

Het grote voordeel van Android en iOS is dat ze beide ondersteuning aanbieden voor het installeren van eigen gemaakte applicaties waarbij je functionaliteit kan toevoegen aan het apparaat. Hierbij kan gewerkt worden met het door het besturingssysteem ondersteunende programmeertaal of met bijvoorbeeld een applicatie welke een browser “nadoet” waardoor je er bijvoorbeeld een website in kan laden welke specifiek voor mobiele browsers is gemaakt. Een voorbeeld hiervan is phonegap, welke de ondersteuning biedt om de mobiele website te draaien in een eigen applicatie met de voordelen daarvan. Wanneer een bedrijf al een (mobiele) website heeft is het maken van de applicatie zeer eenvoudig en zal in de meeste gevallen weinig tijd kosten voor de ontwikkelaar. Veel ontwikkelaars van applicaties zijn het echter er niet mee eens dat applicaties welke puur een mobiele website laden in een eigen schil gebruikt moeten worden. Om de juiste user experience te geven moeten de applicaties specifiek ontworpen worden voor het gebruik van de functies van het operating system met de daarbij behorende talen. Voor veel ontwikkelaars is het leren van de java (In het geval van Android) niet het een probleem, aangezien dit al op talen lijkt zoals bijvoorbeeld javascript. Wat bedrijven vaak als argument gebruiken is dat het gebruiken van een product als phonegap is dat de ontwikkeling goedkoper is aangezien de mobiele website al bestaat. Vaak is dit niet het geval aangezien de mobiele website nog steeds aangepast moet worden aan de werking van phonegap en de specifieke eisen van het besturingssysteem.

Om de applicatie in iOS en Android werkend te krijgen zijn bijna dezelfde stappen benodigd. Voor iOS moet phonegap eerst geïnstalleerd worden en moet er een nieuw project aangemaakt worden specifiek voor phonegap. Voor android hoeft enkel de jar file uitgepakt te worden en in de properties van het project toegevoegd te worden als library file. Hierna moet de HTML en javascript in de www map van phonegap toegevoegd worden en voor Android moet de java file nog aangepast worden om ervoor te zorgen dat phonegap daadwerkelijk gebruikt wordt. Hierna kan de applicatie in de emulator (Of op een telefoon) geïnstalleerd worden om te controleren of hij werkt. In de applicatie zal de mobiele versie van de website geladen worden die in de www map staat.

Werking standaard phonegap applicatie

De standaard phonegap applicatie is een voorbeeld van de mogelijkheden welke phonegap heeft tot het opvragen van gegevens van het apparaat zelf, zoals bijvoorbeeld het unieke nummer van de telefoon en op welk besturingssysteem het draait. Deze gegevens kunnen met een voor phonegap specifiek geschreven stukje javascript opgevraagd worden.

Pure Android applicatie

Om de character conversie werkend te krijgen als android applicatie zonder gebruik te maken van phonegap moet de applicatie geheel herschreven worden. In plaats van gebruik te maken van javascript met php of java moet er nu gebruik gemaakt worden van puur java. Het deel wat we eerder gebruikt hebben voor in week 1 wat we in java hebben geschreven kunnen we wel hergebruiken in de Android applicatie. Om de Android applicatie te maken moeten we eerst Eclipse en de SDK voor Android installeren. Hierna kan je een nieuw Android project maken, met de gewenste Android versie. In dit geval maakt het niet uit welke versie gekozen wordt aangezien de functies welke ik gebruik in iedere Android versies aanwezig zijn. Wanneer een nieuw project aangemaakt is kan de code worden toegevoegd welke reageert wanneer de button wordt aangeraakt en welke de waarde weergeeft op het scherm. Om de button en velden voor de weergave te laten zien moet de layout in de XML aangepast worden zodat. Hierna kan met de emulator getest worden of de code daadwerkelijk werkt.

Emulator problemen

De Android emulator staat er om bekend dat hij niet altijd even goed werkt en dit zorgt voor veel problemen bij de ontwikkeling van nieuwe applicaties. In veel gevallen kan er beter getest worden op de daadwerkelijke hardware waarop de applicatie zal gaan draaien zodat je zeker weet dat de applicatie ook correct werkt. De applicatie ondersteunt niet alle features welke in Android aanwezig zijn en het is bijvoorbeeld zeer lastig te testen of GPS goed werkt in je applicatie. Ook bij het gebruik van OpenGL in een applicatie zal het aantal frames per second zeer laag zijn, terwijl het op de meeste telefoons wel goed werkt. Om deze redenen maken veel ontwikkelaars (haast) geen gebruik van de emulator en testen ze enkel op echte hardware.

Conclusie

Wil je als bedrijf een applicatie die daadwerkelijk iets toevoegd aan de promotie naast de website en de mobiele website is het vaak het best om een eigen, losse, applicatie te maken voor Android en/of iOS. Met een losse applicaties kan een bedrijf meer halen uit het totaalbeeld zonder daar veel in te moeten investeren. De ontwikkeling van een native applicatie voor beide operating systemen kost meer tijd als de ontwikkeling van puur een mobiele website, maar vaak moet die mobiele website alsnog aangepast worden om goed gebruik te kunnen maken van phonegap. Wat de beste keuze is is echt afhankelijk van wat een bedrijf (Of de ontwikkelaar) zelf wilt bereiken met de applicatie, en de keuze om voor een native applicatie of een phonegap applicatie zal afhankelijk daarvan gemaakt moeten worden.

Week 5

Onderdrukken van geolocation melding

Volgens de netiquete is het niet netjes om meldingen die gerelateerd zijn aan security of privacy gerelateerde gegevens te onderdrukken en data te gebruiken waarvoor een applicatie standaard geen toegang toe heeft. Wanneer een applicatie dit wel doet is het zullen veel gebruikers dit niet waarderen en niet langer de applicatie gebruiken. Dit soort praktijken geven gebruikers vaak een reden om te denken dat de applicatie nog meer data opvraagd en eventueel doorstuurd naar een externe locatie zonder hierbij toestemming te vragen van de gebruiker. Applicaties welke dit toch doen zullen vaak niet geaccepteerd worden in bijvoorbeeld de App store van Apple en in de market van Android verwijderd worden. In Android is het standaard tevens ook niet mogelijk om zonder de benodigde permissies in de Android manifest toegang te krijgen tot de GPS gegevens. De gebruiker moet bij de installatie toestemming geven dat de applicatie toegang heeft tot bepaalde gegevens welke de applicatie kan gebruiken. Wanneer de gebruiker geen toegang geeft kan de applicatie niet geïnstalleerd worden.

De geolocatie in een browser op een PC maakt gebruik van een database met wifi accesspoints en bepaald aan de hand daarvan de locatie waar de PC zich op dat moment bevindt.

Nauwkeurigheid van GPS

De nauwkeurigheid van GPS hangt van een aantal factoren af waarbij vooral gekeken moet worden naar de hardware welke aanwezig is in bijvoorbeeld de telefoon. Wanneer er geen GPS hardware aanwezig zal de telefoon gebruik maken van aGPS, welke kijkt naar bijvoorbeeld GSM masten en wifi accesspoints in de buurt. aGPS is aanzienlijk minder nauwkeurig als GPS, maar geeft nog wel een globale indicatie van de locatie waar je bent. Met GPS kan je wanneer je buiten bent en niet in de buurt van grotere gebouwen die het signaal blokkeren of vervormen in principe een nauwkeurigheid van 5 tot 10 meter behalen.

Bereken van afstand

Met de resultaten welke je van de GPS krijgt in de vorm van een longitude en latitude kan je eenvoudig berekenen op welke locatie je bent en welke afstand je hierbij afgelegd hebt. Dit is echter wel hemelsbreed, om de afstand welke je aflegt over de weg te berekenen zijn er meer coördinaten nodig om te kijken wanneer er ergens een bocht gemaakt wordt en hoeveel hiertussen de afstand is. Om deze metingen te doen moet de tijd niet te groot zijn anders krijg je dezelfde soort problemen welke je al had bij originele methode.

Localstorage

Om de data op te slaan heb ik in dit geval gebruik gemaakt van cookies aangezien de data welke opgeslagen moet worden maar tijdelijk is en niet zeer groot. Wanneer de browser nu gesloten wordt is alle data verloren, maar die is verder ook niet nodig bij een latere verwerking.

Conclusie

Het gebruik van geolocatie is simpel om uit te voeren in zowel Android, iOS als in een browser via javascript. Door gebruik van geolocatie kunnen bedrijven hun website specifiek richten op de locatie van een gebruiker en dus bijvoorbeeld een aanbieding voor die regio geven. Op deze manier kunnen bedrijven dus nog meer profiteren van de locatie. Berekeningen met de locatie zijn in principe relatief eenvoudig uit te voeren door wanneer de locatie bekend is, echter zal dit in een browser relatief weinig voorkomen.

Week 6

websockets

Da Sinds de lessen over websockets is het protocol hiervoor aangepast waarbij de voorbeelden welke gebruikt zijn niet meer werken. In eerste instantie werd er gebruik gemaakt van 2 keys welke benodigwaren voor de encryptie van de verbinding. Bij de huidige situatie is er nog maar 1 key nodig voor de handshake, echter is het nog niet exact duidelijk hoe dit dan werkt. Het is mij niet gelukt om de websockets werkend te krijgen waardoor ik een deel van deze opdracht niet hebben kunnen voltooien door het gebruik van websockets.

In plaats van websockets heb ik er voor gekozen om zodra er een verbinding beschikbaar is een AJAX request uit te voeren naar een remote server welke de data uitleest en dan opslaat. Dit heeft als eindresultaat hetzelfde effect als wanneer er gebruik wordt gemaakt van websockets.

localstorage

In de vorige week heb ik gebruik gemaakt van tijdelijke storage voor de locatie. Aangezien de data nu langer moet bewaard blijven als de sessie duurt heb ik gekozen om gebruik te maken van meer permanentere software. In plaats van cookie gegevens heb ik gebruik gemaakt van localStorage om de data op te slaan. Hierbij heb ik de data opgeslagen als JSON string. Ik heb er express niet voor gekozen om gebruik te maken van een lokale database op te slaan aangezien er diverse implementaties zijn in de browsers welke allemaal verschillend werken. Hierdoor is er een te grote kans op fouten wanneer een browser besluit het alsnog aan te passen.

Conclusie

De originele opdracht kon niet worden uitgevoerd aangezien browsers waren aangepast naar een nieuwe manier van communiceren. Door gebruik te maken van een alternatieve manier van communiceren kan de data alsnog opgeslagen worden. Om dit soort data op te slaan moet er officieel toestemming gevraagd worden aan de gebruiker welke op de pc zit te werken, dit aangezien dit om privacy gevoelige informatie gaat en je dit niet zonder toestemming mag opslaan. Tevens moet de opslag van privacy gevoelige gegevens voldoen aan de voorwaarde gesteld door het CPB. Om deze reden is het gebruiken en opslaan van deze data voor een bedrijf niet te doen. Ook als privépersoon heb je je te houden aan deze wetgeving.

Door de verschillende implementaties van localStorage is er niet op te vertrouwen dat de data correct opgeslagen wordt en de code in nieuwere versies van de browser daadwerkelijk zou blijven

werken. Door dit soort problemen zijn de technieken welke hier gebruikt zijn nog niet geschikt om daadwerkelijk te gebruiken op een website voor een bedrijf, aangezien bij een browser update de site (deels) niet meer werkt, wat uiteraard niet acceptabel is voor een bedrijf.

Bronnen

[1] "internet explorer 6". http://en.wikipedia.org/wiki/Internet_Explorer_6.