

Uitleg

Bedankt voor je deelname aan het reviewen van de front-end Mimoto code. Ik ben hier de afgelopen maanden druk mee bezig geweest. Als richtlijn heb ik hier de volgende design challenge voor gehanteerd:

‘Hoe kan ik Mimoto updaten en tegelijkertijd zo opzetten dat het qua techniek herbruikbaar en bij de tijd is?’

De bedoeling van deze expert review is om te valideren of mijn design challenge behaald is. Hou bij het beoordelen van mijn code dan ook de design challenge in je achterhoofd.

De expert review is in twee delen opgedeeld. Het eerste deel heeft betrekking tot de ErrorHandling en FormValidation libraries en het tweede deel tot de Mimoto formulier componenten.

Als je meer context wilt hebben lees dan eens de volgende pagina:

<http://afstudeerproject.lusenet.com/>

De project code kan hier gevonden worden:

<https://github.com/SebastianKersten/Mimoto.Aimless>

Review ErrorHandling en FormValidation

De eerste twee bestanden die beoordeeld dienen te worden zijn *ErrorHandling.js* en *FormValidation.js*. Deze bestanden zijn te vinden in *src/userinterface/app/javascript/utils*.

Op de volgende plekken kan er gedetailleerde uitleg gevonden worden over de desbetreffende libraries:

- <http://afstudeerproject.lusenet.com/#werking-formvalidation>
- <http://afstudeerproject.lusenet.com/#werking-errorhandling>

Het wordt aangeraden om dit te lezen aangezien het dan een stuk duidelijker is wat de doelen zijn van deze libraries. Nadat je dit hebt gelezen kun je beginnen met de eerste review. Op de volgende vragen zou ik graag een antwoord krijgen:

Doet de code alles wat het claimt te kunnen doen of mist er functionaliteit?

De code lijkt vrij compleet en te voldoen aan de beloofde functionaliteit. Enige functionaliteit die ik nog zou kunnen bedenken is lokalisatie, hiermee bedoel ik het eenvoudiger toevoegen van validatie meldingen in andere talen.

Is de code zo geschreven dat het relatief makkelijk verder ontwikkeld kan worden door andere developers?

Ja, code is compact en overzichtelijk in opzet, mogelijk kun je nog een versie

overwegen waarbij je meer uitleg als commentaar in de code geef, dan wijst het zich vanzelf.

Zijn de libraries (met relatief kleine aanpassingen) herbruikbaar voor andere projecten?

Ja, dat lijkt me wel, zeker aangezien formulieren in de meeste projecten grotendeels dezelfde eisen stellen. Omdat deze code en functionaliteit heel generiek opgezet is, kan deze makkelijk gebruikt worden in meerdere projecten.

Is de code zo geschreven dat hij schaalbaar is? Werkt de code bijvoorbeeld voor 100 formulieren maar ook voor 10.000 formulieren?

Ik heb het niet getest, hangt een beetje af van functionele variatie in formulieren, maar ik zou zeggen van wel.

Zijn de gebruikte technieken modern en toekomstbestendig?

De code zelf lijkt mij toekomstbestendig, mede omdat deze in “vanilla JS” is geschreven en daarmee geen afhankelijkheid is gecreëerd van externe libraries (zoals jQuery).

Daarbij zijn formulieren in mijn ervaring doorgaans niet onderhevig aan grote ontwikkelingen. Ook reguliere expressies zijn een vorm van een beproefde techniek die niet snel zal veranderen.

Mogelijk zullen in de toekomst de “native” / html5 validatie technieken (afgehandeld door webbrowser zelf) verder op gaan komen waarbij gebruikers een voorkeur zouden kunnen ontwikkelen voor deze vorm van validatie.

Overige feedback?

Dit was het eerste deel van de review. We gaan nu beginnen met het tweede gedeelte.

Review Mimoto formulier componenten

Het tweede gedeelte gaat om de Mimoto formulier componenten. De HTML en CSS van deze componenten is te vinden in

`src/userinterface/MimotoCMS/components/forms/input`. De volgende componenten dienen gereviewd te worden:

- Checkbox
- Dropdown
- Radio Button
- Textblock
- Textline

De bijbehorende JavaScript wordt gevonden in

`src/userinterface/app/javascript/views/form-components`. Vergeet vooral niet om de file `Form.js` te reviewen aangezien hier alles bij elkaar komt.

Op <http://afstudeerproject.lusenet.com/#demos> kunnen er verschillende demo's gevonden worden van de Mimoto formulier elementen in combinatie met de ErrorHandler en FormValidation libraries. Ik raad je dan ook aan om deze te bekijken en hier eens mee te spelen.

De volgende vragen zou ik graag beantwoord hebben:

Doet de code alles wat het claimt te kunnen doen of mist er functionaliteit?

Ik mis hier geen functionaliteit, de meest benodigde formulier elementen lijken erin te zitten.

Zijn de formulier componenten modulair opgezet? Zijn ze dus makkelijk herbruikbaar?

Ja, dit lijkt mij voldoende modulair opgezet, dus ook makkelijk herbruikbaar en uitbreidbaar.

Wordt er een CSS methodologie toegepast? Zo ja, wat vind je van deze methodologie?

Ik herken er niet direct een specifieke methodologie in. daarentegen zijn de opgesplitste css bestanden voor de formulier elementen zodanig eenvoudig dat er nauwelijks een misverstand kan ontstaan.

Is de code zo geschreven dat het relatief makkelijk verder ontwikkeld kan worden door andere developers?

Ja, code is compact en overzichtelijk in opzet. de implementatie voorbeelden zijn uiteraard wel heel erg project specifiek. hier zou je evt. een generieker voorbeeld van kunnen geven.

Zijn de gebruikte technieken modern en toekomstbestendig?

Ja, de gebruikte code is modern in opzet en daarmee zo toekomstbestendig als mogelijk. Daarbij zijn formulieren in mijn ervaring doorgaans niet onderhevig aan grote ontwikkelingen

Overige feedback?

Dit was de review.

Bedankt voor je deelname!