Documentatie Proftaak Defensie

Wat zijn Angular & Ionic

Angular.JS en Ionic zijn bij onze proftaak gebruikt om onze applicatie te ontwikkelen. Met behulp van Cordova, Angular en Ionic is het mogelijk om meerdere platforms tegelijk te ondersteunen. Angular is een client-side framework gebruikt om dynamische webapplicaties te bouwen. Door zaken te ondersteunen als databinding en dependency injection is dit een ideaal framework om samen te werken met server applicaties.

Ionic Framework wordt gebruikt om HTML5 mobile apps te creëren. Door gebruik van Ionic is het mogelijk om hybride applicaties te maken, deze applicaties kunnen op meerdere platforms gebruikt worden. Ionic stemt de user interface af op mobile door touch interacties te gebruiken voelt het aan als een native applicatie, maar het heeft de voordelen om op meerdere platforms tegelijk te ontwikkelen. Daarnaast levert Ionic ook een aantal scripts waarmee eenvoudig gedeployed kan worden naar mobile devices.

JSON

De gegevens die wij hebben gebruikt zijn afkomstig van de geleverde informatiefolder op de gitrepo. De NSN gegevens bestonden in het CSV (comma seperated value) formaat, dit hebben wij omgezet naar het JSON formaat. JSON staat voor Javascript Object Notation. JSON is een schrijfwijze waarmee je objecten kunt vastleggen. JSON is gemakkelijk te lezen door zowel mens als computer. Door het succes van dit formaat wordt het ondersteund in haast alle talen, hierdoor is het een ideaal formaat voor het communiceren tussen verschillende talen. Het is bijvoorbeeld goed mogelijk om JSON via http op te vragen. Wij hebben ervoor gekozen de JSON enkel intern te houden. De gegevens staan dus in een lokaal bestand, genaamd “Data.json”.

Data binding

Om de data vanuit de controllers naar de views te sturen gebruiken wij de zogeheten data binding. Zodra deze binding opgezet wordt, wordt de initiële waarde van het ene variabele toegekend aan het te binden variabele. Deze binding werkt two-way, dat betekend als we in de view of in de controller aanpassen dat de andere aangepast wordt. Dit is speciaal handig bij het gebruiken van input velden. Zo kan bijvoorbeeld data gefetched worden, getoond worden in een input veld en vervolgens na aanpassing kan de data in de controller weer opgeslagen worden. In angular zijn er een aantal opties voor data binding. Waaronder: ng-model en ng-repeat. Ng-model doet precies wat hierboven beschreven is en is speciaal bedoeld voor input. Ng-repeat kan een element herhalen voor een array, dit is handig wanneer er een lijst getoond wordt. In deze lijst kan elk object in de array los benaderd worden en getoond worden door middel van data binding.

OCRad

Voor het herkennen van de tekst binnen de foto’s hebben wij een OCR library gebruikt. Omdat wij het proces binnen het programma af wilde handelen moest dit een Javascript library zijn. Er is eigenlijk 1 optie als het gaat om een native Javascript OCR libraries. Dit is OCRAD. OCRAD is geen geweldige OCR library, hij heeft problemen met veel fonts en heeft moeite met lagere resoluties. Echter is de toepassing binnen de app voldoende voor ons doeleinde.

Wanneer er een geavanceerdere OCR geïmplementeerd kan worden kan dit eenvoudig gebeuren. De OCR implementatie wordt namelijk opgezet via app.Factory. Ons advies zou zijn het starten van een selectie traject. In dat traject worden de relevante fonts op een rijtje gezet en getest binnen verschillende (mogelijk niet native javascript) OCR pakketten. Deze OCR pakketten zouden kunnen functioneren via een webAPI of een javascript wrapper voor een C++ library, zoals tesseract-js. Het testen van de verschillende OCR pakketten viel echter buiten onze scope.