

Hack the Future 2015

Idee

We zitten opgesloten in een ruimte, de elektriciteit is uitgevallen en we kunnen de ruimte niet verlaten want de deur is geblokkeerd. Om de deur terug open te krijgen moeten we de elektriciteit herstellen en dit gebeurt symbolisch door een aantal stappen te doorlopen. Functioneel - De verschillende stappen

Stap1: Integratie met Philips Hue

e niet noodzakelijk een b

Intelligent (IoT) lightbulbs van Philips. Op elk bureau staat een ronde lamp waarin een Philips Hue lamp voorzien is.

In Stap1 moet er een integratie gebouwd worden met een Philips Hue lamp, dit houdt in dat er eventueel een backend maar zeker een frontend moet gebouwd worden. De frontend is een mobiele /web applicatie die kan gebouwd worden in de technologie naar keuze:

- (Angular) / HTML5 / CSS3
- Ionic
- Native
- JSP
- JSF
- ...

In deze stap hoeft je niet noodzakelijk een backend te gebruiken als je alles in Javascript wil doen, dan is dit ook ok. Moest je toch een backend maken dan zal je die backend moeten gebruiken om een connectie te maken met de Philips hue. Als frontend verwachten we een applicatie die je toelaat om de kleuren van de lamp te wijzigen. De kleuren die je zal nodig hebben zijn: Rood, geel, blauw en groen. Er mag uiteraard ook aandacht besteed worden aan de looks van de applicatie.

Tijdens de beoordeling kan je altijd een woordje uitleg geven waarom je een bepaalde technologie gekozen hebt. Philips biedt zowel voor IOS en Android ook de Philips Hue app aan. Die mag je eventueel wel gebruiken om de lampen te testen, maar je moet tijdens het demo moment wel degelijk bewijzen dat je via code een integratie gebouwd hebt.

Er zijn verschillende manieren om te connecteren naar een Philips Hue waaronder rest interface, Java backend, Node.js, De applicatie die connecteert met de Philips Hue dient wel op een machine te draaien op hetzelfde netwerk als de Hue lamp, dus lokaal op je laptop of op een smartphone / tablet die verbonden is met dezelfde WIFI als de lampen. Dit onderdeel dient dus niet gedeployed te worden naar een centrale server.

Voor elk team is er één Hue lamp voorzien (op je bureau). De globale opdracht bestaat uit een 4 tal stappen. Als je een bepaalde stap afgerond hebt, dan zal dit enerzijds zichtbaar worden in het dashboard, maar anderzijds dient je dan ook de kleur van je lamp te wijzigen via de applicatie die je gemaakt hebt.

De zijn de kleuren codes:

Stap1 (beginsituatie) : Rood

Stap2: Geel

Stap3: Blauw

Stap4: Groen

Bovenstaande wijzigingen in kleur moet je dus kunnen afdwingen door de applicatie die je gebouwd hebt en zal je ook moeten kunnen laten zien tijdens het demo moment. Documentatie over integratie nodig ? Schuim gerust het internet eens af en als je er echt niet uit geraakt, dan kunnen we je eventueel wel op weg helpen. Het doel is dus om de lamp groen te krijgen en dan gaat de symbolische deur open.

Bij integratie met de Hue lamp dien je een bepaalde IP adres te gebruiken (van de hub) en een bepaalde ID. Elk lamp is voorzien van een bepaalde ID. De ID van je lamp kan je terugvinden in de Google Spreadsheet.

Na deze stap dient de kleur van je lamp **groen** te zijn.

Stap2: Registratie van je team

De status van elk team wordt weergegeven in het Dashboard. Een 2de stap bestaat erin om je naam van je team en de personen van je team te registreren in het dashboard, naam van het team is vrij te kiezen, voor de namen van deelnemers gebruik je gewoon best je eigen namen. Om je team te registreren hebben we een API voorzien. Je kan dus gebruik maken van deze API om je team te registreren. Dit kan door een JSON message te posten naar de API. Zo een JSON message kan er als volgt uitzien:

```
{
  "name": "Team 8",
  "members": [{ "name": "Tom" }, { "name": "Attila" } ]
}
```

De URI die je kan gebruiken om je JSON naartoe te sturen is: <https://hackthefuture.herokuapp.com/dashboard/team>

Bedoeling is om de API via code aan te roepen en ook de Json message via code op te bouwen. Dus liefst niet gewoon via String parsing, maar via een framework. Dit mag in Java / Node / Javascript / ... zijn, maar het is niet de bedoeling om je team te registreren via tools zoals Restclient of Postman (of iets dergelijks). De code zal ook moeten gedemonstreerd worden tijdens het demo moment.

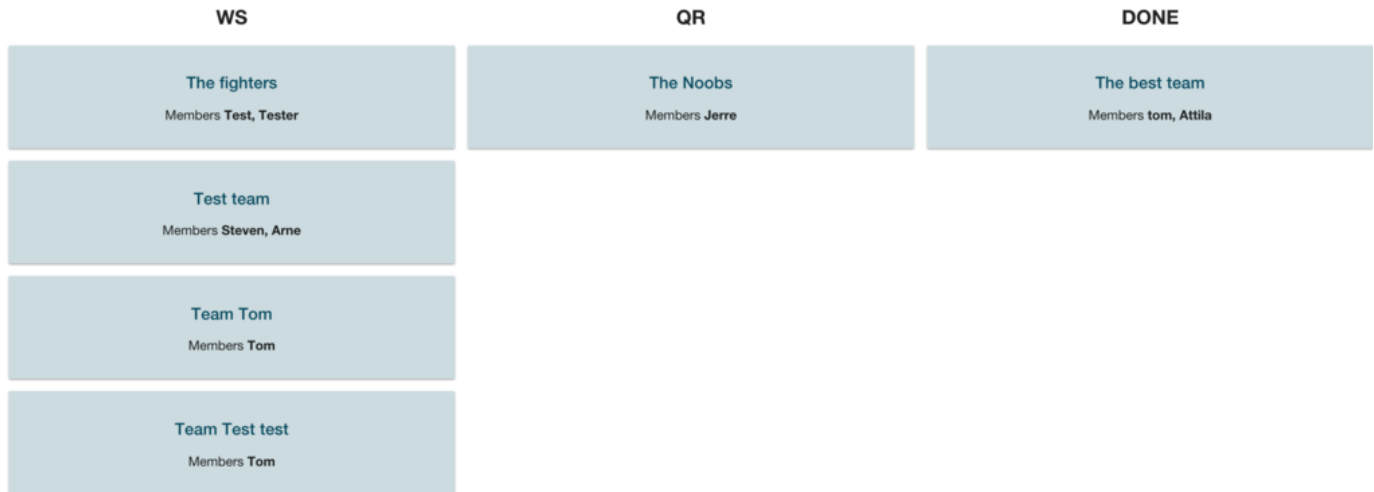
De informatie die jullie via de API doorsturen wordt automatisch gesaved in een database en zou ook moeten getoond worden in het dashboard (na refresh).

Na deze stap dient de kleur van je lamp **geel** te zijn.

Stap3: Implementeer een JSON Webservice

Er zal een algemeen dashboard geprojecteerd worden waarin te zien zal zijn in welke stap welk team zit. Om de status van een team te achterhalen zal het dashboard constant een URL pollen. Dit is een URL naar een webservice die jullie zullen moeten maken en in de cloud zullen moeten deployen. Er zal een Google Compute Engine voorzien worden voor elke team. De Webservice moet JSON teruggeven waarin onder andere de naam van team zal komen te staan en ook de verschillende members. Eigenlijk zal de JSON er hetzelfde uitzien als de JSON die je in stap 2 doorstuurde naar de API.

Het Dashboard zal er ongeveer zo uitzien (zie screenshot).



Het dashboard zal gedisplayed worden op een beamer, maar je kan zelf ook naar het Dashboard surfen door volgende URL te gebruiken: <https://hackthefuture-dashboard.herokuapp.com/>

Alle centrale servers (zowel Java als Node) zijn Linux servers en zijn gehost bij Google (Google Compute Engine). Je kan eventueel remote inloggen op server, maar dan ga je mij wel een public key certificaat moeten bezorgen. Indien dit te veel problemen met zich meebrengt dan kan ik de deploy wel zelf doen. Moeten we dan tijdens de dag zelf afspreken.

Je dient zelf geen status weer te geven, de poll service zal het team in een bepaalde status zetten aan de hand van wat logica:

- Als de webservice URL een 404 teruggeeft wil dit zeggen dat jullie nog geen webservice geïmplementeerd hebben en wordt de status nog rechstreeks uit de database gelezen, dus dan jullie ze nog in status 1 (WS).
- Als jullie een correct Webservice geïmplementeerd en gedeployed hebben en als de WS een correct JSON teruggeeft dan gaan het team naar de volgende status.
- En als de QR code ingelezen wordt dan ga je naar DONE.

Na deze stap dient de kleur van je lamp **Blauw** te zijn.

Stap4: Inlezen van een QR code

Je zal zien dat er een centrale QR code scanner voorzien wordt (via een tablet). Bedoeling is dat je per team een QR code uit een database ophaalt en deze visualiseert op een scherm. Dit mag een scherm zijn van een smartphone, tablet of PC.

Voor deze stap zal je dus ook een stuk frontend en backend moeten schrijven en ben je ook volledig vrij van technologie keuze. Je kan wederom tijdens het demo moment misschien even kort uitleggen waarom je een bepaalde technologie gekozen hebt, dit mag eventueel dezelfde technologie zijn als in stap1.

Bedoeling is dat je via backend code gegevens uit de database leest. De gegevens moeten gelezen worden uit de tabel qr_code, hier is voor elk team een qr_code voorzien (één record per team). In de tabel qr_code staan 2 velden. Eentje ervan is img_title, dit veld bevat de naam van het team, dus zal je hier ook op moeten queryen. De QR code zal dan getoond moeten worden op een scherm en ingescand moeten worden op de centrale QR code scanner. Als de correcte QR code ingescand wordt dan wordt ook de status in het Dashboard geupdate.

IP: 173.194.105.180

port: 3306

user: student

password: mulestudent

Het is niet het team dat als eerste de opdracht geklaard krijgt dat ook de winnaar is. De winnaar zal het team zijn dat uiteraard met een mooie oplossing komt, maar er zal zeker ook beoordeeld worden:

- in hoeverre er samengewerkt wordt in het team
- structuur van de code
- manier van aanpak, ...