

INSTALLATION GUIDE

Colleague Finder

Elaine Heijboer
Versie 1.1
10-01-2020

Table of Contents

<i>Inleiding</i>	2
<i>Installatie</i>	2
<i>Mappen structuur</i>	6
<i>SVG aanpassen</i>	7
<i>Bronnen</i>	9

Inleiding

In dit document staat de installatie guide voor de webapplicatie Colleague Finder. Als aanvulling op de installatie zijn ook de belangrijkste functies en code gedocumenteerd en een lijst met bronnen toegevoegd die gebruikt kunnen worden voor het verder ontwikkelen van Colleague Finder.

Wat is Colleague Finder?

Colleague Finder heeft als belangrijkste functie, collega's en vrije bureaus beter zichtbaar te maken. Werknemers binnen Greenhouse kunnen in Colleague Finder zoeken naar hun collega's en krijgen zoekresultaten terug waarbij ze kunnen zien waar hun collega zich die dag heeft ingecheckt. Ook is het mogelijk om op een plattegrond per verdieping te zien welke plaatsen er vrij of bezet zijn. Op deze manier kunnen collega's elkaar makkelijker vinden. Voor het in kaart brengen van de zitplekken van de werknemers is een inchecksysteem nodig. Dit wordt met hetzelfde platform gemaakt. Met NFC/QR wordt er elke dag door de werknemers ingecheckt op Colleague Finder.

Installatie

Colleague Finder is gemaakt met het Laravel framework. Laravel maakt gebruik van Composer voor het beheren van de dependencies. Voordat je Laravel kan gebruiken moet je eerst composer hebben geïnstalleerd op je computer. Voor meer informatie over het installeren van Laravel raad ik de Laravel documentatie aan <https://laravel.com/docs/6.x>

Om Colleague Finder lokaal te kunnen ontwikkelen zijn er een aantal programma's/tools nodig.

Benodigde tools

- XAMPP
- IDE
- Composer
- Git
- Node.js en npm
- Laravel
- Docker (optioneel)

XAMPP/WAMP

Omdat Colleague Finder werkt met een database heb je een lokale server nodig zoals [XAMPP](#) of [WAMP](#). Hiermee kan je lokaal de Laravel website draaien met een lokale database. Let op dat de PHP-versie 7.3.1 of hoger is, dit is de minimale PHP-versie vereist voor de Laravel applicatie. Voor deze installatie guide maak ik gebruik van XAMPP.

Download op deze pagina de juiste installatie voor Windows/Linux/OS X Geef tijdens de installatie aan waar je de XAMPP-map wilt hebben.

Installeer XAMPP op Windows 10: <https://www.youtube.com/watch?v=f8N4FEQWYy>

Installeer XAMPP op Mac OS X Mojave: https://www.youtube.com/watch?v=PrBJtd_wXlo

Installeer XAMPP op Ubuntu 18.04: <https://www.youtube.com/watch?v=R5CUn5wGQGg>

IDE

PHP ontwikkelen kan met de meeste IDE's, ik maak gebruik van [PHP-storm](#) van JetBrains.

Tip: als student of docent mag je gratis gebruik maken van alle JetBrains tools.

Meer informatie: <https://www.jetbrains.com/student/>

Composer

Composer is een tool voor het managen van dependencies in PHP. Je kan [composer](#) installeren via de setup.exe of via de commandline.

Installeer composer voor Windows: <https://www.youtube.com/watch?v=HBDJsc2cXR4>

Installeer composer voor Mac OS X: <https://www.youtube.com/watch?v=HOaj6mjF-nl>

Installeer composer voor Ubuntu 18.04: <https://www.youtube.com/watch?v=tH14liAyHeY>

Git

Git is een version control system voor webapplicaties. Hiermee kan je de versie van de applicatie beheren. Ik maak gebruik van [Git](#) voor de commandline maar je kunt ook gebruik maken van een Git GUI zoals bijvoorbeeld [Gitkraken](#).

Installeer Git command line voor Windows: https://www.youtube.com/watch?v=J_Clau1bYco

Installeer Git command line voor Mac OS X: <https://www.youtube.com/watch?v=sJ4zr0a4GAs>

Installeer Git command line voor Ubuntu 18.04: <https://www.youtube.com/watch?v=ZMg LZUYd8Cw>

Node.js en npm

Voor het compileren van de SCSS heb je npm nodig. Met `Node -v` en `npm -v` kan je controleren of je Node en npm al hebt geïnstalleerd. Zo niet dan kan je [hier](#) Node downloaden. npm is uitgegeven door Node.js wanneer je deze installeert wordt ook npm automatisch geïnstalleerd op je computer.

Aangeraden wordt om ook al heb Node.js net geïnstalleerd te controleren of je de laatste npm versie hebt. Npm wordt vaker geüpdatet dan Node en is ook los te updaten.

```
$ npm install npm@latest -g
```

Docker (optioneel)

Indien je al ervaring hebt met Docker en de installatie van Docker, raad ik het gebruik van een Docker container aan.

Bouw de Docker image:
`docker build -t "ColleagueFinder" .`

Start een nieuwe Docker container op basis van de hierboven gecreëerde image:
`docker run -p 80:80 (-e DB_DATABASE=colleaguefinder -e DB_USER=root -e DB_PASSWORD= -e OAUTH_APP_ID=fda0e6a7-aca9-4c91-983d-949f5fe71c8b -e OAUTH_APP_PASSWORD='=i3.J8dLUrgt=nXoeclfhXxQRZl1Y9[' -e OAUTH_AUTHORITY=https://login.microsoftonline.com/51ba71a5-c6f8-434b-8010-7827596459c4) ColleagueFinder`

Website clonen

Om de website volledig te laten draaien moeten we door een aantal stappen heen.

Als eerste heb je een account nodig op [GitLab](#) en daarna heb je toegang nodig tot de Colleague Finder hiervoor moet je contact opnemen met de MEH zij kunnen je ook toegang geven tot de database gegevens van Colleague Finder.

Clone de website naar de gewenste map met Git (of Git kraken). Ik heb de website in de htdocs map gezet. Vergeet niet om de apache en MySQL aan te zetten in XAMPP.

Open de het project in je IDE, voordat de applicatie volledig kan draaien moeten de `“.env.example”` (in de main directory) file worden aangepast. Ik raad aan deze te kopiëren en te hernoemen naar `“.env”` Pas in deze file de volgende gegevens aan:

```
OAUTH_APP_ID=fda0e6a7-aca9-4c91-983d-949f5fe71c8b
OAUTH_APP_PASSWORD='=i3.J8dLUrgt=nXoeclfhXxQRZl1Y9['
OAUTH_AUTHORITY=https://login.microsoftonline.com/51ba71a5-c6f8-434b-8010-7827596459c4
```

Deze gegevens zorgen ervoor dat de applicatie toegang heeft tot de Greenhouse Microsoft Office Azure, dit betekent dat je lokaal kan inloggen met je Greenhouse email adres. Het kan zijn dat je bij het inloggen nog toestemming moet vragen aan de ICT-afdeling.

Je kan eventueel ook je eigen Azure opzetten en die gegevens gebruiken voor het testen. De `“.env”` file wordt bij het committen niet meegenomen.

Laravel maakt gebruik van composer deze moeten we nog installeren in het project.

```
$ composer install
```

Laravel heeft ook nog een application key nodig deze kan je genereren (nadat je een ".env" file hebt gemaakt). Deze key is nodig om gebruikers sessies en andere encrypted data te beschermen. Deze key word als je in de ".env" file "APP_KEY=" hebt staan automatisch toegevoegd.

```
$ php artisan key:generate
```

Nu kunnen we de website laten draaien en lokaal bekijken navigeer naar localhost:8001 als het goed is zie je nu de homepagina van colleague finder. **De poort moet 8001 zijn** omdat localhost:8001/authorize de link is in de Greenhouse Azure deze link moet gelijk zijn anders kan je niet inloggen.

```
$ php artisan serve --port 8001
```

Voordat we werkelijk in kunnen loggen missen we nog een database waar de gegevens opgeslagen worden.

Ga in je browser naar <http://localhost/phpmyadmin> en maak een nieuwe database aan genaamd colleaguefinder (de naam moet overeenkomen met de DB_DATABASE in de .env file). Je hoeft hier verder niks aan te maken omdat Laravel met migrations werkt. In de applicatie moet je het volgende command uitvoeren:

```
$ php artisan migrate
```

Met migrate worden de database tabellen automatisch in de database gegenereerd zodat ze hetzelfde zijn als op productie.

Als laatste heb je voor het compilen van de SCSS npm nodig. Installeer npm met:

```
$ npm install
```

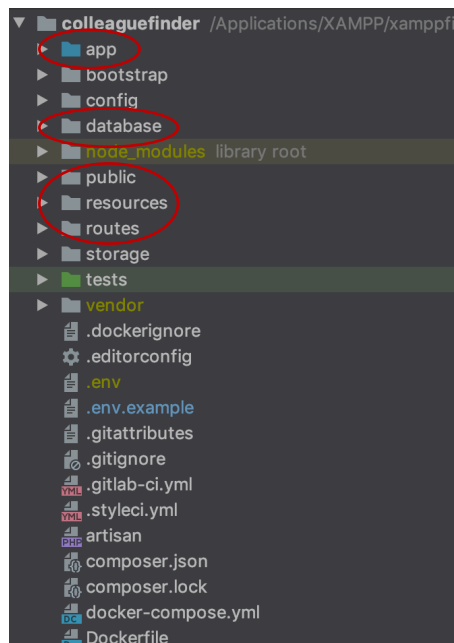
Om de scss automatisch te compilen wanneer er een aanpassing wordt gedaan kan je het volgende command uitvoeren.

```
$ npm run watch
```

Als alles goed is gegaan werkt de applicatie nu volledig en kan je inloggen met je Greenhouse email-adres.

Mappen structuur

De mappen structuur van Laravel kan in het begin chaotisch overkomen. Vandaar dat ik een aantal belangrijke mappen wil toelichten waar je het meeste in zal werken. Wanneer dit je eerste keer is dat je werkt met Laravel raad ik aan een korte cursus te zoeken op bijvoorbeeld Udemy of Youtube om de basis te begrijpen. In de bronnenlijst vind je een aantal goede cursussen.



App folder

De eerste map die je tegen komt is de app folder. In de app folder vind je vooral de bestanden die figureren als "back-end" in de "app/Http" folder vind je de controllers en middleware. De controllers zal je het meeste gebruiken dit is waar de functies staan voor o.a. de login. In de controllers staan de als het ware alle SQL queries en worden ook de verschillende tabellen met elkaar gekoppeld.

Database folder

In de database folder vind je de folder migrations, wanneer de Tabel structuur moet worden aangepast kan je dat in de migrations aanpassen, vervolgens moet je wel de migrations opnieuw uitvoeren. Meer documentatie over migrations is te vinden in de [Laravel documentatie](#).

Public

In deze folder staat alles wat op de front-end getoond wordt. Ook alle images worden hier in de image map geplaatst. In de CSS-map vind je de gecompileerde CSS-file. In de Resources folder

Resources

Hier vind je de javascript, Sass en de views. De views zijn de pagina's waar de html wordt geplaatst de main view pagina staat in resources/views/layouts/app.blade.php in deze file worden alle andere files ingeladen met de @yield('content'). Deze file hoeft je eigenlijk niet aan te passen. De content voor deze pagina staan in de views map zoals info.blade.php.

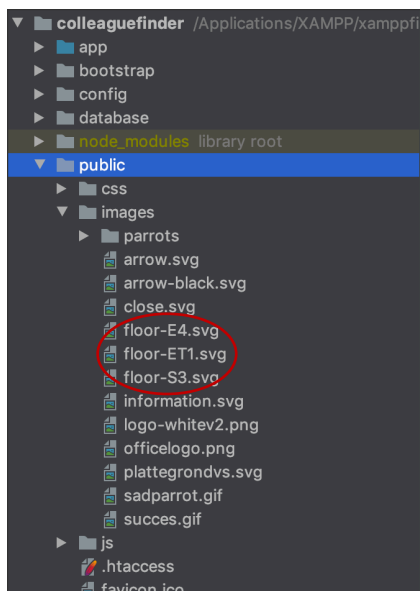
Routes

De belangrijkste file in de routes folder is de web.php hier wordt een view gekoppeld aan een controller.

Plattegrond (SVG) aanpassen

Op dit moment is de SVG-plattegrond nog niet makkelijk beheersbaar in een admin panel maar moet het aangepast worden in de code.

Benamingen van de SVG's



Voorbeeld van plattegrond images in de applicatie

Achter de benamingen van de SVG's zit een structuur, zodat de applicatie herkent over welke locatie en verdieping het gaat.

Aantal voorbeelden:

floor-E4.svg = Eindhoven 4^{de} verdieping

floor-ET1.svg = Eindhoven Toren 1^{ste} verdieping

floor-S3.svg = Sittard 3^{de} verdieping

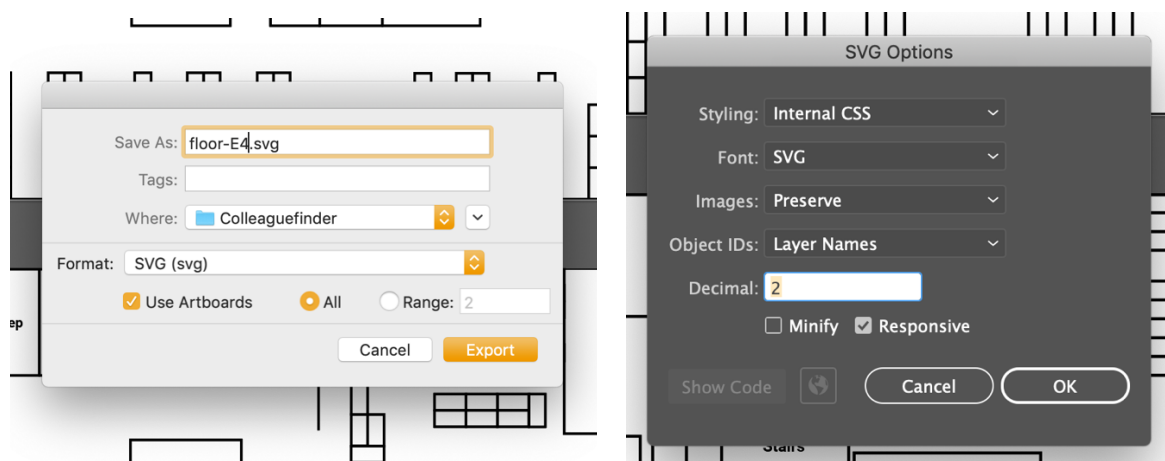
Het is belangrijk dat deze benaming wordt aangehouden als er nieuwe verdiepingen bijkomen. Wanneer er nieuwe locaties bij komen dan zal ook de code aangepast moeten worden.

SVG aanpassen

Zodra de indeling van de tafels veranderd is het belangrijk dat de svg wordt aangepast. Wanneer het een kleine aanpassing is bijvoorbeeld 1 bureau is verplaatst of weggehaald kan je dit in de code van de svg zelf doen door de coördinaten aan te passen:

```
<rect id="159" class="desk" x="427.23" y="359.78" width="10.84" height="20.78"><title>01.003</title></rect>
<rect id="157" class="desk" x="438.07" y="359.79" width="10.84" height="20.78"><title>01.001</title></rect>
<rect id="160" class="desk" x="427.23" y="338.97" width="10.84" height="20.78"><title>01.004</title></rect>
<rect id="158" class="desk" x="438.07" y="338.98" width="10.84" height="20.78"><title>01.002</title></rect>
```

Echter als de complete indeling wordt veranderd is het beter om een nieuwe svg te maken. In een programma zoals bijvoorbeeld Illustrator kan je svg's maken. Je kan dan gemakkelijk de bureaus verplaatsen en daarna het bestand opnieuw exporteren als een svg. Er zijn eps bestanden van alle plattegronden.



Het aanpassen van de svg op deze manier heeft wel een nadeel, de id's die nodig zijn in de svg om een koppeling te maken met de database staan er nog niet in. Deze moeten handmatig in de juiste regels in de svg worden gezet.

De id nummering is afhankelijk van de NFC-tag id die op de tafels zijn geplakt. Als er bij de verhuizing goed rekening gehouden is met de tafel id's zou er een plattgrond moeten zijn van de oude indeling en de nieuwe indeling met id nummers.

```
<rect id="159" class="desk" x="427.23" y="359.78" width="10.84" height="20.78"><title>01.003</title></rect>
<rect id="157" class="desk" x="438.07" y="359.79" width="10.84" height="20.78"><title>01.001</title></rect>
<rect id="160" class="desk" x="427.23" y="338.97" width="10.84" height="20.78"><title>01.004</title></rect>
<rect id="158" class="desk" x="438.07" y="338.98" width="10.84" height="20.78"><title>01.002</title></rect>
```

Zoals je in dit voorbeeld zit moet er aan elke regel dus een id komen die overeenkomt met het id dat op de NFC-tag staat en een class="desk". Deze class desk gaat dus alleen op alle <rect> elementen die bureaus zijn. Controleer goed of alles goed staat en niet op de verkeerde elementen. Je kan een oude plattgrond svg pakken als reference.

De CSS voor de desk is als volgt:

```
.desk:hover{
fill:rgba(108, 108, 108, 0.7)!important;
}
.desk{
fill: #fff;
stroke: #000;
stroke-miterlimit: 10;
stroke-width: 2px;
}
```

Wanneer alle id's op de goede plek staan in de svg moeten de desks nog gekoppeld worden in de database. De database gegevens kan je opvragen bij het MEH-team.

+ Opties										
← T →										
			id	desk_nr	status	screen_type	floor_nr	user_id	expiry_date	
<input type="checkbox"/>				1	E4.104	1	1	4	11	2020-01-09 23:56:00
<input type="checkbox"/>				2	E4.105	NULL	1	4	NULL	NULL
<input type="checkbox"/>				3	E4.106	NULL	NULL	4	NULL	NULL
<input type="checkbox"/>				4	E4.107	NULL	NULL	4	NULL	NULL
<input type="checkbox"/>				5	E4.108	1	NULL	4	NULL	NULL
<input type="checkbox"/>				6	E4.109	1	1	4	NULL	NULL
<input type="checkbox"/>				7	E4.110	NULL	1	4	NULL	NULL

Zorg ervoor dat de desk nummers overeenkomen met de id's. Ook belangrijk is dat het floor_nr nog klopt met de desk_nr. Als laatste moet er gekeken worden op welke desk designschermen staan deze krijgen een '1'.

Bronnen

Laravel documentatie: <https://laravel.com/>

Laravel cursus Udemy: <https://www.udemy.com/course/php-with-laravel-for-beginners-become-a-master-in-laravel/learn/lecture/9526516?start=90#overview>

XAMPP: <https://www.apachefriends.org/index.html>

WAMP: <http://www.wampserver.com/en/>

PHPstorm: <https://www.jetbrains.com/phpstorm/>

Composer: <https://getcomposer.org/>

Git: <https://git-scm.com/>

GitLab: https://gitlab.com/users/sign_in

Node.js(npm): <https://www.npmjs.com/get-npm>

Docker: <https://www.docker.com/>