

# Cloud Applications

## Auth0 Authentication

### Backend

Voor de backend heb ik de tutorial hieronder gevolgd.

<https://auth0.com/docs/quickstart/backend/aspnet-core-webapi/01-authorization>

### Frontend

Voor de Frontend heb ik de quick start gevolgd voor ionic 3 native application op Auth0.

The screenshot shows the Auth0 dashboard interface. On the left is a sidebar with navigation links: Dashboard, Applications, APIs, SSO Integrations, Connections, Users, Rules, Hooks, Multi-factor Auth, Hosted Pages, Emails, Logs, Anomaly Detection, Extensions, and Get support. The main content area displays the 'RiskAntwerp' application details, including a 'Quick Start' tab and a 'Client ID'. Below this, the 'Ionic 3' quickstart guide is featured, showing a list of steps: 1. Configure Auth0, 2. Install the Dependencies, 3. Integrate Auth0 in your Application, 4. Add Authentication with Auth0, and 5. Troubleshooting. There are also buttons for 'DOWNLOAD SAMPLE' and 'VIEW ON GITHUB'.

Bij deze stap had ik bepaalde problemen.

#### Add Platform and Run the App

You will now need to allow Ionic / Cordova to install the necessary plugins and update your `config.xml` appropriately for the platform you wish to run on.

Use the following commands to add your desired platform(s) (e.g., `ios` or `android`) and run the app:

```
# Add platform (e.g., ios or android)
$ ionic cordova platform add {platform}

# Run on desired platform (e.g., ios or android)
$ ionic cordova run {platform} --livereload
```

Did it help? [Yes](#) / [No](#)

This will then launch your app in the local emulation environment for the platform you chose (Xcode Simulator for iOS or the Android emulator of your choice).

Don't be alarmed if you *cannot authenticate* successfully yet at this stage! There was some configuration generated by running the app that you still need to add to your Auth0 settings.

Na Troubleshooting en opzoek werk ben ik er achter gekomen dat met deze commando:

```
ionic cordova run {platform} --livereload
```

```
ionic cordova run android --livereload
```

De toepassing niet werkt , en dat het alleen werkt als het echt op device wordt uitgevoerd.

Dat heb ik gedaan met deze commando: `ionic cordova run android --device`

Hierbij ondeckte ik ook meerdere errors die te maken hadden met de android sdk.

Na dat ik alle errors hiervan had opgelost kon ik het succesvol op mijn gsm testen.

Hier kwam ik dan nieuwe errors tegen die te maken hadden met de Storage.

Na wat opzoek werk heb ik de oplossing hiervoor gevonden:

Er was een probleem met de versie die ik gebruikte bij ionic/storage en dan heb ik deze stappen moeten volgen.

- 1] Run `npm install @ionic/storage@2.0.0 --save --save-exact`
- 2] Remove Storage from your providers in `app.module.ts`
- 3] `import { IonicStorageModule } from '@ionic/storage'` instead of `import { IonicStorage } from '@ionic/storage'` in `app.module.ts`
- 4] Add `IonicStorageModule.forRoot()` to the imports array in `app.module.ts`

```
],  
imports: [  
  BrowserModule,  
  IonicModule.forRoot(MyApp),  
  IonicStorageModule.forRoot(),  
  HttpClientModule,  
  HttpModule,
```

```
"@ionic/storage": "2.0.0",
```

```
import { IonicStorageModule } from '@ionic/storage';
```

Dan paar errors verder heb na op te zoeken gevonden dat ik in de `node_modules` ook nog bepaalde dingen moest veranderen in de `@ionic/storage/es2015`

In de `storage.d.ts` heb ik vanonder de `Token` moeten veranderen naar `InjectionToken`.

```
export interface StorageConfig {  
  name?: string;  
  storeName?: string;  
  driverOrder?: string[];  
}  
export declare function provideStorage(storageConfig?: StorageConfig): Storage;  
export declare const StorageConfigToken: InjectionToken<object>;
```

In de `storage.js` heb ik dit ook moeten doen

```
export function provideStorage(storageConfig) {  
  var config = !!storageConfig ? storageConfig : getDefaultConfig();  
  return new Storage(config);  
}  
export var StorageConfigToken = new InjectionToken('STORAGE_CONFIG_TOKEN');
```

Hierna werkte de code pas van auth0 op mijn device.

## Frontend to Backend

Het doorsturen van Auth0 Bearer Token van de applicatie naar de API.

Met de code hieronder haal ik de token op van Auth0.

```
GetToken(){
    var request = require("request");

    var options = { method: 'POST',
        url: 'https://inias.eu.auth0.com/oauth/token',
        headers: { 'content-type': 'application/json' },
        body:
        '{"client_id":"OH6LtWPu6p2u4VSnsWnp4PnaexbUUgzW","client_secret":"tRpfMy--Vv3b2r1NFNLwMqVey2YAVj4W0dy3C33VaJrfaPdZGOInqbXbDtijinn0","audience":"https://AntwerpRisk/api","grant_type":"client_credentials"}' };

    request(options, function (error, response, body) {
        if (error) throw new Error(error);

        console.log(body);

    });
}
```

En deze token die ik dan terug krijg stuur ik mee naar de API in de header op deze manier.

```
private httpHeader = {
    headers: new HttpHeaders({
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${this.AuhtToken}`
    })
}

getplayers(): Observable<Players[]>{
    return this.http.get<Players[]>(`${this.baseApi}/player`,this.httpHeader);
}
```