

Research report



**well NETWORKED
top CONNECTED**

business models • smart services • strategies • solutions

Yordi Kremer
DATUM: 07/09/2023

Table of Contents

| | |
|--|----|
| Table of Contents | 1 |
| 1 Onderzoeksvragen | 2 |
| 1.1 Hoofdvraag..... | 2 |
| 1.2 Deelvragen | 2 |
| 2 Onderzoek 1..... | 3 |
| 2.1 Vraag | 3 |
| 2.2 Hoe ben ik aan de slag gegaan? | 3 |
| 2.2.1 Library Research - Literature Study | 3 |
| 2.2.2 Library Research – Community research | 4 |
| 2.2.3 Workshop Research - IT architecture sketching | 5 |
| 2.2.4 Workshop Research – Prototyping..... | 6 |
| 2.3 Resultaat | 6 |
| 2.4 Qualiteit van het resultaat | 6 |
| 3 Onderzoek 2..... | 8 |
| 3.1 Vraag | 8 |
| 3.2 Hoe ben ik aan de slag gegaan? | 8 |
| 3.2.1 Field Research – Problem analysis | 8 |
| 3.2.2 Field Research – Explore user requirements..... | 8 |
| 3.3 Resultaat | 9 |
| 3.4 Qualiteit van het resultaat | 9 |
| 4 Onderzoek 3..... | 10 |
| 4.1 Vraag | 10 |
| 4.2 Hoe ben ik aan de slag gegaan? | 10 |
| 4.3 Resultaat | 11 |
| 4.4 Qualiteit van het resultaat | 11 |

1 Onderzoeksvragen

1.1 Hoofdvraag

Hoe kan een integratie van third-party gezondheidsdata (apple, google) op een veilige manier in de huidige bedrijfssoftware gerealiseerd worden, zodat meer klanten van move4vitality gebruik kunnen maken van het delen van data met hun fysiotherapeut?

1.2 Deelvragen

- Hoe kan ik de API's van Apple Health en Google Health aanroepen?
- Welke technische en functionele vereisten zijn nodig voor het ontwikkelen van een systeem dat in staat is om gegevens van externe API's te verzamelen en deze gegevens weer te geven?
- Hoe moet omgegaan worden (uit technisch en regelgevings oogpunt) met privé/gezondheidsdata?

2 Onderzoek 1

2.1 Vraag

Hoe kan ik de API's van Apple Health en Google Health aanroepen?

2.2 Hoe ben ik aan de slag gegaan?

2.2.1 Library Research- Literature Study

Als eerste ben ik gaan uitvogelen wat apple en google health inhoud, bij apple heb je de [gezondheids app](#) en bij google heb je de [google fit app](#). Deze apps houden data bij zoals:

- Activiteiten
- Blood glucose
- Bloeddruk
- Lichaamsafmetingen
- Lichaamstemperatuur
- Hartslag
- Locatie
- Voeding
- Zuurstofverzadiging
- Slaap

Na dit overleg ben ik begonnen met het bestuderen van de documentatie van de [Google Fit API](#) en de [Apple Health API](#). Aangezien de integratie van deze API's aanzienlijk verschilt, en omdat Apple doorgaans beperkte toegang verleent tot hun software, was het duidelijk dat de Apple Health-integratie waarschijnlijk uitdagender zou zijn. Ik heb deze kwestie besproken met mijn stagebegeleider en we hebben besloten om ons in eerste instantie te richten op de Google Fit API.

Ik ben vervolgens de stappen gaan volgen zoals beschreven in de Google Fit Rest API-handleiding. Ik heb een OAuth 2.0-client-ID aangevraagd en een project aangemaakt waarin ik de Fitness API heb ingeschakeld. Tijdens dit proces kwam ik erachter wat 'scopes' zijn, aangezien je bij het aanmaken van je project specifieke toegangsniveaus moet definiëren voor de gegevens die je project van een Google-account wil gebruiken. Ik heb alle Fit-gegevens die ik nodig heb geselecteerd. Het is belangrijk op te merken dat deze scopes beperkt zijn, wat betekent dat ze toegang verlenen tot gevoelige informatie, en ik niet zomaar toestemming krijg om deze gegevens te gebruiken. Daarom zal ik bezoekers van de webpagina moeten vragen om toegang tot deze gegevens te verlenen. Ik heb besloten om [Google Sign-In](#) te gebruiken, aangezien dit automatisch het autorisatieverzoek voor je afhandelt.

Daarnaast heb ik de Google Fit-app gedownload om alvast mijn eigen gegevens te verzamelen, zodat ik later kan testen of de gegevens die ik ophaal daadwerkelijk correct zijn. Tijdens dit proces ontdekte ik ook dat je via Google Fit Apple Health kunt integreren, waardoor de aparte koppeling met Apple Health overbodig wordt. Daarom heb ik besloten om deze integratie niet langer op te nemen in het project.

2.2.2 Library Research – Community research

Op dit moment kan ik succesvol inloggen met mijn Google-account en ontvang ik de profielinformatie. Echter, nu sta ik voor de uitdaging om toegang te krijgen tot gegevens van de beperkte 'scopes'. Met behulp van Google Sign-In op mijn website ontvang ik na een succesvolle gebruikersaanmelding een toegangstoken. Nu wil ik graag weten hoe ik dit toegangstoken met een POST-verzoek naar de Google Fit API kan sturen, zodat ik de benodigde gegevens kan verkrijgen.

Ik heb geprobeerd informatie hierover te vinden, maar helaas kon ik niet veel relevante documentatie vinden. Daarom heb ik mijn [eigen vraag op Stack Overflow geplaatst](#). Helaas heb ik tot nu toe weinig reacties op mijn vraag ontvangen. Na uitgebreid zoeken ben ik echter gestuit op een [vraag van iemand anders](#) waarin nuttige informatie stond. Met deze informatie als leidraad ben ik begonnen met het maken van een GET-verzoek naar de Google API.

```
axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources/derived:com.google.step_count.delta:com.google.android.gms:estimated_steps/datasets/1694124000000000000-1694210400000000000")
```

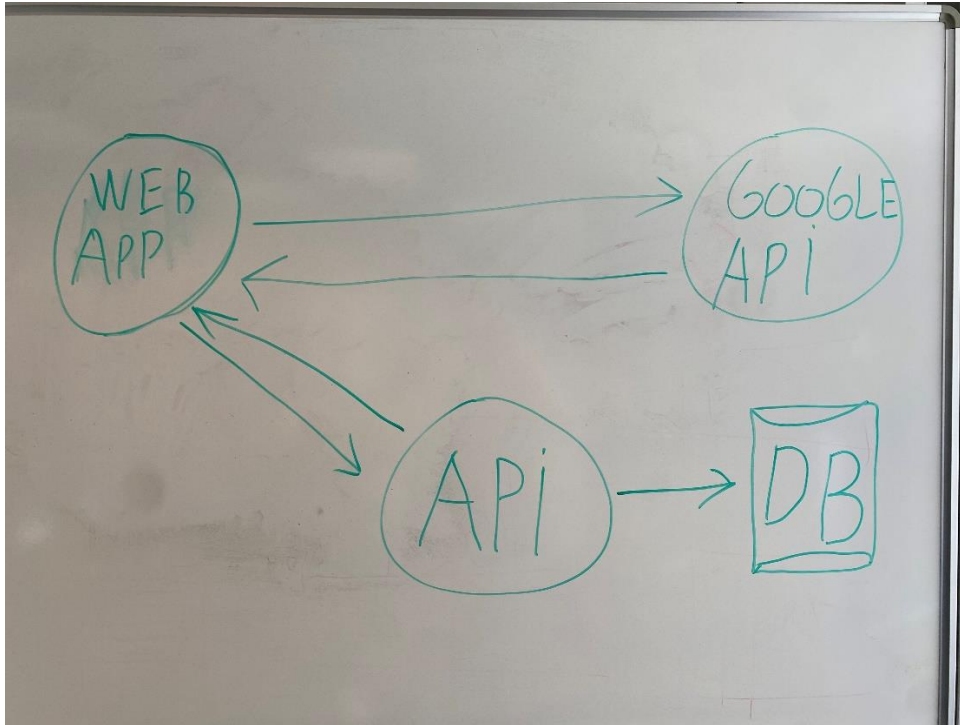
'me' verwijst naar de ingelogde gebruiker via Google. Vervolgens geef ik aan dat ik de gegevensbronnen van deze gebruiker wil bekijken, specifiek de geschatte stappentelling, en aan het einde geef ik met behulp van die nummers aan binnen welk tijdsbestek in milliseconden ik deze gegevens wil zien. Echter, ik bleef een '401 Unauthorized'-foutmelding ontvangen omdat ik geen token meestuurde. Toen herinnerde ik me dat ik tijdens mijn project in het derde semester van React een JWT-token moest meesturen, meestal via een 'Authorization'-header. Tot mijn verbazing werkte dit ook in dit geval.

```
function scopesRequest(){
  axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources/derived:com.google.step_count.delta:com.google.android.gms:estimated_steps/datasets/1694124000000000000-1694210400000000000", {
    headers: { Authorization: 'Bearer ' + token }
  })
  .then(function (response) {
    setStepRecords(response.data.point)
    calculateDailySteps();
  });
}
```

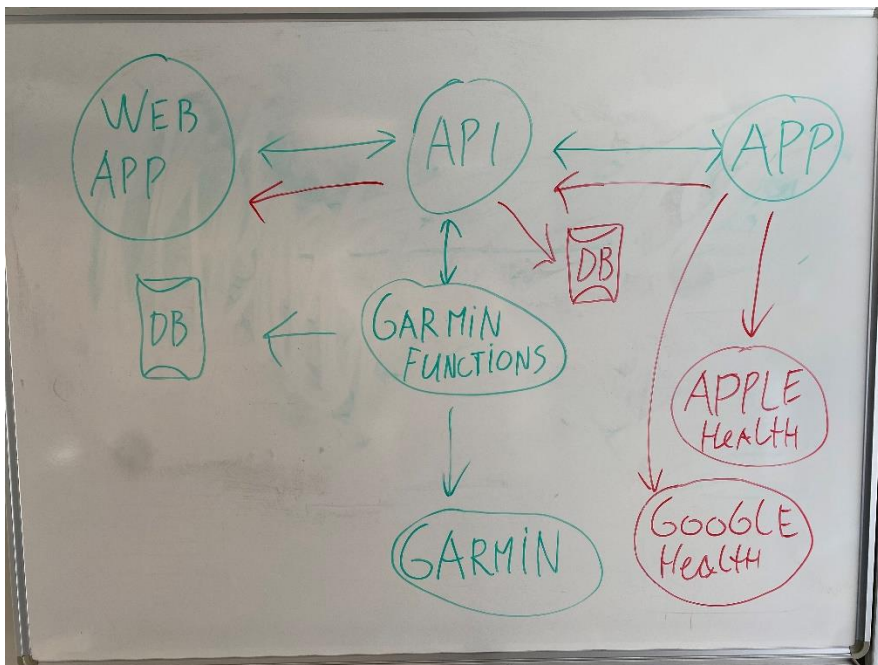
In eerste instantie riep ik de API aan en voerde ik berekeningen uit op de ontvangen gegevens om het totale aantal te verkrijgen. Later ontdekte ik echter dat je met behulp van een ander verzoek naar de [Google API](#) direct het totale aantal kunt verkrijgen.

2.2.3 Workshop Research - IT architecture sketching

Om een helder beeld te krijgen van de reikwijdte van het project, heb ik schetsen gemaakt op een whiteboard. Dit heeft me geholpen om duidelijkheid te krijgen over de aspecten waaraan ik zal werken, en het stelt mijn stagebegeleider ook in staat te zien dat ik het project begrijp. Dit is het systeem dat ik zal ontwikkelen in een testomgeving.



Dit is de gewenste uiteindelijke werking van het systeem.



2.2.4 Workshop Research – Prototyping

Mijn testomgeving zal dienen als een high-fidelity prototype waarin ik de mogelijkheden van de integratie kan verkennen en testen. Zodra ik tevreden ben met de staat van het prototype, zal ik het implementeren in het daadwerkelijke project

2.3 Qualiteit van het resultaat

Een prototype dat de Google API aanroept via de frontend-webpagina en vervolgens gegevens verzamelt.

2.4 Resultaat

Allereerst voer ik een POST-verzoek uit naar de Google Activity API en voer ik een aggregatie uit om het totaal te verkrijgen

```
function stepsRequest(){
    axios.post("https://www.googleapis.com/fitness/v1/users/me/dataset:aggregate",
    {
        "aggregateBy": [{
            "dataTypeName": "com.google.step_count.delta",
            "dataSourceId":
"derived:com.google.step_count.delta:com.google.android.gms:estimated_steps"
        }],
        "bucketByTime": { "durationMillis": 86400000 },
        "startTimeMillis": startTime,
        "endTimeMillis": endTime
    }, {
        headers: { Authorization: 'Bearer ' + token }
    })
    .then(function (response) {
        //console.log(response.data.bucket[0].dataset[0].point[0].value[0].intValue)
        setStepRecords(response.data.bucket[0].dataset[0].point[0].value[0].intValue)
    });
}
```

dataTypeName: Dit is de naam van het gegevenstype zoals te vinden op de Google API.

dataSourceId: Dit is de ID van de gegevensbron die je wilt verzamelen. Je kunt deze ID vinden door eerst een GET-verzoek uit te voeren om alle gegevensbronnen van een bepaald gegevenstype te bekijken.

```
axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources?dataTypeName=com.google.step_count.delta")
```

bucketByTime: Dit is de tijdsduur waarin je de gegevens wilt verzamelen.

startTimeMillis: Dit is het starttijdstip van de 'bucketByTime'.

endTimeMillis: Dit is het eindtijdstip van de 'bucketByTime'.

```
"bucketByTime": { "durationMillis": 86400000 },
                  "startTimeMillis": startTime,
                  "endTimeMillis": endTime
```

Vervolgens stuur ik de accesstoken mee met de autorizatie header

```
headers: { Authorization: 'Bearer ' + token }
```

en hierna sla ik het totaal wat ik terugkrijg van de api op in een locale variabele

```
setStepRecords(response.data.bucket[0].dataset[0].point[0].value[0].intVal)
```

Vervolgens controleer ik met behulp van een useEffect of de variabele is ingevuld.

```
useEffect(() => {
    if(stepRecords > 0){
        saveSteps();
    }
}, [stepRecords]);
```

Vervolgens stuur ik de data naar mijn API

```
function saveSteps(){
    axios.post('https://localhost:7212/api/Step', {
        DailySteps: stepRecords,
        StartTimeNanos: startTime,
        EndTimeNanos: endTime,
        userEmail: user.email
    })
    .catch(function (error) {
        console.log(error);
    });
}
```


3 Onderzoek 2

3.1 Vraag

Welke technische en functionele vereisten zijn nodig voor het ontwikkelen van een systeem dat in staat is om gegevens van externe API's te verzamelen en deze gegevens weer te geven?

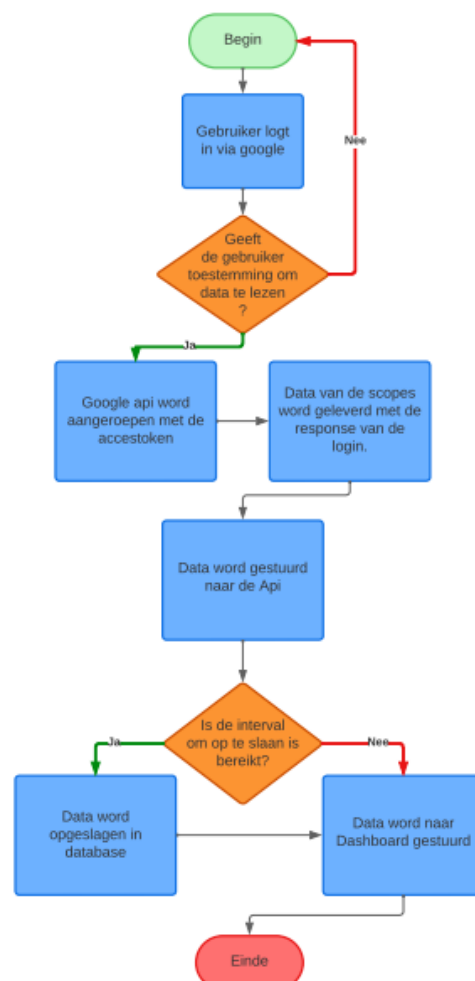
3.2 Hoe ben ik aan de slag gegaan?

3.2.1 Field Research – Problem analysis

Ik heb een gesprek gehad met mijn stagebegeleider en mijn collega Ruben, die heeft gewerkt aan het Move4Vitality-project. Zij hebben mij uitgelegd hoe het huidige systeem functioneert en waar de mogelijke verbeteringen voor mijn project zich bevinden. Ruben heeft de technische vereisten van het project met mij besproken, waaronder de programmeertalen en de benodigde software. In eerste instantie dachten mijn stagebegeleider en Ruben eraan om de Google API aan te roepen via mijn eigen API-backend, zoals bij de Garmin-functionaliteit is gedaan. Echter, ik kwam met het idee om dit mogelijk via de frontend te realiseren. Hierdoor zouden we de extra Garmin API kunnen elimineren, aangezien Garmin ook kan worden gekoppeld aan Apple en Google Health.

3.2.2 Field Research – Explore user requirements

Daarna heb ik een flowchart opgesteld om een visuele representatie te creëren van hoe gebruikers door de applicatie zouden navigeren.



Met behulp van deze flowchart heb ik een lijst van gebruikersverhalen samengesteld om de taken van elke gebruiker te definiëren. Ik heb vervolgens Planning Poker toegepast om een nauwkeurige schatting te maken van de complexiteit van deze functionaliteiten, zodat ik ze kon opnemen in mijn projectplanning. Daarnaast heb ik use cases ontwikkeld voor mijn gebruikersverhalen om te begrijpen hoe gebruikers in aanraking komen met deze functionaliteiten. Deze diagrammen en documentatie heb ik verwerkt in een Functioneel en Technisch ontwerp als resultaat

3.3 [Qualiteit van het resultaat](#)

Een functioneel & technisch ontwerp die de functionele en technische eisen van het project bevatten.

3.4 [Resultaat](#)

4 Onderzoek 3

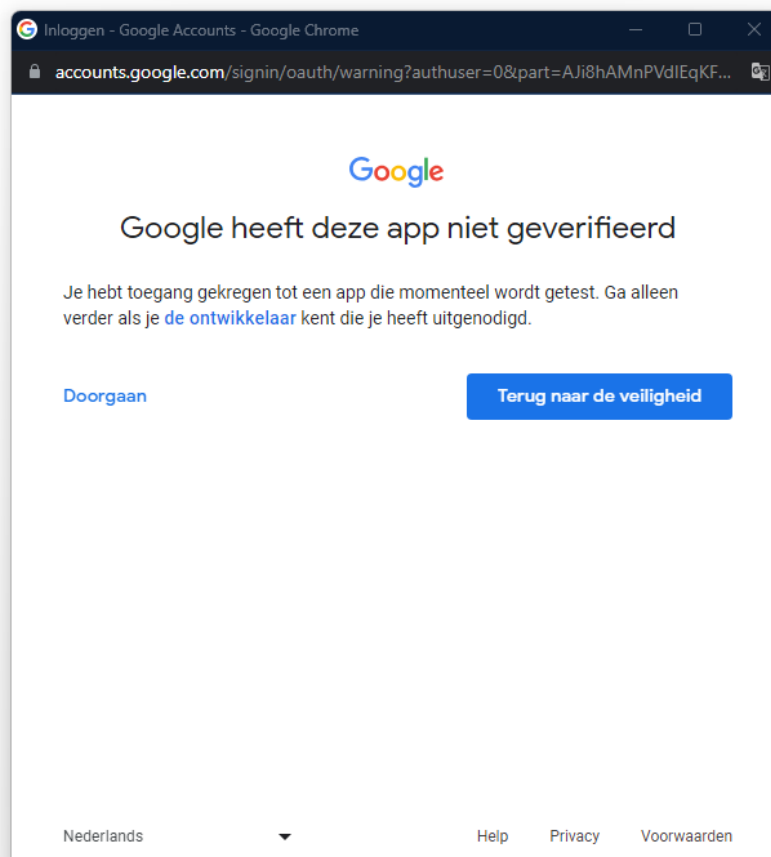
4.1 Vraag

Hoe moet omgegaan worden (uit technisch en regelgevings oogpunt) met privé/gezondheidsdata?

4.2 Hoe ben ik aan de slag gegaan?

4.2.1 Literature study – Library Research

Als eerste ben ik gaan onderzoeken hoe ik aan de beveiligde data van een google gebruiker kom in de google api, dit werkt doormiddel van het aanvragen van scopes. Gezondheidsdata van een gebruiker is een restricted scope dit betekent dat je autorisatie van de gebruiker nodig hebt om deze data te kunnen lezen. Je kunt ook je applicatie laten verifiëren door google, om je te laten verifiëren moet je applicatie voldoen aan de [Google Fit Developer and User data Policy](#) hierin staat hoe je moet omgaan met gevoelige gebruikers informatie en beperkingen waaraan je moet houden zoals niet voor scopes vragen die je niet nodig hebt en niet onnodig teveel gebruikers data gebruiken. Als je je applicatie niet verifiëerd kun je nog steeds data ophalen maar dan krijgt de gebruiker een veiligheids melding op je applicatie zie hieronder:



4.2.2 Expert Interview – Library Research

Ik heb een lijst gemaakt met alle activiteiten en gezondheids data die google aanbied, hier ben ik vervolgens met mijn stagebegeleider over in gesprek gegaan om te bepalen welke data ik ga gebruiken in de applicatie zodat ik het aantal scopes beperkt kan houden tot wat ik daadwerkelijk ga gebruiken. Ook heeft mijn stagebegeleider mij verteld om een lijst te maken wat er nog aan de applicatie moet gebeuren om de verificatie van google te krijgen.

4.2.3 Data Analytics – Lab Research

Ook heb ik Ruben gecontacteerd om de data in te kunnen zien van de huidige connectie met Garmin zodat ik weet wat ze nu gebruiken om te weergeven aan de fysiotherapeuten en hoe ik mogelijk op een later moment deze data kan samenvoegen.

4.2.4 Explore user requirements- Field Research

Ik ga in gesprek met mijn stagebegeleider, Ruben en hopelijk iemand werkzaam bij Move4Vitality om erachter te komen wat ze precies willen gaan doen op zo een applicatie hierdoor kan ik bepalen hoe ik de data wil verwerken en weergeven aan de fysiotherapeut.

4.3 Resultaat

4.4 Qualiteit van het resultaat