



Global Knowledge®

ilionx

Short recap day #1

“Inspiratiesessie”

Application Development

WORLDWIDE LOCATIONS

BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

2010



2022

Banking

Insurance

Government

Entertainment

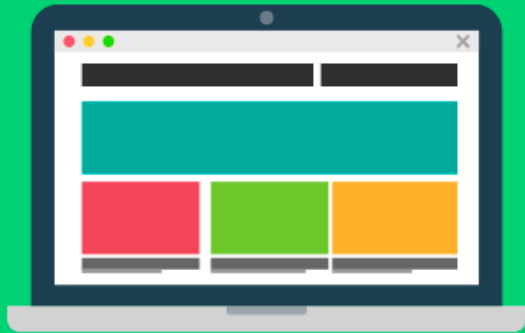
On-the-go navigation

Outdoor activities

Social media

And so on, and so on...

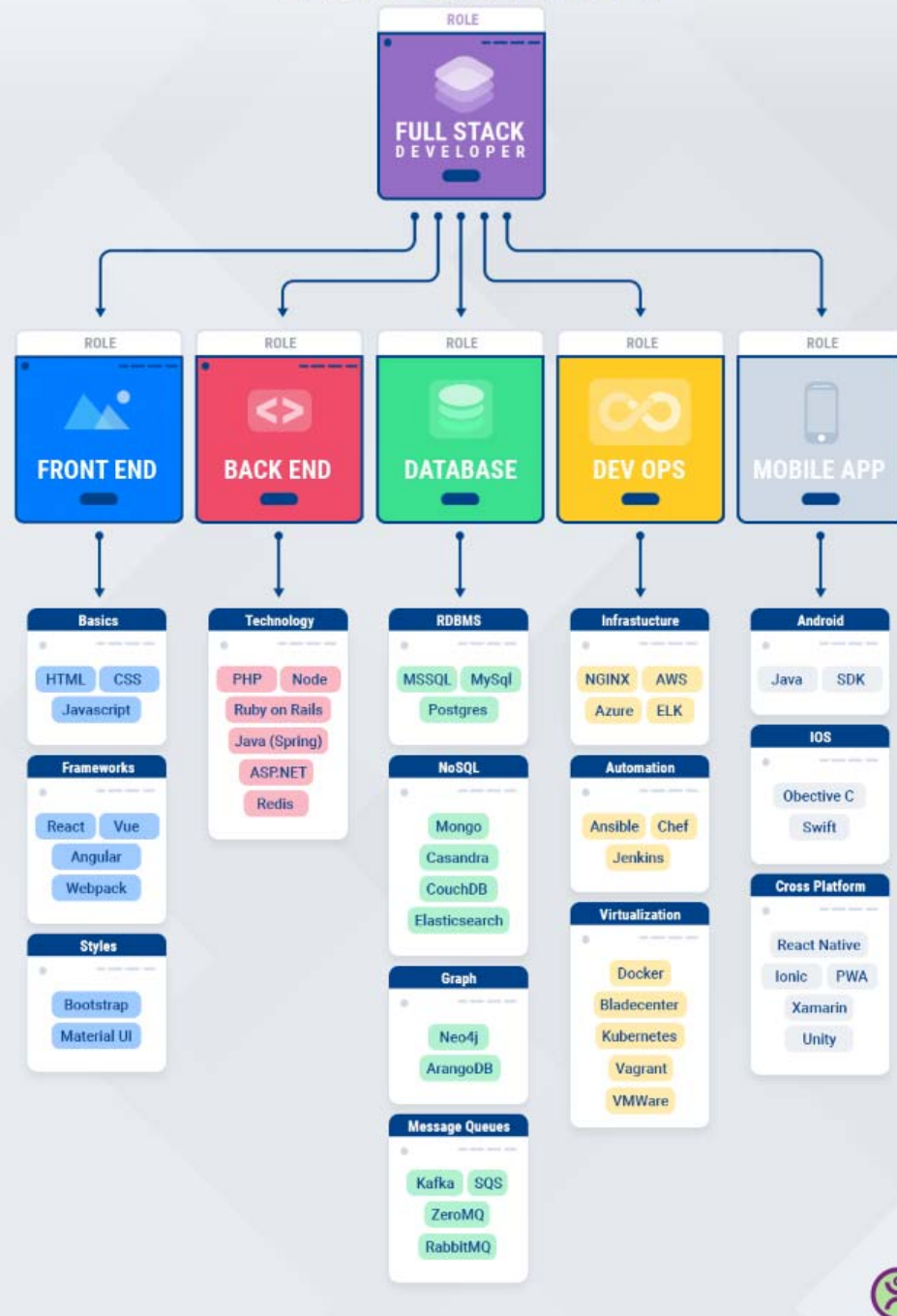


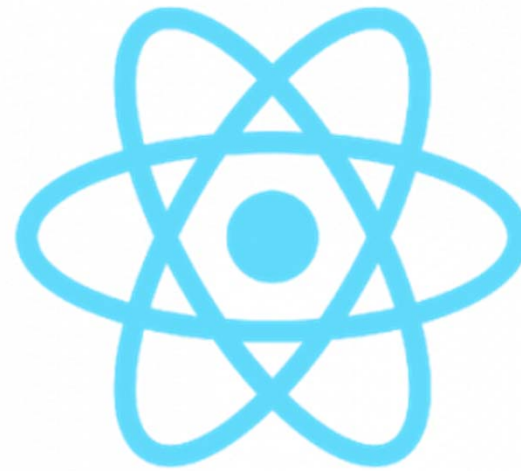


FRONTEND



BACKEND

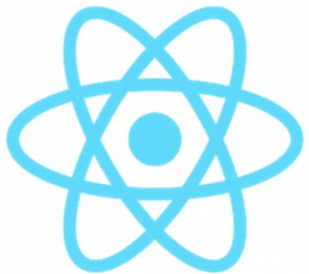
FULL STACK DEVELOPER
SKILLS BREAKDOWN



React



Similarities



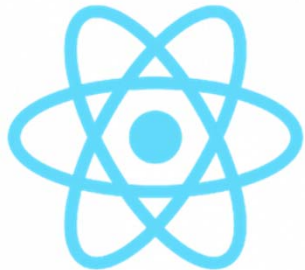
1. Single Page Applications
2. Components
3. Data binding, props, events
4. Routing
5. State management
6. Testing
7. Huge ecosystem
8. Huge community
9. High adaptation rate

Differences (apart from syntax)



Point of departure: *HTML template*, enhanced with framework specific tags and attributes

Aim: *one-stop-shop* / solution



React

Point of departure: *JavaScript* (JSX)

Build-all-yourself, possible *elective stress*

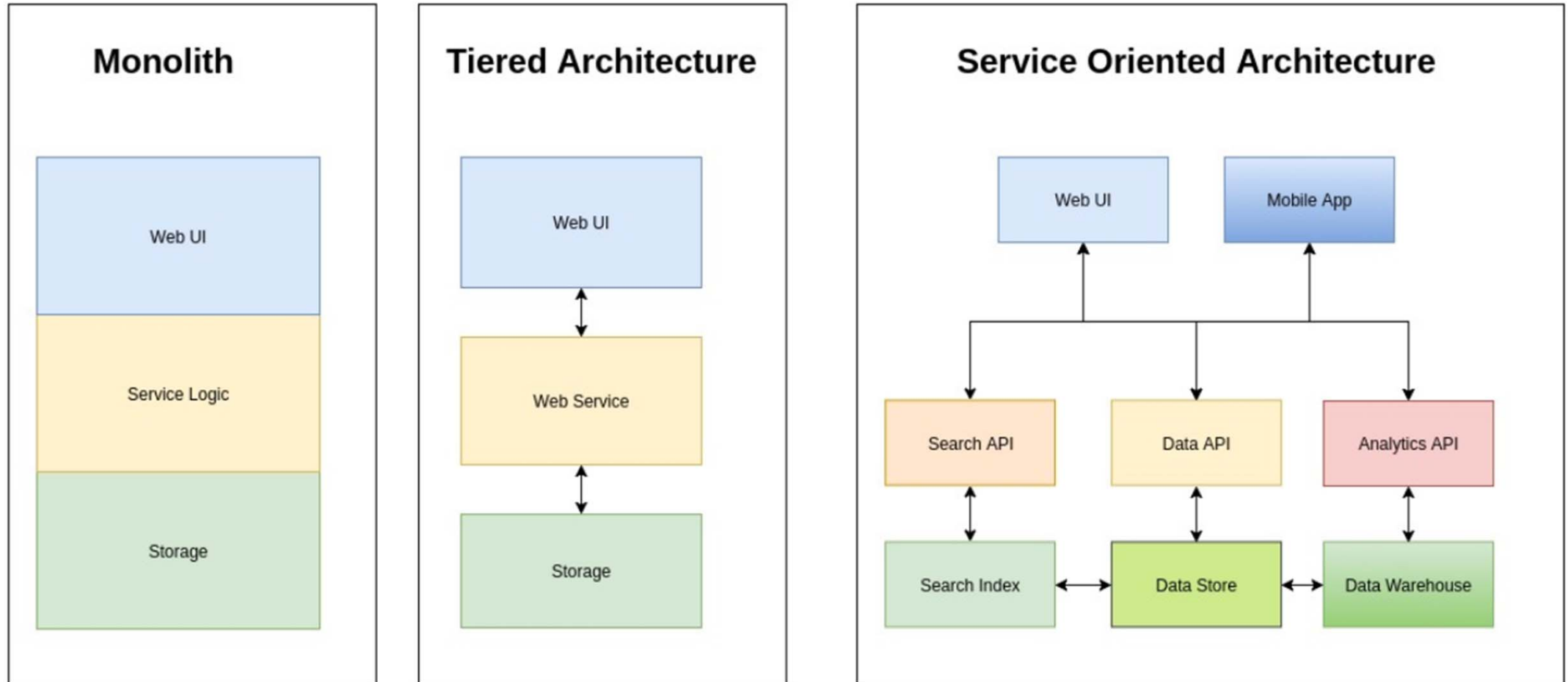


SVELTE

Point of departure: *no framework*
in the browser

Compile all code to plain JavaScript

Evolution towards a Service Oriented Architecture



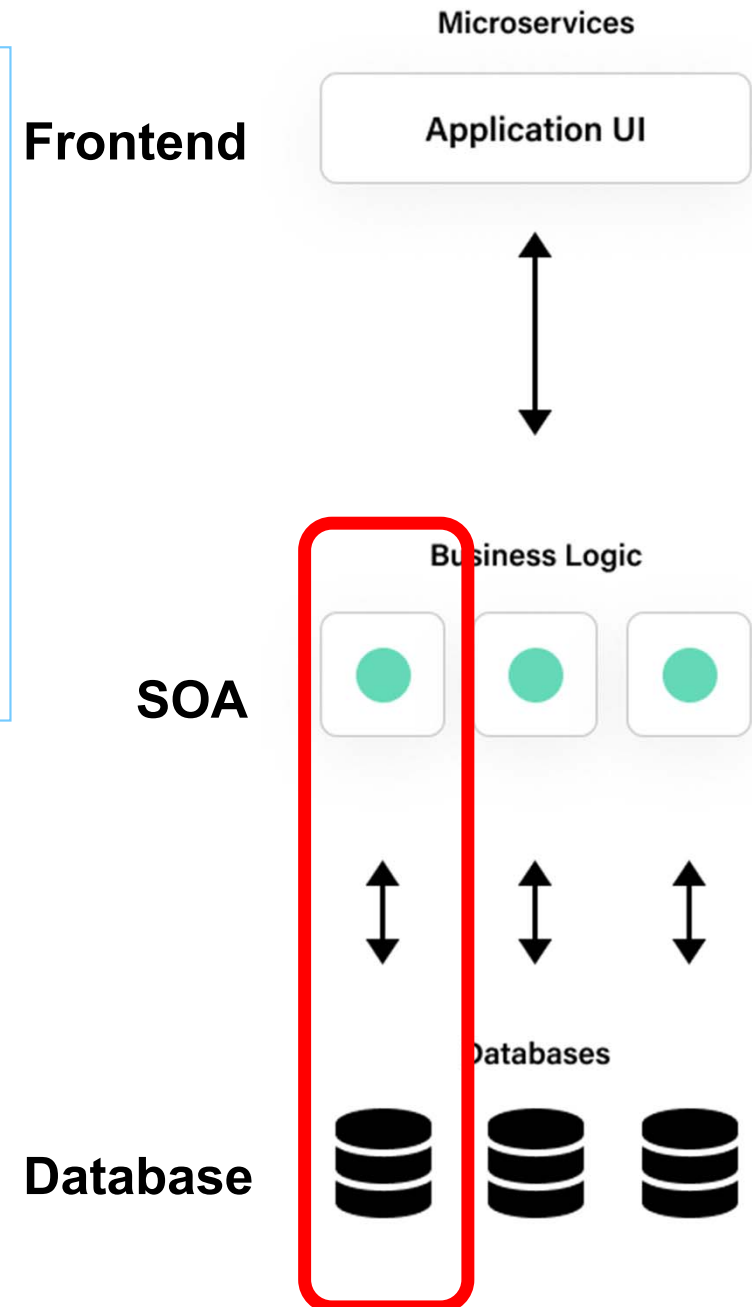
Example – Node.js small server + API

```
// server.js
// 1. Import and create express() application
const express = require('express');
const app = express();

// 2. Import routes and pass in the created app to bind the
// routes to the app.
const routes = require('./api-routes/routes');
routes(app);

// 3. Create a port to listen to
const port = process.env.PORT || 3000;

// 4. Start the server on the given port
app.listen(port, () => {
  console.log(`Listening on port http://localhost:${port}`);
});
```



Cases (2x)

case 01 - ilionx.pdf - Adobe Acrobat Pro

Bestand Bewerken Beeld Venster Help

Maken

1 / 3 119% Gereedschappen Opmerking Delen

ilionX

CASE 01 – INSPIRATIESESSIE - ILIONX

01 - Rijksmuseum applicatie

Bouw een applicatie die gebruikmaakt van de Rijksmuseum API. Met de applicatie kunnen mensen zoeken op naam van kunstenaar (*Rembrandt, Vermeer*) en zien ze een lijst met kunstwerken van deze kunstenaar. Vervolgens kunnen gedetailleerde gegevens over het betreffende kunstwerk worden getoond.

Je kunt dit opzetten volgens het principe van microservices:

1. Je app praat met jouw (gateway-) server.
2. Server stuurt request naar de API en transformeert/manipuleert desgewenst de resultaten.
3. Server stuurt kant-en-klare gegevens terug naar je app.

De applicatie heeft de volgende requirements:

- o De app gebruikt een front-end [framework] naar keuze: Angular, Vue, React, Flutter, Nuxt, Nest.js, Vanilla HTML/CSS/JavaScript, of nog anders.
- o De architectuur van de app gebruikt het API Gateway-principe. De Rijksmuseum-API is geschikt om rechtstreeks aan te spreken vanuit je frontend en JSON retour te geven, maar probeer dit op te zetten volgens het in de sessie besproken microservices-principe.
- o Zorg voor aantrekkelijke vormgeving. Als je een webapp maakt, mag je een CSS-framework zoals Bootstrap gebruiken, maar dit is niet verplicht. Je mag ook aanvullende libraries als Angular Material of PrimeNG gebruiken. Dit is ook niet verplicht. Bij Flutter/Ionic/... gebruik je natuurlijk de mobiele widgets die voor dat platform beschikbaar zijn.
- o De app moet bruikbaar zijn op mobiele apparaten (responsive).
- o De app bevat een zoekvak en een knop met voldoende aanwijzingen in de userinterface.
 - o Als de bezoeker een kunstenaar invult en op *Zoeken* klikt, wordt een lijst met kunstwerken getoond.
 - o Bij elke beschrijving wordt een kleine afbeelding van het kunstwerk getoond.
- o Selectie van een kunstwerk zorgt ervoor dat meer details voor dat kunstwerk worden getoond.
- o Zorg ervoor dat de app schaalbaar is. Gebruik het Single responsibility-principe voor componenten. Gebruik services voor communicatie met het backend.
- o Optioneel: gebruikers kunnen een kunstwerk als Favoriet aanmerken. Het wordt dan bewaard in HTML5 localStorage op het eigen device.

WERKWIJZE:

- o Frontend: gebruik een CLI van jouw voorkeur om het project te starten. Voeg eventueel aanvullende

case 02 - ilionx.pdf - Adobe Acrobat Pro

Bestand Bewerken Beeld Venster Help

Maken

1 / 3 119% Gereedschappen Opmerking Delen

ilionX

CASE 02 – INSPIRATIESESSIE - ILIONX

02 – Postcode en Google Maps applicatie

Bouw een applicatie die gebruikmaakt van een Postcode API en Google maps. Met de applicatie kunnen mensen een postcode intypen, waarna detailgegevens van die postcode (straat, stad, provincie) worden getoond. Vervolgens wordt een Google Maps-kaartje geopend waarin de betreffende locatie inclusief een marker wordt getoond.

Je kunt dit opzetten volgens het principe van microservices:

1. Je app praat met jouw (gateway-) server.
2. Server stuurt request naar de API en transformeert/manipuleert desgewenst de resultaten.
3. Server stuurt kant-en-klare gegevens terug naar je app.

De applicatie heeft de volgende requirements:

- o De app gebruikt een front-end [framework] naar keuze: Angular, Vue, React, Flutter, Nuxt, Nest.js, Vanilla HTML/CSS/JavaScript, of nog anders.
- o De architectuur van de app gebruikt het API Gateway-principe. Beide API's zijn geschikt om rechtstreeks aan te spreken vanuit je frontend en JSON retour te geven, maar probeer dit op te zetten volgens het in de sessie besproken microservices-principe.
- o Zorg voor aantrekkelijke vormgeving. Als je een webapp maakt, mag je een CSS-framework zoals Bootstrap gebruiken, maar dit is niet verplicht. Je mag ook aanvullende libraries als Angular Material of PrimeNG gebruiken. Dit is ook niet verplicht. Bij Flutter/Ionic/... gebruik je natuurlijk de mobiele widgets die voor dat platform beschikbaar zijn.
- o De app moet bruikbaar zijn op mobiele apparaten (responsive).
- o De app bevat twee zoekvakken (postcode en huisnummer) en een knop *Zoek*.
 - o Als de bezoeker een postcode invult en op *Zoek* klikt, wordt het bijbehorende adres, de stad en de provincie getoond.
 - o Onder het adres wordt een Google Map van de betreffende locatie getoond.
- o Er is een knop *Leegmaken* die de gegevens en de invoervelden wist, en de Google Map verbergt.
- o Zorg ervoor dat de app schaalbaar is. Gebruik het Single responsibility-principe voor componenten. Gebruik frontend services voor communicatie met het backend.

WERKWIJZE:

- o Frontend: gebruik een CLI van jouw voorkeur om het project te starten. Voeg eventueel aanvullende libraries of packages toe.

Questions?

Today

- Presentation and discussion of cases
- Comparison of frontend frameworks, 2nd half
- Continuous Integration / Continuous Deployment
- Example + workshop CI / CD
- Q & A, closing

Agenda

28 Feb. 2022 – Monday

~ 18:00 start

~ 19:30 short coffee break

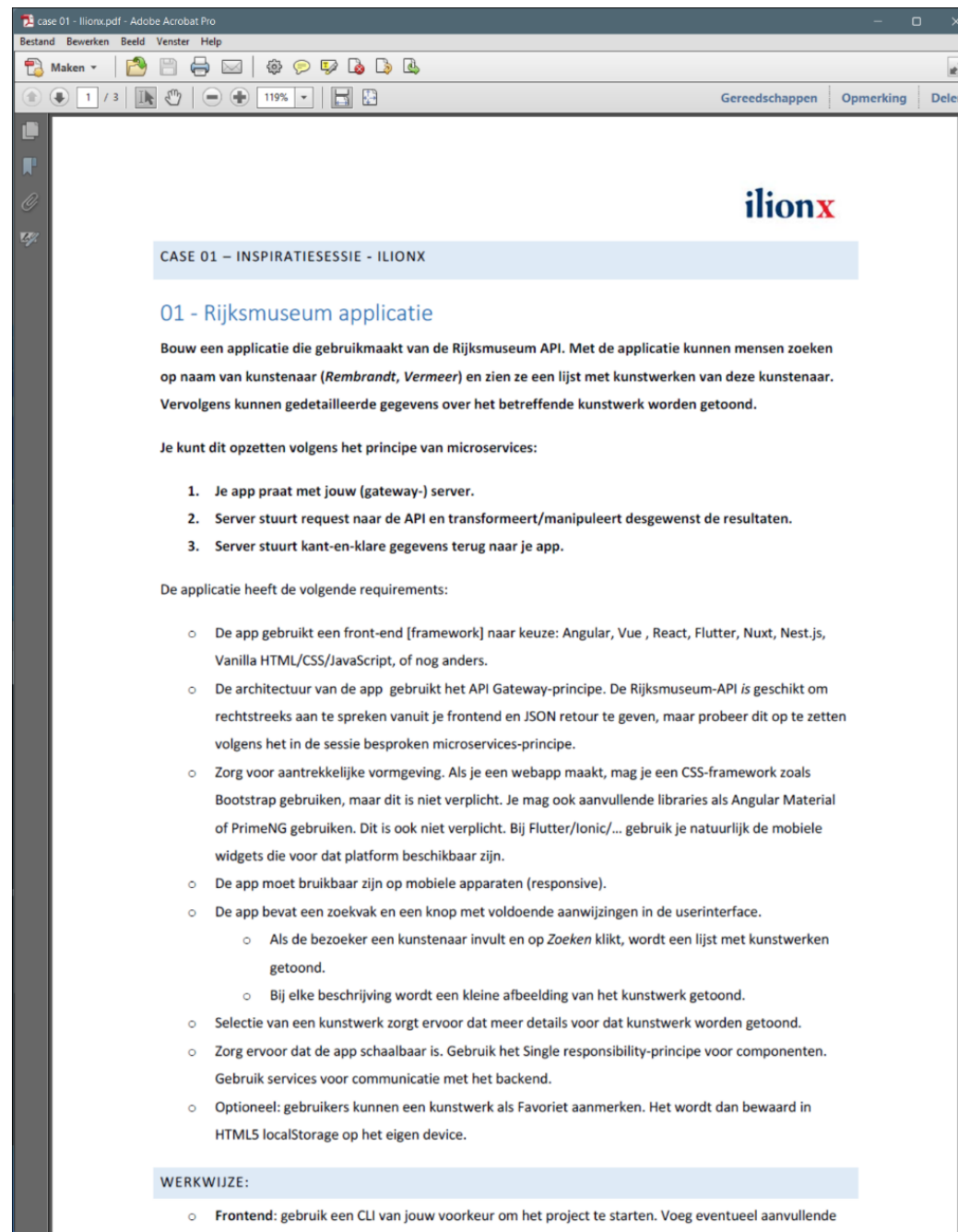
~ 21:00-21:15 Closing



Discussion of cases

Volunteers?

Case – Rijksmuseum app



case 01 - ilionx.pdf - Adobe Acrobat Pro

Bestand Bewerken Beeld Venster Help

Maken

1 / 3 119% Gereedschappen Opmerking Delen

ilionx

CASE 01 – INSPIRATIEESSIE - ILIONX

01 - Rijksmuseum applicatie

Bouw een applicatie die gebruikmaakt van de Rijksmuseum API. Met de applicatie kunnen mensen zoeken op naam van kunstenaar (*Rembrandt, Vermeer*) en zien ze een lijst met kunstwerken van deze kunstenaar. Vervolgens kunnen gedetailleerde gegevens over het betreffende kunstwerk worden getoond.

Je kunt dit opzetten volgens het principe van microservices:

1. Je app praat met jouw (gateway-) server.
2. Server stuurt request naar de API en transformeert/manipuleert desgewenst de resultaten.
3. Server stuurt kant-en-klare gegevens terug naar je app.

De applicatie heeft de volgende requirements:

- De app gebruikt een front-end [framework] naar keuze: Angular, Vue, React, Flutter, Nuxt, Nest.js, Vanilla HTML/CSS/JavaScript, of nog anders.
- De architectuur van de app gebruikt het API Gateway-principe. De Rijksmuseum-API is geschikt om rechtstreeks aan te spreken vanuit je frontend en JSON retour te geven, maar probeer dit op te zetten volgens het in de sessie besproken microservices-principe.
- Zorg voor aantrekkelijke vormgeving. Als je een webapp maakt, mag je een CSS-framework zoals Bootstrap gebruiken, maar dit is niet verplicht. Je mag ook aanvullende libraries als Angular Material of PrimeNG gebruiken. Dit is ook niet verplicht. Bij Flutter/Ionic/... gebruik je natuurlijk de mobiele widgets die voor dat platform beschikbaar zijn.
- De app moet bruikbaar zijn op mobiele apparaten (responsive).
- De app bevat een zoekvak en een knop met voldoende aanwijzingen in de userinterface.
 - Als de bezoeker een kunstenaar invult en op *Zoeken* klikt, wordt een lijst met kunstwerken getoond.
 - Bij elke beschrijving wordt een kleine afbeelding van het kunstwerk getoond.
- Selectie van een kunstwerk zorgt ervoor dat meer details voor dat kunstwerk worden getoond.
- Zorg ervoor dat de app schaalbaar is. Gebruik het Single responsibility-principe voor componenten. Gebruik services voor communicatie met het backend.
- Optioneel: gebruikers kunnen een kunstwerk als Favoriet aanmerken. Het wordt dan bewaard in HTML5 localStorage op het eigen device.

WERKWIJZE:

- Frontend: gebruik een CLI van jouw voorkeur om het project te starten. Voeg eventueel aanvullende

Case 2 – Postcode & Google Maps

case 02 - ilionx.pdf - Adobe Acrobat Pro

Bestand Bewerken Beeld Venster Help

Maken

1 / 3 119% Gereedschappen Opmerking Delen

ilionx

CASE 02 - INSPIRATIESESSIE - ILIONX

02 – Postcode en Google Maps applicatie

Bouw een applicatie die gebruikmaakt van een Postcode API en Google maps. Met de applicatie kunnen mensen een postcode intypen, waarna detailgegevens van die postcode (straat, stad, provincie) worden getoond. Vervolgens wordt een Google Maps-kaartje geopend waarin de betreffende locatie inclusief een marker wordt getoond.

Je kunt dit opzetten volgens het principe van microservices:

1. Je app praat met jouw (gateway-) server.
2. Server stuurt request naar de API en transformeert/manipuleert desgewenst de resultaten.
3. Server stuurt kant-en-klare gegevens terug naar je app.

De applicatie heeft de volgende requirements:

- De app gebruikt een front-end [framework] naar keuze: Angular, Vue , React, Flutter, Nuxt, Nest.js, Vanilla HTML/CSS/JavaScript, of nog anders.
- De architectuur van de app gebruikt het API Gateway-principe. Beide API's zijn geschikt om rechtstreeks aan te spreken vanuit je frontend en JSON retour te geven, maar probeer dit op te zetten volgens het in de sessie besproken microservices-principe.
- Zorg voor aantrekkelijke vormgeving. Als je een webapp maakt, mag je een CSS-framework zoals Bootstrap gebruiken, maar dit is niet verplicht. Je mag ook aanvullende libraries als Angular Material of PrimeNG gebruiken. Dit is ook niet verplicht. Bij Flutter/Ionic/... gebruik je natuurlijk de mobiele widgets die voor dat platform beschikbaar zijn.
- De app moet bruikbaar zijn op mobiele apparaten (responsive).
- De app bevat twee zoekvakken (postcode en huisnummer) en een knop Zoek.
 - Als de bezoeker een postcode invult en op Zoek klikt, wordt het bijbehorende adres, de stad en de provincie getoond.
 - Onder het adres wordt een Google Map van de betreffende locatie getoond.
- Er is een knop *Leegmaken* die de gegevens en de invoervelden wist, en de Google Map verbergt.
- Zorg ervoor dat de app schaalbaar is. Gebruik het Single responsibility-principe voor componenten. Gebruik frontend services voor communicatie met het backend.

WERKWIJZE:

- **Frontend:** gebruik een CLI van jouw voorkeur om het project te starten. Voeg eventueel aanvullende libraries of packages toe.

Cont'd

*Comparison of Frontend
Frameworks (2nd half)*



Global Knowledge®

ilionx

Closing

WORLDWIDE LOCATIONS

BELGIUM CANADA COLOMBIA DENMARK EGYPT FRANCE IRELAND JAPAN KOREA MALAYSIA MEXICO NETHERLANDS NORWAY QATAR
SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

We covered...

- The **current state** of application development
- Frontend **frameworks**
 - Features
 - Similarities
 - Differences
- **Microservice** architecture
- **CI / CD**

As always...

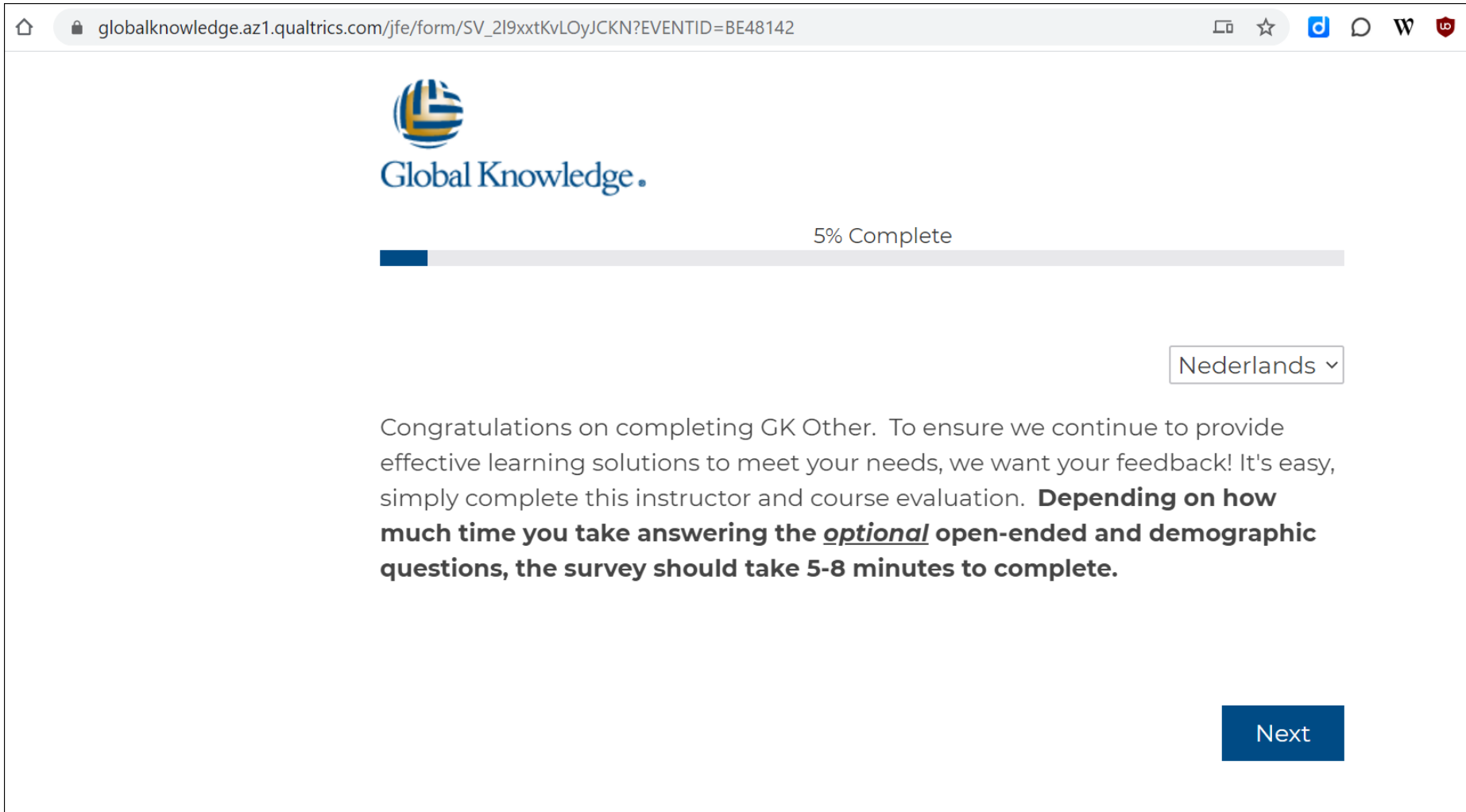
*There is much, much
more to learn!*

`/* LIFE RUNS ON CODE */`



Eval

<https://bit.ly/3IthVJD>



The screenshot displays a web browser window with the URL `globalknowledge.az1.qualtrics.com/jfe/form/SV_2l9xxtKvLOYJCKN?EVENTID=BE48142`. The page features the Global Knowledge logo at the top left. A progress bar indicates that the survey is 5% complete. On the right side, there is a dropdown menu currently showing 'Nederlands'. The main text of the survey reads: 'Congratulations on completing GK Other. To ensure we continue to provide effective learning solutions to meet your needs, we want your feedback! It's easy, simply complete this instructor and course evaluation. **Depending on how much time you take answering the optional open-ended and demographic questions, the survey should take 5-8 minutes to complete.**' At the bottom right, there is a blue button labeled 'Next'.

Let's connect

LinkedIn profile page for Peter Kassenaar. The page shows a search bar, navigation icons (Home, My Network, Jobs, Messaging, Notifications, Me, Work), and a profile section with education details. The 'Featured Skills & Endorsements' section is highlighted with a red circle. It lists skills and the number of endorsements: Teaching (3), TypeScript (3), and Angular (2). The right sidebar includes a button to 'Add new profile section' and a 'Messaging' button at the bottom.

duursporten (lopen, zwemmen, fietsen, roeien, schaatsen).

Laatste jaar in combinatie met studie Bewegingswetenschappen, Rijksuniversiteit Groningen (RUG).

See more education ▾

Featured Skills & Endorsements Add a new skill ✎

Teaching · 3 Youri Pattynama and 2 connections have given endorsements for this skill

TypeScript · 3 Youri Pattynama and 2 connections have given endorsements for this skill

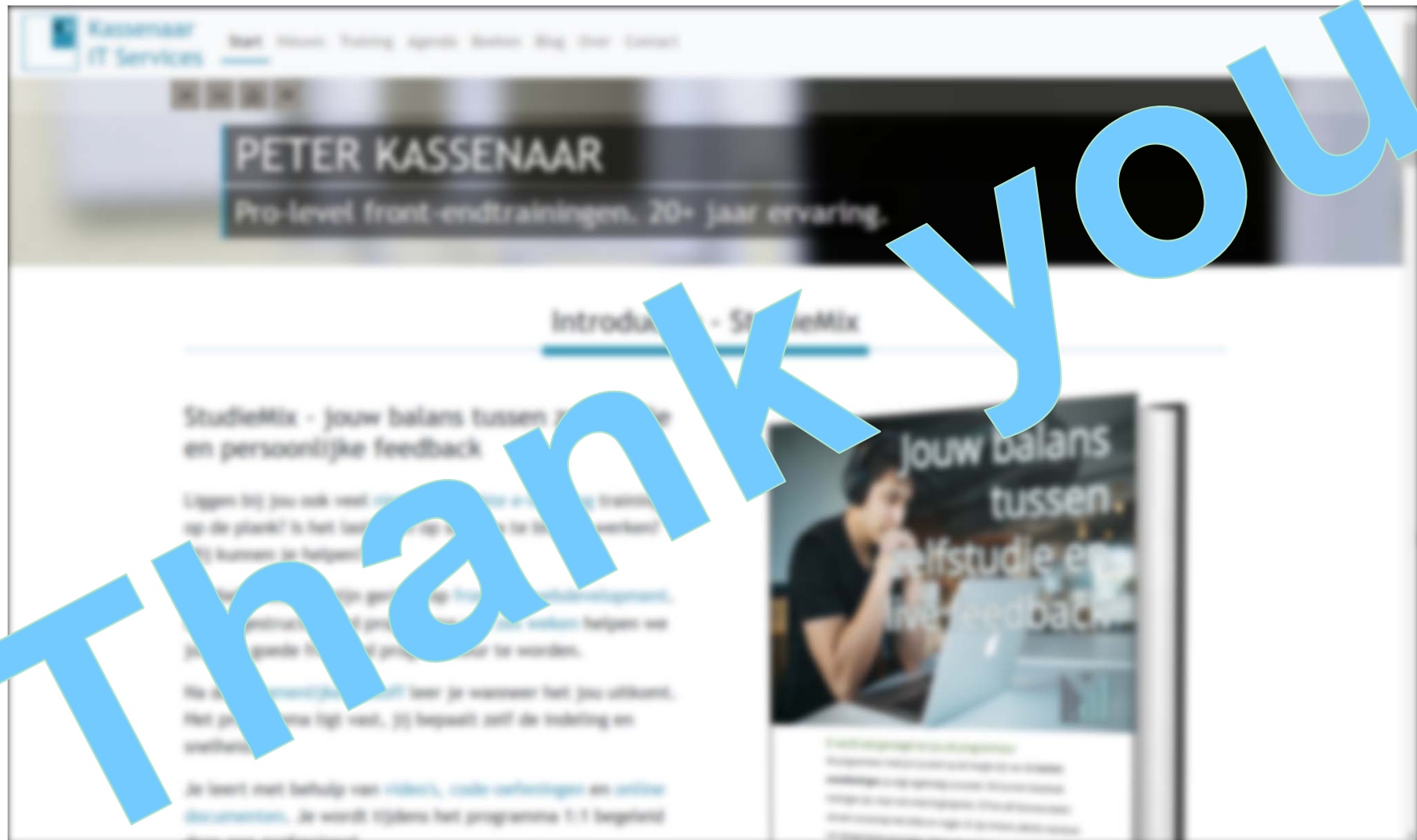
Angular · 2 Momen El Kamri and 1 connection have given endorsements for this skill

See 26 more skills ▾

Recommendations Ask to be recommended ✎

Messaging

<https://www.linkedin.com/in/peterkassenaar/>



www.kassenaar.com