

REACT NATIVE CASE 01 – PROUD NERDS

01 – Country Picker

Bouw een React Native applicatie waarmee gebruikers informatie over landen kunnen opzoeken. Je gebruikt hiervoor de (open) RestCountries API. Met de applicatie kunnen mensen zoeken op (een deel van de) naam van een land zoals `Bel` (resultaat: `Belgium`, `Belize`, etc) en zien ze een lijst met landen die met de betreffende letters beginnen. Vervolgens kunnen gedetailleerde gegevens over het betreffende land worden getoond.

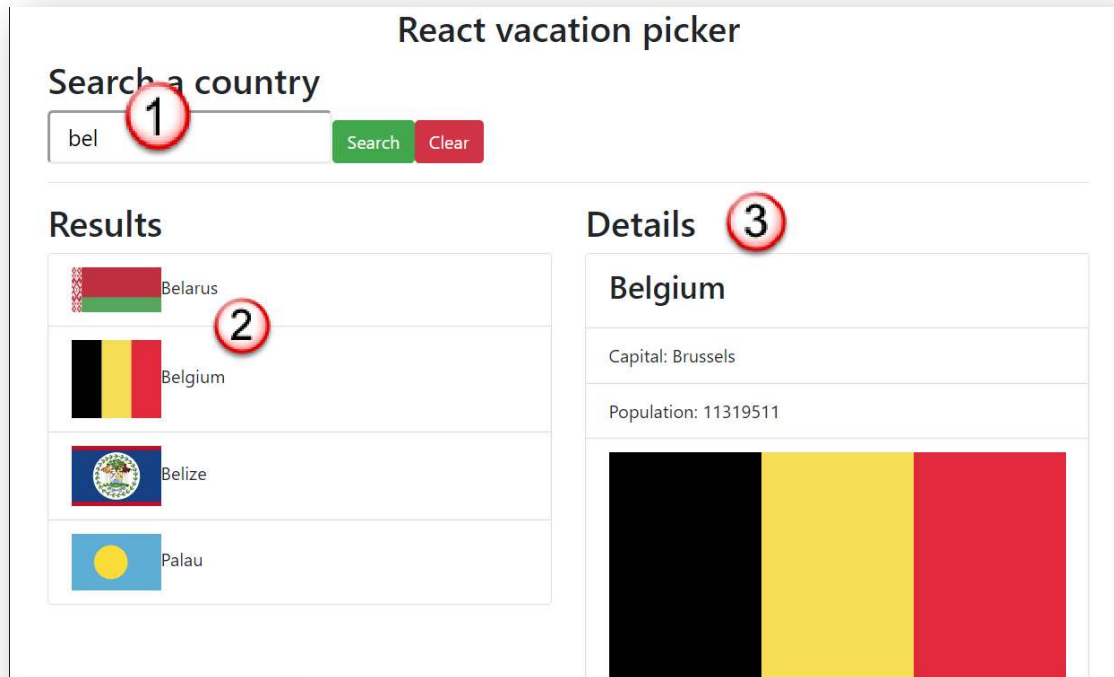
De applicatie heeft de volgende requirements:

- De app is gemaakt in React Native en draait op een Android of iOS emulator en/of device via de Expo Go-app.
- Naar keuze mag een aanvullend UI-component framework zoals *React Native UI Lib*, *React Native Elements* of anders worden gebruikt. De applicatie mag echter ook gebruik maken van de standaard React Native Core Components zoals `<View>`, `<Text>`, `<Image>` enzovoort.
- De app bevat een zoekvak en een knop en voldoende aanwijzingen in de userinterface.
 - Als de bezoeker (een deel van) een landnaam invult en op *Zoeken* klikt, wordt een lijst met landen getoond.
 - In de lijst wordt de vlag van het land, de naam, de hoofdstad en het inwoneraantal getoond.
- Er moet een knop zijn om de lijst weer leeg te maken/te verbergen.
- Als een land wordt geselecteerd, worden extra details voor dat land getoond.
 - Denk aan onderdelen als regio, subregio, telefooncode, munteenheden (currency), native name enzovoort.
 - De wijze waarop details worden getoond staat vrij. In een `alert()` venster (=eenvoudig) in een modale popup (=meer complex), of via routing naar een detailpagina.
- Optioneel: een land kan als *Favoriet* worden gemarkeerd. Favoriete landen worden opgeslagen in `localStorage` (of via andere opslagwijze) op het apparaat.
 - Er moet in dat geval een voorziening zijn om de lijst met favoriete landen te tonen, te verbergen, leeg te maken enzovoort.

WERKWIJZE:

- Begin een nieuwe React Native app – gebruik de standaard structuur en -architectuur.
- Gebruik de REST Countries API, op <https://restcountries.com/>.
 - Gebruik het endpoint
`https://restcountries.com/<versienummer>/name/{name}` om landinformatie te vinden.
 - Lees de documentatie op de website.

- Gebruik naar keuze `fetch()` of `axios` voor communicatie met de API.
- Gebruik functional components en React Hooks om de landen in de state op te slaan. Je code wordt zoiets als `const [countries, setCountries] = useState([])`.
- Een voorbeeld van hoe de app er uit zou kunnen zien (hier: web, bouw dit om naar React Native):



HINTS/TIPS

- Enkele hints/tips die je verder kunnen helpen bij ontwikkeling van de app:
 - De API bevat verwijzingen naar `.png`-versies van de vlaggen. Als je de (standaard) `.svg`-versies van de vlag wilt tonen, kijk dan naar `react-native-svg` (want React Native kan van zichzelf geen SVG-bestanden tonen). Meer info: <https://docs.expo.dev/versions/latest/sdk/svg/>
 - Als je routing wilt gebruiken voor een Homepage, Detailpagina en bijvoorbeeld een About-pagina, kijk dan naar <https://reactnavigation.org/>.
 - Als je een UI-framework wilt gebruiken, kijk dan bijvoorbeeld naar:
 - RNUI - <https://wix.github.io/react-native-ui-lib/>
 - React Native Elements: <https://reactnativeelements.com/>
 - De documentatie van React Native is beschikbaar op <https://reactnative.dev/>.
 - *Google is your friend. Stack Overflow is your neighbour!*

Veel succes!

Peter Kassenaar, info@kassenaar.com
