React Native

Projet développement mobile

PARTIE 4: Requêtes HTTP:

Formateur : Djemai Samy

1/ Drawer Navigation:

1.1/ Installation du package:

```
expo install @react-navigation/drawer
```

1.2/ Création du composant:

Créer le composant **./Components/Drawer/GlobalDrawer/GlobalDrawer.jsx**:

```jsx
import { createDrawerNavigator } from "@react-navigation/drawer";
import ProfilStack from "../../stacks/ProfilStack";

const Drawer = createDrawerNavigator();
const GlobalDrawer = () => {
  return (
    <Drawer.Navigator>
      <Drawer.Screen name='profilstack' component={ProfilStack} />
    </Drawer.Navigator>
  );
};

export default GlobalDrawer;
```

1.3/ Utiliser le composant:

Dans **./App.js**:

```
import { useState } from "react";
import { StyleSheet, View } from "react-native";
import Auth from "./components/page/Auth/Auth";
import { UserContext } from "./contexts/UserContext";
import { NavigationContainer } from "@react-navigation/native";
import GlobalDrawer from "./components/Drawer/GlobalDrawer/GlobalDrawer";
export default function App() {
  const [user, setUser] = useState(null);
  return (
    <UserContext.Provider value={{ user, setUser }}>
      <View style={styles.container}>
        <NavigationContainer>
          {user ? <GlobalDrawer /> : <Auth />}
        </NavigationContainer>
      </View>
    </UserContext.Provider>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
});
```

1.4/ Personnaliser le Header:

Nous avons toujours la possibilité de personnaliser le header grâce aux attributs options et screenOptions.

Cette fois, nous allons voir comment remplacer le header par un de nos composants.

Création du composant Navbar:

Créer le composant: **./Components/Container/Navbar/Navbar.jsx**:

```
import React, { useContext } from "react";
import { View, Text, StyleSheet, TouchableOpacity } from "react-native";
import { AntDesign } from "@expo/vector-icons";
import { UserContext } from "../../../contexts/UserContext";

const Navbar = ({navigation, options}) => {
  const { setUser } = useContext(UserContext);

  function openMenu() {
    navigation.toggleDrawer();
  }

  function logout() {
    setUser(null);
  }

  return (
    <View style={styles.container}>
      <TouchableOpacity onPress={openMenu}>
        <AntDesign name='menu-fold' size={32} color='whitesmoke' />
      </TouchableOpacity>

      <Text style={styles.title}>{options.title}</Text>

      <TouchableOpacity onPress={logout}>
        <AntDesign name='logout' size={32} color='whitesmoke' />
      </TouchableOpacity>
    </View>
  );
};

export default Navbar;
```

```
const styles = StyleSheet.create({
  container: {
    backgroundColor: "#2C3E50",
    display: "flex",
    flexDirection: "row",
    justifyContent: "space-between",
    padding: 16,
    alignItems: "center",
    paddingTop: 35,
  },
  title: {
    color: "whitesmoke",
    fontSize: 25,
  },
});
```

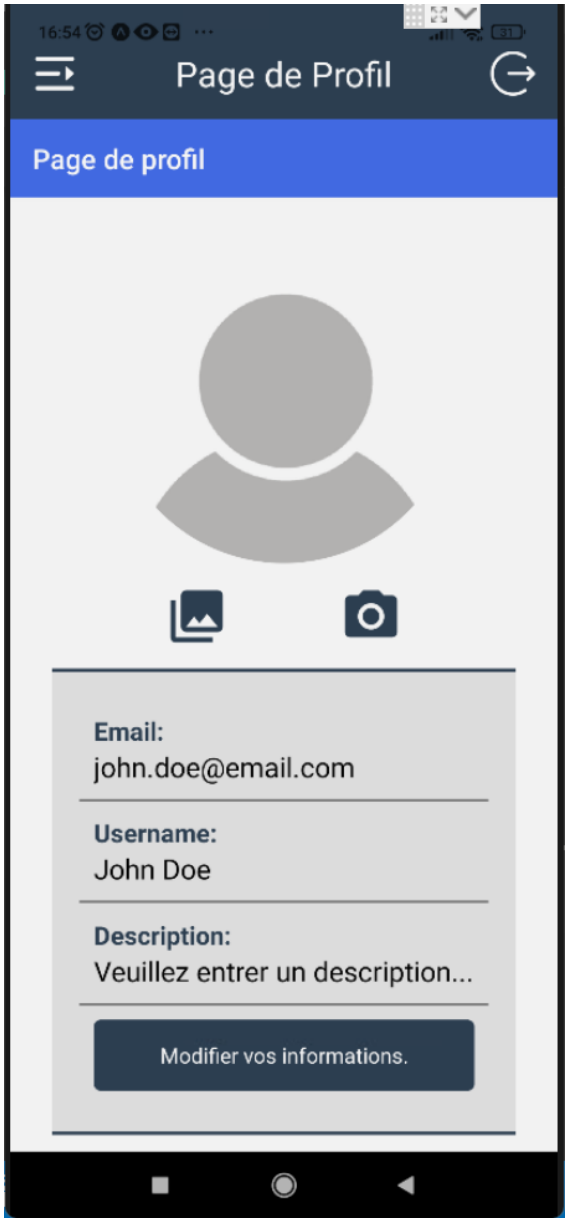
Rendre le composant dans le Drawer:

```
import { createDrawerNavigator } from "@react-navigation/drawer";
import Navbar from "../Containers/Navbar/Navbar";
import ProfilStack from "../Stacks/ProfilStack/ProfilStack";

const Drawer = createDrawerNavigator();

const GlobalDrawer = () => {
  return (
    <Drawer.Navigator
      screenOptions={{
        header: (navProps) => <Navbar {...navProps} />,
      }}
    >
      <Drawer.Screen
        name='profilstack'
        component={ProfilStack}
        options={{
          title: "Page de profil",
        }}
      />
    </Drawer.Navigator>
  );
};

export default GlobalDrawer;
```


16:54
Page de Profil
Page de profil
Email:
john.doe@email.com
Username:
John Doe
Description:
Veuillez entrer un description...
Modifier vos informations.

1.5/ Exercice: Map:

1- Créer un composant: **./Components/Pages/Map/Map.jsx**

2- Ajouter un écran dans **GlobalDrawer** avec le composant Map.

3- Utiliser la documentation (**MapView**), pour afficher une carte.

4- Utiliser la documentation (**Location**) pour:

4.1- Demander la permission de connaître la position de l'utilisateur.

4.2- Demander la position de l'utilisateur, une fois la permission acceptée.

4.3- Utiliser dans **MapView** l'attribut:

region={{longitude, latitude, longitudeDelta, latitude}}, pour afficher la position de l'utilisateur.

1.6/Solution: Map:

```
import { useEffect, useState } from "react";
import { View, Text, StyleSheet } from "react-native";
import MapView from "react-native-maps";
import * as Location from "expo-location";
export default function Map () {
  const [userLocation, setUserLocation] = useState({
    longitude: 2,
    latitude: 48,
    longitudeDelta: 1,
    latitudeDelta: 1,
  });

  useEffect(() => {
    (async () => {
      let positionPermission =
        await Location.getForegroundPermissionsAsync();
      if (positionPermission.granted) {
        let position = await Location.getCurrentPositionAsync({});
        setUserLocation({
          longitude: position.coords.longitude,
          latitude: position.coords.latitude,
          longitudeDelta: 0.1,
          latitudeDelta: 0.1,
        });
      }
    })();
  }, []);
  return (
    <View style={styles.container}>
      <MapView style={styles.map} region={userLocation} />
    </View>
  );
};
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "#2c3e50",
  },
  map: {
    width: "100%",
    height: "100%",
  },
});
```

2/ Requêtes HTTP:

Nous allons utiliser une API gratuite pour effectuer des requêtes HTTP et recevoir la liste des personnages de Game of thrones . Pour cela nous allons utiliser un package populaire qui permet d'envoyer des requêtes:

```
expo install axios
```

2.1/ Mise en place de la structure:

Créer le composant: **./Components/Page/GOT/GOT.jsx**

```
import React, { Component } from 'react';
import { View, Text, StyleSheet } from 'react-native';

const GOT = () => {
  return (
    <View style={styles.container}>
      <Text>GOT</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#2c3e50',
  },
});

export default GOT;
```

Ajouter un écran dans le **GlobalDrawer**:

```
import { createDrawerNavigator } from "@react-navigation/drawer";
import Navbar from "../Containers/Navbar/Navbar";
import GOT from "../Pages/GOT/GOT";
import Map from "../Pages/Map/Map";
import ProfilStack from "../Stacks/ProfilStack/ProfilStack";

const Drawer = createDrawerNavigator();

const GlobalDrawer = () => {
  return (
    <Drawer.Navigator
      screenOptions={{
        header: (navProps) => <Navbar {...navProps} />,
      }}
    >
      <Drawer.Screen
        name='profilstack'
        component={ProfilStack}
        options={{
          title: "Page de profil",
        }}
      />

      <Drawer.Screen
        name='map'
        component={Map}
        options={{
          title: "Carte",
        }}
      />

      <Drawer.Screen
        name='got'
        component={GOT}
        options={{
          title: "Personnages de Game of thrones",
        }}
      />
    </Drawer.Navigator>
  );
};
export default GlobalDrawer;
```

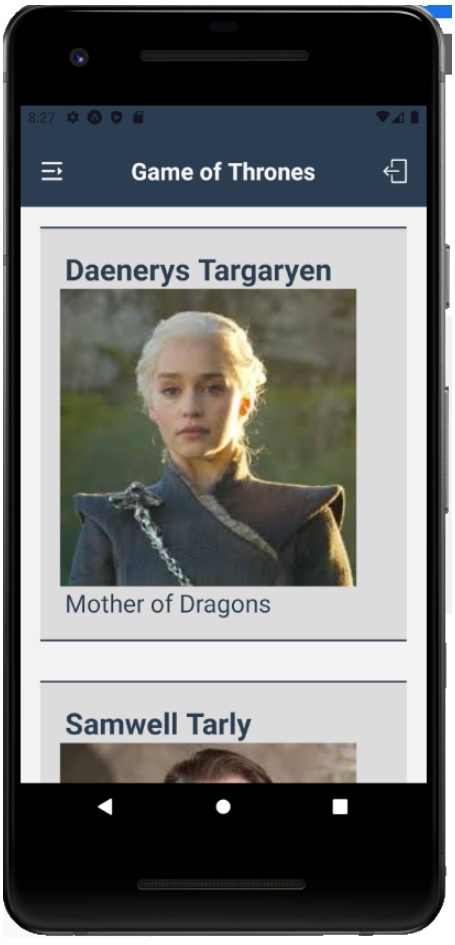
2.2/ Envoyer une requête:

```
import axios from "axios";
import { useEffect, useState } from "react";
import {
  View,
  Text,
  StyleSheet,
  Image,
  TouchableOpacity,
  ScrollView,
} from "react-native";
const GOT_GET_CHARACTERS_URL = "https://thronesapi.com/api/v2/Characters";

export default function GOT(){
  const [listChar, setListChar] = useState([]);
  useEffect(() => {
    axios.get(GOT_GET_CHARACTERS_URL).then((reponse) => {
      setListChar(reponse.data);
    });
  }, []);

  return (
    <ScrollView>
      {listChar.map((char) => {
        return (
          <View
            style={styles.charContainer}
            key={char.id}
          >
            <Text style={styles.title}>
              {char.fullName}
            </Text>
            <Image
              style={{ width: 300, height: 300 }}
              source={{ uri: char.imageUrl }}
            />
            <Text style={styles.container}>
                {char.title}
            </Text>
          </TouchableOpacity>
        );
      })}
    </ScrollView>
  );
};
```

```
const styles = StyleSheet.create({
  charContainer: {
    backgroundColor: "rgb(220,220,220)",
    margin: 20,
    padding: 20,
    borderBottomWidth: 2,
    borderTopWidth: 2,
    borderColor: “#2C3E50”,
    maxWidth: 500,
    alignContent: "center",
  },
title: {
    color: "#2C3E50",
    fontSize: 20,
    fontWeight: “bold”,
  },
content: {
    color: "#2C3E50",
    fontSize: 15,
  },
});
```



- On effectue la requête dans le **useEffect()** pour recevoir les données quand le composant est monté dans le DOM.
- On utilise la fonction **get()** d'axios en fournissant en paramètre l'url de la requête: https://thronesapi.com/api/v2/Characters
- Une fois la réponse obtenue, nous mettons à jour la variable d'état listChar, avec la liste des personnages reçues.
- Enfin, nous affichons tous les personnages en utilisant la fonction **map()**, du tableau.

2.3/ Exercice: News:

Le but de l'exercice est:

D'utiliser l'api newsapi.org, pour afficher les articles du jour, et de pouvoir faire une recherche sur des articles en utilisant un mot clé.

1- Créer le composant: ./Components/Pages/News/News.jsx.

2- Ajouter un écran dans le GlobalDrawer avec le composant News.

3- Dans le composant News.jsx:

3.1- Utiliser useEffect et axios pour chercher les articles top-healines: https://newsapi.org/v2/top-headlines?country=fr&apiKey=VOTRE_KEY

3.2- Mettre la liste des articles dans une variable d'état.

3.3- Afficher les articles.

3.4- Styliser les composants.

4- Dans le composant News.jsx:

4.1- Ajouter un input permettant à l'utilisateur de taper un mot clé.

4.2- Ajouter un bouton pour exécuter une fonction qui:

4.2.1- Fait une requête GET: https://newsapi.org/v2/everything?q=bitcoin&apiKey=VOTRE_KEY

4.2.2- Met à jour les articles dans la variable d'état

```
import { useEffect, useState } from "react";
import {
  View,
  Text,
  StyleSheet,
  Image,
  TouchableOpacity,
} from "react-native";
import axios from "axios";
import InputWithError from "../../UI/InputWithError/InputWithError";
import { AntDesign } from "@expo/vector-icons";
const NEWS_GET_TOP_HEADLINES =
"https://newsapi.org/v2/top-headlines?country=fr&apiKey=xxxxxxxxxxxxx";

const NEWS_GET_QUERY = (q) =>
`https://newsapi.org/v2/everything?language=fr&q=${q}&apiKey=xxxxxxxxxx`;

const News = () => {
  const [listNews, setListNews] = useState([]);

  const [query, setQuery] = useState("");
  const [queryError, setQueryError] = useState("");

  function handleQuery(text) {
    setQuery(text);
    setQueryError("");
  }

  function search() {
    if (query.length > 1) {
      axios.get(NEWS_GET_QUERY(query)).then((reponse) => {
        setListNews(reponse.data.articles);
      });
    } else {
      setQueryError("Veuillez entrer un mot clé...");
    }
  }

  useEffect(() => {
    axios.get(NEWS_GET_TOP_HEADLINES).then((reponse) => {
      setListNews(reponse.data.articles);
    });
  }, []);
```

```
  return (
    <View style={styles.container}>
      <View
        style={{
          display: "flex",
          flexDirection: "row",
          alignItems: "center",
          width: "80%",
          justifyContent: "space-between",
        }}
      >
        <InputWithError
          holder={"Chercher..."}
          valeur={query}
          action={handleQuery}
          errorMessage={queryError}
          type='default'
        />
        <TouchableOpacity onPress={search}>
          <AntDesign name='search1' size={30} color='black' />
        </TouchableOpacity>
      </View>
      {listNews.map((article) => {
        return (
          <View style={styles.articleContainer}>
            <Text>
              {article.title}
            </Text>
            <Text >
              {article.description}
            </Text>
            <Image
              style={{ width: "100%", height: 300 }}
              source={{ uri: article.urlToImage }}
            />
            <Text>
              {article.content}
            </Text>
          </View>
        );
      })}
    </View>
  );
};
```

```
const styles = StyleSheet.create({
  articleContainer: {
    padding: 20,
    backgroundColor: "rgb(220,220,220)",
    marginVertical: 30,
    borderBottomWidth: 2,
    borderTopWidth: 2,
    borderColor: "rgb(220,220,220)",
    maxWidth: 700,
  },

  container: {
    alignItems: "center",
  },
});

export default News;
```