

Exercise19-20 – Biometrics & SecureStore

1. Fortsæt på dit projekt fra exercise11-12 firebase login. Hvis du ikke har færdiggjort øvelserne fra exercise11-12, kan du clone den ned fra github.

Opsætning af projektet

2. BiometricAuthRegistrationScreen.js

- a. Download BiometricAuthRegistrationScreen.js fra Learn og kopier den ind i components/pages mappen i dit projekt
- b. Filen indeholder blot deklareringen af de funktioner du kommer til at lave og en render() med et TextInput, hvor man skal skrive sin adgangskode for at verificere sig selv og en knap til verificeringen. Endvidere indeholder den et Overlay der bliver kaldt, når mobilen venter på brugerens scan af biometric for at øge UX.

3. Home.js

- a. Importer BiometricAuthRegistrationScreen i Home.js
- b. Inde SettingsStack, tilføjer du en ny route til BiometricAuthRegistrationScreen.

i.

```
15 const SettingsStack = createStackNavigator({
16   Settings: { screen: SettingsScreen },
17   Details: { screen: DetailsScreen },
18   BiometricAuthRegistration: { screen: BiometricAuthRegistrationScreen }
19 });
```

4. LoginForm.js

- a. Importer SecureStore og LocalAuthentication fra expo
- i.

```
import {SecureStore, LocalAuthentication} from 'expo';
```
- b. I din **signIn()** funktion gemmer du email variablen i SecureStore før du kalder **firebase.auth().signInWithEmailAndPassword:**

i.

```
async signIn() {  
  const { email, password } = this.state;  
  await SecureStore.setItemAsync('email', email);  
  this.setState({ error: '', loading: true });
```

Registrering af biometrics

5. SettingsScreen.js

- a. I SettingsScreen.js skal vi lave en knap der giver adgang til skærbilledet, hvor man kan registrere sin biometric. Dog findes der telefoner, der hverken har en fingeraftryksscanner eller faceID som hardware *eller* brugere der ikke benytter sig af disse, og derfor slet ikke har noget biometric registreret på telefonen. SettingsScreen skal tjekke om der både er hardwaren og om brugeren har registreret biometrics, før den lader én navigere til BiometricAuthRegistrationScreen
- b. Importer LocalAuthentication fra expo.
- c. Opret SettingsScreens constructor og lav to variabler i dens initial state kaldet hasBiometricsHardware og hasBiometricsRegistered og sæt dem begge til false.
- d. **componentDidMount()**
 - i. Opret componentDidMount() life-cycle funktionen.
 - ii. Den skal kalde to *async* funktioner (slide 9 fra exercise07-08) der tjekker om telefonen har hardwaren og om brugeren har registreret biometrics på telefonen og sætter staten ud fra det:

iii.

```
componentDidMount(){
  this.checkDeviceForHardware();
  this.checkForBiometrics();
}

async checkDeviceForHardware() {
  let hasBiometricsHardware = await LocalAuthentication.hasHardwareAsync();
  this.setState({hasBiometricsHardware})
}

async checkForBiometrics() {
  let hasBiometricsRegistered = await LocalAuthentication.isEnrolledAsync();
  this.setState({hasBiometricsRegistered})
}
```

- e. Lav en ny Button i render() med titlen "Activate Biometrics" og en onPress handler, der kalder funktionen navigateToBiometricsAuthRegistration.

f. **navigateToBiometricsAuthRegistration()**

- i. Funktionen skal både tjekke om hasBiometricsHardware og hasBiometricsRegistered er true. Hvis ikke kan den vise en fejlmeddelelse med alert(), ellers kan den navigere til BiometricAuthRegistration ved hjælp af this.props.navigation.navigate("BiometricAuthRegistration"). (Kommer an på hvad du kaldte routen i opgave 3.b)

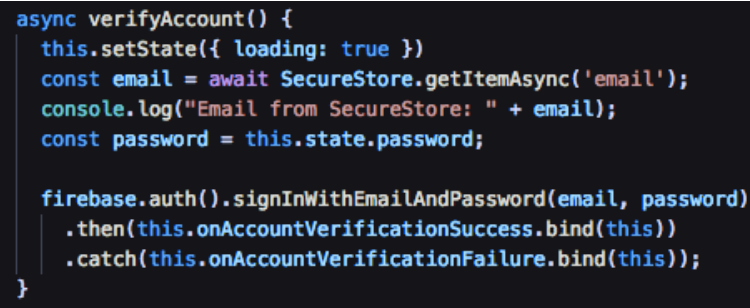
6. BiometricAuthRegistration.js

- a. Importer LocalAuthentication og SecureStore fra expo. Importer også firebase samt Overlay fra react-native-elements. **Note: kræver mindst react-native-elements@1.0.0-beta2**
- b. Flowet i funktionerne foregår således
- verifyAccount()** bliver kaldt fra din Button onPress.
 - verifyAccount()** henter det indskrevne password i staten og henter email fra SecureStore fra den bruger der er logged in (som vi gemte i LoginForm), og authenticator email og password i firebase med signInWithEmailAndPassword.

- iii. **verifyAccount()** kalder så enten **onAccountVerificationSuccess()** eller **onAccountVerificationFailure()** baseret på hvad firebase returnerer.
- iv. **onAccountVerificationSuccess()** tjekker om den **currentUser** der er logged in i firebase er den samme bruger som der lige er returneret fra firebase (man kunne jo blot skrive en anden persons login fx). Er der et match, kalder den **scanBiometrics()**.
- v. **scanBiometrics()** venter på at personen scanner sin biometric – enten fingeraftryk eller faceID på de nye iPhones. Hvis der er et match med det gemte biometric i telefonen, så gemmer vi det indtastede password samt en variabel der blot fortæller, at vedkommende kan logge ind med biometrics i SecureStore. Ved at gøre dette, kan vi ved log ind, når der er match på biometric, få fat i både brugernavn(email) og password, som vi så kan sende til firebase for authentication.

c. **verifyAccount()**

- i. Hent email fra SecureStore ved brug af funktionen **LocalAuthentication.getItemAsync('email')** og password fra state.
- ii. Kald **firebase.auth().signInWithEmailAndPassword(email, password)**

iii. The screenshot shows a JavaScript function named verifyAccount() with the following code:


```

async verifyAccount() {
  this.setState({ loading: true })
  const email = await SecureStore.getItemAsync('email');
  console.log("Email from SecureStore: " + email);
  const password = this.state.password;

  firebase.auth().signInWithEmailAndPassword(email, password)
    .then(this.onAccountVerificationSuccess.bind(this))
    .catch(this.onAccountVerificationFailure.bind(this));
}
  
```

d. **onAccountVerificationSuccess(credentials)**

- i. Få fat i **currentUser** ved at kalde **firebase.auth().currentUser**.
- ii. **credentials** returnerer et user objekt, der indeholder en uid (unique id) variable, hvilket **currentUser** også gør. Lav en if-statement der tjekker om de matcher, som så kalder **scanBiometrics()** og sætter **waitForBiometrics** til true i staten.

```

onAccountVerificationSuccess(credentials) {
  const currentUser = firebase.auth().currentUser;
  if (credentials.user.uid === currentUser.uid) {
    this.setState({ waitForBiometrics: true });
    this.scanBiometrics();
  }
}

```

iii.

e. scanBiometrics()

- i. Opret en variabel result og sæt den lig kaldet LocalAuthentication.authenticateAsync() som en async funktion for at aktivere hardwaren, der venter på brugerens scan af fingeraftryk eller ansigt.
- ii. Lav en if-statement der tjekker om result.success er lig true. Hvis true, skal password og en variabel, der definerer at biometrics er registreret i denne app sættes til true (som en String¹)

```

async scanBiometrics() {
  let result = await LocalAuthentication.authenticateAsync('Password verified. Now, use your biometric.');
```

iii.

f. onAccountVerificationSuccess(credentials)

- i. Blot set loading state til false.

Login med biometrics

7. LoginForm.js

- a. Tilføj en variabel waitForBiometrics til initial state og sæt den lig false.

¹ Der kan kun gemmes Strings i SecureStore

- b. Kopier **renderWaitForBiometrics()** funktionen fra BiometricAuthRegistrationScreen.js, og sæt den i bunden af LoginForm.js. Kald den på samme måde som i BiometricAuthRegistrationScreen i starten af View-componenten i **render()**
- c. Tilføj en Button med titlen "Login with biometrics", og en onPress der kalder en funktion kaldet **"checkBiometricsRegistered()"**
- d. **checkBiometricsRegistered()**
 - i. Sørg for at have funktionen som en async funktion.
 - ii. Lav en variabel kaldet biometricsRegistered der er lig en String "false" (husk det **ikke** skal være en boolean).
 - iii. Sæt den nu lig et await kald fra SecureStore.getItemSync("biometricsRegistered") for at tjekke om brugeren har registreret biometrics til appen og gemt det i SecureStore (fra opgave 6.e.ii)
 - iv. Lav en if statement der tjekker om biometricsRegistered er lig String "true". Hvis true, sæt waitForBiometrics i state til true og kald en funktion kaldet **scanBiometrics()**

```
async checkBiometricsRegistered() {  
  let biometricsRegistered = "false";  
  biometricsRegistered = await SecureStore.getItemAsync("biometricsRegistered");  
  if(biometricsRegistered === "true"){  
    this.setState({waitForBiometrics: true})  
    this.scanBiometrics();  
  } else {  
    alert("No biometrics registered. You cannot login with biometrics");  
  }  
}
```

v.

- e. **scanBiometrics()**
 - i. Byg funktionen op på samme måde som i BiometricsAuthRegistration, hvor du i stedet for **setItemAsync**, bruger **getItemAsync** til at få fat i email og

password gemt i SecureStore. Kald dernæst **signInWithEmailAndPassword()** som du gør ved normalt login.

```
async scanBiometrics() {  
  let result = await LocalAuthentication.authenticateAsync('Verifying biometrics.');
```



```
  if(result.success === true) {  
    let email = await SecureStore.getItemAsync("email");  
    let password = await SecureStore.getItemAsync("password");  
    alert("Biometrics success");  
  
    firebase.auth().signInWithEmailAndPassword(email, password)  
      .then(this.onLoginSuccess.bind(this))  
      .catch(this.onLoginFail.bind(this));  
  } else {  
    alert("biometrics failed");  
    this.setState({waitForBiometrics: false})  
  }  
}
```

ii.