Version: 6.x

Preventing going back

Sometimes you may want to prevent the user from leaving a screen, for example, if there are unsaved changes, you might want to show a confirmation dialog. You can achieve it by using the `beforeRemove` event.

The event listener receives the `action` that triggered it. You can dispatch this action again after confirmation, or check the action object to determine what to do.

Example:

```
function EditText({ navigation }) {
  const [text, setText] = React.useState('');
  const hasUnsavedChanges = Boolean(text);

  React.useEffect(
    () =>
      navigation.addListener('beforeRemove', (e) => {
        if (!hasUnsavedChanges) {
          // If we don't have unsaved changes, then we don't need to do anything
          return;
        }

        // Prevent default behavior of leaving the screen
        e.preventDefault();

        // Prompt the user before leaving the screen
        Alert.alert(
          'Discard changes?',
          'You have unsaved changes. Are you sure to discard them and leave the
screen?',
          [
            { text: "Don't leave", style: 'cancel', onPress: () => {} },
            {
              text: 'Discard',
              style: 'destructive',
              // If the user confirmed, then we dispatch the action we blocked
earlier
              // This will continue the action that had triggered the removal of
the screen
              onPress: () => navigation.dispatch(e.data.action),
            },
          ]
        );
      }),
    [navigation, hasUnsavedChanges]
  );

  return (
    <TextInput
      value={text}
      placeholder="Type something…"
      onChangeText={setText}
    />
  );
}
```

Try this example on Snack

Previously, the way to do this was to:

- Override back button in header
- Disable back swipe gesture
- Override system back button/gesture on Android

However, this approach has many important differences in addition to being less code:

- It's not coupled to any specific buttons, going back from custom buttons will trigger it as well
- It's not coupled to any specific actions, any action that removes the route from state will trigger it
- It works across nested navigators, e.g. if the screen is being removed due to an action in parent navigator
- User can still swipe back in the stack navigator, however, the swipe will be cancelled if the event was prevented
- It's possible to continue the same action that triggered the event

Limitations

There are couple of limitations to be aware of when using the `beforeRemove` event. The event is **only** triggered whenever a screen is being removed due to a navigation state change. For example:

- The user pressed back button on a screen in a stack.
- The user performed a swipe back gesture.
- Some action such as `pop` or `reset` was dispatched which removes the screen from the state.

This event is **not** triggered when a screen is being unfocused but not removed. For example:

- The user pushed a new screen on top of the screen with the listener in a stack.
- The user navigated from one tab/drawer screen to another tab/drawer screen.

The event is also **not** triggered when the user is exiting the screen due to actions not controlled by the navigation state:

- The user closes the app (e.g. by pressing the back button on the home screen, closing the tab in the browser, closing it from app switcher etc.). You can additionally use `hardwareBackPress` event on Android, `beforeunload` event on Web etc. to handle some of these cases.
- A screen gets unmounted due to conditional rendering, or due to a parent component being unmounted.
- A screen gets unmounted due to usage of `unmountOnBlur` options with `@react-navigation/bottom-tabs`, `@react-navigation/drawer` etc.

In addition to the above scenarios, this feature also doesn't work properly with `@react-navigation/native-stack`. To make this work, you need to:

- Disable the swipe gesture for the screen (`gestureEnabled: false`).
- Override the native back button in the header with a custom back button (`headerLeft: (props) => <CustomBackButton {...props} />`).

Edit this page