Rakamin

# React Native Expo SDK and Native Modules

## Utilizing Expo SDK Library and Writing Custom Native Module

Outline Pembelajaran

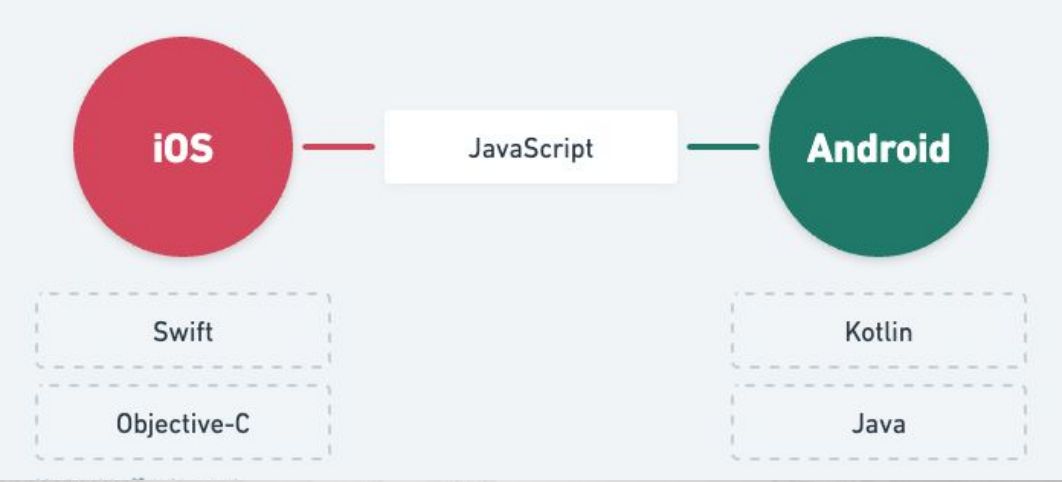
Rakamin Academy

# Objectives

- [ ] Getting Familiar with Expo SDK Library.
- [ ] Understands how to write custom Native Module.

# React Native Architecture

As we know from previous class that React Native consists of two sides: JavaScript and Native.



Rakamin Academy

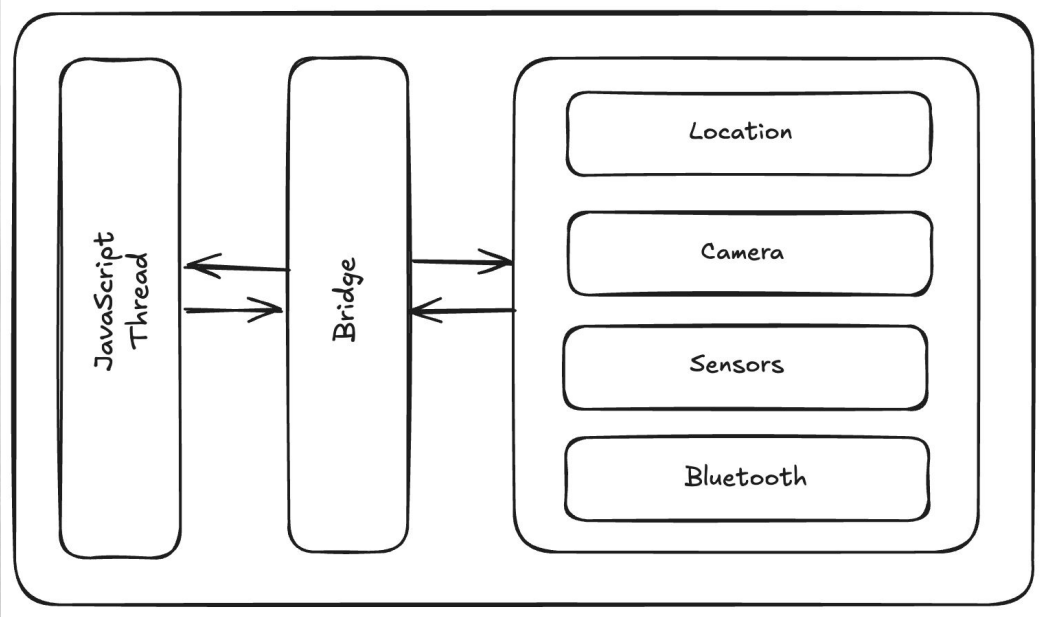
# Native Modules

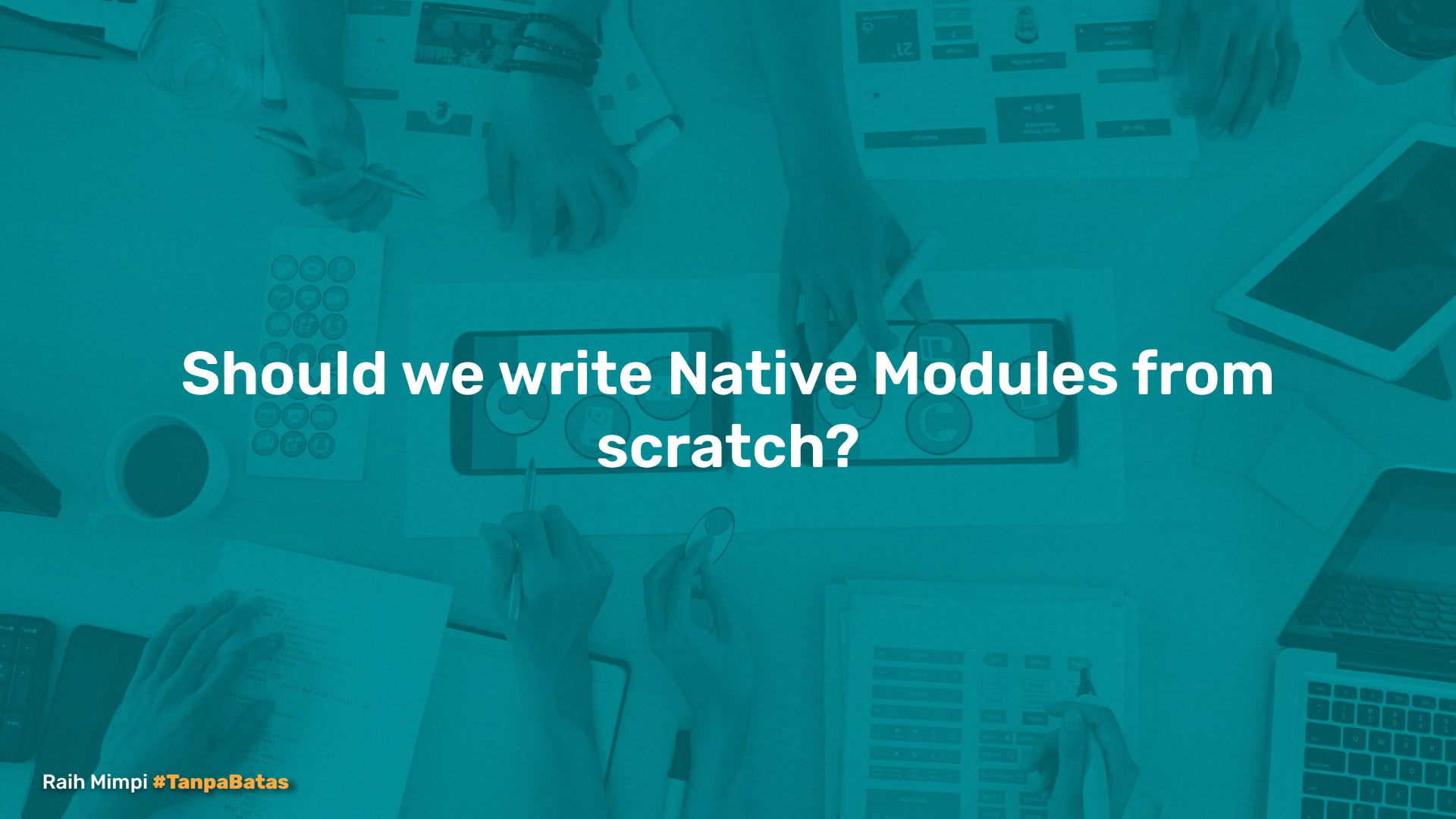
System functionality such as contacts, camera, gyroscope, GPS location, and so on, is on the Native side.


How do we use the Native functionality by using JavaScript?

# React Native Bridge

JavaScript could access native functionalities via “bridge”.

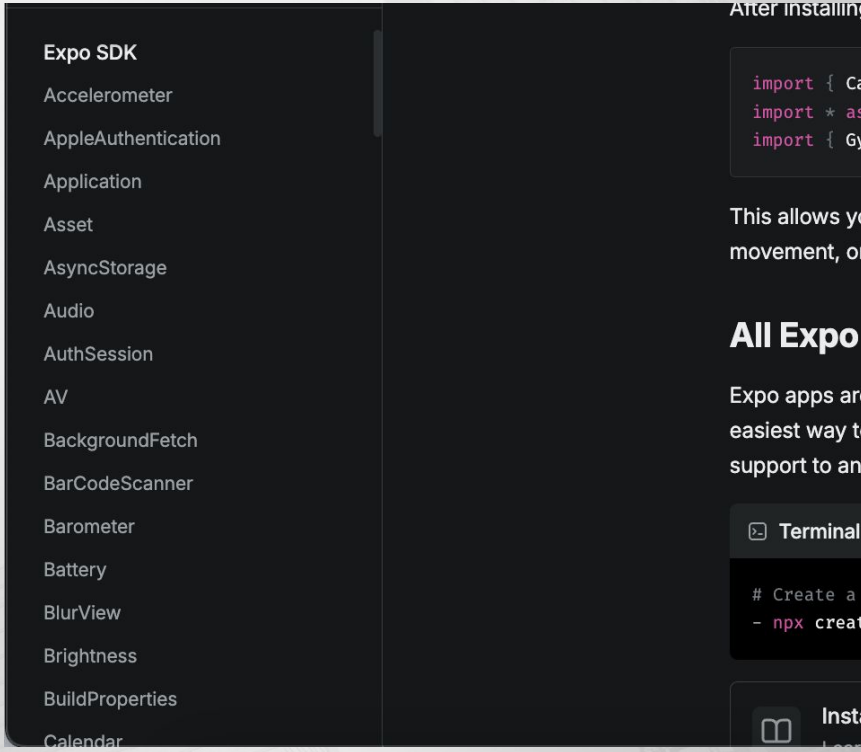



Should we write Native Modules from scratch?


No!

# Expo SDK

The Expo SDK provides access to device and system functionality such as contacts, camera, gyroscope, GPS location, and so on, in the form of packages.



# Expo Camera

expo-camera provides a React component that renders a preview of the device's front or back camera. The camera's parameters such as zoom, torch, and flash mode are adjustable.

```
export default function App() {
  const [facing, setFacing] = useState<CameraType>('back');
  const [permission, requestPermission] = useCameraPermissions();

  if (!permission) {
    // Camera permissions are still loading.
    return <View />;
  }

  if (!permission.granted) {
    // Camera permissions are not granted yet.
    return (
      <View style={styles.container}>
        <Text style={styles.message}>We need your permission to show the
camera</Text>
        <Button onPress={requestPermission} title="grant permission" />
      </View>
    );
  }

  function toggleCameraFacing() {
    setFacing(current => (current === 'back' ? 'front' : 'back'));
  }

  return (
    <View style={styles.container}>
      <CameraView style={styles.camera} facing={facing}>
        <View style={styles.buttonContainer}>
          <TouchableOpacity style={styles.button} onPress={toggleCameraFacing}>
            <Text style={styles.text}>Flip Camera</Text>
          </TouchableOpacity>
        </View>
      </CameraView>
    </View>
  );
}
```

Rakamin Academy

# Expo Contacts

expo-contacts provides access to the device's system contacts, allowing you to get contact information as well as adding, editing, or removing contacts.

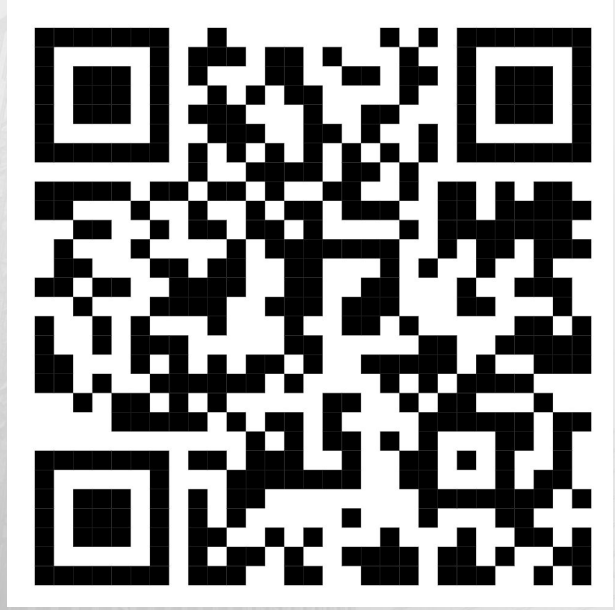
```
export default function App() {
  useEffect(() => {
    (async () => {
      const { status } = await Contacts.requestPermissionsAsync();
      if (status === 'granted') {
        const { data } = await Contacts.getContactsAsync({
          fields: [Contacts.Fields.Emails],
        });

        if (data.length > 0) {
          const contact = data[0];
          console.log(contact);
        }
      }
    })();
  }, []);

  return (
    <View style={styles.container}>
      <Text>Contacts Module Example</Text>
    </View>
  );
}
```

Rakamin Academy

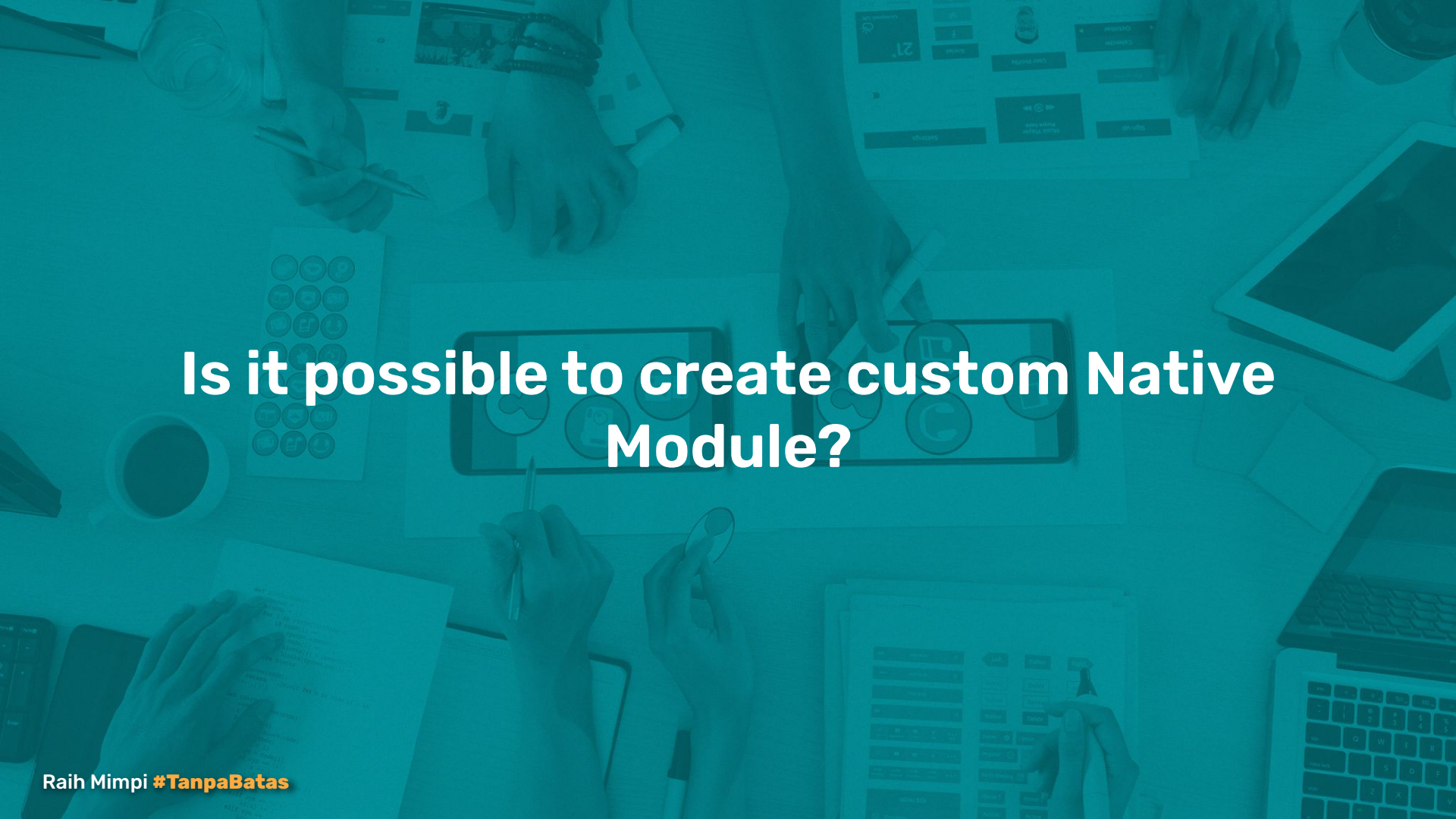
# Expo Image Picker

expo-image-picker provides access to the system's UI for selecting images and videos from the phone's library or taking a photo with the camera.

```tsx
export default function ImagePickerExample() {
  const [image, setImage] = useState<string | null>(null);

  const pickImage = async () => {
    // No permissions request is necessary for launching the image library
    let result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.All,
      allowsEditing: true,
      aspect: [4, 3],
      quality: 1,
    });

    console.log(result);

    if (!result.canceled) {
      setImage(result.assets[0].uri);
    }
  };

  return (
    <View style={styles.container}>
      <Button title="Pick an image from camera roll" onPress={pickImage} />
      {image && <Image source={{ uri: image }} style={styles.image} />}
    </View>
  );
}
```



https://docs.expo.dev/versions/latest/



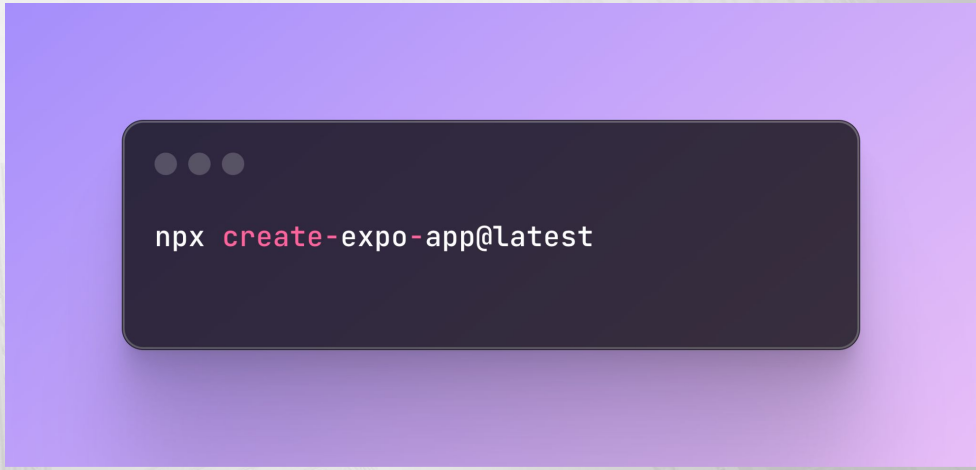
# Is it possible to create custom Native Module?


Raih Mimpi #TanpaBatas

Rakamin Academy

# Simple Custom Native Module

First, create a new expo project.



# Simple Custom Native Module

Next, change to the project directory, and initialize new module.

Rakamin Academy

# Simple Custom Native Module

Clean up unused files inside the native module directory.

```
cd expo-settings
rm ios/ExpoSettingsView.swift
rm android/src/main/java/expo/modules/settings/ExpoSettingsView.kt
rm src/ExpoSettingsView.tsx src/ExpoSettings.types.ts
rm src/ExpoSettingsView.web.tsx src/ExpoSettingsModule.web.ts
```

Rakamin Academy

# Simple Custom Native Module

Write, iOS Native Module.

ios/ExpoSettingsModule.swift

```swift
import ExpoModulesCore

public class ExpoSettingsModule: Module {
  public func definition() -> ModuleDefinition {
    Name("ExpoSettings")

    Function("getTheme") { () -> String in
      "system"
    }
  }
}
```

Rakamin Academy

# Simple Custom Native Module

Write, Android Native Module.

android/src/main/java/expo/modules/settings/ExpoSettingsModule.kt

```kotlin
package expo.modules.settings

import expo.modules.kotlin.modules.Module
import expo.modules.kotlin.modules.ModuleDefinition

class ExpoSettingsModule : Module() {
  override fun definition() = ModuleDefinition {
    Name("ExpoSettings")

    Function("getTheme") {
      return@Function "system"
    }
  }
}
```

Rakamin Academy

# Simple Custom Native Module

Access Native method from JavaScript.

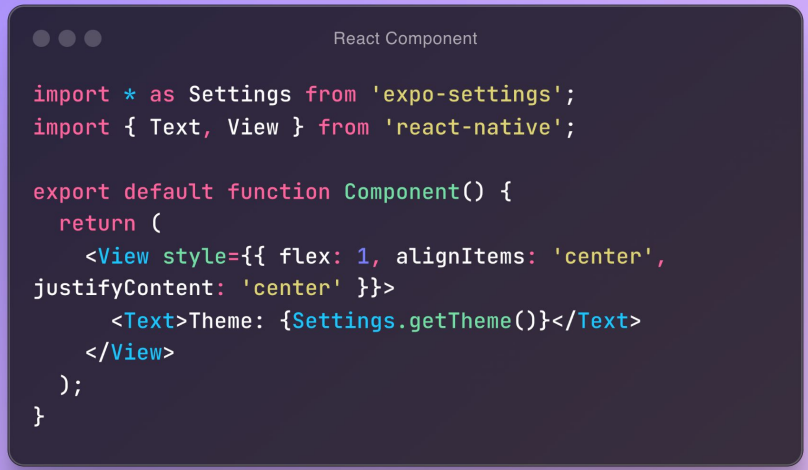
src/index.ts

```ts
import ExpoSettingsModule from './ExpoSettingsModule';

export function getTheme(): string {
  return ExpoSettingsModule.getTheme();
}
```

Rakamin Academy

# Simple Custom Native Module

Use the Native Module in your React Component.



Rakamin Academy

https://docs.expo.dev/modules/native-module-tutorial/


Rakamin
Academy
Thank you
Raih Mimpi #TanpaBatas