microsoft /
fluentui-react-native

Code 59 Issues 14 Pulls Actions 1 Projects 3 Wiki

A react-native component library that implements the Fluent Design System.

developer.microsoft.com/fluentui

MIT license

Code of conduct

Security policy

1k stars 145 forks 48 watching Activity

Public repository

main

Branches Tags

dependabot[bot] Bump react-native-windows from 0.71.35 t... ... ✓ 16 hours ago 3,903

View code

FluentUI React Native

npm package 0.37.24 Azure Pipelines succeeded Azure Pipelines never built

FluentUI React Native is a javascript component library that provides developers with controls that are part of the Fluent Design System. These controls are built on React Native and fully customizable.

FluentUI React Native is still in the alpha stages of development for both the components and the repo. We encourage anyone who is interested in getting an early glimpse of our plans to download and use our components, but please note that you may hit bumps along the way. Please leave us feedback or file issues if you run into bumps, and we will continue to improve the quality of the repo.

Development status on each platform:

Windows	macOS	iOS	Android
Alpha (in progress)	Alpha (in progress)	Alpha (in progress)	Alpha (in progress)

Getting Started

If you have an existing React Native project, it's easy to begin using FluentUI React Native. If you need to setup a new React Native project, please see the React Native Windows Getting Started documentation.

README.md

Prerequisites

- Standard React Native dependencies
- Node.js
- Setting up your React Native Development Environment

Create New React Native project (if needed)

1. Follow the instructions on the React Native Windows Getting Started documentation to create a React Native project.
2. Navigate to the root folder of your project, and use npm to install the package:

```
npm i @fluentui/react-native
```

3. After successful installation, you can test the package by importing components at the top of your app's entry file, e.g. `App.js` :

```
import { Checkbox } from '@fluentui/react-native';
```

4. After importing the @fluentui/react-native package, you can use components such as `Text` and `Checkbox` in your JSX.

```
// In App.js in a new project
import React from 'react';
import { View, Text } from 'react-native';
import { Checkbox } from '@fluentui/react-native';
function HelloWorldApp() {
  return (
    <View
      style={{
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center',
      }}
    >
      <Text>Hello, world!</Text>
      <Checkbox label="Hello World Checkbox" />
    </View>
  );
}
export default HelloWorldApp;
```

If you run into an error that says `pragma and pragmaFrag cannot be set when runtime is automatic` , you can try switching to classic runtime.

Documentation

Components and Controls

Our component documentation can be found in SPEC.md files for each component. The current list can be found in our Wiki's sidebar. They will be uploaded to a website at a future time.

Expanding Component documentation

Our SPEC.md files should reflect the current state of controls that have established the *v1* set of properties on any one platform.

Since the FluentUI React Native controls are cross-platform, but represented by a single page, it's important to distinguish platform differences and limitations. Examples include:

- If the component is not available on all supported platforms.
- If the component has properties not available on all supported platforms.

- If the component has limited support for a given property on any supported platforms.
- If the component has distinguishable behavior on a supported platform that must be minded while used.

Theming framework

Documentation for Theming can be found in our docs file, or for more in depth documentation, along side the implementation.

- Theming Overview
- StyleSheets
- Customizing Theme Settings
- Theme Reference
- Tokens
- Slots
- Customize and Compose

Developing in the repo

Yarn + Lage

This repo is set up as a monorepo using Yarn workspaces. To install yarn, please follow instructions in the Yarn documentation.

For running tasks the repo has switched to using Lage for task running. The primary tasks that can be executed at the root are:

- `yarn build` - does the typescript build for all packages in the repository
- `yarn test` - will build, lint, and run any applicable tests on all packages in the repo
- `yarn bundle` - will bundle all packages in the repo
- `yarn buildci` - will build, lint, run tests, and bundle everything in the repo

Note that Lage uses caching to avoid redundant steps and has very minimal output. To avoid caching add `--no-cache` as a command line argument. Similarly adding `--verbose` will give more detailed output.

Setup your development environment

To start developing in the repository you can:

1. `git clone https://github.com/microsoft/fluentui-react-native.git`
2. `cd fluentui-react-native`
3. `yarn`
4. `yarn build`

After a successful yarn build, you can explore FluentUI Tester, our demo application to play with each of the controls. To run FluentUI Tester, please follow instructions in the FluentUI Tester readme.

Note: If your repo is located on either your Desktop or Documents folder, you may encounter the error: "Watchman error... Operation not permitted". Clone it in a different directory to avoid Watchman permission issues.

Prettier

This repo is set up to run Prettier. To run Prettier in fix mode on the repo, run `yarn prettier-fix` at the root of the Repo.

If you are using VSCode as your editor, you can configure it to run Prettier on save. Prettier is a recommended extension for the repo. You can configure it to run by:

1. Installing the Prettier extension for VSCode
2. Going to Settings > Text Editor > Formatting > Check Format On Save

Contributing

Please visit our contribution guide for more information on contributing to this repo.

This project has adopted the Microsoft Open Source Code of Conduct. For more information see the Code of Conduct FAQ or contact opencode@microsoft.com with any additional questions or comments.

Releases

13,749 tags

Used by 31

+ 23

Contributors 60

● TypeScript 92.3% ● JavaScript 3.7% ● Objective-C 1.4% ● Swift 0.9% ● C++ 0.8% ● Shell 0.4%
● Other 0.5%

Report repository