

Configuration types

List of all @brinkninja/widgets configuration types [Suggest Edits](#)

```
types.ts ** Theme ** interface BrinkWidgetThemeConfig {theme ? : {mode ? : 'light' | 'dark' colors ? : {primary ? :BrinkWidgetThemePaletteOptions // Primary color secondary ? :BrinkWidgetThemePaletteOptions // Secondary color
neutral ? :BrinkWidgetThemePaletteOptions // Neutral color informative ? :BrinkWidgetThemePaletteOptions // Informative
color error ? :BrinkWidgetThemePaletteOptions // Error color success ? :BrinkWidgetThemePaletteOptions // Success color
warning ? :BrinkWidgetThemePaletteOptions // Warning color background ? : {paper ? :string base100 ? :string base200 ? :string base300 ? :string },// Widget & components base color text ? : {primary ? :string ,secondary ? :string ,disabled ? :string }// Text colors },shape ? : {roundedBtn ? :string | number roundedBox ? :string | number roundedFormField ? :string | number }// Adding shape to common components config ? : {syncFieldColorWithText ? :boolean // If set to true, form field label color will be same as main color }slot ? : {tabsInContainer ? :boolean // If set to true, widget tabs will render inside form container } } }interface BrinkWidgetThemePaletteOptions { '100' :string , '200' :string , '300' :string , '400' :string ,main :string , '500' :string , '600' :string , '700' :string } ** App config ** interface BrinkWidgetsProviderAppConfig extends BrinkWidgetThemeConfig {integratorId ? :string appName :string // Your appName apiKey :string // apiKey which is provided by Brink team apiKeyName :string // apiKeyName which is provided by Brink team rpcUrl ? :string // rpcUrl used by wagmi provider under the hood axiosBaseUrl ? :string // Setting the base API url walletConnectProjectId ? :string // Wallet connect project id which will be used by wagmi provider infiniteApproval ? :boolean // If set to true, when approving token it will always ask for max approval enableReactQueryDevtools ? :boolean // If set to true, widget will display RQ dev tools for the RQ client used in widget tabCacheKey ? :string // If provided, when clicking on tabs in widget, selected tab will be stored in localStorage, so it can be selected again on page reload events ? :WidgetEvents // If you want to do some analytics tracking or something else on specific form event }export interface WidgetEvents {onNextButtonClick ? : ()=> Promise < void | void // Event emitted if user proceeds to Sign step onSignAndSubmit ? : ()=> Promise < void | void // Event emitted after user successfully Signed and Submitted Intent } ** Providers config ** // You can disable providers used by the widget if you want to provide them from your application export type BrinkWidgetsProviderSubProviderConfig = WagmiProviderConfigPropTypes & MuiProviderConfigPropTypes & FormikProviderConfigPropTypes & RainbowKitProviderConfigPropTypes & QueryClientProviderConfigPropTypes export interface MuiThemeProviderProps < Theme = DefaultTheme {children ? :React .ReactNode theme :Partial < Theme | ((outerTheme :Theme )=> Theme ) }export type WagmiProviderConfig = ReturnType < typeof createConfig export type MuiProviderConfig = MuiThemeProviderProps export type FormikProviderConfig = FormikConfig < FormikValues export type RainbowKitProviderConfig = RainbowKitProviderProps export type QueryClientProviderConfig = QueryClient export type WagmiProviderConfigPropTypes = | {disableWagmiProvider ? :false | undefined | never wagmiProviderConfig ? :ReturnType < typeof createConfig } | {disableWagmiProvider ? :true wagmiProviderConfig ? :never }export type MuiProviderConfigPropTypes = | {disableMuiProvider ? :false | undefined | never muiProviderConfig ? :MuiThemeProviderProps } | {disableMuiProvider ? :true muiProviderConfig ? :never }export type FormikProviderConfigPropTypes = | {disableFormikProvider ? :false | undefined | never formikProviderConfig ? :FormikConfig < FormikValues } | {disableFormikProvider ? :true formikProviderConfig ? :never }export type RainbowKitProviderConfigPropTypes = | {disableRainbowkitProvider ? :false | undefined | never rainbowKitProviderConfig ? :RainbowKitProviderProps } | {disableRainbowkitProvider ? :true rainbowKitProviderConfig ? :never }export type QueryClientProviderConfigPropTypes = | {disableQueryClientProvider ? :false | undefined | never queryClientProviderConfig ? : {client :QueryClient } } | {disableQueryClientProvider ? :true queryClientProviderConfig ? :never }export type BrinkWidgetsProviderConfig = BrinkWidgetsProviderAppConfig & BrinkWidgetsProviderSubProviderConfig Updated26 days ago
```

[Theme Customization](#) [Provider customization](#) Did this page help you?Yes No