Packaging

Get oriented - TELUS Brand

The Universal monorepo offers five packages for TELUS teams. All packages are distributed publicly on NPM with the `@telus-uds/` prefix.

```
— packages
    ├── palette-allium      # TELUS design token palette
    ├── theme-allium        # TELUS theme for UDS Base Components
    ├── ds-allium           # TELUS component library (Multi-
platform + Web only)
    ├── themes-community.theme-allium # TELUS theme for community
components & additional core variants.
    └── components-community # Community component library
```

`palette-allium`

This package exports a `json` file containing design tokens that represents the smallest, most atomic design options that make up the TELUS brand.

`theme-allium`

This package exports a json file containing a list of objects, each representing a given component. These objects contain the design decisions around each UDS component (split between `appearances`, `rules` and `tokens`) that dictate how the tokens from the palette are applied and consequently how the component should look.

`ds-allium`

This package contains the core components that are built specifically for the TELUS brand. It uses the palette to ensure that styling decisions in the component library adhere to TELUS' brand standards. This package also imports and re-exports all UDS Base Components so that it can act as a single point of entry for all developers building with the TELUS brand.

⚠ CAUTION

This package contains both base and web components and will be renamed to `web-components` in the future. Teams consuming it will have to update the import on their applications.

`themes-community.theme-allium`

This package is similar to `theme-allium` but it is intended for theming community components & adding new variants for existing core components. This package exports a json file containing a list of objects, each representing either a community component or a base component. These objects contain the design decisions and dictate how the tokens from the palette are applied and consequently how a community component should look. It can also be used to extend new appearances for core components. Teams consuming this package will have to use the `mergeThemes` utility from `@telus-uds/system-constants` to merge it run-time over the relevant main theme, which in this case is `theme-allium`. This can be done by importing `theme-allium` and `themes-community.theme-allium` on the frontend, then pass the merged theme returned from `mergeThemes(alliumTheme, alliumCommunityTheme)` to the `ThemeProvider`.

`components-community`

This package is similar to `ds-allium` but have individually versioned component packages that are maintained exclusively by the community. Teams wanting to build a allium specific web or multi-platform compnonent can create a component in this space and consume it by importing it individually on their applications.

Get oriented - Koodo Brand

The Universal monorepo offers four packages for Koodo teams. All packages are distributed publicly on NPM with the `@telus-uds/` prefix.

```
- packages
    ├── palette-koodo       # Koodo design token palette
    ├── theme-koodo         # Koodo theme for UDS Base Components
    ├── themes-community.theme-allium # Koodo theme for community
```

```
components & additional core variants.
    └── components-community # Community component library
```

`palette-koodo`

This package exports a `json` file containing design tokens that represents the smallest, most atomic design options that make up the Koodo brand.

`theme-Koodo`

This package exports a json file containing a list of objects, each representing a given component. These objects contain the design decisions around each UDS component (split between `appearances`, `rules` and `tokens`) that dictate how the tokens from the palette are applied and consequently how the component should look.

`themes-community.theme-koodo`

This package is similar to `theme-koodo` but it is intended for theming community components & adding new variants for existing core components. This package exports a json file containing a list of objects, each representing either a community component or a base component. These objects contain the design decisions and dictate how the tokens from the palette are applied and consequently how a community component should look. It can also be used to extend new appearances for core components. Teams consuming this package will have to use the `mergeThemes` utility from `@telus-uds/system-constants` to merge it run-time over the relevant main theme, which in this case is `theme-koodo`. This can be done by importing `theme-koodo` and `themes-community.theme-koodo` on the frontend, then pass the merged theme returned from `mergeThemes(koodoTheme, koodoCommunityTheme)` to the `ThemeProvider`.

`components-community`

This package has individually versioned component packages that are contributed exclusively by the community. Teams wanting to build a koodo specific web or multi-platform compnonent can create a component in this space and consume it by importing it individually on their applications.

Get oriented - Public Mobile Brand

The Universal monorepo offers four packages for Public Mobile teams. All packages are distributed publicly on NPM with the `@telus-uds/` prefix.

```
— packages
    ├── palette-public-mobile      # Koodo design token palette
    ├── theme-public-mobile        # Koodo theme for UDS Base
Components
    ├── themes-community.theme-public-mobile # Public mobile
theme for community components & additional core variants.
    └── components-community       # Community component library
```

`palette-public-mobile`

This package exports a `json` file containing design tokens that represents the smallest, most atomic design options that make up the Public Mobile brand.

`theme-public-mobile`

This package exports a json file containing a list of objects, each representing a given component. These objects contain the design decisions around each UDS component (split between `appearances`, `rules` and `tokens`) that dictate how the tokens from the palette are applied and consequently how the component should look.

`themes-community.theme-public-mobile`

This package is similar to `theme-public-mobile` but it is intended for theming community components & adding new variants for existing core components. This package exports a json file containing a list of objects, each representing either a community component or a base component. These objects contain the design decisions and dictate how the tokens from the palette are applied and consequently how a community component should look. It can also be used to extend new appearances for core components. Teams consuming this package will have to use the `mergeThemes` utility from `@telus-uds/system-constants` to merge it run-time over the relevant main theme, which in this case is `theme-public-mobile`. This can be done by importing `theme-public-mobile` and `themes-community.theme-public-mobile` on the frontend, then pass the merged theme returned from `mergeThemes(pmTheme, pmCommunityTheme)` to the `ThemeProvider`.

`components-community`

This package has individually versioned component packages that are maintained exclusively by the community. Teams wanting to build a public mobile specific web or multi-platform compnonent can create a component in this space and consume it by importing it individually on their applications.

Edit this page