Develop a community theme

Themes are an abstract layer that sits between the brand palettes and the components, making it possible to customize the look and feel of components freely in a variety of different contexts. Learn more about brand themes. Currently, only UDS Base components use themes, but in principle, any new component in the UDS ecosystem should be leveraging them to style components. Universal Design System encourages the community to create themes in the community space. This guide covers everything you need to know to create a new theme or extend an existing theme.

We have three core themes that you could extend and create a new theme from: `@telus-uds/theme-allium` is the main theme for TELUS, `@telus-uds/theme-koodo` is for Koodo and `@telus-uds/theme-public-mobile` is for Public Mobile.

> INFO
>
> - If this is the first time you are contributing to the Universal Design System repository
>   - You will need Github access
>   - You may need to request write access for you and your team
> - If you are contributing to the UDS project, please follow the instructions to develop for UDS

Using core theme vs extending themes vs creating new themes

If you are using `styled-components`, you will not use any theme, the styles will be defined on the component itself.

Using core themes

You should use a `core theme` if you are working with `core components` and want to be aligned with one of the core brands in the system (TELUS, Koodo or Public Mobile). The core themes contain all the appearance rules for the *core*

components. Please refer to the brand orientation section of the guide for specific information.

Extending themes

> INFO
>
> Theme extension should be used with discretion, please assess your use case with the UDS core team before creating a theme to avoid unnecessary themes and have the contribution blocked downstream.

Extending themes should be used for:

1. Defining appearances for new community components.
2. Creating new variants for existing components.
3. Overriding styles of existing variants and states of core components.

Creating a new community theme

> INFO
>
> Creating new themes should be used with discretion, please assess your use case with the UDS core team before creating a theme to avoid unnecessary themes and have the contribution blocked downstream.

If your team is going to work with a new brand, or if your product will not share the options from one of the existing brands, you should consider creating a new theme. New themes are created from scratch, your team will have to refer to an existing palette or create a new one (if you need to have control over the values) and your team will have to define each and every value on it.

Scaffold your theme

We provide easy scripts to generate the basic scaffolding of a theme. From the root directory of the monorepo, run the following script. You'll be prompted with a few questions to scaffold a package that best suits your needs. Make sure to select a core theme if you choose to extend and provide a core / community palette the theme would reference.

```
npm run create-theme
```

Each theme created or extended will live in the folder `packages/themes/community`. The script will create the necessary files to get you started on a folder that concatenates the pattern palette and the name provide (e.g. If the name informed is UDS, the new folder will be called theme-UDS) to keep the consistency across the system.

The script will create the necessary files to get you started. Now you are all set to start adding entries to the `theme.json`. If you chose to extend an existing theme, the build process would create a merged `build/theme.js`. So by changing the `theme.json` you get to create new or override existing base theme rules, default appearances and component variants. The `system-tokens` package implements the tooling that validates, builds, and merges these themes.

See it live

You can check out the scaffolded theme and see your changes live by running Storybook. For the theme to be picked up by Storybook:

- Export the newly created theme in `themes-community/index.js`.
- Update the list of paths to the component stories in `storybook/.storybook/paths.json` and `.storybook/paths.json`.

Run the following script and make sure to select the platform and brand you'll be working on. If you have selected the Community Theme option for the brand, you'll also be prompted to type in the community theme you want to apply to the components:

```
npm run storybook
```

Edit this page