



Treasury Board of Canada  
Secrétariat

Secrétariat du Conseil du Trésor  
du Canada

Canada

# Government of Canada Design System

## Documentation plan

Sep 18, 2019

# Documentation plan objectives

- Deliver a minimum viable design system for the next iteration of WET and GC web guidance
- Plan MVP feature/component development
- Establish documentation layouts to support ease of use
- Establish navigation to make documentation easy to find

# Documenting page layouts (templates), patterns and components

# Objective

**Objective:** Determine how best to document components, styles and design patterns to make them easy to understand and use

- Review approaches other design systems have used
- Consider how to allow for experimentation and quicker improvements to live services
- Express thoughts on the design system as a stand-alone site or part of Canada.ca, its navigation system and labels
- Review template examples for components and suggest possible user issues or gaps that could prevent easily consolidating various current documentation systems
- Discuss ways the community might be able to engage and gaps in facilitating engagement
- Plan timelines towards a live production of the Design System documentation

# Expected audiences/top tasks

## Developers

### Publishers:

- What's available
- HTML to copy
- Using beta design

### Contributors:

- Submitting new code
- New documentation
- Experimentation criteria
- Adding variants (ie react)

### Maintainers:

- Updates
- Breaking changes

## Designers

### Communicators:

- What's available
- Constraints + what is mandatory
- When to use
- Writing style
- How to display choices or data meaningfully
- Theming (apps, intranets, etc)

### Researchers/designers:

- Known limitations
- Research and testing
- More on styles

## 3rd parties

### Collaborators:

- Opportunities to reuse research
- Designs and code
- Contacting both GC theme and specific services

### External WET community:

- Using or contributing as a non-GC theme partner
- Theming
- Adding variants

# Documenting components and patterns

## Principles

- Ensure example and code is easily findable
- Include semantic versioning and change log
- Provide design for code variants
- Provide design for component variants
- Provide beta variants where available, with necessary CSS and JS overrides
- Ensure it's clear how to:
  - Report issues or gaps (and see overview of backlog)
  - Volunteer to experiment on a new pattern
  - Submit a beta with code
  - Move from beta to PR process for stable

# Documenting variants

**Who are your providers?**

Select all that apply.

You must choose at least one provider.

☐ Aardv  
☐ Bumb  
☐ Caterp

HTML

SilverStripe / Common Web Platform (CWP)

Mustache / Handlebars

React

React (JavaScript)

React (TypeScript)

React with Styled Components (JavaScript)

React with Styled Components (TypeScript)

Vue

Vue (JavaScript)

Vue (TypeScript)

Code ^

Template format: HTML

Copy code

<!--  
Remember to add these styles:  
in CSS: FieldsetBlockWithError.css, H1.css, CheckboxBlock.css  
OR in Sass (SCSS): FieldsetBlockWithError.scss, H1.scss, CheckboxBlock.scss  
-->

Choose a React  
vs Vue variant -  
NZ model

Design variants of a  
component all in same  
example & code block -  
US model

**If pregnant or considering pregnancy - both partners**

Avoid travel to Costa Rica if you are pregnant or planning to conceive a child in the next 3 months.

Code

HTML CSS

```
<section class="alert alert-danger">
  <h3>If pregnant or considering pregnancy - both partners</h3>
  <p>Avoid travel to Costa Rica if you are pregnant or planning to conceive a child in the next 3 months.</p>
</section>
```

Provide beta  
code with  
override CSS/JS

!

**Success status**

Lorem ipsum dolor sit amet, [consectetur adipiscing](#) elit, sed do eiusmod.

!

**Warning status**

Lorem ipsum dolor sit amet, [consectetur adipiscing](#) elit, sed do eiusmod.

!

**Error status**

Lorem ipsum dolor sit amet, [consectetur adipiscing](#) elit, sed do eiusmod.

i

**Informative status**

Lorem ipsum dolor sit amet, [consectetur adipiscing](#) elit, sed do eiusmod.

Component code

—

```
1 <div class="usa-alert usa-alert--success" >
2   <div class="usa-alert__body">
3     <h3 class="usa-alert__heading">Success status</h3>
```

# Documenting templates

Provide link to the generic page layout

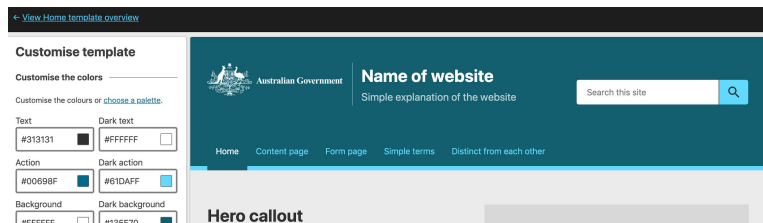


Beta institutional landing page

Template with guidance

► Code

Update will move the guidance toggle into a sticky banner with back to guidance link - AUS model



Toggle between embedded guidance and clean template

[Institution name]

Short description of the institution's mandate

Guidance in expand collapses for page layouts

Super task link [optional]

▼ **Guidance:** breadcrumb, title, image and super task link ⓘ

Breadcrumb

**Mandatory**

The breadcrumb is: [Home](#) > [Departments and agencies](#).

Title

# Documenting variants/configurations without examples

## Image

### Optional

The 1200x726 space behind the H1 can be used for a banner or image. The image does not have to take up the whole space and can be customised as needed.

The image and H1 should be visually distinct from the [Canada.ca homepage](#) to prevent confusion between institutional landing pages and Canada.ca.

**Variation:** Apply the `ip-cover-hide-mobile` class in addition to `ip-cover-img` to hide the image in mobile view.

In page layouts, variants on a component can be mentioned within guidance blocks

Option	Description	How to configure
Disabled state	Disabled inputs will not be clickable. Visual indications include greying out the control and adding a 'stop' cursor icon on hover.	To toggle a disabled state: <ul style="list-style-type: none"><li>• Add the class <code>is-disabled</code> to the list item of the checkbox or radio item</li><li>• Add <code>disabled</code> attribute to the input field</li></ul>
Inline state	Inline radio buttons or checkboxes will display side by side on larger screens and wrap to new lines, rather than vertically.	Add the <code>form-label-inline</code> class to each checkbox or radio button label.
Success checkboxes (green)	Success checkboxes use a green background within the checkbox when it is selected to support user confidence that they are ready or successful. Recommended application is for checklists.	Add the <code>checkbox-green</code> class to each checkbox list item.

In components, variants can be documented in tables if less frequently used, or in a set of similar variations (but option+description must be plain language)

# Documenting versions and changes

▼ Previous version

v4.0.28.1 to v4.0.27	• Clearfix removed
<del>testCase-</del> v4.0.27 (2.0)	• Doormat component explicit
	• More services and information
v4.0.26 to v4.0.22	• Separated column layout for
<del>testCase-</del> v4.0.22 (1.2)	

Current WET shows version impact of component update on overall WET4 build

## Here's the proposal:

- Single version per component that would increment with changes to the pieces (API version - would not mention WET version on component page)
  - These would then have version notes + a label saying what the update affects.
- Specs on the visual rendering will only be made visible or changed in alpha design patterns so do not need to be in the design system proper
  - These changes then get reflected in the other categories
- A progressive disclosure pattern would be used to get to technical markup of the schema' or template near the bottom of the page.

### API (Version 3.0)

CSS Class	Template	Visual rendering	Schema
Version 1.3	Version 3.0	Version 1.1	Version 1.1

### CSS Class (v1.3)

<b>gc-srvinfo</b>	Component: Service and information
<b>gc-drm</b>	Component: (Deprecated) Doormat link. Defined inside a Service and information compo

### ▼ Version notes

Version 1.3	• Added: <b>gc-srvinfo</b>
Version 1.2 (between v4.0.28.1 and v4.0.27)	• Deprecated: <b>gc-drm</b>

Current WET gives breakdown of component into 4 categories each with versioning

# Prioritising development of the WET5 core and documentation

# Prioritising components and patterns

## Principles

- Must include core designs to support adoption of Canada.ca
- Must include patterns from recent projects showing top features supporting top task success
  - contact us patterns, decision points, interactive questions/forms, sign in and authentication
- Must include a pilot collaboration to fork another design system's pattern

## Next steps

- DTO will present a components list based on the WET, recent and spec
- Determine what we fork from gov.uk or other design system
- Documentation and WET5 dev

# Component priorities - first cut

- Generic page template semantic markup
  - Grids and layout, JSON-LD support for schema.org, revised outline markup
- Components for decision points and interactive questions
  - Expand/collapse, radio/checkboxes, toggle, responsive tables, tabbed interface, auto-complete/polyfill, buttons, overlays, checkbox display logic, postal code finder
- Components for nav options
  - Sticky left hand 'on this page' nav, circles nav for SIT, Canada.ca menu, long document pattern
- Support for consistent authentication experience
  - Sign in pattern, customisable form validation messages, contextualising content based on region and URL parameters
- Support for responsive tables
- Nicer data visualisation - charts and graphs

# Design system nav and getting started/collaboration guidance

# Design system comparison (see annex A)

	Name	Landing	Stand alone	Top nav?	Code
<b>UK</b>	Descriptive	Doormats	Yes	Yes	Included w/ex
<b>Australia</b>	Descriptive	Doormats	Yes	Yes	Included w/ex - templates are separate page examples and code only in github - component examples are in a second column right of the title, always top of page
<b>US</b>	Descriptive	Doormats	Yes	Yes	Included w/ex
<b>New Zealand</b>	Descriptive	Doormats	Yes	Yes	Included w/ex
<b>Material Design</b>	Branded	Doormats	Yes	Yes	Only style and examples
<b>Clone</b>	Branded	No doormat	Yes	No	Included first
<b>Aurora</b>	Branded	Role based	Yes	Yes	Included w/ex
<b>Shopify</b>	Branded	Doormats	Yes	Yes	Included w/ex

# Branding and navigation principles

- Unified design and component documentation
  - How do we reconcile frequent publishing with technical/updating guidance? Updating as smaller use case at bottom of page?
- Distinct design system branding
  - Use a broad, official, descriptive name (e.g. GC Design System)
  - Build as a “stand-alone” site with top nav (tabs) and left nav conventions
- Support for non-Canada.ca implementations (intranet, authenticated spaces and sign in, non-GC users)

# What we need to support use and collab

- Documented get started process to use the system
  - CDS and code variant distributions
- Experiment requirements and support:
  - Easy get started experimenting, how to do it well
  - Modularised component implementation and changes
  - Clear exit requirements to get to beta and integrate into WET5
- Tracking component development
  - Issues tracking in github repo
  - Backlog: currently working with a documentation oriented project board, might need to add more component dev phases: [GCDS backlog](#)
  - Documenting easy to follow contribution guidelines
  - Documenting easy how to meet standards guidelines

# 90-day plan

# 90-day plan - Documentation

Activity	Deadline
Present 90 plan	Oct 31
Present documentation page layout and example	Nov 14
MVP patterns in new library (and translated)	Dec 15
Design system in production (at least limited distribution) with contribution guidance	Jan 15
Host in Drupal/Github integrated environment	Mar 2020
Update to Design System branding and nav	Mar 2020
Elaborated Getting started and Experimentation guidance and support	Mar 2020

# Annex A: Comparative analysis of other design systems

# Other design systems

- [GOV.UK Design System](#)
- [Australian Government Design System](#)
- [United States Web Design System](#)
- [New Zealand Government Design System](#)
- [Google Material Design](#)
- [Clone](#)
- [Aurora](#)
- [Polaris \(Shopify\)](#)

# Landing pages with doormats

## Design your service using GOV.UK styles, components and patterns

Use this design system to make your service consistent with GOV.UK. Learn from the research and experience of other service teams and avoid repeating work that's already been done.

[Get started >](#)

### Styles

Make your service look like GOV.UK with guidelines for applying layout, typography, colour and images.

[Browse styles](#)

### Components

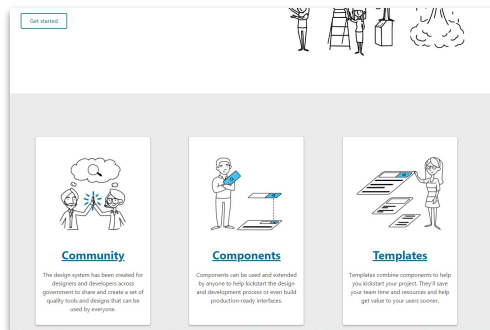
Save time with reusable, accessible components for forms, navigation, panels, tables and more.

[Browse components](#)

### Patterns

Help users complete common tasks like entering names and addresses, filling in forms and creating accounts.

[Browse patterns](#)



### Components

Browse all USWDS components and get UX, accessibility, and implementation guidance.

[Browse the components](#)



### Design tokens

Learn how to get started using design tokens, the building blocks of USWDS component design.

[View design tokens](#)



### Utilities

Adapt your designs and deliver prototypes quickly and consistently, without touching a line of CSS.

[Build with utilities](#)



### Page templates

Use our basic page templates as a starting point for your design and development process.

[Start from a template](#)



### CONTENT

Learn how to use language to design a more thoughtful product experience.

[Read content guidelines](#)



### DESIGN

Find out how we approach the visual elements of our interface with purpose.

[Read design guidelines](#)



### COMPONENTS

Use components as building blocks as you develop new products and features.

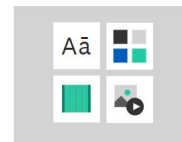
[Browse components](#)



### PATTERNS

See how we put the pieces together to create meaningful product experiences.

[Browse patterns](#)



### Basics

Guidance for applying basic elements, like colour and typography. We're working on layout, iconography, images, and media now.

[Browse some basics](#)



### Components

Reusable multiple components for navigation, panels, forms, tables, and more. We're working on these now.

[Browse some components](#)



### Patterns (coming soon)

Help users by providing easy flows to complete specific tasks, like setting up accounts, filling in forms, and checking eligibility. We'll be inviting you to co-design some patterns with us soon.

## Design guidance and code

Use our most popular design and development resources to jumpstart your latest project



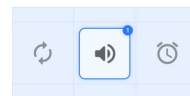
### Material Design guidelines

Material Design principles, styles, and best practices



### Components

Design guidance and developer documentation for interactive UI building blocks



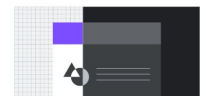
### Icons

Access five sets of stylized system icons, available in a range of formats and sizes



### Material Components for the web

Implement and customize Material web apps with our code and documentation



### Material dark theme

Create a dark theme version of your Material UI

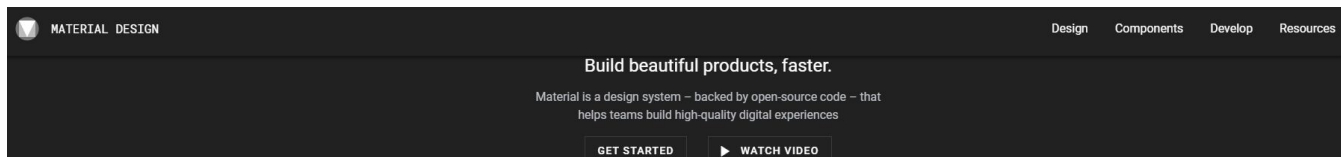
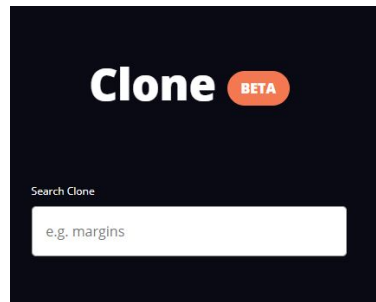
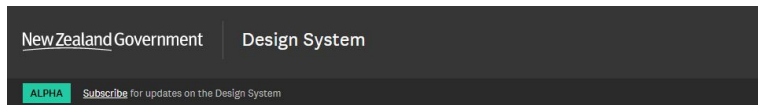


### Color Tool

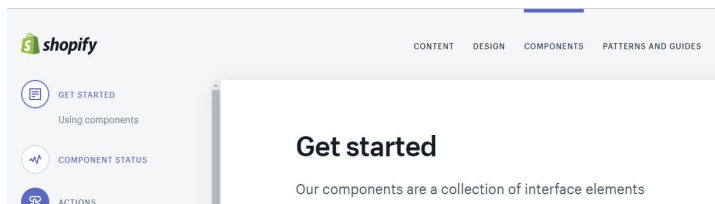
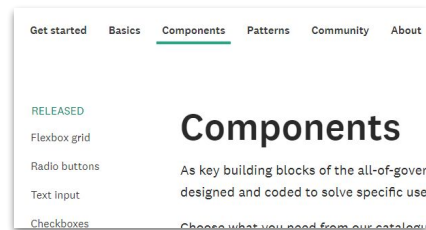
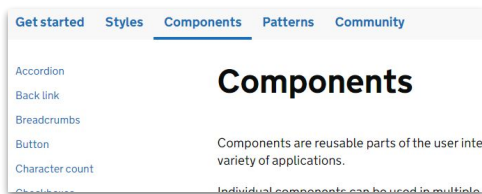
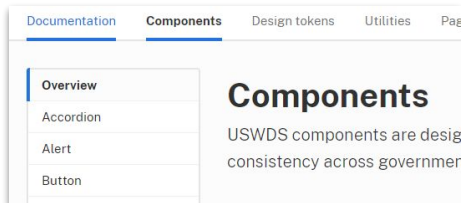
Create and share color palettes for your UI, and measure the accessibility of color combinations

# Stand-alone sites, with additional branding

All government design systems are stand-alone sites, not connected to the overall site navigation, with their own search.



# Using top and left menus



# NZ specific component - how to present code

## Who are your providers?

Select all that apply.

**You must choose at least one provider.**

☐ Aardvark  
☐ Bumblebee  
☐ Caterpillar

HTML

SilverStripe / Common Web Platform (CWP)

Mustache / Handlebars

**React**

React (JavaScript)

React (TypeScript)

React with Styled Components (JavaScript)

React with Styled Components (TypeScript)

**Vue**

Vue (JavaScript)

Vue (TypeScript)

[Code](#) ^

Template format: 

HTML

Copy code

```
<!--  
Remember to add these styles:  
in CSS: FieldsetBlockWithError.css, H1.css, CheckboxBlock.css  
OR in Sass (SCSS): FieldsetBlockWithError.scss, H1.scss, CheckboxBlock.scss  
-->  
...
```