



# Test Frontend

Instructions

# Mise en contexte

Le but premier de ce test est d'évaluer:

- le respect des requis
- la reproduction des designs
- le souci du détail
- la propreté du code

Nous pensons que, grâce à ton expérience, tu pourras compléter cette composante en environ 5h.

N'hésites pas à nous écrire en cours de route si tu as des questions.

## Objectifs

À partir du design fourni et des spécifications ci-dessous, tu devras assurer le développement d'une composante VueJS qui se retrouverait normalement dans notre design system et qui pourrait être utilisé dans diverses applications Centiva.

Cette composante a réellement été développée dans notre design system et les requis techniques + les maquettes sont les vraies exigences qui ont été transmises aux développeurs.

# Requis techniques

## Informations générales

Nom de la composante: Modal

Design: [Maquette descriptive et responsive de la composante «Modal»](#)

## Spécifications techniques

- La Modal doit être paramétrable via des props et slots de manière à permettre :
  - De gérer le contenu de la Modal à partir de son parent (via un *slot*)
  - D'afficher un X dans le coin en haut à droite pour permettre de fermer la Modal
  - De la fermer ou non lors d'un clic à l'extérieur de la Modal
  - D'inclure un footer personnalisé pour remplacer celui contenant les boutons «Annuler» et «Confirmer» présents par défaut (ex. : le footer où l'on voit la checkbox «Ne plus afficher» dans le design)
  - De donner un état à la Modal (info, success, alert et danger) afin :
    - D'afficher un trait de couleur en haut
    - De changer la couleur du bouton de confirmation présent par défaut
    - D'afficher une icône à gauche du titre de la Modal
      - Infos : [info-circle](#)
      - Success : [check-circle](#)
      - Alert : [exclamation-triangle](#)
      - Danger : [exclamation-circle](#)
- La Modal doit émettre trois événements à son parent :
  - «Open» lorsqu'elle s'ouvre

- «Close» lorsqu'elle se ferme - via le X, le clic à l'extérieur ou le bouton «Annuler»
- «Confirm» lorsque le bouton de confirmation présent par défaut dans le footer est cliqué
- Le déclencheur d'ouverture de la Modal doit être à l'extérieur de la composante de la Modal de manière à ce qu'elle soit complètement indépendante de son environnement.
- La largeur de la Modal dépend de son contenu, mais est limitée à 580 pixels ou à 90% de la largeur de la fenêtre dans le cas où la taille de l'écran est inférieur à 640 pixels.

## Props

Voici, à titre informatif, une liste de *props* à créer. Tu es libre d'en ajouter ou de les modifier si cela te semble pertinent.

active	<ul style="list-style-type: none"> <li>● Boolean</li> <li>● Facultatif</li> <li>● Default : false</li> </ul>
title	<ul style="list-style-type: none"> <li>● String</li> <li>● Requis</li> </ul>
closeOnOutsideClick	<ul style="list-style-type: none"> <li>● Boolean</li> <li>● Facultatif</li> <li>● Default : true</li> <li>● Est-ce que la Modal se ferme lors d'un clique à l'extérieur?</li> </ul>
state	<ul style="list-style-type: none"> <li>● String</li> <li>● Facultatif</li> <li>● Default : «info»</li> </ul>

## Slots

Voici, à titre informatif, une liste de slots à créer. Tu es libre d'en ajouter ou de les modifier si cela te semble pertinent.

content	Représente le contenu de la Modal. Il est obligatoire; s'il n'est pas fourni, nous n'affichons pas la Modal.
footer	Représente le footer personnalisé. Il est facultatif; s'il n'est pas fourni, nous affichons le footer par défaut (boutons «Annuler» et «Confirmer»)

## Liste des couleurs

Bleu (info) : correspond à la variable `$color-accent1-normal`

Vert (success) : correspond à la variable `$color-accent6-normal`

Jaune (alert) : correspond à la variable `$color-accent8-normal`

Rouge (danger) : correspond à la variable `$color-accent9-normal`

## SCSS

Notre CSS est écrit à l'aide du préprocesseur SCSS de manière à bénéficier de ses nombreux avantages : utilisation de variables, nesting, mixins, etc.

Nous écrivons le code SCSS dans un fichier externe à la composante. Le fichier peut être scoped de manière à garder les mêmes avantages d'encapsulation que nous aurions si le code était écrit directement dans la composante.

Un fichier SCSS nommé « `variables.scss` » contenant les variables normalement disponibles dans le design system est fourni.

## Notes

Les boutons et icônes sont normalement d'autres composantes de notre design system. Dans le cadre de ce test, elles peuvent être créées directement dans la Modal.

L'environnement comportant la Modal pour l'évaluation du test est à ton choix, ex. :

- Plusieurs boutons ouvrant chacun une Modal avec des options différentes
- Un seul bouton ouvrant une Modal et nous modifions manuellement les props
- Un formulaire qui permet de modifier les props de la Modal et un bouton pour l'ouvrir
- Etc.