SHOP.APP.MODULES

ANGULAR | MATERIAL | NGRX | RXJS

– K –

Sommaire

1. Liste
2. Description

01
Liste

Liste

Module

- **@action.** Standardise le nom des actions Ngrx et la façon de les créer
- **@alert.** Gère les alertes pour l'utilisateur avec différents niveaux de criticité
- **@dialog.** Permet de d'ouvrir une dialog (=popup ou modal)
- **@enum.** Centralise la récupération des énumérations métiers
- **@form.** Gère les formulaires (état, validation, persistance…) (Basé sur ngrx-forms)
- **@loader.** Gère l'état de chargement de l'application et l'affichage des indicateurs correspondants
- **@material.** Fournit des composants graphiques (Bouton, tableau…) (Basé sur angular.material)
- **@router.**
- **@timer.**
- **@token.**

02

Description

@alert

Module

Fonctionnalité

- **Thread.** Zone d'écran ou s'affiche des alertes
  - **Nb thread affichable :** 1
  - **Nb Alert / Thread :** 1
- **Dismissal.**
  - Manuelle : OUI -> Appui icone « X »
  - Auto : OUI -> Après 'chgt de route' | 'X sec'
  - Style (Instant / Fade) : Fade
- **Component.**
  - Style : Couleur selon Criticité
  - Criticité : « success, info, warn, error »
  - Contenu : message

Implémentation

- Material snackbar
- Couleur selon criticité : Bootstrap

Action

- `triggerAlertAction`
- `dismissAlertAction`
- `keptAfterRouteChangeAction`

@dialog & @enum

Module

@dialog

Fonctionnalité

- Permet de

Action

- `openDialogAction`
- `closeDialogAction`

Implémentation

- Injecter un component dynamiquement dans « mat-dialog »

@enum

Fonctionnalité

- Gère des énumérations dynamiques

Action

- `loadAllSuccessAction`
- `loadAllFailureAction`

Implémentation

- 1 Api call

@form | Présentation

Module

Concept

- **Form.**
  - Est un **composant configurable**
  - Peut contenir [1..*] **Field**
  - **A un état** résultant de l'état de ses fields
  - Est **persistable** & **validable**
  - Emet une action à la soumission
- **Field.**
  - Est un **composant configurable**
  - **Type :** Input, Select, Date, Checkbox…
  - **Validation :**
    - Statique & Dynamique
    - Configurable
    - Message d'erreur **-> injecter module ?**
  - **Persistance :** désactivable (ex : Password)

Composant

- **Form.** <k-form>
- **Field.** <k-form-field-***<type>***>
- **GroupField.** <k-form-group-field-***<name>***> (password)
- **Button.** <k-button>

Selector

- `selectForm(formId)`
- `selectFormValue(formId)`
- `selectControl(formId, ctrlId)`
- `selectControlValue(formId, ctrlId)`

Layout

- 1 grid | 1 column
- Field
  - Min-width : **240px**
  - Width : **100%**

Action

- `resetFormAction`
- `clearFormValueAction`
- `formValidatedAction`
- `submitFormAction`
- `validateFormAction…`

Validation

- Fonction « ngrx-forms »
- HomeMade
  - mustMatch(ctrl, ctrlRef)

@form | Utilisation

Module

1. Module

- Importez le module « FormModule »
- Créez un **component** (x.component.ts + .html)
- Créez un **effect** (x.effect.ts)

2. Formulaire

- Placez un `<k-form>` et spécifiez :
  - **(obligatoire)** un identifiant unique **[formId]**
  - la persistance dans le global state (`no-persist`)
  - la validation des champs (`no-validate`)
  - `Ex : <k-form [formId]="'my_form'" no-persist no-validate></k-form>`

3. Layout

- Par défaut, un `<k-form>` est une grid d'une colonne
- Vous pouvez redéfinir le layout du formulaire en ajoutant un et en utilisant 'angular-flex-layout'

4. Contenu

- **Champ(s).** Placez [1..*] `<k-form-field-XXX>` & spécifiez :

Identification

- **(obligatoire)** un nom unique ds le formulaire **[ctrlName]**
- un [formId] (si != du <k-form>)

Persistance

- La persistante dans le global state : [unpersist]="true"

Usage Info

- Label
- Placeholder
- Value

Validation Rules

- Format (password | email | number)
- Required
- [1..*] Fonction de Validation

- **Bouton(s).** Placez [1..*] <k-button> de soumission

5. Effet (x.effect.ts)

- Ecoutez une action (Ex : `formValidatedAction`)
- Filtrez sur l'**identifiant du formulaire** (`filter()`)
- Récupérez les **valeurs du formulaires** dans l'**action** ou via un **selector**

@form | Implémentation

Module

Dependancy Module

- LoaderModule (Pour BasicButton)
- LuxonModule (Pour DateField)
- MaterialModule (mat-form-field, mat-button…)
- NgrxFormsModule (FormGroupState, …)

Technique

- **Angular.**
  - Component, Dependancy Injection, Directive
  - Content Projection, Component Interaction
- **Typescript.** Classe/Héritage
- **Ngrx-forms.** UI Ex | Actions | Code | Code Ex
  - * actions existantes ds « ngrx-forms » ont été redéfinies dans le module @form (la maj de l'état en résultant aussi via un reducer), Ceci parce que les actions 'ngrx-forms' sont implémentées avec la méthode de Ngrx V<8.x (Version actuelle |) alors que les actions de l'app sont basés sur la version post 8.x
  - /!\ ngrxEnableFocusTracking non utilisé

Contenu

- **Composant**
  - Model. FormComponent & FieldComponent
  - Button.
  - Field.
  - Field-group
- **Service**
  - **error-message.** Gère les messages des erreurs
  - **validation-fns.** Gère les fonctions de validation
- **Store**

@loader

Module

Fonctionnalité

- **Activation / Désactivation**
  - **Activation :** à l'envoi d'une requête HTTP
  - **Désactivation :**
- **Identification**

Implémentation

- **Component**
- **Store**

Utilisation

- **Quand ?**
  - Now : dès qu'il y a un appel HTTP (via interceptor http)
- **Ou ?**
  - Dans les Submit Bouton des « @Form »

@material

Module

Bouton & Lien

Icon / Font

- **Icon :** liste **/** user guide

Form

Table

- **Contenu** datasource
- **Colonne et lignes**
- **Pagination**
- **Tri**
  - S'appliquer au contenu affiché, (Si * Page ?)
  - Est sélectionnable pour une ou *colonne)
- **Filtre**
- **Sélection**

Autre

- **SideNav, Toolbar, Accordéon…**

@timer & @token

Module

@timer

Fonctionnalité

- Déclaration d'un Timer
- Emet l'action fourni à la fin du Timer

Action

- defineTimerAction
- deleteTimerAction
- timerDefinedAction
- timerDeletedAction
- timerEndedAction

@token

Fonctionnalité

- Déclaration d'un Token
- Appel l'Api de validation des Token

Action

- validateTokenAction
- deleteTokenAction
- tokenValidatedAction
- tokenInvalidatedAction

@Search Todo

Fonctionnalité

- **Critère de Recherche**
- **Résultat de Recherche**
  - **Pagination**

Implémentation

- **Composant « Critères de Recherche »**
  -
  -
- **Composant**
  - **filter-shell :** enveloppe standard des filtres
  - **filter-select-tree-view :** (Instance) - arbre de checkbox
  - **filter-search-tag-box :** (Instance) - Présente un champ libre avec 'autocomplétion'

Merci !