

- Etre de type (*string*)
- <Entité>Id

VERSION ACTUELLE

```
ENUM
+ @action/emitter-type (api, component, page, store)
+ @action/module (action, alert, enum...)
+ @alert/alert-type (success, error, warning, info)
+ @button/type (submit, button)
+ @form/field-format (email, password, number)
+ @material/theme (deeppurple-amber, ...)
+ @token/token-status (validating, valid, invalid)
+ @account/role (admin, user)
```

Token
+ tokenId [PK]
+ status ( <i>Enum</i> )
+ value ( <i>string</i> )

Enum
+ name ( <i>string</i> ) [PK] + value(s) ( <i>string[]</i> )

ShoppingList
+ shoppingListId [PK]
+ active ( <i>bool</i> )
+ name ( <i>string</i> )
+ description ( <i>string</i> )
+ account [FK]
+ catUsedProduct(s) [FK]

Account
+ accountId <b>[PK]</b>
+ email ( <i>string</i> )
+ pwd ( <i>string</i> )
+ jwtToken ( <i>string</i> )
+ userName ( <i>string</i> )
+ role ( <b>Enum</b> )

UsedProduct
+ usedProductId [PK]
+ category ( <i>string</i> )
+ subCategory ( <i>string</i> )
+ name ( <i>string</i> )
+ bought ( <i>bool</i> )
+ quantity ( <i>double</i> )
+ note ( <i>string</i> )

```
classDiagram
    class CatUsedProduct {
        +category string
        +subCatProducts FK
    }
```

SubCatUsedProduct
+ subCategory ( <i>string</i> )
+ <b>products</b> [FK]

Form
+ formId ( <i>string</i> )
+ unpersist ( <i>bool</i> )
+ validate ( <i>bool</i> )

Field
+ fieldId ( <i>string</i> ) <b>[Calculated]</b>
+ type (input, date...)
+ value ( <i>string</i>   <i>bool</i>   <i>number</i> )
+ label ( <i>string</i> )
+ placeholder ( <i>string</i> )
+ format ( <b><i>Enum</i></b> )
+ unpersist ( <i>bool</i> )

FormValue
+ [controlName: string] :

<Field/Form>State
+ id ( <i>string</i> )
+ enabled/disabled ( <i>bool</i> )
+ valid/invalid ( <i>bool</i> )
+ dirty/pristine ( <i>bool</i> )
+ touched/untouched ( <i>bool</i> )
+ focus/unfocus ( <i>bool</i> )

Validation
<ul style="list-style-type: none"> <li>+ id (<i>String</i>)</li> <li>+ statique / dynamique</li> <li>+ fonction (required, min, max...)</li> <li>+ synchrone / asynchrone</li> </ul>

Button
<ul style="list-style-type: none"> <li>+ <b>buttonId</b> (<i>string</i>)</li> <li>+ <b>type</b> (<b>Enum</b>)</li> <li>+ <b>icon</b> (<i>string</i>)</li> <li>+ <b>color</b> (<b>Enum</b>)</li> <li>+ <b>action</b> (<i>TypedAction</i>)</li> </ul>

BasicButton
+ label ( <i>string</i> ) + customClass ( <i>string</i> )

Timer
<ul style="list-style-type: none"> <li>+ timerId (string)</li> <li>+ time (number)</li> <li>+ timeoutHandler (NodeJS.Timeout)</li> <li>+ action (TypedAction)</li> </ul>

Alert
+ alertId ( <i>string</i> )
+ type ( <i>AlertTypeEnum</i> )
+ message ( <i>string</i> )
+ autoClose ( <i>bool</i> )
+ keepAfterRouteChange ( <i>bool</i> )
+ fade ( <i>bool</i> )

NEXT VERSION

The diagram shows a yellow box labeled "ENUM" at the top. Below it, a list of attributes is shown, each preceded by a plus sign (+). The attributes are: @action/emitter-type (api, component, page, store), @action/module (action, alert, enum...), @alert/alert-type (success, error, warning, info), @button/type (submit, button), @button/geometry (round, rectangle...), @form/field-format (email, passworld, number), @material/theme (deeppurple-amber, ...), @token/token-status (validating, valid, invalid), and @account/role (admin, user).

```
graph TD; ENUM[ENUM]; ENUM --- A["@action/emitter-type (api, component, page, store)"]; ENUM --- B["@action/module (action, alert, enum...)"]; ENUM --- C["@alert/alert-type (success, error, warning, info)"]; ENUM --- D["@button/type (submit, button)"]; ENUM --- E["@button/geometry (round, rectangle...)"]; ENUM --- F["@form/field-format (email, passworld, number)"]; ENUM --- G["@material/theme (deeppurple-amber, ...)"]; ENUM --- H["@token/token-status (validating, valid, invalid)"]; ENUM --- I["@account/role (admin, user)"];
```

- + @action/emitter-type (api, component, page, store)
- + @action/module (action, alert, enum...)
- + @alert/alert-type (success, error, warning, info)
- + @button/type (submit, button)
- + @button/geometry (round, rectangle...)
- + @form/field-format (email, passworld, number)
- + @material/theme (deeppurple-amber, ...)
- + @token/token-status (validating, valid, invalid)
- + @account/role (admin, user)

ShoppingList
+ shoppingListId [PK]
+ active ( <i>bool</i> )
+ name ( <i>string</i> )
+ description ( <i>string</i> )
+ account [FK]
+ item(s) [FK]

Item
+ itemId [PK]
+ name ( <i>string</i> )
+ bought ( <i>bool</i> )
+ quantity ( <i>double</i> )
+ note ( <i>string</i> )
+ TreeNode [FK]

```

classDiagram
    class Product {
        +productId PK
        +name string
        +description string
        +dimension
        +weight
    }

```

ButtonState
+ <b>buttonId</b> ( <i>string</i> )
+ value (bool)
+ disable/enable

Button
<ul style="list-style-type: none"> <li>+ <b>buttonId</b> (<i>string</i>)</li> <li>+ <b>UIDesign:color</b> (<i><b>Enum</b></i>)</li> <li>+ <b>UIDesign:raised</b> (<i>bool</i>)</li> <li>+ <b>UIDesign:geometry</b> (<i><b>Enum</b></i>)</li> <li>+ <b>Content:icon</b> (<i><b>Enum</b></i>)</li> <li>+ <b>Content:label</b></li> <li>+ <b>Effect:customAction</b> (<i>TypedAction</i>)</li> </ul>

Button
<ul style="list-style-type: none"> <li>+ <b>buttonId</b> (<i>string</i>)</li> <li>+ <b>type</b> (<b>Enum</b>)</li> <li>+ <b>icon</b> (<i>string</i>)</li> <li>+ <b>color</b> (<b>Enum</b>)</li> <li>+ <b>action</b> (<i>TypedAction</i>)</li> </ul>

BasicButton
+ label ( <i>string</i> ) + customClass ( <i>string</i> )

Timer
<ul style="list-style-type: none"> <li>+ timerId (string)</li> <li>+ time (number)</li> <li>+ timeoutHandler (NodeJS.Timeout)</li> <li>+ action (TypedAction)</li> </ul>

Alert
+ alertId ( <i>string</i> )
+ type ( <i>AlertTypeEnum</i> )
+ message ( <i>string</i> )
+ autoClose ( <i>bool</i> )
+ keepAfterRouteChange ( <i>bool</i> )
+ fade ( <i>bool</i> )