

Règles sur les classes

Version : 04-2023

Nommage des classes de donnée (modèle)

Classe correspondant à une table des BDDs

Élément	Règle	Exemple
Préfixe	TSQL + nom de la table	TSQLPersonne
Type	Classe représentant une table BDD	TSQLFamille
Nom du fichier	record + nom de la classe	recordTSQLPersonne
Emplacement	ngx-logeasweb-ui\src\lib\models\BDD\NomBDD\	—

—

Classe interne au front

Règle	Description	Exemple
Préfixe	Lettre T suivie d'un nom parlant	TPersonnePlus
Extension	Utiliser l'héritage plutôt qu'une copie	extends TSQLPersonne
Propriété calculée	Ajouter une méthode de calcul	getNomComplet()
Usage unique	Utiliser une interface	Interface locale

Exemple :

```
class TPersonnePlus extends TSQLPersonne {
    nomDeJeuneFille: string;
}
```

Exemple propriété calculée :

```
getNomComplet() {
    return `${this.titre} ${this.prenom} ${this.nom}`;
}
```

—

Classes dérivées pour affichage

Type	Règle	Exemple
Nature	Interface uniquement (pas de logique)	—
Nommage	Nom de la table + Grid	IPersonneGrid
Utilisation	Type mis à plat pour une grille	DataGrid

Les classes issues des BDDs (TSQL...)

Quelques règles génériques

Règle	Description	
Chargement	Toujours via les fonctions de la classe	Comportement homogène
Couplage	Une classe ne doit pas appeler une autre classe métier	Utiliser un service
Logique transverse	Centralisée dans un service	Vérifications croisées

Exemple interdit :

```
TSQLFamille appelle TSQLPersonne  
TSQLPersonne appelle TSQLFamille
```

⇒ À gérer dans un **service**

Fonctions normées correspondant au CRUD

READ

Nom de la fonction :

```
chargeFromBDD$
```

Signature :

```
static async chargeFromBDD$(  
  filtre: TSQLPersonne = new TSQLPersonne(),  
  base?: TBaseLoGeAs  
) : Promise<TSQLPersonne[]>
```

Paramètre	Type	Description
filtre	Objet modèle	Permet de sélectionner un sous-ensemble
base	TBaseLoGeAs	Base alternative optionnelle

Exemple :

```
this.personnes = await TSQLPersonne.chargeFromBDD$();
```

Utilisation du filtre

```
const filtre = new TSQLPersonne({
  Codepostal: "31800"
});

this.personnes = await TSQLPersonne.chargeFromBDD$(filtre);
```

Résultat attendu :

* uniquement les personnes ayant le code postal 31800

—

CREATE / UPDATE

Nom de la fonction :

```
sauveToBDD$
```

Signature :

```
async sauveToBDD$(base?: TBaseLoGeAs)
```

Cas	Condition	Action
Création	ID = 0	Insertion en base
Mise à jour	ID ≠ 0	Update en base
Préparation	Aucune	Gérée par la méthode

—

DELETE

Nom de la fonction :

```
effaceToBDD$
```

Signature :

```
async effaceToBDD$(
  ID: number,
  base?: TBaseLoGeAs
)
```

Paramètre	Type	Description
ID	number	Identifiant de l'objet
base	TBaseLoGeAs	Base optionnelle

Tableau récapitulatif du CRUD

Opération	Méthode	Type	Description
READ	chargeFromBDD\$	static async	Charge une liste d'objets depuis la base
CREATE	sauveToBDD\$	async	Crée un objet
UPDATE	sauveToBDD\$	async	Met à jour un objet
DELETE	effaceToBDD\$	async	Supprime un objet

From:

<https://wiki-logeas.fr/certif/> - **dokuwiki-certif**

Permanent link:

<https://wiki-logeas.fr/certif/doku.php?id=certif:procedure:usageclasseinterface&rev=1777030804>

Last update: **2026/04/24 13:40**

