Retour au Dossier Organisationnel	
Sujets connexes	Cartographie fonctionnelle de LoGeAs Protection des données dans le développement Règles sur les classes Conventions de nommage Angular Mise en place des dx-validator

Proc#179 - Développement : règles et bonnes pratiques

Informations qualité

Suivi des modifications majeures		
Date	**Auteur**	**Modifications**
1 octobre 2026	Nicolas MARCHAND	Création : la page sommaire devient une procédure ; les pages d'installation, de référentiel et d'astuces deviennent des documents techniques rattachés
2 octobre 2026	Nicolas MARCHAND	Procédure étendue à tous les développements (ancien titre : « Développement Angular : règles et bonnes pratiques ») : règles communes, dont la protection des données (point de vigilance n°24 de l'audit NF 552 de novembre 2025), puis règles propres à Angular

Suivi des approbations	Cartographie fonctionnelle – fiche de la procédure
Objet	Fixer les règles que respecte tout développement de LoGeAs (interfaces Angular, serveurs, client lourd, bibliothèques), afin que le code reste homogène, lisible, maintenable par toute l'équipe et conforme aux exigences de certification.
Destinataires	**- Validation des modifications :** Chef de projet **- Approbation du document :** Équipe Développement

Principe

Tout développement respecte les **règles communes** ci-dessous ; un développement Angular (applications, bibliothèque `ngx-logeasweb-ui`) respecte en plus les **règles Angular**. Le détail et les exemples sont dans les documents techniques ; en cas de doute, ils font foi sur la manière de faire.
Une règle nouvelle ou modifiée est d'abord inscrite dans le document technique concerné, puis, si elle est générale, résumée ici (modification majeure : nouvelle approbation).

Règles communes

Domaine	Règle	Détail
Protection des données	toutes les données personnelles sont traitées comme sensibles ; une donnée personnelle n'est ajoutée que pour une finalité identifiée, déclarée « nominative » dans la cartographie, couverte par les droits des personnes ; aucune donnée personnelle dans les journaux, les traces ni les jeux de test ; aucun secret dans le code	Protection des données dans le développement
Tickets	toute correction ou évolution part d'un ticket qualifié (type, gravité)	Proc#8
Versions et code	travail sur branche personnelle, intégration, tags et dépôt légal selon les règles de gestion des versions	Proc#145, Proc#170
Qualification et livraison	aucune version n'est livrée sans rapport qualité validé portant sur son code exact	Proc#168

Règles Angular

Domaine	Règle	Détail
Nommage	fichiers en kebab-case selon le modèle `nom.type.ts` ; classes en PascalCase avec le suffixe de leur rôle (`…Component`, `…Service`) ; sélecteurs préfixés ; un fichier `.spec.ts` par fichier testé	Conventions de nommage
Classes de données	une table de la base = une classe `TSQL<Table>` ; une classe interne au front est préfixée T et hérite de la classe `TSQL` plutôt que de la copier ; une structure d'affichage est une interface `I<Table>Grid`	Règles sur les classes
Accès au back	on ne charge, n'enregistre et n'efface les données qu'au travers des services (`chargeFromBDD$`, `sauveToBDD$`, `effaceFromBDD$`) ; une classe de données n'appelle jamais un service	Règles sur les classes
Validation des saisies	les contrôles de saisie utilisent les `dx-validator` regroupés par `validationGroup` ; un validateur maison est déclaré dans la bibliothèque (`custom-validation-rule.ts`) et appelé par `customValidatorService`	Mise en place des dx-validator

Documents techniques

Document	Portée	Contenu
Protection des données dans le développement	tous les développements	minimisation, déclaration des données nominatives, droits des personnes, conservation, journaux, données de test, sécurité
Installation	Angular	installation du poste de développement Angular
Conventions de nommage Angular	Angular	fichiers, classes, sélecteurs, tests
Règles sur les classes	Angular	classes de données et fonctions d'accès au back
Mise en place des dx-validator	Angular	validateurs de saisie, organisation des contrôles
Graphe de dépendances	Angular	astuce : tracer les dépendances et repérer les boucles (dependency-cruiser, Graphviz)

Responsabilités

Qui	Quoi
Chef de projet	valide les règles et leurs modifications

Qui	Quoi
Équipe Développement	applique les règles, tient les documents techniques à jour

From:
https://wiki-logeas.fr/certif/ - **dokuwiki-certif**

Permanent link:
https://wiki-logeas.fr/certif/doku.php?id=certif:procedure:develop:angular

Last update: **2026/10/02 07:10**