

# R4.10 Compléments Web

BTP TD6

## Consignes :



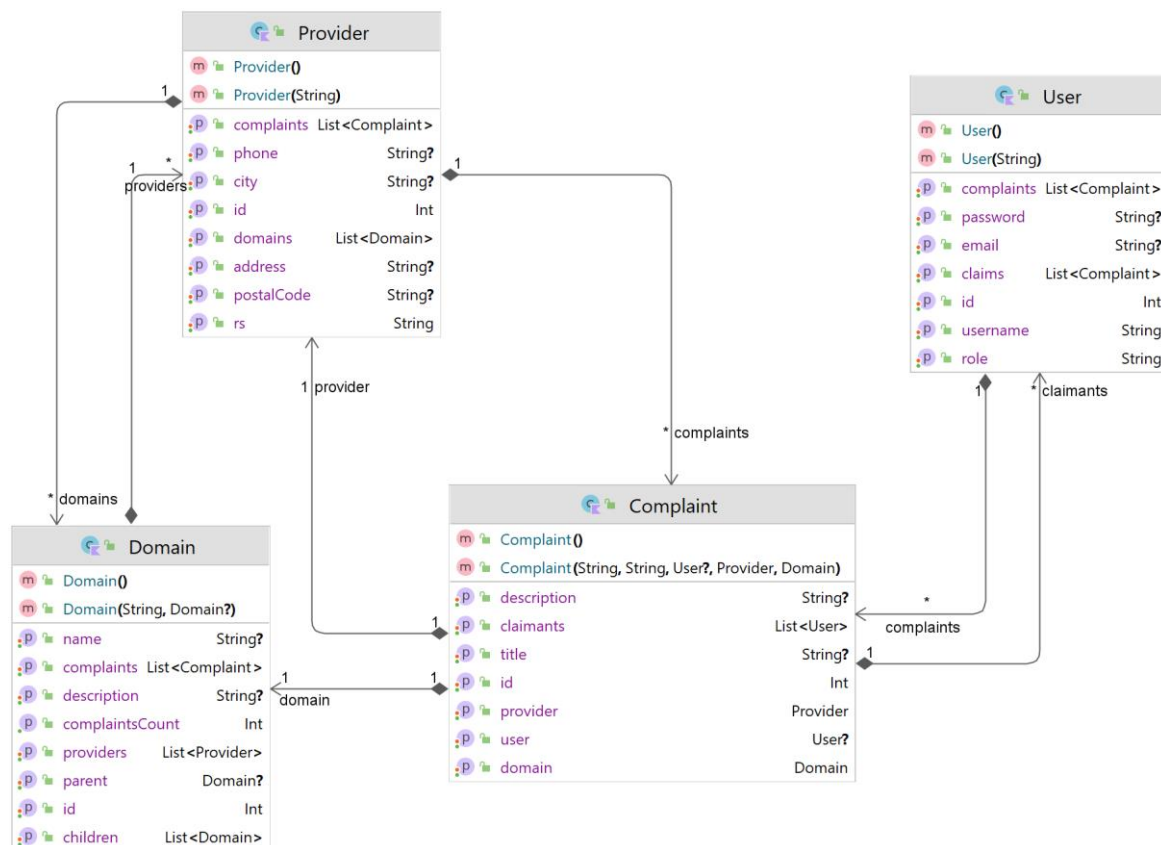
- Respecter les nommages de routes.
- Commiter régulièrement.
- **Laissez votre repository en private** (ajouter **jcheron** aux collaborators).

## Initialisation

- Télécharger le projet <https://github.com/s4-dut-info/btp-td> et dézipper le dans la branche **td6** de votre repository.

## Entites

Vérifier/compléter l'implémentation des entites dans le package **edu.spring.btp.entities**



## Règles de gestion :

- Un domaine peut avoir un domaine parent (excepté le Root). Les domaines ont éventuellement des domaines enfants.
- Le provider (entreprises) est rattaché à un ensemble de domaines.
- Les utilisateurs peuvent « balancer » un provider (complaints) qui intervient dans un domaine.
- Les utilisateurs ayant été confrontés à une situation similaire à une plainte déjà déposée peuvent se joindre à la plainte (claimants).

Méthodes spécifiques :

#### Domain

- **getComplaintsCount**: Retourne le nombre cumulé de plaintes par domaine, en incluant les sous-domaines.
- **hasChildren** : Détermine si le domaine a des enfants.

#### User

- **preRemove** : méthode permettant de mettre à null le user d'une plainte si le user est supprimé, à marquer avec l'annotation **@PreRemove**.

### Repositories

A compléter/vérifier dans le package **edu.spring.btp.repositories**

#### DomainRepository

- Recherche par name (1 seul)
- Recherche par name du parent (plusieurs)
- Recherche des domaines n'ayant pas de parent (plusieurs)
- Recherche des domaines par id du parent (plusieurs)
- Retourne un domaine aléatoire à partir d'un id de parent (1 seul)

#### UserRepository

- Recherche d'un utilisateur par username ou email (à utiliser pour l'authentification)

### Contrôleur & routes

A définir dans **edu.spring.btp.controllers** :

#### IndexController

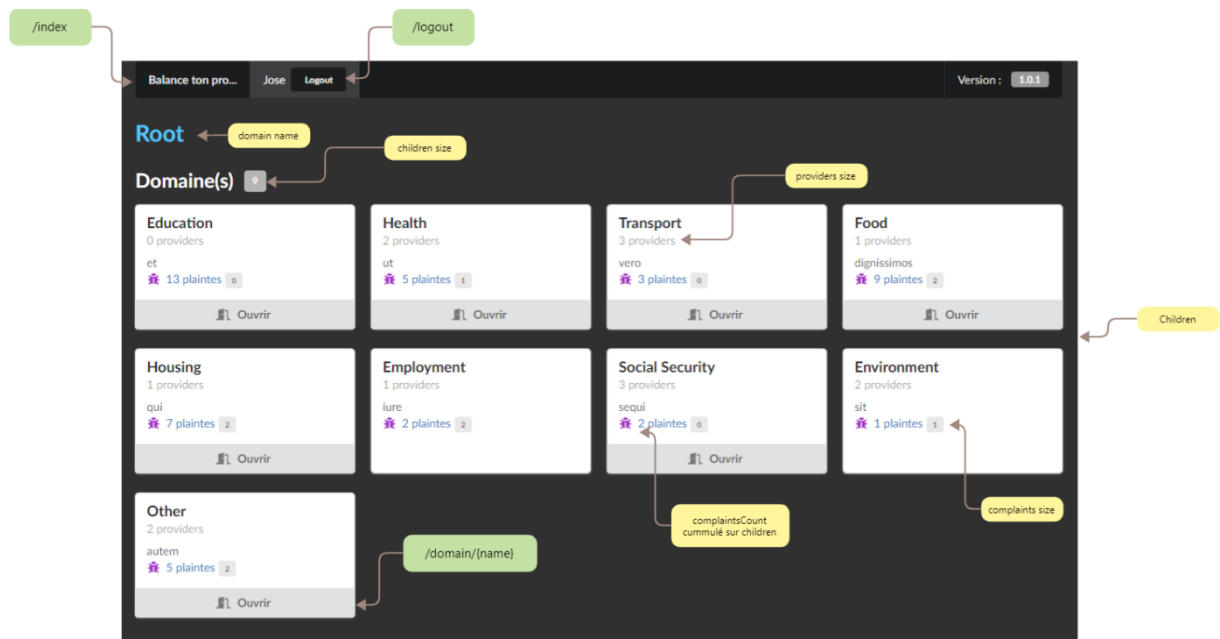
Route	Methode	Description	Template/redirect:	Role
/init/all/{count}	get	Peuple la BDD à partir de kotlin faker	/	
"/" ou "" ou "/index"	*	Affiche la page principale listant les domaines depuis Root	index	GUEST
/domain/{name}	get	Liste les domaines enfants du domaine de nom name	index	GUEST
/complaints/{domain}	get	Liste les plaintes du domaine	complaints	USER
/complaints/{domain}/sub	get	Liste les plaintes du domaine et de sa descendance (children)	complaints	USER
/complaints/{domain}/new	get/post	Ajout de nouvelle plainte	/forms/complaint redirect:/complaints/{domain}	USER
/login	get/post	Formulaire de login	/forms/login	GUEST
/signup	get/post	Formulaire de création de compte puis création de compte avec message	Redirige vers /	GUEST
/logout		Déconnexion et affichage d'un message	Redirige vers /	

En vert : **conception libre, mais respectant le diagramme UC ci-dessous**

En bleu : **déjà implémenté**

## UI

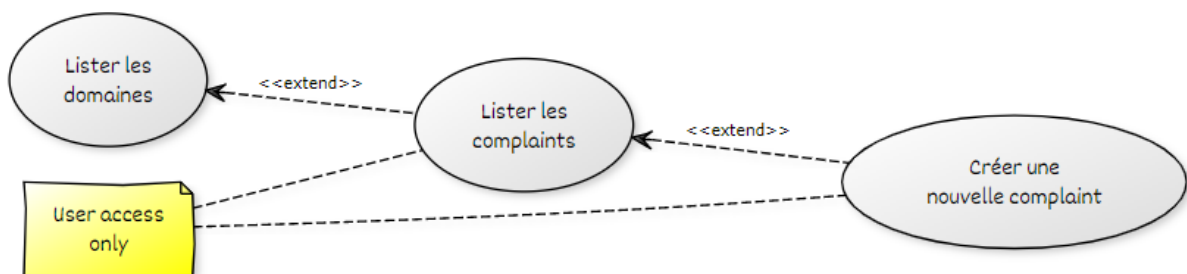
### /index



### /domain/{name}



### /complaints et /complaints/{domain}/new



On veillera à afficher le domaine des complaints :

- Dans l'affichage des complaints, en mode récursif ou non
- Dans le formulaire de création de nouvelle complaint

## Authentification

A mettre en place avec Spring security

The screenshot shows a dark-themed login page titled "Connexion - BTP". At the top, there are three navigation links: "Balance ton pro...", "Se connecter", and "Créer un compte", followed by a "Version : 1.0.1" label. The main form contains an email input field with the placeholder "ahmed.beer@gmail.com", a password input field with masked characters ".....", and a "Se connecter" button. Below the form, there is a link "Pas de compte ? ya plus qu'à..." and a footer note "icônes créées par ultimatearm - Flaticon".

/signup

The screenshot shows a dark-themed registration page titled "Création de compte - BTP". It features four input fields: "Nom d'utilisateur", "Adresse email", "Mot de passe", and "Confirmez votre mot de passe". Each field has a placeholder text. Below the fields is a "Créer le compte" button. At the bottom, there is a link "Déjà un compte ? Te connecter..." and a footer note "icônes créées par ultimatearm - Flaticon".

## Gestion des erreurs et messages

Gérer par message (semantic-ui) les erreurs :

- 403 - Forbidden
- 404 – Fichier non trouvé
- 500 – Erreur serveur

Afficher des messages :

- Sur erreur de connexion
- Sur connexion
- Sur déconnexion



Faire un dernier commit & push en fin de séance.

Créer une release **v2.0.0** sur la branche.

Poursuivre après la séance jusqu'à la date limite fixée et créer une release **v2.1.0** une fois terminé.