

# MARQUENTIN DARDOUILLET REMY

**TITRE CONCEPTEUR DEVELOPPEUR  
D'APPLICATION**

**DOSSIER PROJET**

**Septembre 2024 - Juin 2025**



## SOMMAIRE

COMPÉTENCES.....	3
CAHIER DES CHARGES.....	4
PRÉSENTATION DE L'ENTREPRISE .....	5
Histoire.....	5
L'équipe .....	6
Services proposés .....	8
GESTION DU PROJET.....	10
Planning.....	10
Outils et logiciels utilisés .....	11
<b>Phuton</b> .....	13
Suivi.....	13
LE PROJET.....	14
Jeu de données .....	14
Présentation de l'existant.....	14
Analyse de l'existant .....	15
Amélioration à apporter et maquettage.....	16
Azure dev ops.....	17
Frontend du projet .....	18
Backend du projet.....	20
Charger la page sans détail.....	21
Charger les détails au minimum (accordéon « type » ) .....	24
Chargement du détail .....	30
Sécurisation.....	32
Effet de bord : Page Réorientation .....	34
Vérification des performances sur un serveur distant.....	36
TEST.....	38
CONCLUSION .....	40
REMERCIEMENT .....	41
ANNEXES .....	42

## COMPÉTENCES

<b>Développer une application sécurisée</b>	Installer et configurer son environnement de travail en fonction du projet	
	Développer des interfaces utilisateur	
	Développer des composants métier	
	Contribuer à la gestion d'un projet informatique	
<b>Concevoir et développer une application sécurisée en couches</b>	Analyser les besoins et maquetter une application	
	Définir l'architecture logicielle d'une application	
	Concevoir et mettre en place une base de données relationnelle	
	Développer des composants d'accès aux données <b>SQL</b> et <b>NoSQL</b>	
<b>Préparer le déploiement d'une application sécurisée</b>	Préparer et exécuter les plans de tests d'une application	
	Préparer et documenter le déploiement d'une application	
	Contribuer à la mise en production dans une démarche DevOps	

# CAHIER DES CHARGES

Mon travail a été d'aider **NEOLINK** à améliorer les performances de leurs services.

Pour les aider, on m'a proposé de choisir une page sur laquelle travailler puis faire une procédure complète :

- L'analyse de l'existant
- Proposition d'amélioration
- Maquettage et analyse **SQL**
- Développement
- Vérifier les effets de bords
- Test fonctionnels sur ma page

En fonction du temps que je consacrerai à la procédure, **NEOLINK** me permet d'effectuer des tests fonctionnels et automatique sur d'autres pages

# PRÉSENTATION

## Contexte

Pour mon stage de **concepteur développeur d'application**, je l'ai effectué chez **NEOLINK** à Blois, dans les locaux de la fabrique du numérique, d'une durée de 13 semaines du **24 mars** jusqu'au **18 juin**.

## Histoire de NEOLINK

**NEOLINK** est une entreprise fondée en 2012 par Akli BRAHIMI, spécialisée dans l'édition en web social avec pour objectif de développer « l'internet qui réunit ».

L'entreprise se concentre sur la conciliation de l'activité économique avec l'utilité sociale.

**NEOLINK** s'inscrit résolument dans une démarche responsable et citoyenne qui se traduit par diverses actions telles que le choix d'implantation en zone urbaine défavorisée dans un but de mixité sociale, son attachement profond à la notion d'amélioration sociale, ainsi que son engagement à réinvestir chaque année 20% de son chiffre d'affaires en R&D

Forte de son expertise digitale et de sa connaissance du monde des plateformes, **NEOLINK** est reconnue pour ses innovations numériques. Elle s'appuie sur une équipe en constante évolution, composée de professionnels issus principalement du monde web et de l'analyse des données, avec pour certaines personnes des parcours atypiques.

Tous partagent l'ambition de construire des outils digitaux qui soient avant tout au service de chacun d'entre nous.

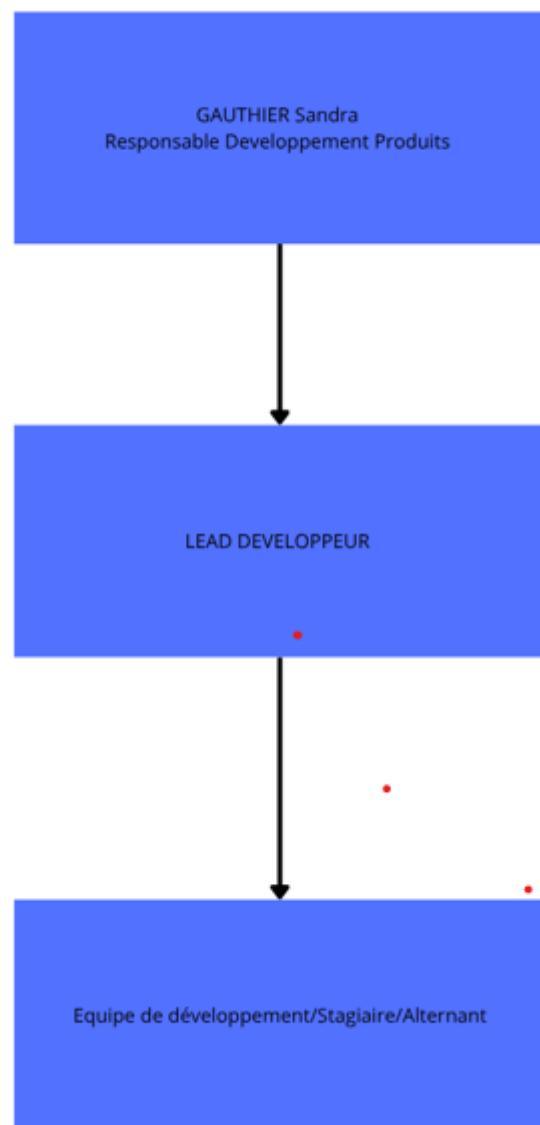
L'entreprise a été rachetée en 2019 par le groupe **Berger-Levrault**, et son nouveau directeur général délégué se nomme Guy BEAUDET.

## L'équipe

Il y a actuellement 37 personnes qui travaille chez **NEOLINK** :

- 2 testeurs
- 4 qualités
- 5 infrastructures
- 12 managers
- 1 chargé de communication
- 2 développeurs frontend
- 13 développeurs : 2 frontend, 11 backend dont 4 on le rôle de lead développeurs :
  - COLOMBEL Guillaume
  - BEAUGEARD Emilie
  - PIERRE Thomas
  - AUZURET Clement

## Organigramme de l'équipe de développement



## Services proposés

### **Neo Job360**

**NEOLINK** répond à la demande du plan de lutte contre la pauvreté et propose une approche globale et sans rupture, qui prend en compte toutes les potentialités et dimensions des allocataires du RSA et des acteurs économiques présents sur chaque territoire.

Les objectifs sont les suivant :

- Géolocalisation des offres
- Données candidats centralisés
- Matching des compétences
- Mise en relation
- Propositions d'emplois

### **Neo Gestion**

Cette solution, lancée courant 2020 en collaboration avec le Conseil Départemental du Bas-Rhin, propose une approche globale et sans rupture qui prend en compte toutes les potentialités et dimensions des allocataires du RSA et des partenaires du Département.

Les objectifs sont les suivant :

- Récupération des données
- Echanges simplifiés et coordonnés
- Suivi en ligne, global et sans rupture
- Réalisation d'un diagnostic socio-professionnel complet

- Orientation vers les dispositifs de retour à l'emploi simplifiée
- Optimisation de la prise de rendez-vous
- Digitalisation du CER
- Production des indicateurs nécessaire

## Neo Insertion

Cette plateforme permet la gestion et l'accès en ligne à l'ensemble des dispositifs insertion du PDI/PTI. Grâce à ce module, agissez sur tous les leviers pour accélérer le retour à l'emploi des allocataires du RSA par des actions d'insertion mises en place sur vos territoires.

Les objectifs sont les suivant :

- Développer la mobilisation de l'offre d'insertion dans les parcours
- Rendre lisible et accessible l'offre départementale d'insertion
- Favoriser l'autonomie et le pouvoir d'agir des bénéficiaires du RSA
- Sécuriser et accélérer l'entrée dans une action d'insertion
- Permettre une vision partagée du parcours pour le référent et l'opérateur
- Optimiser le suivi conventionnel avec les opérateurs
- Faciliter l'évaluation des dispositifs et des actions ainsi que leur impact dans les parcours des BRS

# GESTION DU PROJET

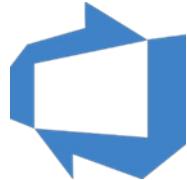
## Planning

Semaines	1	2	3	4	5	6	7	8	9	10	11	12	13
Découvertes													
Documentations													
Mise en place des outils													
Phase d'analyse													
Maquettage													
Développement													
Test													

## Outils et logiciels utilisés



NEOLINK utilise **teams** pour communiquer



**Azure dev ops** pour l'aspect gestion de projet et pour les tickets



**Git** pour le versionning



**Adminer** est le logiciel de gestion de base de données

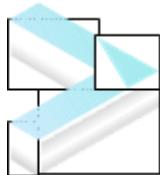


**PHP** avec leurs « framework maison » du nom de **Phuton** actuellement, ils ont pour objectif d'utiliser Symfony donc leur travail va de plus en plus dans le sens de ce dernier.



**CSS** et **HTML** et javascript (avec la bibliothèque **Jquery**) pour le front

**Tabby** pour l'accès au serveur local.



J'ai utilisé **VSCode** comme IDE avec un ensemble d'extension.



Une extension **VSCode** et du navigateur internet,  
**Xdebug** permet de faire du pas à pas pour faciliter la visualisation du transit des données au seins du code.



Le PC prêté par **NEOLINK** possède **Windows 11** comme OS

## Framework Maison : Phuton

NEOLINK ont leur propre **Framework maison** et son fonctionnement se base sur une structure **MVC**.

- 1) Le **Controller** fera appel à des **traitements**.
- 2) Les **traitements** gèrent tout ce qui est en rapport avec la base de données, et toutes les données sont renvoyés au **Controller**
- 3) Le **Controller** avec les informations récupérées du **traitement**, les envois au **Template**.
- 4) Le **Template** récupère toutes les informations et génère une partie de l'html en faisant appel à d'autres **Template** et/ou en l'écrivant directement. Le **Template** créer des pseudos balises qui seront utilisable dans la **vue**
- 5) La **vue** contient le minimum de HTML possible et c'est les pseudos balises qui s'occupe de générer une grosse partie de ce dernier.

Au cours des années, le **Framework** ayant été pensé pour du développement de 2012-2013, certaines adaptations ont été ajoutées, comme l'usage des objets pour adopter une approche de développement plus moderne durant le projet.

## Suivi

Mon Tuteur est **LOIR Thomas** avec qui j'ai pu échanger tout le long du stage et qui m'a permis de comprendre les manières de travailler dans l'entreprise.

NEOLINK utilises la **méthode Scrum** avec des sprints de 2 semaines et un Daily chaque matin dont **GAUTHIER Sandra** s'occupe de la préparation.

# LE PROJET

## Jeu de données

Le jeu de données que je vais utiliser tout au long du projet est basé sur l'environnement de production. Mais toutes les données sont modifiées et anonymisées sur l'environnement local et de développement. Les données sont mises à jour tous les week-ends. Cela permet de travailler avec une quantité de données réelles.

## Présentation de l'existant

Mon but lors de ce stage a été de choisir une page et d'analyser les problèmes qu'elle comportait. J'ai choisi celle des sanctions plus précisément la partie demande en attente.

The screenshot shows a web interface for managing administrative requests. The main header is 'Sanction / les demandes en attente de positionnement'. The 'DEMANDES DE SANCTION' section contains the following data:

Nom	Motif
LEBLANC Maïl	Le bénéficiaire ne respecte pas les engagements du contrat d'engagement ou du PPNE
HILARE Maissa	Le bénéficiaire ne respecte pas les engagements du contrat d'engagement ou du PPNE
HOREAU Martin	Le bénéficiaire n'a pas débordé de contrat d'engagement ou de PPNE dans les délais prévus après le dépôt de sa demande ou n'a pas renouvelé son contrat d'engagement ou son PPNE
HILARE Maya	Le bénéficiaire n'a pas débordé de contrat d'engagement ou de PPNE dans les délais prévus après le dépôt de sa demande ou n'a pas renouvelé son contrat d'engagement ou son PPNE
PROUST Sofiane	Le bénéficiaire respecte les engagements du contrat d'engagement ou du PPNE
CHARVET Odile	Le bénéficiaire ne respecte pas les engagements du contrat d'engagement ou du PPNE
FOUBERT Raphael	Le bénéficiaire ne respecte pas les engagements du contrat d'engagement ou du PPNE

Cette page affiche les allocataires rangés en 3 types de demandes :

- Demande de réouverture de droits
- Demande de Sanction
- Demande de Fin de sanction

L'accordéon à chaque allocataire permet de voir ses informations

## Analyse de l'existant

Le problème principal de cette page est que toutes les données sont chargées dès l'initialisation de la page, ce qui peut causer des ralentissements en cas forte influence.

Toutes les données non affichées à l'écran se mettent en « display : none ».

```
<h2 class="titre-type">Demandes de sanction</h2>
<ul id="ul_liste_demandes_ctrsa-sanction" class="liste_demandes_ctrsa liste-demande-ctrsa" data-type="sanction">flex
  <li class="animated fadeInLeft demande_ctrsa demande_ctrsa--demandeattente demande-instance" data-nature="Demandes de sanction" data-secteur="STRASBOURG" data-territoire="STRASBOURG"
    data-cleps="2130812" data-date="" data-perclep="546196" data-etreref="" data-reunionclep="" data-annulation="" data-type_ctrsa_valeur="sanction">flex
      <div class="conteneur_entete_demande_complete conteneur-entete-demande"></div> event grid
      <div id="details-informations-demande-2130812" class="conteneur-details-infos-demande dropdown-details--content" style="display:none;"></div>
    </li>
  <li class="animated fadeInLeft demande_ctrsa demande_ctrsa--demandeattente demande-instance" data-nature="Demandes de sanction" data-secteur="UTAMS SUD" data-territoire="UTAMS SUD"
    data-cleps="2146828" data-date="" data-perclep="554200" data-etreref="" data-reunionclep="" data-annulation="" data-type_ctrsa_valeur="sanction">flex
    <div class="conteneur_entete_demande_complete conteneur-entete-demande"></div> event grid
    <div id="details-informations-demande-2146828" class="conteneur-details-infos-demande dropdown-details--content" style="display:none;"></div>
  </li>
```

Un utilisateur aura rarement besoin de consulter en détail toutes les personnes, ce qui rend certaines données inutiles au chargement initial de la page.

## Amélioration à apporter et maquettage

Ma proposition principale a été de charger les données progressivement plutôt que toutes d'un coup, afin d'éviter un ralentissement au lancement de la page. Seules les données utiles à l'utilisateur sont chargées lors du dépliement de l'accordéon, ce qui réduit la charge immédiate sur la base de données et devrait améliorer les performances. Une fois chargées, les données restent disponibles pendant toute la navigation sur la page.

Dans un premier temps, je me suis chargé de réaliser des **maquettes** proposant une présentation sous forme d'accordéons, avec un chargement des données déclenché uniquement lors de leur dépliement, dans le cadre du maquettage 'chargement des données'. Ces **maquettes** (cf. annexe « Maquettage ») ont été validées par le pôle frontend.

The screenshot shows a web-based application interface for managing requests. At the top, there's a header with the logo 'QD JOB' and several navigation links: Accès, Contacts, Agenda, Docs, Images, and Compte. Below the header, the main title is 'Tableau de bord des instances > Sanction / les demandes en attente de positionnement'. The main content area is titled 'Sanction Les demandes'. It features a search bar with dropdowns for 'Tous les types de demande', 'Territoire', and 'Secteur', along with a 'Réinitialiser les filtres' button. There are also buttons for 'Sélectionner toutes les demandes', 'Rattacher les demandes sélectionnées à une instance existante', 'Créer une instance pour les demandes sélectionnées', and 'Afficher le détail de toutes les demandes'. A checkbox for 'Afficher uniquement les demandes des allocataires dont je suis le référent' is also present. Below these controls, there are three tabs: 'En attente de positionnement (33)', 'Refusées / annulées ce mois (7)', and 'Demandes en attentes sanction couple (0)'. Underneath the tabs, there are three accordéon-like sections: 'DEMANDES DE RÉOUVERTURE DE DROITS', 'DEMANDES DE SANCTION', and 'DEMANDES DE FIN DE SANCTION'. At the bottom of the page, there are two footer sections: 'LES RUBRIQUES DU SITE' and 'INFORMATIONS'. The 'LES RUBRIQUES DU SITE' section includes links to 'Espace accompagnateur', 'Espace Emploi', 'Community Manager', and 'Mon espace événements'. The 'INFORMATIONS' section includes links to 'Espace recruteurs', 'Mon espace aide', and 'Espace Entreprises'.

## Azure dev ops

Après cette phase d'analyse, j'ai dû créer mes premiers tickets sur **azure devops** pour pouvoir organiser le développement.

Je présente l'exemple de mon premier ticket :

- « CD XX » indique la plateforme concernée. Même si je travaille actuellement sur le CD67, mon code est conçu pour être réutilisable sur d'autres plateformes et a déjà un impact sur certaines d'entre elles.
- « Front : Accordéon Type » indique que je travaille seulement sur le Front de l'accordéon type
- Il y a une description expliquant ce qui est fait dans ce ticket avec des images de mes **maquettes** (cf. annexe « Azure Dev ops »).

Work items						
Assigned to me		New Work Item	Open in Queries	Column Options	Import Work Items	Recycle Bin
Filter by keyword				Types	States	Area
ID	Title			State	Area Path	Tags
14356	CD XX - Ajax : Accordéon Type			<input checked="" type="radio"/> Active	Dev	
14357	CD XX - Ajax : Accordéon Allocataire			<input type="radio"/> New	Dev	
14246	CD XX - Back : Séparation en 3 méthode			<input checked="" type="radio"/> Active	Dev	
14226	CD XX - Front: Accordéon Type			<input checked="" type="radio"/> Active	Dev	
14222	PERPLAT - Optim performance "instances"			<input type="radio"/> New	Dev	

## Frontend du projet

Avant de passer à la partie backend du projet et d'adapter le chargement des requêtes **SQL**, il m'a été conseillé de me concentrer d'abord sur l'aspect frontend de l'accordéon, afin de faciliter ensuite l'intégration côté backend

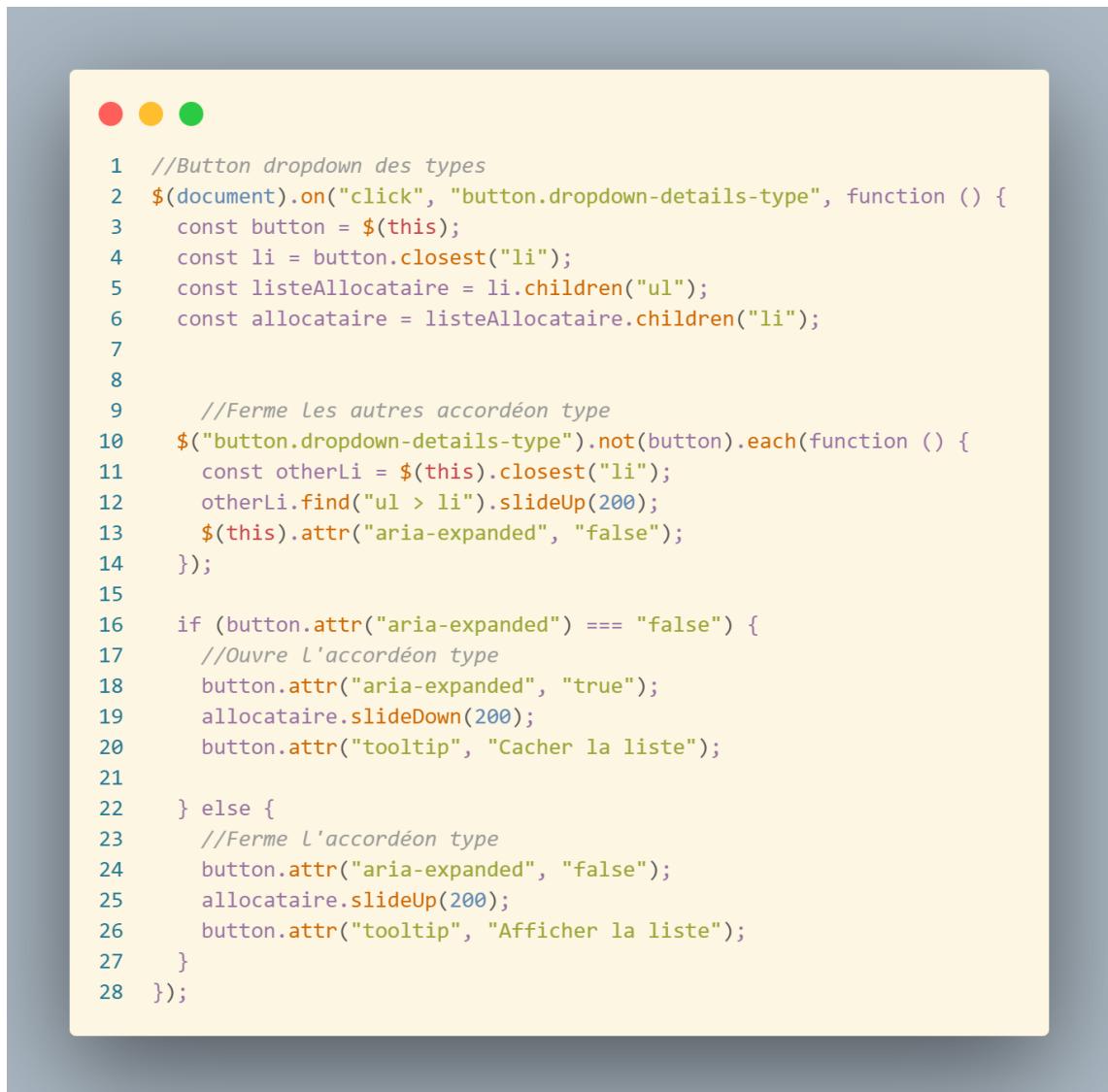
J'ai modifié le HTML/CSS en ajoutant un bouton déroulant. J'ai créé une div de la classe conteneur-type qui m'a permis de pouvoir placer le code du bouton dropdown déjà existant sur celui que j'ai créé dans le html avec la classe dropdown-detail-type.



```
1  > div.conteneur-type {
2    display: grid;
3    grid-template-rows: min-content 1fr;
4    grid-auto-flow: column;
5    align-items: center;
6    padding-right: 1.5rem;
7    cursor: pointer;
8    button.dropdown-details-type {
9      grid-column: ~"3 / 4";
10     grid-row: ~"1 / 3";
11     display: flex;
12     align-items: center;
13     margin-inline: auto 1rem;
14     padding: 0;
15     border-radius: 0;
16     aspect-ratio: 1;
17     &:before {
18       font-family: @fontIcones;
19       font-size: 1.5rem;
20     }
21
22   &[aria-expanded="true"] {
23     background: @couleurBoutons;
24     color: @couleurTexteBoutons;
25     &:before {
26       content: @pictoChevronHaut;
27     }
28   }
29
30   &[aria-expanded="false"]::before {
31     content: @pictoChevronBas;
32   }
33
34   span {
35     display: none;
36   }
37 }
38 }
```

Je me suis occupé de la partie **JS (Jquery)**, avec l'attribut aria-expanded réglé à **false** dans le HTML. J'ai fait une condition aux cliques du bouton qui passe aria-expanded a **true** et affiche la variable allocataire et dans le **else** l'effet inverse ce produit

J'ai également ajouté un comportement permettant de refermer automatiquement les autres accordéons actifs, afin qu'un seul puisse être ouvert à la fois.



The screenshot shows a presentation slide with a light gray background. In the top left corner, there are three colored circles: red, yellow, and green. Below them is a code block in a white box with a thin black border. The code is written in JavaScript (using jQuery syntax) and is numbered from 1 to 28. The code handles the click event on a button with the class "button.dropdown-details-type". It finds the closest li element and its ul children. It then iterates over other li elements to close their ul children and set their aria-expanded attribute to false. Finally, it checks if the clicked button's aria-expanded attribute is false; if so, it opens the ul and changes the tooltip to "Cacher la liste". If it's true, it closes the ul and changes the tooltip to "Afficher la liste".

```
1 //Button dropdown des types
2 $(document).on("click", "button.dropdown-details-type", function () {
3     const button = $(this);
4     const li = button.closest("li");
5     const listeAllocataire = li.children("ul");
6     const allocataire = listeAllocataire.children("li");
7
8
9     //Ferme les autres accordéon type
10    $("button.dropdown-details-type").not(button).each(function () {
11        const otherLi = $(this).closest("li");
12        otherLi.find("ul > li").slideUp(200);
13        $(this).attr("aria-expanded", "false");
14    });
15
16    if (button.attr("aria-expanded") === "false") {
17        //Ouvre l'accordéon type
18        button.attr("aria-expanded", "true");
19        allocataire.slideDown(200);
20        button.attr("tooltip", "Cacher la liste");
21
22    } else {
23        //Ferme l'accordéon type
24        button.attr("aria-expanded", "false");
25        allocataire.slideUp(200);
26        button.attr("tooltip", "Afficher la liste");
27    }
28});
```

## Backend du projet

Pour cette partie, j'ai dû séparer le code en trois **méthodes** différentes de base il y avait un seul **Controller** qui fessais appelle à un seul **Traitement** qui génère toutes les données. Après des échanges pendant les points et avec mon tuteur de stage je suis parti sur l'idée de séparer en 3 **Controller** et 3 **Traitements** :

- Le **Controller** demandeattente() utilise le **traitement** demandeAttenteSansDetail(). Ce **Controller** sera celui qui sera appelé au moment du chargement de la page il ne charge que c'est nécessaire comme les types de demande/les compteurs/les territoires/les secteurs. Du a la construction de **Phuton** je suis obligé de laisser le **Controller** demandeattente() en tant que **Controller** de base pour le système de routing.
- Le **Controller** demandesAttentes() pour cette **méthode** j'ai utilisé le système de **Traitement/dépôt** un système plus orienté sur les objets. Ce **Controller** sera celui qui sera appelé via une requête **Ajax** après avoir cliqué sur le bouton des bandeau types il récupère le nom/prénom/numéro d'allocataire et le motif de tous les allocataires liés au types déplié.
- Le **Controller** demandeAttenteParAllocataire() pour cette **méthode** j'ai utilisé le système de **Traitement/dépôt** un système plus orienté sur les objets. Ce **Controller** sera celui qui sera appelé via une requête **Ajax** après avoir cliqué sur le bouton pour voir le détail d'un allocataire il récupère le reste des informations de l'allocataire.

## Charger la page sans détail

Pour ce **Controller** je suis parti sur la même base déjà existante que voici :



```
1  /**
2   * Demandes en attente
3   * @created 02/02/2022
4   * @param mixed $params
5   */
6  public static function demandeattente($params)
7  {
8      /** @var Dashboard $tDashboard */
9      $tDashboard = self::traitement('Dashboard');
10     $acces = false;
11     // Internaute connecté
12     $personne = $tDashboard::obtenirPersonne();
13     if (!empty($params[0])) {
14         // Configuration de l'internaute
15         $cfg = $tDashboard::config($personne, $params[0]);
16         $cfg['affichage'] = 'demande';
17         if (!empty($cfg['types']) && !empty($cfg['role'])) {
18             // Demandes en attente
19             $data = $tDashboard::demandeattente($cfg, $params[0]);
20
21             $dataDashboardManageur = new DataDashboardManageur($cfg);
22             $compteurs = $dataDashboardManageur->getData('demandes');
23
24             self::parametre('compteurs', $compteurs);
25             self::parametre('type', $params[0]);
26             self::parametre('typeDemande', 'demandeattente');
27             self::parametre('data', $data);
28
29             // Droits de l'internaute
30             $droits = $cfg['droits'] ?? false;
31             self::parametre('droits', $droits);
32
33             list($formUri, $formData, $formSubmit) = $tDashboard::préparerFormulaireCreationSanctionReorientation();
34             self::parametre('formUri', $formUri);
35             self::parametre('formData', $formData);
36             self::parametre('formSubmit', $formSubmit);
37
38             $acces = true;
39         }
40     }
41     // Crédit d'une page standard
42     self::creerPage('defaut_react');
43     if ($acces) {
44         // Ajout du template de demandeattente comme template principal
45         self::template('listes' . DS . 'demande_passage_instance');
46     } else {
47         // Ajout du template erreur comme template principal
48         self::template('erreur');
49     }
50 }
```

Ce **Controller** fait appelle à la **méthode** `demandeattente()` du **Traitement** `Dashboard` qui fait une grosse requête **SQL** et comme dit plus tôt charge de beaucoup de données pas spécialement utiles au chargement de la page.

Donc j'ai refactoriser ce code en enlevant tout ce qui n'est pas utile au chargement de la page et j'ai créé une **méthode** dans le **Traitemet** du nom de `demandeAttenteSansDetail()`.



```
1  /**
2   * Demandes en attente
3   * @created 02/02/2022
4   * @param mixed $params
5   */
6  public static function demandeAttente($params)
7  {
8      /** @var Dashboard $tDashboard */
9      $tDashboard = self::traitement('Dashboard');
10     $acces = false;
11     // Internaute connecté
12     $personne = $tDashboard::obtenirPersonne();
13     if (!empty($params[0])) {
14         // Configuration de l'internaute
15         $cfg = $tDashboard::config($personne, $params[0]);
16         $cfg['affichage'] = 'demande';
17         if (!empty($cfg['types']) && !empty($cfg['role'])) {
18             // Demandes en attente
19             $data = $tDashboard::demandeAttenteSansDetail($cfg, $params[0]);
20             $dataDashboardManageur = new DataDashboardManageur($cfg);
21             $compteurs = $dataDashboardManageur->getData('demandes');
22
23             self::parametre('compteurs', $compteurs);
24             self::parametre('type', $params[0]);
25             self::parametre('typeDemande', 'demandeattente');
26             self::parametre('data', $data);
27
28             // Droits de l'internaute
29             $droits = $cfg['droits'] ?? false;
30             self::parametre('droits', $droits);
31             list($formUri, $formData, $formSubmit) = $tDashboard::preparerFormulaireCreationSanctionReorientation();
32             self::parametre('formUri', $formUri);
33             self::parametre('formData', $formData);
34             self::parametre('formSubmit', $formSubmit);
35         }
36     }
37     $acces = true;
38 }
39 // Crédit d'une page standard
40 self::creerPage('defaut_react');
41 if ($acces) {
42     // Ajout du template deamndeattente comme template principal
43     self::template('listes' . DS . 'demande_passage_instance_accordeon');
44 } else {
45     // Ajout du template erreur comme template principal
46     self::template('erreur');
47 }
48 }
```

J'ai enlevé la partie qui récupère les infos de l'accompagnement car ce n'est pas utile au chargement de la page.

**Traitemet** demandeAttenteSansDetail() récupère les 3 valeurs utile pour la suite elle fait appelée à la **méthode** demande() qui a déjà une base de requête je n'ajoute juste les 3 colonne que je veux récupérer avec les **méthodes** proposées par **Phuton** la clep/per\_clep(équivalent des id) et type et \$retour me renvois ça dans le **Controller** demandeattente()

```
1  /**
2   * Demande en attente sans détails
3   * @param array $cfg Configuration
4   * @param string $type Le type de suivi
5   * @return array
6   * @throws \Exception
7   */
8  public static function demandeAttenteSansDetail(array $cfg, string $type): array
9  {
10
11      $req = self::demandes($type, $cfg, "attente");
12      $req->columns(['ep.clep']);
13      $req->columns(['ep.per_clep']);
14      $req->columns([$req::alias('rvtype.valeur', 'type_ctrsa_valeur')]);
15      $retour = $req();
16      if ($retour === false) {
17          throw new \Exception(__FILE__ . "::demandeAttenteSansDetail dans la requête :" . $req);
18      }
19      return $retour ? $retour : [];
20  }
21
```

### **Charger les détails au minimum (accordéon « type »)**

A partir de cette étape je n'ai plus utilisé les **Traitements** de base de **Phuton** mais j'ai utilisé un système créé récemment qui tourne autour des objets du a une optique de passage vers Symfony dans le futur.

Je récupère mes infos envoyées par requête **Ajax**. La requête ne sera exécutée que une fois dès qu'un accordéon a déjà été déroulé j'ai fait en sorte que ça ne relance pas l'appel **Ajax** inutilement avec comme condition s'il y a un li qui a une longueur strictement égale à 0 ça lance la requête **Ajax** alors cela crée avec append le html que j'ai envoyé via **Ajax** avec mon **Controller** demandesAttentes().

```

if(button.attr("aria-expanded") === "false") {
    button.attr("aria-expanded", "true");
    button.attr("tooltip", "Cacher la liste");
    //Ne rappelle pas l'appel ajax si les données des allocataires sont déjà chargé
if (enteteDemandes.children("li").length > 0) {
    enteteDemandes.children("li").slideDown(200);
    //Appelle un appel ajax si les données des allocataires ne sont pas chargé
} else if (enteteDemandes.children("li").length === 0) {
    Neolink.ui.demarrerAnimationChangement(enteteDemandes, "Changement...");
    NL.ajax.post(
        "/ctrsa/default/demandesAttentes",
        {
            cleps: dataClepsValues,
            type: type,
        },
        function (reponseAjax) {
            if (reponseAjax.etat === "succes") {
                enteteDemandes.append(reponseAjax.html);
            } else if (reponseAjax.etat === "echec") {
                Neolink.ui.messageAjax(reponseAjax.message);
            }
        }
    );
}

}
} else {
    button.attr("aria-expanded", "false");
    button.attr("tooltip", "Afficher la liste");
    enteteDemandes.children("li").slideUp(200);
}
}

```

Ensute via la **méthode** de **Phuton** parametreRequete() et grâce au cleps récupéré je vais pouvoir aller chercher le nom/prénom/numéro allocataire et le motif du bandeau que j'ai déplié via la **méthode** recupererDemandesPassageInstanceResumees() que j'ai créé dans le manageur. Après cela le **Controller** envois le **Template** qui est appeler avec la **méthode** de **Phuton** genererTemplate() html avec les paramètres nécessaire et un état « succes » après cela la **méthode Phuton** parametre() contient **Ajax** pour lui indiquer que c'est une requête **Ajax** et le tableau réponse et creePage() est obligatoire pour la génération d'un **Template**.



```
1  /**
2   * Récupère plusieurs demandes avec Les données minimal
3   * @return void
4   */
5  public static function demandesAttentes()
6  {
7      $params = self::parametresRequete();
8
9      $cleps = !empty($params["cleps"]) ? $params["cleps"] : [];
10     $type = !empty($params["type"]) ? $params["type"] : '';
11
12     // Appeler un traitement pour récupérer les objets demandes
13     $demandeInstanceManageur = new DemandeInstanceManageur();
14     $demandes = $demandeInstanceManageur->recupererDemandesPassageInstanceResumees($cleps);
15
16     $reponse['html'] = self::genererTemplate('listes' . DS . 'blocs' . DS . 'entete_bloc_demande', [
17         'demandes' => $demandes,
18         'type' => $type,
19         'page' => 'demandeattente'
20     ]);
21     $reponse['etat'] = 'succes';
22
23
24     self::parametre('ajax', $reponse);
25     self::creerPage('ajax');
26 }
```

Dans la **méthode** recupererDemandesPassageInstanceResumees() je fais appelle à différente **méthode** qui font appel à des requête **SQL** (cf. annexe « codes supplémentaire utilisés ») la première me permet de rérécupérer les cleps des allocataires et la deuxième me permet de récupérer leurs motifs ces **méthodes** été déjà existante pour une allocataire j'ai juste adapté les requêtes avec l'opérateur logique IN en **SQL**.

```
/*
 * Récupéré des demandes de manière résumé nom/prenom/numero allocataire/motifs à partir des cleps demandes
 * @param array $cleps
 * @return array
 * @throws \Exception
 */
public function recupererDemandesPassageInstanceResumees(array $cleps): array
{
    // Depot pour récupération données
    $depot = new DepotDemandeInstance();
    $demandes = $depot->recupererDonneesEvenements($cleps);
    $motifs = $depot->recupererMotifsDemandes($cleps);

    // Récupération des infos allocataires via les per_clep présent dans le tableau demandes
    foreach ($demandes as &$demande) {

        if (!empty($demande["per_clep"])) {
            $infosAllocataires = $depot->recupererInfosAllocataire($demande["per_clep"]);

            if (!empty($infosAllocataires)) {
                $demande['nom'] = $infosAllocataires[0]['nom'];
                $demande['prenom'] = $infosAllocataires[0]['prenom'];
                $demande['num_allocataire'] = $infosAllocataires[0]['num_allocataire'];
            }
        }
    }
}
```



```
/*
 * Récupération : per_clep/nom/prenom/numero allocataire d'un allocataire via sa cclep
 * @param int $clep
 * @return array
 * @throws \Exception
 */
public static function recupererInfosAllocataire(int $clep): array
{
    $req = self::serviceNew('Requete', 'select', 'select_allocataire');

    $req
        ->table('████████', 'pers')
        ->columns(['pers.nom', 'pers.prenom', 'pers.num_allocataire'])
        ->where([
            ['pers.per_clep', '=', $clep],
        ]);
    $donnees = $req();
    if ($donnees === false) {
        throw new \Exception(__FILE__ . " ::recupererInfosAllocataire dans la requête :" . $req);
    }
    return $req();
}
```

Après cela j'ai fait une boucle foreach dans donneesEvenements avec l'opérateur de référence « & » qui me permet d'enregistrer directement dans le tableau \$donnees evenement les informations de l'allocataire que je vais récupérer via la **méthode** que j'ai créé dans le dépôt qui me permet de récupérer les informations de l'allocataire via sa clep j'ai ajouté un retour d'exception en cas d'erreur qui remplace le système de try/catch traditionnelle car cela ne fonctionne pas en **Phuton**.

```
foreach ($demandes as &$demande) {
    foreach ($motifs as $motif) {
        if ((!empty($demande["clep"]) && !empty($motif["clep"]) && $demande["clep"] === $motif["clep"])) {
            $demande = array_merge($demande, $motif);
        }
    }
}

return $demandes;
}
```

Après cela j'ai fait des foreach imbriqué pour parcourir les tableaux \$donneesEvenements qui contient du coup en plus des cleps/per\_cleps ect toutes les données des allocataire (cf.annexe « Autre ») et \$motifs qui contient les motifs ces 2 tableaux sont lié par les cleps commune ce qui me permet de les fusionné à l'aide de array\_merge() dans le tableau \$retour.

Ensuite le renvoyer au **Controller** via le tableau \$demandes qui est ensuite envoyé au **Template**

Pour la partie templating avec **Phuton** une bonne partie du html ce crée coté PHP avec le moins de choses possible coté **Vue** qui ce trouve dans un dossier html ici j'ai créé une variable \$html a vide qui permettra de stocké tout mon html a renvoyé a la **Vue** j'ai bouclé sur le tableau \$demandes que mon **Controller** a envoyé que j'ai récupéré via la **méthode** parametre() et j'ai fait un foreach dessus qui me permet de récupérer les données dans une variable réutilisable j'ai pris le soins de faire une condition ternaire au cas ou si le tableau est vide cela renvois autres choses.

```
{
    $demandes = self::parametre('demandes');
    $page = self::parametre('page');
    $html = '';
    foreach ($demandes as $donnees) {
        self::$htmlRelatif(
            'datachangepr',
            isset($donnees['exchange_pluri']) ? "data-changepr" . $donnees['exchange_pluri'] . '=""' : ''
        );
        $clep = !empty($donnees['clep']) ? $donnees['clep'] : '';
        $perClep = !empty($donnees['per_clep']) ? $donnees['per_clep'] : '';
        $nom = !empty($donnees['nom']) ? $donnees['nom'] : 'Non renseigné';
        $prenom = !empty($donnees['prenom']) ? $donnees['prenom'] : 'Non renseigné';
        $numAlloc = !empty($donnees['num_allocataire']) ? '' . $donnees['num_allocataire'] . '' : 'Non renseigné';
        $motifDemande = !empty($donnees['motif']) ? '

>span class="motif-entete-demande">Motif ' . ($page === 'demandeannule' ? 'initial' : '') . '</span>' . $donnees['motif'] . '</p></div>' : 'Non renseigné';

        $divSelectionAllocataire = '<div class="selection-demande" >
            <label for="checkbox-demande-' . $donnees['clep'] . '">Sélectionner cette demande</label><input id="checkbox-demande-' . $donnees['clep'] . '" type="checkbox" class="checkbox_demande_ctrsa" value="'.($page === 'demandeannule' ? 'initial' : '') .'></input>
        </div>';
        $infosAnnulation = '';
        if ($page === 'demandeannule') {
            $infosAnnulation =
                '<span class="infos-date">' . (isset($donnees['type_titre']) ? '' : $donnees['type_titre']) . '<span>' . ($donnees['type_titre'] === 'annulée' ? 'Refusée' : ucfirst($donnees['type_titre'])) . '</span>' . '<br>' .
                (isset($donnees['date_complete']) ? '' : ' le ' . $donnees['date_complete']) . '</span>';
        }
        $html .= '<li class="animated fadeInLeft demande_ctrsa-' . $page . ' demande-instance" >'.($page === 'dataChangepr' ? $htmlRelatif('datachangepr', $donnees['exchange_pluri']) : $html) . '</li>';
    }
}


```

J'ai ensuite ajouté les données dynamiquement avec le html déjà existant certaine donnée sont manquante mais ce sera traiter dans la partie test. Ensuite la **méthode** htmlRelatif crée une pseudo balise utilisable dans la **Vue** ma **Vue** ne contient seulement cette dernière.

```
$html .= '<li class="animated fadeInLeft demande_ctrsa demande_ctrsa--" . $page .' demande-instance" <./dataExchange>
data-nature="<./nature>" data-secteur="<./secteur>" data-territoire="<./territoire>" data-clep="'.$clep.'" data-date="<./date>"'
data-perclep="'.$perclep.'" data-etreref="<./etre_referent>" data-reunionclep="<./reunion_clep>"'
data-annulation="<./type_titre>" data-type_ctrsa_valeur ="<./type_ctrsa_valeur>"'
<div class="conteneur_entete_demande_complexe conteneur-entete-demande" >'.
$divSelectionAllocataire .
<div class="identite-personne">
<h3 class="nom_prenom">' .
$nom .' '. $prenom .
'</h3>' .
$numAllo .
'</div>' .
$motifDemande .
$infosAnnulation .
<button id="btn-show-details-'.$clep.'" tooltip="Afficher le détail de la demande"
class="dropdown-details bouton bouton-outline" aria-label="Afficher/Cacher du détail de la demande"
aria-expanded="false" aria-controls="details-informations-demande-'.$clep.'"
><span>Afficher/Cacher le détail de la demande</span>
</button>
</div>';
}
self::htmlRelatif('html', $html);
}
```

## Chargement du détail

Pour cette étape le raisonnement a été le même que pour le **Controller** demandesAttentes(). Je fais appel à un système d'objet a la différence que j'ai appelle à des **Traitements** déjà existants (cf.annexe « codes supplémentaire utilisés »)



```
1 /**
2  * Retourne Les informations détaillées d'une demande dans Le template detail_bloc_demande
3  * @return void
4  */
5 public static function demandeAttenteParAllocataire()
6 {
7
8     $params = self::parametresRequete();
9     $reponse['etat'] = 'echec';
10    if (!empty($params)) {
11
12        $internaute = new Personne((new Utilisateur())->per_clep);
13        $cfg = Dashboard::config($internaute);
14
15        $cfg['affichage'] = 'demande';
16
17        $droits = !empty($cfg['droits']) ? $cfg['droits'] : [];
18        $droitBoutons = $droits['gestion_demande'];
19
20        $droits = !empty($cfg['droits']) ? $cfg['droits'] : [];
21        $droitReo = iset($droits['instance']['reorientation']['categorie']['demande']['attente'])
22        ? $droits['instance']['reorientation']['categorie']['demande']['attente'] === true
23        : false;
24        $droitSanction = iset($droits['instance']['sanction']['categorie']['demande']['attente'])
25        ? $droits['instance']['sanction']['categorie']['demande']['attente'] === true
26        : false;
27
28
29        $clep = !empty($params["clep"]) ? $params["clep"] : [];
30        $type = !empty($params["type"]) ? $params["type"] : [];
31        $typesSanction = $type === "sanction" || $type === "reouverture_droits" || $type === "fin_sanction";
32
33        if (($type === 'reorientation' && $droitReo) || ($typesSanction && $droitSanction) && !empty($cfg['role'])) {
34
35            $typeDemandeManager = new TypeDemandeInstanceManager();
36            $typeDemande = $typeDemandeManager->recupererTypeDemandeDepuisClepDemande($clep);
37
38            $demandeInstanceManager = new DemandeInstanceManager();
39            $demandeInstance = $demandeInstanceManager->recupererDemandePassageInstance($clep, $typeDemande);
40
41            $reponse['html'] = self::genererTemplate('listes' . DS . 'blocs' . DS . 'detail_bloc_demande', [
42                'demande' => $demandeInstance,
43                'droits' => $droits,
44                'droitBouton' => $droitBoutons,
45                'visible' => true,
46                'page' => 'demandeattente'
47            ]);
48            $reponse['etat'] = 'succes';
49        } else {
50            $reponse['message'] = 'Vous n\'avez pas les droits requis';
51        }
52    } else {
53        $reponse['message'] = 'Erreur dans la requête';
54    }
55    self::parametre('ajax', $reponse);
56    self::creerPage('ajax');
57}
```

J'ai juste eu à modifier le **Traitement recupererDemandePassageInstance()** en ajoutant la récupération du secteur car il ne le fesais pas a la fin cela crée un objet DemandeInstance sur lequel je vais pouvoir agir dans mon **Template** via les getteurs.

```

    return (new DemandeInstance())
        ->setClep($clepDemandeInstance)
        ->setAllocataire($allocataire)
        ->setDemandeur($demandeur)
        ->setType($type)
        ->setSituationFamiliale($situationFamiliale)
        ->setMotif(self::serviceNew('ReferentielValeur', $depot->recupererMotifDemande($clepDemandeInstance)[0]))
        ->setCommentaire($depot->recupererCommentaire($clepDemandeInstance, "EvenementsParcours"))
        ->setTerritoire($territoire)
        ->setSecteur($secteur);
}

```

Pour le **Template** j'ai donc utilisé les getteurs

```

$donnees = self::parametre('demande');
$droits = self::parametre('droits');
$droitBoutons = self::parametre('droitsBouton');
$page = self::parametre('page');

$allocataire = $donnees->getAllocataire();
$territoire = $donnees->getTerritoire();
$secteur = $donnees->getSecteur();
$motifDonnees = $donnees->getMotif();
$CommentaireDonnees = $donnees->getCommentaire();
$demandeur = $donnees->getDemandeur();

```

Pour chaque champ nécessaire j'ai vérifié s'il était vide et j'ai récupéré seulement ce qui m'intéresser dans mon exemple le libelle du motif le html proviens d'un code déjà existant je l'ai adapté à ma situation.

```

$motif = !empty($motifDonnees) ? $motifDonnees->libelle : 'Non renseigné';
self::htmlRelatif('motif', '<li class="ligne_detail_demande ligne_detail_demande--motif" >
    <span class="titre_detail_demande"> Motif :</span>
    <span class="contenu_detail_demande">' . $motif . '</span>
</li>');

```

Certain comme la situation familiale m'a demandé de faire appel à un autre objet et une **méthode** déjà existante.

```

$traitementSituation = new SituationFamiliale();
$situation = $traitementSituation->recupererLibelleValeurSituationFamiliale($perClep, $clep);
$libelleSituation = $situation['libelle'];
self::htmlRelatif('situation_familiale', '<li class="ligne_detail_demande ligne_detail_demande--situation_familiale" >
    <span class="titre_detail_demande">Situation familiale :</span>
    <span class="contenu_detail_demande">' . $libelleSituation . '</span>
</li>');

```

Pour le demandeur j'ai fait un système de tableau car il faut récupérer le nom et le prénom de ce dernier et sa peu avoir une utilité pour le code futur

```
$demandeurDonnees = $donnees->getDemandeur();
if (!empty($demandeurDonnees)) {
    $demandeur = [
        'clep_demandeur' => $demandeurDonnees->clep,
        'nom_demandeur' => $demandeurDonnees->nom,
        'prenom_demandeur' => $demandeurDonnees->prenom
    ];
}
self::htmlRelatif('demandeur', '<li class="ligne_detail_demande ligne_detail_demande--demandeur" >
    <span class="titre_detail_demande">Professionnel demandeur :</span>
    <span class="contenu_detail_demande">' . $demandeur['nom_demandeur'] . ' ' . $demandeur['prenom_demandeur'] . '</span>
</li>');
```

## Sécurisation

Pour la partie sécurisation des systèmes ont été déjà mis en place avec des règle de développement dans l'entreprise comme par exemple il faut toujours qu'un tableau ou une chaîne de caractère renvois quelque chose pour cela j'ai découvert le fonctionnement des condition ternaire dans l'exemple si dessous par exemple pour \$type :

Si le tableau params["type"] n'est pas vide alors (le ?) on envoie la chaîne de caractère qui ce trouve dans type sinon(les :) renvois une chaîne vide

J'ai appliqué ce raisonnement à chaque fois que c'était nécessaire tout au long du projet

```
$cleps = !empty($params["cleps"]) ? $params["cleps"] : [];
$type = !empty($params["type"]) ? $params["type"] : '';
```

Pour les requêtes **SQL** j'en ai déjà parler quelques pages avant la manière d'écrire les requêtes avec le framework **Phuton** fait le try/catch ne fonctionne pas à la place j'ai utilisé une condition si la req() est fausse alors cela enclenche une erreur volontaire et ça évite de faire cracher le script.

```
));
$donnees = $req();
if ($donnees === false) {
    throw new \Exception(__FILE__ . "::recupererInfosAllocataire dans la requête :" . $req);
}
return $req();
```

Pour la sécurisation générale il y a déjà un système de droit existant dans la code base donc je l'ai utilisé pour parler plus en détail de cette étapes je vais prendre en exemple mon **Controller** demandesAttentes() que j'ai modifié.

```

1 /**
2  * Récupère plusieurs demandes avec les données minimal
3  * @return void
4 */
5 public static function demandesAttentes()
6 {
7     $params = self::parametresRequete();
8     $reponse['etat'] = 'echec';
9     if (!empty($params)) {
10         $internaute = new Personne((new Utilisateur())->per_clep);
11         $cfg = Dashboard::config($internaute);
12
13         $cfg['affichage'] = 'demande';
14
15         $droits = !empty($cfg['droits']) ? $cfg['droits'] : [];
16         $droitReo = isset($droits['instance']['reorientation']['categorie']['demande']['attente'])
17             ? $droits['instance']['reorientation']['categorie']['demande']['attente'] === true
18             : false;
19         $droitSanction = isset($droits['instance']['sanction']['categorie']['demande']['attente'])
20             ? $droits['instance']['sanction']['categorie']['demande']['attente'] === true
21             : false;
22
23
24         $cleps = !empty($params["cleps"]) ? $params["cleps"] : [];
25         $type = !empty($params["type"]) ? $params["type"] : '';
26         $typesSanction = $type === "sanction" || $type === "reouverture_droits" || $type === "fin_sanction";
27
28         if (($type === 'reorientation' && $droitReo) || ($typesSanction && $droitSanction) && !empty($cfg['role'])) {
29
30             // Appeler un traitement pour récupérer les objets demandes
31             $demandeInstanceManager = new DemandeInstanceManager();
32             $demandes = $demandeInstanceManager->recupererDemandesPassageInstanceResumees($cleps);
33
34             $reponse['html'] = self::genererTemplate('listes' . DS . 'blocs' . DS . 'entete_bloc_demande', [
35                 'demandes' => $demandes,
36                 'type' => $type,
37                 'page' => 'demandeattente'
38             ]);
39             $reponse['etat'] = 'succes';
40         } else {
41             $reponse['message'] = 'Vous n\'avez pas les droits requis';
42         }
43
44     } else {
45         $reponse['message'] = 'Erreur dans la requête';
46     }
47     self::parametre('ajax', $reponse);
48     self::creerPage('ajax');
49 }

```

J'ai ajouté une condition qui vérifie si \$params n'est pas vide s'il l'est cela renvoie un message "Erreur dans la requête".

en ajoutant la variable \$cfg qui fait appel à la **méthode config()** du dashboard et qui prend en paramètre \$internaute qui vérifie si un utilisateur est bien connecté.

Je ne peux pas montrer la **méthode config()** pour des raisons de confidentialité, mais dedans ce trouve plusieurs appels à différentes **méthodes** qui permettent de vérifier si \$cfg['droits'] vérifie les droitsCtrsa qui est un tableau associatif avec plusieurs entrées qui vérifient si l'internaute a le droit d'accéder à ces droits et renvoie une réponse en booléen. Cette partie sur les droits vérifie surtout si la personne qui accède à ce **Controller** se trouve bien sur une page de type sanction ou réorientation.

## Effet de bord : Page Réorientation

La modification du fonctionnement de ma page a affecté une autre page la page réorientation qui utilise le même **Controller** principale demandeattente(). J'ai donc adapté et ajouter des nouveautés dans mon code pour prendre en compte cette page.

Dans un premier temps dans le **Controller** chargement des détails j'ai ajouté une condition qui vérifie si le type est bien réorientation et fait appel à des nouveau **Traitements** (cf.annexe « page réorientation ») qui vont récupérer les informations nécessaires pour la page réorientation.

```
if ($type === 'reorientation') {  
    $ecPluri = $demandeInstanceManageur->recupererDemandeEchangePluri(clep: $clep);  
    $structReoProp = $demandeInstanceManageur->recupererStructureReoProp(clep: $clep);  
    $autreMotif = $demandeInstanceManageur->recupererAutreMotif(clep: $clep);  
}
```

Une fois les informations récupérer je les ajoute à mon appel **Ajax** pour pouvoir les réutiliser dans le **Template** ici aussi avec une condition qui vérifie si la page est bien réorientation pour éviter l'apparition de ces nouvelles informations dans ma page initiale

```
● ● ●  
1  if ($type === "reorientation") {  
2      $typeSuiviStructurePr = $structureDeSuivi->recupererTypeSuivi();  
3      self::htmlRelatif('structure_suivi_type', '<li class="ligne_detail_demande ligne_detail_demande--structure_suivi_type" >  
4          <span class="titre_detail_demande">Type de suivi actuel :</span>  
5          <span class="contenu_detail_demande">' . $typeSuiviStructurePr . '</span>  
6      </li>');  
7      if (!empty($autreMotif)) {  
8          self::htmlRelatif('autre_motif', '<li class="ligne_detail_demande ligne_detail_demande--autre_motif" >  
9              <span class="titre_detail_demande">Précision du motif :</span>  
10             <span class="contenu_detail_demande">' . $autreMotif . '</span>  
11         </li>');  
12     }  
13     $structureReoProp = !empty($structureReoPropDonnees) ? $structureReoPropDonnees[0]['StructureProp'] : 'Non Défini';  
14     self::htmlRelatif('structure_propose_suivi_type', '<li class="ligne_detail_demande ligne_detail_demande--structure_propose" >  
15         <span class="titre_detail_demande">Structure de réorientation proposée :</span>  
16         <span class="contenu_detail_demande">' . $structureReoProp . '</span>  
17     </li>');  
18     $ecPluri = !empty($ecPluriDonnees) ? $ecPluriDonnees['libelle'] : "Non Défini";  
19     self::htmlRelatif('echange_pluri', '<li class="ligne_detail_demande ligne_detail_demande--echange_pluri" >  
20         <span class="titre_detail_demande">Echange pluridisciplinaire souhaité :</span>  
21         <span class="contenu_detail_demande">' . $ecPluri . '</span>  
22     </li>');  
23 }
```

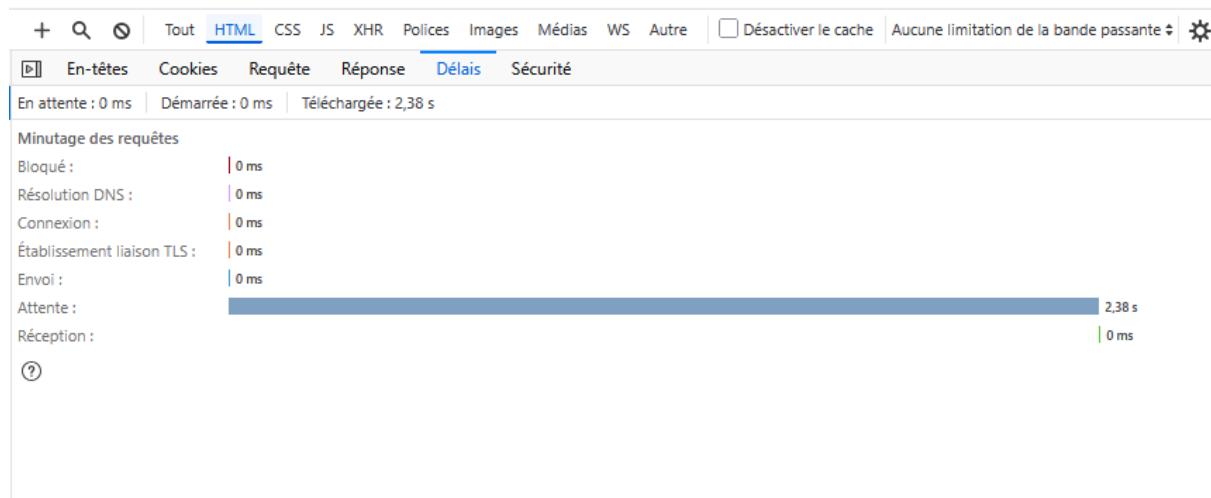
La seule information récupérer de manière différente est le type de suivis vu que structure de suivis est un objet de type Partenaire qui contiens déjà la clep dont j'ai besoin, j'ai ajouté la **méthode** recupererTypeSuivi() dans l'objet directement en récupérant la clep directement avec le mot clés \$this et je récupère exclusivement le libelle.

```
1      /**
2       * fonction qui permet de recuperer les prestation
3       * @return array
4       * @throws \Exception
5       */
6      public function recupererTypeSuivi()
7      {
8          $requete = self::serviceNew('Requete', 'select', 'prestation_type');
9          $requete
10             ->table('██████████', 'fr', true)
11             ->join('██████████', ['rv.clep', '=', 'fr.reponse_clep'], true);
12          $requete
13             ->columns(['rv.libelle'])
14             ->groupby(['rv.libelle'])
15             ->where([
16                 ['fr.nom_champ', '=', 's:type_accompagnement_structure_suivi'],
17                 ['fr.ctx_clep', '=', $this->clep],
18                 ['fr.ctx_type', '=', 's:Partenaire']
19             ]);
20          $donnees = $requete();
21          return empty($donnees) ? $donnees : $donnees[0]['libelle'];
22      }
23  }
24 }
```

## Vérification des performances sur un serveur distant

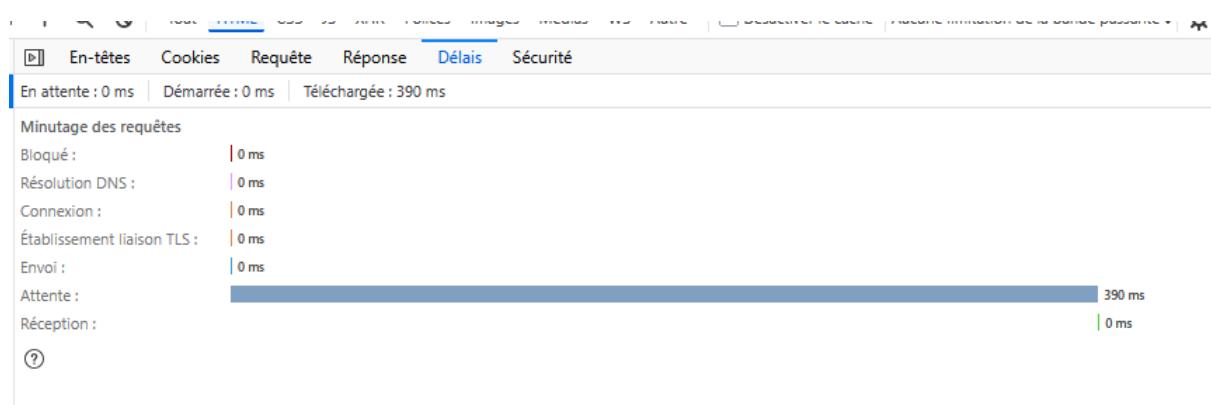
Pour vérifier si mes modifications étaient bénéfique aux pages j'ai utilisé le système de pipeline qui m'ont permis de test sur le serveur distant de développement. Dans mon exemple je ne prendrais que la page sanction mais les résultats de la page réorientation ce trouverons en annexe « résultat page réorientation »

Voilà actuellement la page sur le serveur de développement elle prend en moyenne 2s à ce chargé avant mes modifications

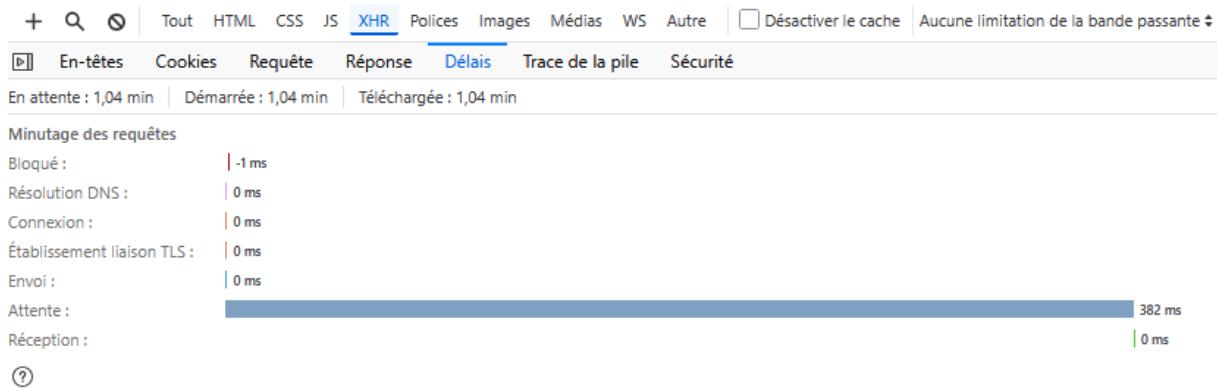


Mes modifications utilisant 3 **Controller** différent j'ai dû regarder au pas à pas le temps de chaque **Controller**

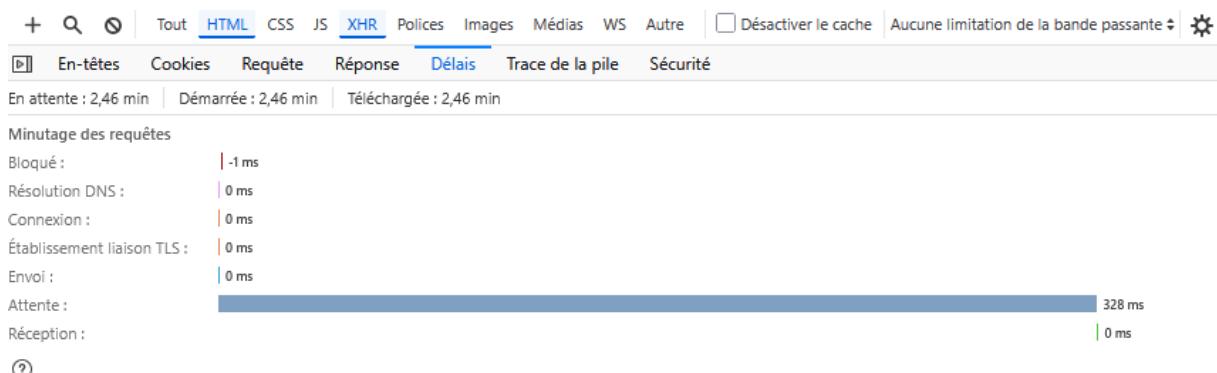
Le temps du **Controller** demandeattente() :



## Le temps du Controller demandeAttente() :



## Le temps du Controller demandeAttenteParAllocataire() :



Si on calcule le résultat des 3 Controller ils ne prennent seulement 1,1s à chargé les détails d'un allocataire ce qui en général sera l'utilisation commune d'un utilisateur. Mais sur une utilisation normal le gain est de 83% de performance.

## **TEST**

J'ai eu l'occasion de me plonger dans la partie test automatisé je n'ai pas travailler sur mon projet même si j'ai fait un cahier de test de ma page.

Cahier de test - demandeAttente			
Dépliement accordéon complet			
Action	Détail de l'action	Résultat attendu des tests	Test validé (O/N)
<a href="https://lob67.home/ctrsa/default/demandeattente/sanction">https://lob67.home/ctrsa/default/demandeattente/sanction</a>	accède à la page déplie un accordéon "type"	La page s'affiche avec les bonnes information affiche la liste des allocataires avec les bonnes information	O
	déplie detail d'un allocataire	clique sur le bouton ou le bandeau d'un allocataire affiche les details de l'allocataire avec les bonnes informations	O

Pour cette partie j'ai dû faire dans un premier temps quatre cahier (cf.annexe « test ») de test validé par la personne en charge des tests et de recette

Cahier de test - Gestion des structures			
Modification des informations d'une structure			
Action	Détail de l'action	Résultat attendu des tests	
<a href="https://cdlnumeroDepartt-katalon-php7.ru-neolink.fr/partenaire/detail/accueil/[clep]">https://cdlnumeroDepartt-katalon-php7.ru-neolink.fr/partenaire/detail/accueil/[clep]</a>	Accède à la page détail d'une structure	la page s'ouvre correctement avec les bonne information afficher une nouvelle page avec un formulaire s'ouvre	
Accès à la page des modifications	Clique sur le bouton modifier		
	Les champs modifiable sont : -Numéro SIRET parent -Code NAF -Nom commercial -Ville -Email entreprise -Site internet -Statut -Tranche d'Effectif -Tranche de Chiffre d'affaire -Description		
Modifier les informations			Les champs fonctionne correctement
Valider les modification	Clique sur le bouton enregistrer		Pop up vert de validation

Après avoir fait ce cahier de test je suis passé sur le développement du test de manière automatisé avec la librairie de test playwright.

Après avoir eu de l'aide d'un testeur pour configurer playwright comme la création de fixtures/Pages/Session et de la page principale de test (cf annexe « playwright ») j'ai créé ma méthode

```

1 import { Page } from "@playwright/test";
2 import { getUserBaseUrl, User } from "@utils/data";
3
4 export class PartenaireDetailPage {
5
6     /**
7      * Modifier le formulaire de modification d'une structure
8      * @param page
9      * @param user
10     * @param clepStructure
11    */
12
13    async modifierFormulaireStructure(page : Page, user: User, clepStructure : number) {
14        await page.goto(getUserBaseUrl(user) + 'partenaire/detail/accueil/' + clepStructure);
15        await page.click('button:has-text("Modifier")');
16
17        await page.locator('#siret-parent').fill('12345678900010');
18
19
20
21        await page.locator('input[name="Code NAF "]').fill('0112Z - Culture du riz');
22        await page.keyboard.press('ArrowDown');
23        await page.keyboard.press('Enter');
24
25        await page.locator('#nom_commercial').fill('Test UTMS DEUX');
26
27
28        await page.locator('input[name="Ville"]').fill('Furchhausen');
29        await page.keyboard.press('ArrowDown');
30        await page.keyboard.press('Enter');
31
32        await page.locator('#email_entreprise').fill('testUTMS2@test2.fr');
33
34        await page.locator('#site_internet').fill('https://testutms2.fr');
35
36
37        await page.locator('input[name="Statut"]').fill('Autre');
38        await page.keyboard.press('ArrowDown');
39        await page.keyboard.press('Enter');
40
41        await page.locator("input[name=\"Tranche d'Effectif\"]").fill("10000 salariés et plus");
42        await page.keyboard.press('ArrowDown');
43        await page.keyboard.press('Enter');
44
45        await page.locator("input[name=\"Tranche de Chiffre d'affaire\"]").fill("De 50 millions à moins de 100 millions d'eurose");
46        await page.keyboard.press('ArrowDown');
47        await page.keyboard.press('Enter');
48
49        await page.getByLabel('Description').fill('Lorem ipsum dolor sit amet, consectetur adipiscing elit TEST ETMS 2.');
50        await page.click('input:has-text("Enregistrer")');
51    }
52 }

```

La **méthode** goto permet d'accéder a la page de détail d'une structure dessus ce trouve un bouton « modifier ».

Ensuite pour chaque champ de type input classique je les ais ciblé sur leurs id avec la **méthode** locator et les remplir avec la **méthode** fill le champ ce remplis

Certains champs du formulaire étant construit différemment sous forme d'un select personnalisé qui est remplissable comme un input classique j'ai ciblé leurs nom et j'ai mis dans le fill une donnée fonctionnelle la **méthode** press permet de simuler le comportement d'une clavier.

## **CONCLUSION**

Ce projet a été très pertinent et très complet.

J'ai beaucoup appris du fait de ne pas me retrouver dans un **framework** que j'avais étudié, comme Symfony ou Laravel. Cela m'a obligé à m'adapter rapidement. Cependant, comme je maîtrisais déjà la structure **MVC**, la logique globale est restée similaire, ce qui m'a facilité l'adaptation.

Les compétences que j'ai pu développées pendant ce stage sont :

- Le travail en équipe
- L'autonomie,
- Meilleure compréhension du travail en entreprise
- La patience
- Adaptation

J'ai pu confirmer ma voie pour ce métier et j'espère pouvoir retrouver une entreprise tout aussi accueillante pour pouvoir m'épanouir.

## **REMERCIEMENT**

Je tiens déjà à remercier **GAUTHIER Sandra** de m'avoir accepté au sein de **NEOLINK** de m'avoir proposé un projet complet et intéressant.

Un remerciement tout particulier à **LOIR Thomas** mon tuteur, pour sa disponibilité constante, sa patience et l'expérience précieuse qu'il m'a partagée tout au long de ce stage.

Et enfin, je remercie toute l'équipe de **NEOLINK** pour sa disponibilité et son aide précieuse.

## ANNEXES

### Maquettage

The image displays two wireframe mockups of a software interface titled "Espace Accompagnateur".

**Top Mockup:** This view shows a general dashboard. At the top, there are filter dropdowns for "Tous les types de demande", "Territoire", and "Secteur", along with a "Réinitialiser les filtres" button. Below these are several action buttons: "Sélectionner toutes les demandes", "Rattacher les demandes sélectionnées à une instance existante", "Créer une instance pour les demandes sélectionnées", and "Afficher le détail de toutes les demandes". A status bar at the bottom indicates "En attente de positionnement (34)", "Refusées / annulées ce mois (9)", and "Demandes en attentes sanction couple (1)".

**Bottom Mockup:** This view is a detailed list of cases under the "Demandes" tab. It includes sections for "DEMANDES DE RÉOUVERTURE DE DROITS", "DEMANDES DE SANCTION", and "DEMANDES DE FIN DE SANCTION". The list shows four entries:

- DUMAS Guillaume: Motif: Réouverture de droit suite à une radiation/sanction
- DUFLOT Adele: Motif: Réouverture de droit suite à une radiation/sanction
- BARRIER Anne: Motif: Réouverture de droit suite à une radiation/sanction
- WEBER Dylan: Motif: Réouverture de droit suite à une radiation/sanction

**Espace Accompagnateur**

Tâches de bord des instances > Sanction | les demandes en attente de positionnement

**Sanction**  
**Les demandes**

Tous les types de demande

Sélectionner toutes les demandes    Afficher le détail de toutes les demandes

Afficher uniquement les demandes des allocataires dont je suis le référent

**Ex attente de positionnement (34)** Refusées / annulées ce mois (11) Demande en attente sanction couple (1)

**DEMANDES DE RÉOUVERTURE DE DROITS**

DUMAS Guillaume N° d'allocataire : 0000544415 Motif : Réouverture de droit suite à une radiation/sanction

DUFLOT Adèle N° d'allocataire : 0003598872 Motif : Réouverture de droit suite à une radiation/sanction

BARRIER Annie N° d'allocataire : 0000591174 Motif : Réouverture de droit suite à une radiation/sanction

WEBER Dylan N° d'allocataire : 0003798798 Motif : Réouverture de droit suite à une radiation/sanction

**Une question ?**

**Espace Accompagnateur**

Tâches de bord des instances > Sanction | les demandes en attente de positionnement

**Sanction**  
**Les demandes**

Tous les types de demande

Sélectionner toutes les demandes    Afficher le détail de toutes les demandes

Afficher uniquement les demandes des allocataires dont je suis le référent

**Ex attente de positionnement (34)** Refusées / annulées ce mois (11) Demande en attente sanction couple (1)

**DEMANDES DE RÉOUVERTURE DE DROITS**

DUMAS Guillaume N° d'allocataire : 0000544415  
[Voir son espace personnel](#)  
[Voir ses données administratives](#)  
 Adresse :  
 voir annotée(67100)STRASBOURG  
 Territoire : STRASBOURG - UT EST  
 Situation familiale : Seul sans enfant  
 Structure de suivi actuelle : CCGC STRASBOURG  
 Professionnel demandeur : D Josseline

Motif : Réouverture de droit suite à une radiation/sanction

Exposé de la situation : Commentaire anonymisé

DUFLOT Adèle N° d'allocataire : 0003598872 Motif : Réouverture de droit suite à une radiation/sanction

BARRIER Annie N° d'allocataire : 0000591174 Motif : Réouverture de droit suite à une radiation/sanction

WEBER Dylan N° d'allocataire : 0003798798 Motif : Réouverture de droit suite à une radiation/sanction

**Une question ?**

## Azure dev ops

### 14226 CD XX - Front: Accordéon Type



MARQUENTIN DARDOUILLET Remy

0 Comments

Add Tag

State	<input checked="" type="radio"/> New	Area	Dev
Reason	New	Iteration	Dev\2025 SPRINT S18 S19

#### Description

Le but de cette tache est de transformé la page des instance en double accordéon comme les maquette ici présente

Dans cette taches seuls l'aspect Front va être géré :

The screenshot shows a web-based application interface. At the top, there is a header with a logo (JOB) and several navigation links: Accès, Demande, Agenda, Dossiers, Import, and Départ. Below the header, a search bar displays the text "Sanction / les demandes en attente de positionnement". A sub-menu titled "Sanction Les demandes" is open, showing various filtering options: "Tous les types de demande", "Territoire", "Secteur", and "Réinitialiser les filtres". There are also buttons for "Sélectionner toutes les demandes", "Relier les demandes sélectionnées à une instance existante", "Créer une instance pour les demandes sélectionnées", and "Afficher le détail de toutes les demandes". A checkbox for "Afficher uniquement les demandes des allocataires dont je suis le détenteur" is also present. Below the search area, there are three tabs: "En attente de positionnement (33)", "Refusées / annulées ce mois (7)", and "Demandes en attente sanction couple (8)". Under these tabs, there are sections for "DEMANDES DE RÉOUVERTURE DE DROITS", "DEMANDES DE SANCTION", and "DEMANDES DE FIN DE SANCTION". At the bottom of the interface, there is a footer with links: "LES RUBRIQUES DU SITE" (Espace accompagnateur, Espace Emploi, Community Manager, Mon espace événements), "INFORMATIONS" (Espace recruteurs, Mon espace offre, Espace Entreprises), and a "S'ABONNER" button.

▼

**14222** PERFPLAT -  
Optim performance  
"instances"

● New

MARQUENTIN DARD...

**14226** CD XX - Front:  
Accordéon Type

● New

MARQUENTIN DARD...

Complete...

Iteration ... 2025 SPRINT S...

▼

## Code Supplémentaire utilisés

```
● ○ ●
1  /**
2   * Récupération des données des demandes
3   * @param array $clepsDemandeInstance
4   * @return array
5   * @throws \Exception
6   */
7  public function recupererDonneesEvenements(array $clepsDemandeInstance): array
8  {
9      $requeteDonneesEvenements = self::serviceNew('Requete', 'select', 'donnees_evenement')
10     ->table('████████', 'ep')
11     ->columns(['clep', 'per_clep', 'meta_per'])
12     ->where(['ep.clep', 'IN', '(' . implode(',', $clepsDemandeInstance) . ')']);
13     $donnees = $requeteDonneesEvenements();
14     if ($donnees === false) {
15         throw new \Exception(__FILE__ . " ::recupererDonneesEvenementsErreur dans la requête :" . $requeteDonneesEvenements);
16     }
17     return $donnees;
18 }
```

```
● ○ ●
1  /**
2   * Récupération des motifs des demandes
3   * @params array $clepsDemande Clep de La demande de passage en instance de l'allocataire
4   * @return array Correspondant au motif de la demande
5   * @throws \Exception
6   */
7  public function recupererMotifsDemandes(array $clepsDemande)
8  {
9      $req = self::serviceNew('Requete', 'select', 'select motif');
10
11     $req
12         ->table('████████', 'motif', true)
13         ->left('████████', 'val_motif', ['val_motif.clep', '=', 'motif.reponse_clep'], true)
14         ->where([
15             ['motif.ctx_clep', 'IN', '(' . implode(',', $clepsDemande) . ')'],
16             ['motif.ctx_type', '=', 's:EvenementsParcours'],
17             ['motif.nom_champ', '=', 's:gst_motif_demande'],
18             ['motif.reponse_type', '=', 's:ReferentielValeur']
19         ])
20         ->columns([
21             $req::alias('motif.ctx_clep', 'clep'),
22             $req::alias('val_motif.libelle', 'motif')
23         ]);
24     $donnees = $req();
25     if ($donnees === false) {
26         throw new \Exception(__FILE__ . " ::recupererMotifsDemandes dans la requête :" . $req);
27     }
28     return $req();
29 }
```

```

1  /**
2   * Création de l'objet DemandeInstance à partir de la clep demande
3   * @param int $clepDemandeInstance
4   * @param TypeDemandeInstanceInterface $type
5   * @return DemandeInstance
6   * @throws \Exception
7   */
8  public function recupererDemandePassageInstance(int $clepDemandeInstance, TypeDemandeInstanceInterface $type) : DemandeInstance
9  {
10     // Initialisation:
11     // Depot pour récupération données
12     $depot = new DepotDemandeInstance();
13     $donneesEvenement = $depot->recupererDonneesEvenement($clepDemandeInstance);
14     $clepTerritoire = $depot->requeteTerritoireDemandeInstance($clepDemandeInstance);
15     $clepSecteur = $depot->requeteSecteurDemandeInstance($clepDemandeInstance);
16     // Variables contenant object personne allocataire et réfèrent ainsi que situation familiale
17     $allocataire = null;
18     $demandeur = null;
19     $situationFamiliale = null;
20     $territoire = null;
21     $secteur = null;
22
23     // Condition: Clep allocataire disponible alors création objet personne et appel fonction pour récupérer
24     // situation familiale pour instance DemandeInstance
25     if (!empty($donneesEvenement[0]['per_clep'])) {
26         $allocataire = self::obtenirPersonne($donneesEvenement[0]['per_clep']);
27         if (!empty($allocataire)) {
28             $situationFamiliale = $this->recupererSituationPersonneDemande($allocataire->clep, $clepDemandeInstance);
29         }
30     }
31
32     // Condition: Clep référent renseignée alors création instance personne
33     if (!empty($donneesEvenement[0]['meta_per'])) {
34         $demandeur = self::obtenirPersonne($donneesEvenement[0]['meta_per']);
35     }
36
37     // Condition: Clep territoire récupéré alors création du Partenaire
38     if (!empty($clepTerritoire)) {
39         $territoire = new Partenaire($clepTerritoire);
40     }
41
42     // Condition: Clep secteur récupéré alors création du Partenaire
43     if (!empty($clepSecteur)) {
44         $secteur = new Partenaire($clepSecteur);
45     }
46
47     return (new DemandeInstance())
48         ->setClep($clepDemandeInstance)
49         ->setAllocataire($allocataire)
50         ->setDemandeur($demandeur)
51         ->setType($type)
52         ->setSituationFamiliale($situationFamiliale)
53         ->setMotif(self::serviceNew('ReferentielValeur', $depot->recupererMotifDemande($clepDemandeInstance)[0]))
54         ->setCommentaire($depot->recupererCommentaire($clepDemandeInstance, "EvenementsParcours"))
55         ->setTerritoire($territoire)
56         ->setSecteur($secteur);
57     }

```

```

1  /**
2   * Requete territoire de La demande CTRSA
3   * @param int $clepDemande
4   * @return string
5   * @throws \Exception
6   */
7  public static function requeteTerritoireDemandeInstance(
8      int $clepDemande
9  ): string {
10
11      $req = self::serviceNew('Requete', 'select', 'territoire_demande');
12      $req
13          ->table(['██████████'], 'fr', true)
14          ->left('██████████', ['rv.clep', '=', 'fr.reponse_clep'], true)
15          ->left('██████████', 'p', ['rv.valeur', '=', 'p.siret', true])
16          ->columns(['p.clep']);
17      $where = [['fr.nom_champ', '=', 's:territoire_instance']];
18      $where[] = ['fr.ctx_clep', '=', $clepDemande];
19      $where[] = ['fr.ctx_type', '=', 's:EvenementsParcours'];
20      $req
21          ->where($where);
22      $res = $req();
23
24      if ($res === false) {
25          throw new \Exception(__FILE__ . " ::requeteTerritoireDemandeInstancedans la requête :" . $req);
26      }
27      return empty($res) ? '' : $res[0]['clep'];
28
29  }
30
31 /**
32  * Requete secteur de La demande CTRSA
33  * @param int $clepDemande
34  * @return string
35  * @throws \Exception
36  */
37  public static function requeteSecteurDemandeInstance(
38      int $clepDemande
39  ): string {
40      $req = self::serviceNew('Requete', 'select', 'secteur_demande');
41      $req
42          ->table(['██████████'], 'fr', true)
43          ->left('██████████', 'rv', ['rv.clep', '=', 'fr.reponse_clep'], true)
44          ->left('██████████', 'p', ['rv.valeur', '=', 'p.siret', true])
45          ->columns(['p.clep']);
46      $where = [['fr.nom_champ', '=', 's:secteur_instance']];
47      $where[] = ['fr.ctx_clep', '=', $clepDemande];
48      $where[] = ['fr.ctx_type', '=', 's:EvenementsParcours'];
49      $req
50          ->where($where);
51      $res = $req();
52      if ($res === false) {
53          throw new \Exception(__FILE__ . " ::requeteTerritoireDemandeInstancedans la requête :" . $req);
54      }
55      return empty($res) ? '' : $res[0]['clep'];
56  }

```

## Page Réorientation



```
1  /**
2   * Fait appelle au dépôt pour récupérer Les Echange Pluri
3   * @param $clep Clep de la demande de passage en instance de l'allocataire
4   * @return array
5   */
6  public function recupererDemandeEchangePluri($clep)
7  {
8      $depot = new DepotDemandeInstance();
9      $ePluri = $depot->recupererEp($clep);
10     return $ePluri;
11 }
13
14 /**
15  * Fait appelle au dépôt pour récupérer la structure réo proposé
16  * @param $clep Clep de la demande de passage en instance de l'allocataire
17  * @return array
18  */
19 public function recupererStructureReoProp($clep)
20 {
21     $depot = new DepotDemandeInstance();
22     $structReoProp = $depot->recupererStructureReoProp($clep);
23     return $structReoProp;
24 }
25
26 /**
27  * Fait appelle au dépôt pour récupérer l'autre motif quand il existe sinon renvois une chaîne vide
28  * @param $clep Clep de la demande de passage en instance de l'allocataire
29  * @return string
30  */
31 public function recupererAutreMotif($clep)
32 {
33     $depot = new DepotDemandeInstance();
34     $autreMotifDonees = $depot->recupererAutreMotif($clep);
35     $autreMotif = !empty($autreMotifDonees) ? $autreMotifDonees[0]['Autre Motif'] : '';
36     return $autreMotif;
37 }
```

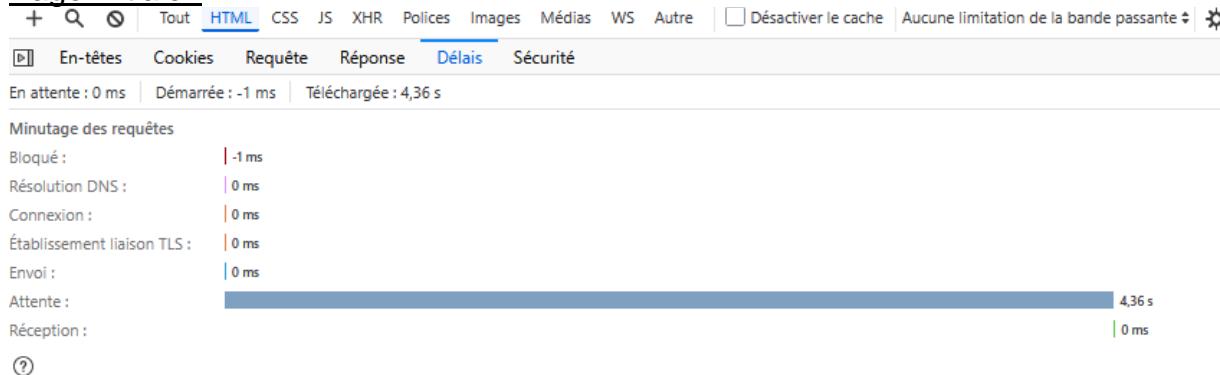
```

1  /**
2   * Récupération des valeurs échanges pluridisciplinaire
3   * @param $clepDemande Clep de La demande de passage en instance de L'allocataire
4   * @return array
5   * @throws \Exception
6   */
7  public function recupererEp(int $clepDemande): array
8  {
9      $req = self::serviceNew('Requete', 'select', 'select ecPluri');
10
11     $req
12         ->table('████████', 'ep', true)
13         ->join('████████', 'val_ep', ['val_ep.clep', '=', 'ep.reponse_clep'], true)
14         ->where([
15             ['ep.ctx_clep', '=', $clepDemande],
16             ['ep.ctx_type', '=', 's:EvenementsParcours'],
17             ['ep.nom_champ', '=', 's:demande_ctrsa_hebdo'],
18             ['ep.reponse_type', '=', 's:ReferentielValeur']
19         ])
20         ->columns([
21             $req::alias('ep.ctx_clep', 'clep'),
22             $req::alias('val_ep.libelle', 'échange_pluri'),
23         ]);
24     $donnees = $req();
25     if ($donnees === false) {
26         throw new \Exception(__FILE__ . "::recupererEp dans la requête :" . $req);
27     }
28     return $req();
29 }
30
31 /**
32 * Récupération des valeurs de la structure de réorientation proposée
33 * @param $clepDemande Clep de La demande de passage en instance de L'allocataire
34 * @return array
35 * @throws \Exception
36 */
37 public function recupererStructureReoProp($clepDemande): array
38 {
39     $req = self::serviceNew('Requete', 'select', 'select strctureReoProp');
40     $req
41         ->table('████████', 'ep', true)
42         ->join('████████', 'np', ['np.clep', '=', 'ep.valeur'], true)
43         ->where([
44             ['ep.ctx_clep', '=', $clepDemande],
45             ['ep.ctx_type', '=', 's:EvenementsParcours'],
46             ['ep.nom_champ', '=', 's:type_ctrsa'],
47         ])
48         ->columns([
49             $req::alias('np.nom_commercial', 'StructureProp'),
50         ]);
51     $donnees = $req();
52     if ($donnees === false) {
53         throw new \Exception(__FILE__ . "::recupererStructureReoProp dans la requête :" . $req);
54     }
55     return $req();
56 }
57
58 /**
59 * Récupération de L'autre Motif
60 * @param $clepDemande Clep de La demande de passage en instance de L'allocataire
61 * @return array
62 * @throws \Exception
63 */
64 public function recupererAutreMotif($clepDemande): array
65 {
66     $req = self::serviceNew('Requete', 'select', 'select autreCommentaire');
67     $req
68         ->table('████████', 'ep', true)
69         ->where([
70             ['ep.ctx_clep', '=', $clepDemande],
71             ['ep.ctx_type', '=', 's:EvenementsParcours'],
72             ['ep.nom_champ', '=', 's:commentaire_motif_autre_ctrsa'],
73         ])
74         ->columns([
75             $req::alias('ep.valeur', 'Autre Motif'),
76         ]);
77     $donnees = $req();
78     if ($donnees === false) {
79         throw new \Exception(__FILE__ . "::recupererAutreMotif dans la requête :" . $req);
80     }
81     return $req();
82 }
83
84 }

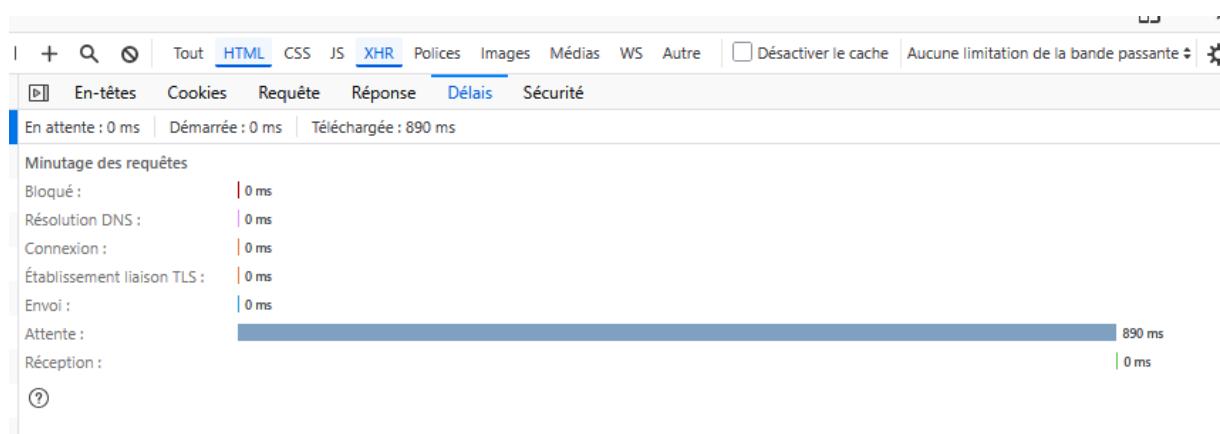
```

## Résultat page réorientation

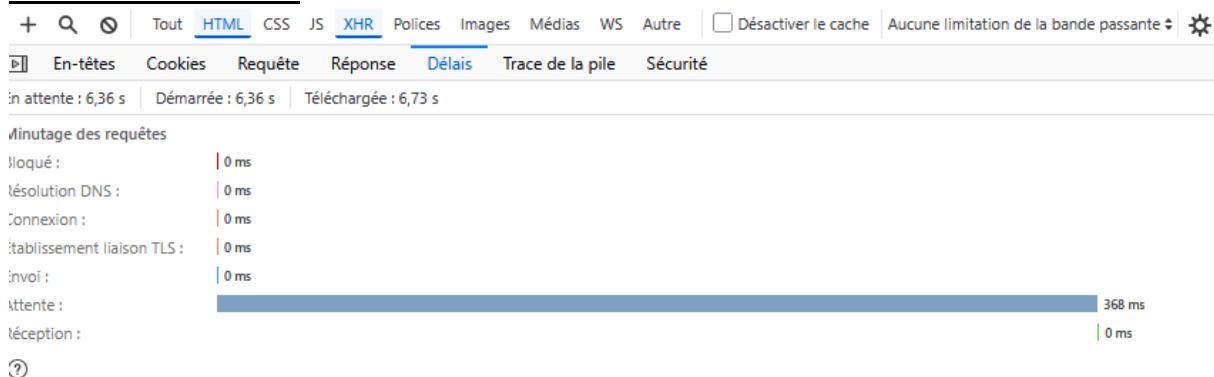
### Page initiale :



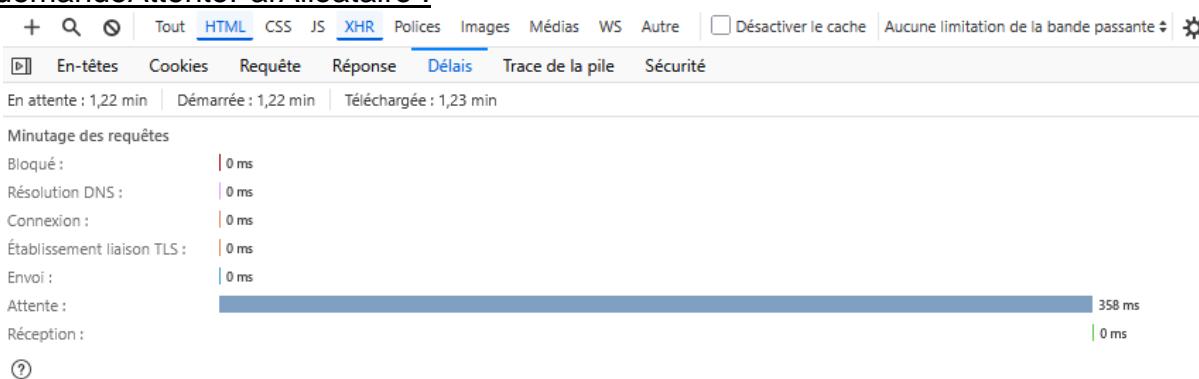
### demandeattente :



## DemandesAttentes :



## demandeAttenteParAlloataire :



Test

## Cahier de test - Gestion des structures

### Création d'une structure fille

Action	Détail de l'action	Résultat attendu des tests
<a href="https://cd[NumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/default/gestionPM/[clepDependantDuTerritoire]">https://cd[NumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/default/gestionPM/[clepDependantDuTerritoire]</a>	Accède à la page Cliquer sur le bouton "+enfant" sur un territoire externe	la page s'ouvre Une fenêtre avec un formulaire s'ouvre
Accès au formulaire de création	Remplir le formulaire avec un numero SIRET de 14 caractères et cliquer sur rechercher puis sur le bouton "créer une personne morale"	
Ajouter une numero Siret inexistant	Les champs modifiable sont: -Numéro SIREN -Numéro SIRET parent -Code NAF -Nom commercial -Raison sociale -Voie/lieu-dit -Code postal -Ville -Email entreprise -Site internet -Statut -Tranche d'Effectif -Tranche de Chiffre d'affaire -Description	Un nouveau formulaire s'ouvre
Remplir les informations		Après avoir cliquer sur suivant un nouveau
Coche la case	Cocher la case "Externe" et cliquer sur suivant	pop-up vert qui valide la creation

## Cahier de test - Gestion des structures

### Ajouter une structure fille

Action	Détail de l'action	Résultat attendu des tests
<a href="https://cd[NumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/default/gestionPM/[clepDependantDuTerritoire]">https://cd[NumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/default/gestionPM/[clepDependantDuTerritoire]</a>	Accède à la page Cliquer sur le bouton "+enfant" sur un territoire externe	la page s'ouvre Une fenêtre avec un formulaire s'ouvre
Accès au formulaire d'ajout	Remplir le formulaire avec un numero SIRET de 14 caractères (existant)	
Ajouter une numero Siret existant	Cliquer sur le résultat apparu précédemment et sur le bouton suivant	Apparition d'un résultat
Accès au formulaire de validation	Cocher "externe" et cliquer sur valider	Apparition d'un nouveau formulaire
Validation		Pop up vert de validation

## Cahier de test - Gestion des structures

### Modification du territoire d'intervention

Action	Détail de l'action	Résultat attendu des tests
<a href="https://cdfnumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/dashboard/structure">https://cdfnumeroDepart]-katalon-php7.ru-neolink.fr/dispositif/dashboard/structure</a>	Accède à la page dashboard d'une structure	la page s'ouvre correctement avec les bonne information afficher
Accès à la carte	Clique sur le lien "Voir la carte de son territoire"	
Accès à la carte choix des territoires	Les bonnes informations passe en "display:none" tandis que la carte s'affiche	
Choix d'un territoire	Clique sur le bouton "Ajouter territoire"	Une nouvelle carte apparait
	Clique sur un territoire sur la carte et sur le bouton ajouter	Pop up vert de validation

## Playwright

```
● ● ●
1 import { mergeTests } from '@playwright/test';
2 import { partenaireDetailFixtures } from '@fixtures/modules/partenaire/detail/partenaire_detail.fixture';
3 import { sessionFixtures } from '@fixtures/sessions/session.fixture';
4
5 const test = mergeTests(
6   partenaireDetailFixtures,
7   sessionFixtures,
8 );
9
10 test.describe('Parcours Modification structure', () => {
11   test('Modification structure', async ({{
12     page,
13     dataUserGestionStructureModifInfoStructure,
14     sessionPage,
15     partenaireDetailPage,
16   }}) => {
17     test.slow();
18
19     const { users_1 } = dataUserGestionStructureModifInfoStructure;
20
21     for (const admin of users_1) {
22       try {
23         // Étape 1 : Connexion avec l'utilisateur actuel
24         await test.step(`Environnement: ${admin.environnement}, Plateforme: ${admin.plateforme}, User: ${admin.email}`, async () => {
25           await sessionPage.naviguer(page, admin);
26           await sessionPage.connexion(page, admin.email, admin.password);
27         });
28
29         // Etape 2 : Modifier le formulaire de modification
30         await partenaireDetailPage.modifierFormulaireStructure(page, admin, 17958873);
31
32         // Etape 3 : Déconnexion
33         await sessionPage.deconnexion(page);
34       } catch (userError) {
35         console.error(`Échec pour l'utilisateur ${admin.email}:`, userError);
36         // Capturer une capture d'écran de l'erreur
37         const userErrorScreenshot = await page.screenshot();
38         test.info().attachments.push({
39           name: `error_${admin.email.replace('@', '-at-')}`,
40           contentType: 'image/png',
41           body: userErrorScreenshot,
42         });
43       }
44     });
45   });
46 });
47
```

## Autre

```

> $demandes = array(2)
  ↘ 0 = array(7)
    clep = "1743893"
    per_clep = "544415"
    meta_per = "1964038"
    nom = "DUMAS"
    prenom = "Guillaume"
    num_allocataire = "9700000544415"
    motif = "Réouverture de droit suit...
  ↘ 1 = array(7)
    clep = "1826494"
    per_clep = "3798756"
    meta_per = "3681478"
    nom = "WEBER"
    prenom = "Dylan"
    num_allocataire = "9700003798756"
    motif = "Réouverture de droit suit...

```

Afficher les données   Afficher la structure   Modifier la table   Nouvel élément

