

Documentation de stage

Mardi 04/01

Arrivé : 9h00

Récupération de nos badges puis rencontre avec Monsieur Laurent JUSTE.

9h30 : Récupération et configuration du poste de travail.

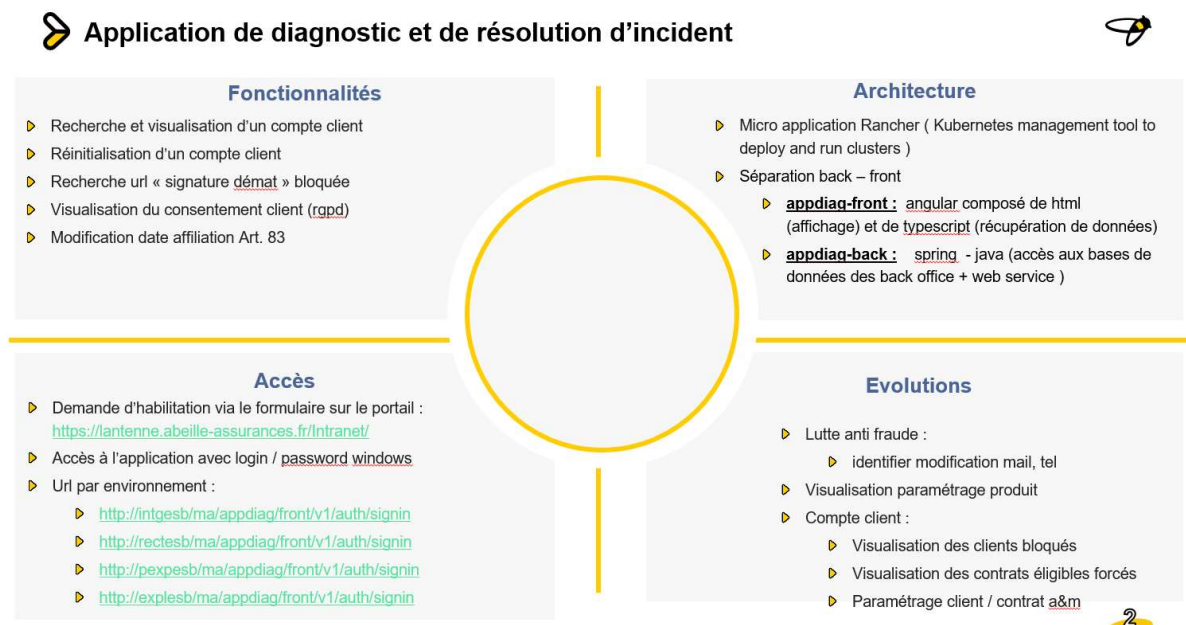
12h10 : Réunion avec Etienne et M. JUSTE.

14h00 : Pause et recherche

Spring Boot est un framework qui facilite le développement d'applications fondées sur **Spring** en offrant des outils permettant d'obtenir une application packagée en JAR, totalement autonome.

En informatique, un fichier JAR est un fichier ZIP utilisé pour distribuer un ensemble de classe Java. Ce format est utilisé pour stocker les définitions des classes, ainsi que des métadonnées, constituant l'ensemble d'un programme.

14h30 : Réunion avec Thierry CREPLET pour nous expliquer nos missions et présenter les différentes applications.



Formulaire de recherche « compte client »



obeille
ASSURANCES

Bienvenue Se deconnecter

Votre espace de recherche

Recherche Client Recherche Lien Art83 GDPR

Recherche d'un client

Numéro client :

Mail client :

Identifiant annuaire (Tam_Id) :

ID person internet (Pri_Id) :

Identifiant A&M (Pri_Username) :

Numéro de contrat :

Search Afer Search Aviva Lancer la recherche

► Ce formulaire permet de rechercher un compte client en base de donnée à partir de multiples critères :

- Numéro de client
- Mail client
- Tam Id (identifiant annuaire ADLDS)
- Pri_id (identifiant Bdd)
- Identifiant client
- Numéro de contrat

► Le bouton Search Aviva appelle le back office du Dommage à partir d'un numéro de contrat abeille.

► Les boutons Search Afer interroge la base de donnée CRM à partir d'un numéro de contrat afer.

3

Résultat de recherche « compte client »



Resultats

Prs Id	Nom Client	Prénom Client	Identifiant A&M	TAM_ID A&M	PRI_ID A&M
2 -9994415359	KLECZEWSKI	MIGA	tcrolept	6000000120	-9998278311 1

1 (current) 4 items per page

Resultats

Prs Id	Nom Client	Prénom Client	Identifiant A&M	TAM_ID A&M	PRI_ID A&M
2 -9999345784	AUBIN	PHILIPPE			

1 (current) 4 items per page

► 1 : si présence d'un pri_id, cliquer sur le numéro sous le pri_id pour accéder au détail du compte client.

► 2 : en cas d'absence du pri_id, cliquer sur le numéro en dessous du prs_id pour accéder au détail client / contrat.

4

Détail du compte client

Détail du compte Client

Etat du compte client en base de donnée

Identifiant : tcreplet
PersonInternet ID : -9998278311
PersonInternet Tam ID : 6000000120
Date de création du compte : 06/09/2022 15:17:50

Retrieve Aviva Retrieve Afer Lecture ADLDS

Information compte : cf Jle

Etat du compte client dans l'annuaire ADLDS

Suppression compte

Détail des Connexions

Téléchargement de l'historique de connexion

Suppression du compte en base de donnée CRM, et dans l'annuaire ADLDS.
Cette double suppression permet de recréer le compte avec le même identifiant.

Chaque suppression est enregistrée en base de donnée dans la table : CRM.fe_purge_person_internet, y compris historique des connexions.

Feuille de calcul Query Builder

```
select * from CRM.fe_purge_person_internet
```

50 lignes extraites en 0,185 secondes

	PPL_ID	PPL_TAM_ID	PPL_PRI_ID	PPL_USERNAME	PPL_CREATETMS	PPL_LISTE_NUM_OLI	PPL_LISTE_NUM_CRT	PPL_CREATEUSER_LOAD	PPL_PFLUX
1	-9999999993	9955007518	-999796444	aligierie91	23/02/22	04480257	142018267	106449242	5041104
2	-9999999999	9900168750	-999001199	belapierre@wanado.fr	04/04/22	07549002	04071737	124493442	5041104
3	-9999999939	9929022216	-9990305143	709992010	04/04/22	6997402	6997932	101102737...	0140369102
4	-9999999479	9977836197	-9990352914	Matheno20	09/05/22	850555942	109312597	30200030906	CS400...

5

Détail client

Détail des Connexions

Détail des Clients

8380034 – AFER_GT MIGA KLECZEWSKI

Statut personne : Client
Civilité : Madame
Nom : KLECZEWSKI
Prénom : MIGA
Date de naissance : 30/05/1960
Numéro de téléphone : 0603126830
Email : cklec@aol.com

eDoc : Oui

date de mise à jour eDoc : Aucune modifications eDoc
date de modification de l'email : Aucune modifications de mail
statut email : Aucun statut

Numéro d'adhérent : 0010330850
Numéro de client GT (Clt Id) : 8380034
Numéro de client Prévoyance (Clt Fils Id) :
Reference : 8380034
Origine du client : AFER_GT
Date du compte provisoire :
Code inter : 713056 **DérogatPat**
Code réseau : 42 **Webcl Client**

Dérogation produit : cf Jle

Info détail client via ws GT

Détail des Contrats

9179185 -- AFER_MONO

13442967 -- AFER_MULTI

A noter qu'une liste de plusieurs clients est possible, par exemple pour un client dommage ayant de l'auto, de la vie et de la prévoyance, on peut avoir 3 lignes clients.

6

Détail contrat



eDoc : Oui

Code réseau : 42

Webclic Client

date de mise à jour eDoc : Aucune modifications eDoc
date de modification de l'email : Aucune modifications de mail
statut email : Aucun statut

Détail des Contrats

9179185 -- AFER_MONO

Numéro de contrat : 9179185
Origine du contrat : AFER_MONO
Eligibilité contrat : DANS LE PERIMETRE, CONSULTATION COMPLETE
Status contrat : En cours
Date d'effet du contrat : 28/05/1993
Situation du contrat : contrat

Code Produit : N08
Date GDPR :
Statut GDPR (P.A.R) :
Valeur de rachat : 142007
Date de sortie :

Webclic_Contrat

Info détail contrat via ws
GT

Tpcli_Mc

Info détail contrat via ws
GT

13442967 -- AFER_MULTI

Numéro de contrat : 13442967
Origine du contrat : AFER_MULTI
Eligibilité contrat :
Status contrat : Clôturé
Date d'effet du contrat : 11/04/2001
Situation du contrat : racheté

Code Produit : N09
Date GDPR :
Statut GDPR (P.A.R) :
Valeur de rachat : 0
Date de sortie : 01/08/2006 00:00:00

Webclic_Contrat

Tpcli_Mc

7

Recherche lien signature démat



Bienvenue

Votre espace de recherche

Recherche Client Recherche Lien Art63 GDPR

Recherche d'un Lien

Numéro travail :

1094760245

Lancer la recherche

URL Client Afer:
https://adherent.gie-afer.fr/hub/index.jsp?ssoTicket=9201ae6f-86e3-4daf-8699-c2be195524a9

URL Client Aviva:
https://www.aviva.fr/hub/?ssoTicket=9201ae6f-86e3-4daf-8699-c2be195524a9

Résultat de la recherche

Permet de copier le lien

➤ A partir de la saisie d'un numéro de travail, cette fonction permet de retrouver le lien de signature démat dans la table CRM_fe_suivi_acte icy afin de débloquent la situation.

➤ A noter, il y a 2 liens de proposés, Afer et Abeille. L'utilisateur prendra celui correspondant à son contexte, pour le jouer dans un navigateur.

8

Modification date d'affiliation Art. 83



Votre espace de recherche

Recherche Client Recherche Lien Art83 GDPR

Recherche d'un Lien

Numéro d'encaissement :

Lancer la recherche

Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

1 (current) 4 items per page

Changer la date d'affiliation

lu	ma	me	je	ve	sa	di
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Valider la date

➤ A partir de la saisie d'un numéro d'encaissement, cette fonction permet de retrouver les affiliés du contrat, mais surtout de vérifier la date d'affiliation erronée.

➤ La page de résultat propose un calendrier permettant de modifier cette date et de la valider.

➤ On peut ensuite relancer la recherche pour vérifier la prise en compte de la nouvelle date.

Résultat de la recherche

9

Recherche consentements client



Votre espace de recherche

Recherche Client Recherche Lien Art83 GDPR

GDPR

Clé client :

Consentement **Historique consentement** **Permission**

```
{
  "personId": "FR-AFER:PPHAER_9209632",
  "platform": "AFER",
  "sourceApplication": "AFER&MOI",
  "question": {
    "consentAviva": false,
    "questionAviva": "GROUPE",
    "consentThirdParty": false,
    "questionThirdParty": "GIE_TIERS"
  },
  "date": "2022-02-21",
  "proof": {
    "type": null,
    "location": null,
    "url": null
  },
  "isFromStock": false
}
```

```
{
  "personId": "FR-AFER:PPHAER_9209632",
  "sourceApplication": "AFER&MOI",
  "date": "2022-02-21",
  "permissionsAviva": {
    "sms": false,
    "post": false,
    "email": false,
    "phone": false,
    "pushNotification": false
  },
  "permissionsThirdParty": {
    "sms": false,
    "post": false,
    "email": false,
    "phone": false,
    "pushNotification": false
  },
  "permissionsMarketingStudies": {
    "sms": false,
    "phone": false,
    "email": false
  }
}
```

➤ A partir de la saisie d'une clé **gdpr**, ces 3 boutons permettent de restituer l'état du consentement du client, avec les médias associés (permission). Le bouton historique permet d'obtenir l'historique des consentements.

➤ La clé **gdpr** est composée d'un préfixe qui fait référence au back office du client, puis on y ajoute le numéro de client.

Voici les différentes clés avec un exemple :

Portail :
FR-PORTAIL:TTCLICLI_51610718

Abeille :
FR-GT:PPH_9123490

Afer :
FR-AFER:PPHAER_9281508

Eurofil :
FR-CIRIARD:PEF01B0_000000001061126

Aviva Direct :
FR-MARK:MKBSIG_4298535002

10

Lors de ce stage, nous aurons accès à certaines plateformes afin de nous diriger ou de nous aider dans nos missions (décrites juste après).



Bitbucket

Connexion

Nom d'utilisateur

Mot de passe

☒ Rester connecté

[Se connecter](#) [Impossible d'accéder à votre compte ?](#)

Cette gestion des dépôts Git est procurée par Atlassian Bitbucket

[Atlassian Bitbucket v5.10.2](#) · [Documentation](#) · [Contacter le service d'assistance](#) · [Demander une fonctionnalité](#) · [À propos de](#) · [Contacter Atlassian](#)

ATLASSIAN

Bitbucket Projets Dépôts

Rechercher du code, des commits ou des dépôts

Espaces de stockage

Filter les répertoires par nom

Nom

- appdiag-back
- appdiag-back-envs
- appdiag-front
- appged-back
- client_service_back
- client_service_back-envs
- ged-documents-back
- ged-documents-back-envs
- hub-direct-back-envs
- hub-direct-esign-front
- hub-direct-esign-front-envs
- hub-dommage-back-envs

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions Git. Il s'agit d'un service freemium dont la version gratuite permet déjà de créer jusqu'à un nombre illimité de dépôts privés, accessibles par cinq utilisateurs au maximum.

Il permet aux équipes de développeurs de disposer d'une plateforme commune pour planifier des projets, effectuer un travail collaboratif autour du code, mais aussi tester et déployer.

Bitbucket dispose des fonctionnalités suivantes :

- **Collaboration sur le code** : vous pouvez approuver la revue de code avec les pulls requests, créer une liste de contrôle des merges avec des approbateurs désignés, et mener des discussions directement dans le code source.
- **Livraison continue** : vous bénéficiez de la centralisation du développement, des tests et du déploiement avec la CI/CD intégrée ainsi que de la configuration sous forme de code et de boucles de feedback accélérées.
- **DevSecOps** : vous obtenez une visibilité en temps réel sur les problèmes de sécurité dans votre code et vos conteneurs. Vous pouvez identifier les corrections de vulnérabilité aux premiers stades du développement et surveiller les risques après déploiement.
- **Définition des rôles** : vous avez la possibilité de limiter l'accès à des utilisateurs et de contrôler leurs actions grâce aux autorisations de branche.





Jenkins






Welcome to Jenkins!


Sign in


☐ Keep me signed in





  KATIA BERTAUX  SE DÉCONNECTER


Jenkins


 Utilisateurs


 Historique des constructions


 Relations entre les builds


 Vérifier les empreintes numériques


 Bulk Builder

 Job Import Plugin

 Mes vues

 Open Blue Ocean

 Utilisation du disque




Supporte JAVA 8/9 & JAKARTA EE

Portable et sans droits ADMIN

Serveurs et outils embarqués

Projets installés et IDE clef en main





JAKARTA EE

Download DEX SHEET

Plateforme CICD [staging] d'Intégrations & Livraisons Continues

CI/CD de la DSI Aviva France

DevSecOps by IT-LABS Aviva |  : bernard.chatin@abeille-assurances.fr |  : <http://cicd-prod>

Splunk new URLS >> IN : <http://intgsplunk> | RC : <http://rectsplunk> | PX : <http://pexpsplunk> | EX : <http://explsplunk>

AVIVAFR-MA

DIGFAC

Tableau de bord Jenkins

Test Statistics Chart

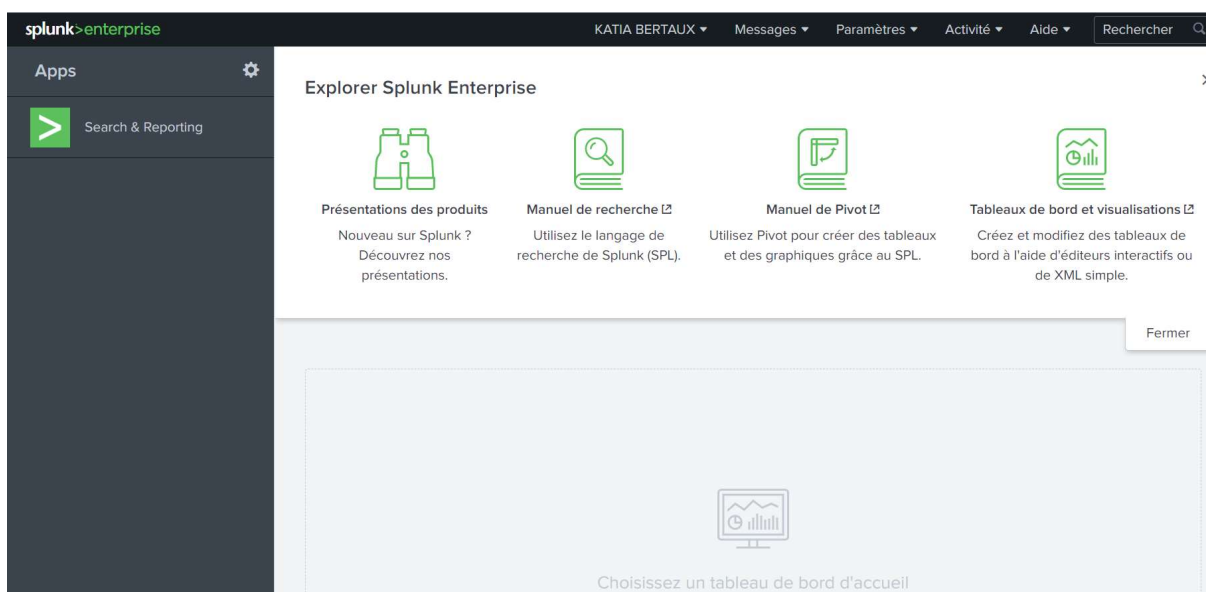
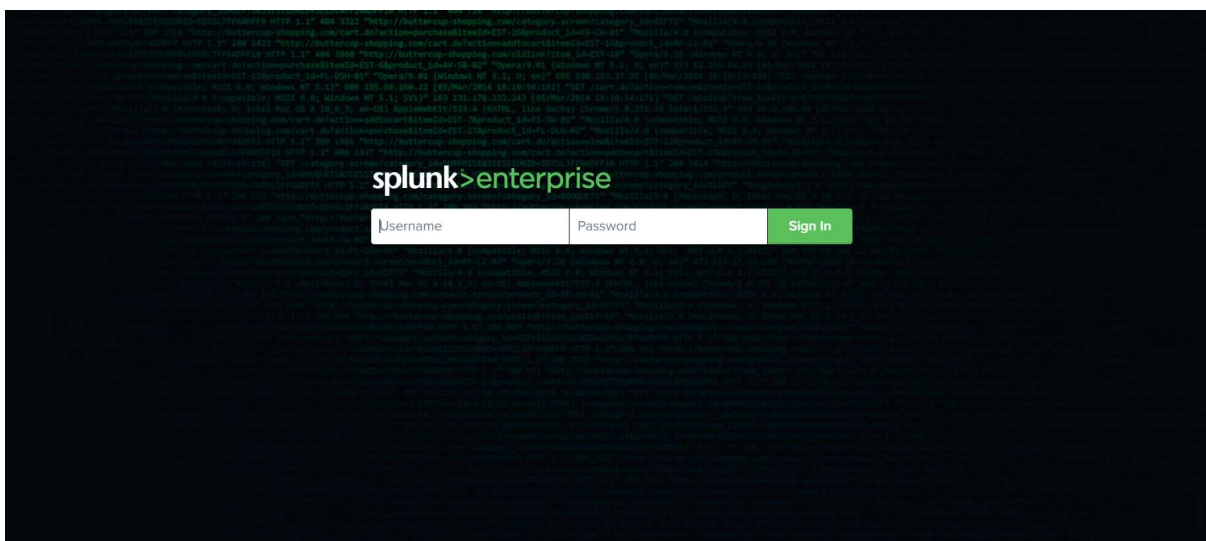
Test Trend Chart

8

Classification : Public

Jenkins est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement logiciel liées au build, aux tests et au déploiement, et facilite l'intégration continue et la livraison continue.




Développé en Java. A chaque modification de code d'une application dans le gestionnaire de configuration, **Jenkins** se charge automatiquement de la recompiler, et de la tester.



Splunk fait partie des plateformes logicielles largement utilisées pour **surveiller, rechercher, analyser et visualiser les données générées par les machines en temps réel**. Ce logiciel a la capacité de lire diverses informations, qu'elles soient non structurées, semi-structurées ou rarement structurées.

Il **capture, indexe et corrèle les data en temps réel dans un conteneur interrogeable**. Ensuite, cette plateforme produit des graphiques, des alertes, des tableaux de bord et des visualisations. Cette opération permet un **accès facile aux données dans toute l'organisation**. De ce fait, le diagnostic et la résolution de divers problèmes d'entreprise deviennent plus aisés.

Jira Software

 Tableaux de bord [Créer](#)   Connexion

Pour toute demande d'habilitation JIRA/Confluence merci de cliquer sur le lien suivant : [Demande Habilitation](#) , Pour toute autre demande : [Demande ACCA](#)

Bienvenue dans Jira Abeille Assurances

Nom d'utilisateur





Mot de passe

☐ Retenir ma connexion sur cet ordinateur

Pas encore membre ? Pour demander un compte,
Veuillez contacter vos administrateurs Jira.

[Connexion](#)

[Vous ne pouvez pas accéder à votre compte ?](#)

 Tableaux de bord [Projets](#) [Tickets](#) [Tableaux](#) [Calendar](#) [Créer](#)   

System Dashboard

Introduction

Bienvenue sur Jira Abeille Assurances
Vous découvrez Jira ? Consultez le Guide utilisateur de Jira.




Qui me sont assignés

Aucun ticket ne vous est pour l'instant assigné. Passez une bonne journée !

Flux d'activité

Your Company JIRA

Aujourd'hui

-  **[REDACTED]** a joint un fichier à JOP-49848 - Signature CSR pour l'environnement SAAS SMAX DEV par sectigo
• dev_esm_abeille-assurances_fr.cer
05/01/2023 11:24 [Comment](#) [Vote](#) [Watch](#)
-  **[REDACTED]** a changé l'état à TECHNICAL VERIFICATION sur JOP-49840 - provocation de Failover des Clusters FW : vmpalofw000, ADC2FWO-SI1PL01, ADC2FWO-SI1CL01 pour test d'envois des logs des Fw Standby
05/01/2023 11:24 [Comment](#) [Vote](#) [Watch](#)
-  **[REDACTED]** a changé l'état à Traitement : Terminé sur IN-66831 - Incident sur mise à jour des coordonnées bancaires du contrat 13118344
Il y a quelques instants [Comment](#) [Vote](#) [Watch](#)

À l'origine, **Jira** a été pensé comme un **outil** de suivi des bugs et des tickets. Mais aujourd'hui, **c'est** devenu un puissant **outil** de gestion du travail pour toutes sortes de cas d'usage, de la gestion des exigences et des cas de test au développement logiciel Agile.

Nous avons donc 2 missions à accomplir pour ce stage (pour le moment)

- **User de recherche de changement d'email et/ou téléphone mobile**

Contexte :

Dans le cadre de la lutte anti-fraude, lorsqu'un email ou un numéro de téléphone mobile est modifié, l'application concernée par cette modification envoie un sms ou un mail sur l'ancien numéro / mail et sur le nouveau numéro / mail.

Cela engendre des appels d'inquiétude de personnes qui ne sont pas à l'origine de la modification, puis de notre service anti-fraude qui nous demande de vérifier.

Ainsi, dans l'application Digital Diag, nous souhaitons **ajouter un nouvel onglet, permettant la recherche d'un numéro de téléphone et/ou d'un email dans certaines tables.**

A noter que cette interface ne concerne que le périmètre Epargne GT Abeille et Afer.

Partie 1 - Interface de recherche :

Ce nouvel onglet sera nommé « Anti-fraude »

Sur cette page de recherche, nous avons besoin dans l'immédiat de 3 zones :

- Numéro de mobile
- Email
- Numéro de client

abeille ASSURANCES

Bienvenue [Se deconnecter](#)

Votre espace de recherche

Recherche Client Recherche Lien Art83 GDPR **Anti-Fraude**

Recherche d'un client

Numéro client :

Mail client :

Numéro mobile :

[Lancer la recherche](#)

Partie 2 - Résultat de la recherche :

Lorsqu'on clique sur le bouton « Lancer la recherche », il faut aller lire dans 3 tables et restituer l'ensemble des champs de ces 3 tables en effectuant **un tri décroissant sur une date**.

Ces tables sont des sas de transit entre les Fronts CRM et le back office Epargne. On les appelle aussi, les tables des « Données Partagées ». Toutes les 15 minutes, un batch est lancé coté front et coté back office pour aller lire les mouvements et mettre à jour les bases de données.o

Le plus simple et pertinent est de saisir un seul champ. Voici les requêtes à implémenter.

Une limite de 100 lignes est arbitrairement fixée pour ne pas remonter trop de données anciennes.

Partie 2.1 - Table front N°1 : CRM.fe_mvcli

Requête Numéro de client :

```
select * from CRM.fe_mvcli where mvt_cli_id = 10112666 and rownum < 100  
order by mvt_enr_d desc ;
```

Requête Email :

```
select * from CRM.fe_mvcli where mvt_mvt_c = 'email' and mvt_val like  
'%abeille@aviva.com' and rownum < 100 order by mvt_enr_d desc ;
```

Requête Numéro de mobile :

```
select * from CRM.fe_mvcli where mvt_mvt_c = 'tel_portable' and mvt_val like  
'%0607080910%' and rownum < 100 order by mvt_enr_d desc ;
```

Données à restituer sur la page de résultat :

Résultat de requête x									
Toutes les lignes extraites : 94 en 4,839 secondes									
	MVT_ID	MVT_TRT_D	MVT_ENR_D	MVT_USR_ORI	MVT_CLI_ID	MVT_CLI_FILS_ID	MVT_MVT_C	MVT_VAL	MVT_TRT_BATCH
72	17014538	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	eai2_jurisdiction	(null)	(null)
73	17014539	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	eai3_jurisdiction	(null)	(null)
74	17014540	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	eai1_nif	(null)	(null)
75	17014541	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	eai2_nif	(null)	(null)
76	17014542	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	eai3_nif	(null)	(null)
77	17014543	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	tranche_patrim_fin	2	(null)
78	17014544	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	tranche_patrim_immob	4	(null)
79	17014545	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	tranche_revenus	1	(null)
80	17014546	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	orig_rev_salaire	1	(null)
81	17014547	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	orig_rev_fonciers	0	(null)
82	17014548	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	orig_rev_retraite	0	(null)
83	17014549	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	orig_rev_autres	0	(null)
84	17014550	20/12/22	20/12/22	Mme GUYOMARD CHRISTELLE	10112666	(null)	orig_rev_bic_bnc	0	(null)
85	17014551	20/12/22	20/12/22	GUYOMARD CHRISTELLE	10112666	(null)	code_csp	33	(null)
86	17014552	20/12/22	20/12/22	GUYOMARD CHRISTELLE	10112666	(null)	secteur_activite	04	(null)
87	17014553	20/12/22	20/12/22	GUYOMARD CHRISTELLE	10112666	(null)	type_profession	1	(null)
88	17014554	20/12/22	20/12/22	GUYOMARD CHRISTELLE	10112666	(null)	profession	(null)	(null)

Exemple pour le rendu :

Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

Partie 2.2 - Table front N°2 : *dcce.dcc_mvtcliocce*

Requête Numéro de client :

*select * from dcce.dcc_mvtcliocce where mvt_cli_id = 10112666 and rownum < 100
order by mvt_enr_d desc ;*

Requête Email :

*select * from dcce.dcc_mvtcliocce where mvt_mvt_c = 'email' and mvt_val like
'%abeille@aviva.com' and rownum < 100 order by mvt_enr_d desc ;*

Requête Numéro de mobile :

*select * from dcce.dcc_mvtcliocce where mvt_mvt_c = 'tel_portable' and mvt_val like
'%0607080910%' and rownum < 100 order by mvt_enr_d desc ;*

Données à restituer sur la page de résultat :

Résultat de requête x							
SQL 100 lignes extraites en 4,584 secondes							
MVT_MVT_C	MVT_TRT_D	MVT_ENR_D	MVT_ID	MVT_CLI_ID	MVT_UTI_ORI	MVT_VAL	MVT_INT_ABV_C
1 tranche_revenus	20221220	20221220	1265816	10112666	Mme GUYOMARD CHRISTE	1	(null)
2 situat_fam	20221220	20221220	1265864	10112666	Mme GUYOMARD CHRISTE	1	(null)
3 regime_matrimonial	20221220	20221220	1265865	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
4 nombre_enfants	20221220	20221220	1265866	10112666	Mme GUYOMARD CHRISTE	0	(null)
5 nombre_enfants_charge	20221220	20221220	1265867	10112666	Mme GUYOMARD CHRISTE	0	(null)
6 impot_revenu	20221220	20221220	1265868	10112666	Mme GUYOMARD CHRISTE	0	(null)
7 impot_fortune_immo	20221220	20221220	1265869	10112666	Mme GUYOMARD CHRISTE	0	(null)
8 fatca_code	20221220	20221220	1265870	10112666	GUYOMARD CHRISTELLE	(null)	(null)
9 fatca_citoyen	20221220	20221220	1265871	10112666	Mme GUYOMARD CHRISTE	Non	(null)
10 fatca_oblig_fisc	20221220	20221220	1265872	10112666	Mme GUYOMARD CHRISTE	Non	(null)
11 fatca_identifiant	20221220	20221220	1265873	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
12 eai1_jurisdiction	20221220	20221220	1265874	10112666	Mme GUYOMARD CHRISTE	FR	(null)
13 eai2_jurisdiction	20221220	20221220	1265875	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
14 eai3_jurisdiction	20221220	20221220	1265876	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
15 eai1_nif	20221220	20221220	1265877	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
16 eai2_nif	20221220	20221220	1265878	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
17 eai3_nif	20221220	20221220	1265879	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
18 tranche_patrim_fin	20221220	20221220	1265880	10112666	Mme GUYOMARD CHRISTE	2	(null)

Exemple pour le rendu :

Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

Partie 2.3 : Table back office N°1 : *dcce.pc_bp_contact*

Requête Numéro de client :

```
select * from dcce.pc_bp_contact where numero_pers = 10112666 and rownum < 100
order by numero_mvt desc;
```

Requête Email :

```
select * from dcce.pc_bp_contact where nom_zone = 'email' and valeur_zone like
'%abeille@aviva.com' and rownum < 100 order by numero_mvt desc;
```

Requête Numéro de mobile :

```
select * from dcce.pc_bp_contact where nom_zone = 'tel_portable' and valeur_zone
like '%0607080910%' and rownum < 100 order by numero_mvt desc;
```

Données à restituer sur la page de résultat :

Résultat de requête x							
SQL 100 lignes extraites en 4,584 secondes							
MVT_MVT_C	MVT_TRT_D	MVT_ENR_D	MVT_ID	MVT_CLI_ID	MVT_UTI_ORI	MVT_VAL	MVT_INT_ABV_C
1 tranche_revenus	20221220	20221220	1265816	10112666	Mme GUYOMARD CHRISTE	1	(null)
2 situat_fam	20221220	20221220	1265864	10112666	Mme GUYOMARD CHRISTE	1	(null)
3 regime_matrimonial	20221220	20221220	1265865	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
4 nombre_enfants	20221220	20221220	1265866	10112666	Mme GUYOMARD CHRISTE	0	(null)
5 nombre_enfants_charge	20221220	20221220	1265867	10112666	Mme GUYOMARD CHRISTE	0	(null)
6 impot_revenu	20221220	20221220	1265868	10112666	Mme GUYOMARD CHRISTE	0	(null)
7 impot_fortune_immo	20221220	20221220	1265869	10112666	Mme GUYOMARD CHRISTE	0	(null)
8 fatca_code	20221220	20221220	1265870	10112666	GUYOMARD CHRISTELLE	(null)	(null)
9 fatca_citoyen	20221220	20221220	1265871	10112666	Mme GUYOMARD CHRISTE	Non	(null)
10 fatca_oblig_fisc	20221220	20221220	1265872	10112666	Mme GUYOMARD CHRISTE	Non	(null)
11 fatca_identifiant	20221220	20221220	1265873	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
12 eail_jurisdiction	20221220	20221220	1265874	10112666	Mme GUYOMARD CHRISTE	FR	(null)
13 eai2_jurisdiction	20221220	20221220	1265875	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
14 eai3_jurisdiction	20221220	20221220	1265876	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
15 eail_nif	20221220	20221220	1265877	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
16 eai2_nif	20221220	20221220	1265878	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
17 eai3_nif	20221220	20221220	1265879	10112666	Mme GUYOMARD CHRISTE	(null)	(null)
18 tranche_patrim_fin	20221220	20221220	1265880	10112666	Mme GUYOMARD CHRISTE	2	(null)

Exemple pour le rendu :

Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

- **User référentiel produit**

Contexte :

Nous sommes régulièrement sollicités pour fournir des informations diverses sur un produit.

L'idée de cette évolution est de **mettre à disposition une fiche produit complète**. Cela permettra également d'en vérifier le paramétrage.

Pour ce faire, on va s'appuyer sur 3 tables : fe_parameter, fe_series_ic et fe_policydict

Partie 1 - Interface de recherche :

Ce nouvel onglet sera nommé « Référentiel Produit »

Sur cette page de recherche, nous avons besoin d'une zone :

- Numéro de série

Bienvenue

Votre espace de recherche

Recherche Client
Recherche Lien
Art83
GDPR
Référentiel produit

Recherche d'un produit

Numéro de série

Lancer la recherche

Partie 2 - Résultat de la recherche :

Lorsqu'on clique sur le bouton « Lancer la recherche », il faut aller lire dans 3 tables et restituer certains champs qui sont spécifiés ci-après dans la requête.

Voici les requêtes à implémenter en fonction des infos recherchées.

Partie 2.1 – Eligibilité de la Série

Requête FE SERIES IC :

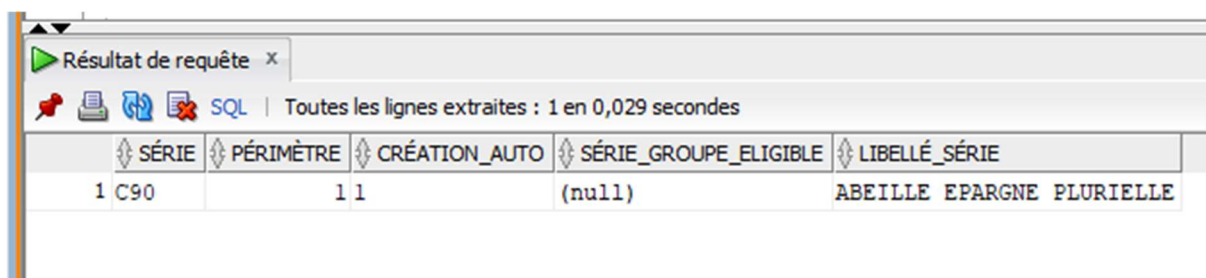
```
select ic_num_ser Série, ic_niveau Périmètre, ic_auto Création_auto, ic_groupe
Série_Groupe_Eligible, key_longlabel Libellé_Série
from crm.fe_series_ic, crm.fe_parameter
where key_class = 'POLICYDICT_OWNED'
and key_language = 'FR'
and ic_num_ser = key_shortlabel
and ic_num_ser = 'C90';
```

La jointure vers fe_parameter permet d'obtenir le libellé.

Il faut restituer les champs du select. En théorie, une ligne est attendue, **mais prévoir l'affichage de plusieurs lignes.**

Si la requête ne remonte aucune ligne, il faut afficher le message suivant :

« Série non trouvée, ou non éligible. »



SÉRIE	PÉRIMÈTRE	CRÉATION_AUTO	SÉRIE_GROUPE_ELIGIBLE	LIBELLÉ_SÉRIE
1 C90	1 1	(null)	(null)	ABEILLE EPARGNE PLURIELLE

Exemple pour le rendu en remplaçant par les champs du select :



Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

Partie 2.2 – Référentiel produit

Requête FE POLICYDICT :

```
select pod_l3_producttype Série,  
key_longlabel Libellé_Série,  
pod_l1_group Série_Groupe,  
pod_pacte_f Pacte,  
pod_fis_pacte_f Fis_Pacte,  
pod_mono_f Mono,  
pod_multi_f Multi,  
pod_ecplus_f Ecplus_f,  
pod_crt_gsm Gsm,
```

```

pod_crt_collectif Collectif
from crm.fe_policydict, crm.fe_parameter
where pod_l3_producttype = 'D12'
and key_class = 'POLICYDICT_OWNED'
and key_language = 'FR'
and pod_l3_producttype = key_shortlabel
;

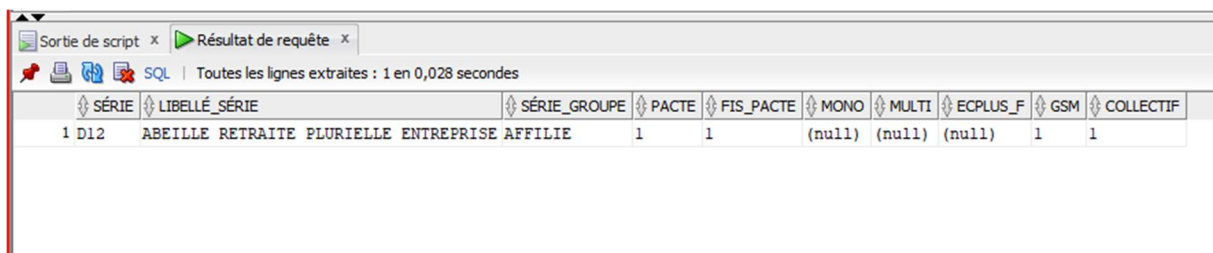
```

La jointure vers fe_parameter permet d'obtenir le libellé.

Il faut restituer les champs du select. En théorie, une ligne est attendue, mais prévoir **l'affichage de plusieurs lignes**. Exemple Série NPA.

Si la requête ne remonte aucune ligne, il faut afficher le message suivant :

« Série non trouvée. »



Sortie de script x Résultat de requête x

Toutes les lignes extraites : 1 en 0,028 secondes

SÉRIE	LIBELLÉ_SÉRIE	SÉRIE_GROUPE	PACTE	FIS_PACTE	MONO	MULTI	ECPLUS_F	GSM	COLLECTIF
1 D12	ABEILLE RETRAITE PLURIELLE ENTREPRISE AFFILIE		1	1	(null)	(null)	(null)	1	1

Exemple pour le rendu en remplaçant par les champs du select :

Nom	Prénom	Affiliation
BOURRIN	YOHAN	01/12/2022
BOURRIN	MATTHIEU	01/12/2022

Partie 2.3 : Export Séries éligibles

Il nous est souvent demandé la liste des séries éligibles.

Nous allons proposer un bouton téléchargement de la table correspondante : fe_series_ic

Requête export table fe_series_ic :

```
select ic_num_ser Série, ic_niveau Périmètre, ic_auto Création_auto, ic_groupe  
Série_Groupe_Eligible, key_longlabel Libellé_Série  
from crm.fe_series_ic, crm.fe_parameter  
where key_class = 'POLICYDICT_OWNED'  
and key_language = 'FR'  
and ic_num_ser = key_shortlabel  
order by ic_num_ser ;
```

Ajouter le bouton télécharger et le texte « Export séries éligibles »

The screenshot shows the Abeille Assurances website interface. At the top, there is a header with the logo and the text 'Bienvenue'. Below this is a section titled 'Votre espace de recherche'. Inside this section, there are several tabs: 'Recherche Client', 'Recherche Lien', 'Art83', 'GDPR', and 'Référentiel produit'. The 'Référentiel produit' tab is selected. Below the tabs, there is a search bar labeled 'Recherche d'un produit'. To the right of the search bar, there is a button labeled 'Export séries éligibles' with a download icon. At the bottom right of the search area, there is a yellow button labeled 'Lancer la recherche'.

L'export doit être au format csv, séparateur ';'

Nom du fichier : « Export series IC.csv »

17h00 : Réunion avec Adil BONO afin d'installer les applications métiers et les configurer sur IntelliJ IDEA 2022.2 + Explication du fonctionnement de l'entreprise au niveau informatique.



IntelliJ IDEA est un *environnement de développement intégré (IDE)* pour les langages JVM (Java, Kotlin, Scala et Sensationnel) conçu pour maximiser la productivité des développeurs. Il effectue les tâches routinières et répétitives pour vous en fournissant une complétion de code intelligente, une analyse de code statique et des refactorisations, et vous permet de vous concentrer sur le bon côté du développement logiciel, ce qui en fait une expérience non seulement productive mais aussi agréable.

Un **environnement de développement intégré (IDE)** est une application logicielle qui aide les programmeurs à développer efficacement le code logiciel.

Un **Framework** est une **boîte à outils pour un développeur web**. Frame signifie cadre et work se traduit par travail. Un Framework contient des composants autonomes qui permettent de faciliter le développement d'un site web ou d'une application.

Quelle est la différence ?

Un **IDE** est un notepad amélioré, un éditeur de texte et pleins d'outils pour faciliter des tâches comme la compilation ou le débogage. Un IDE n'est pas nécessairement associé à un langage, exemple Notepad+ qui peut être utilisé pour à peu près tous les langages.

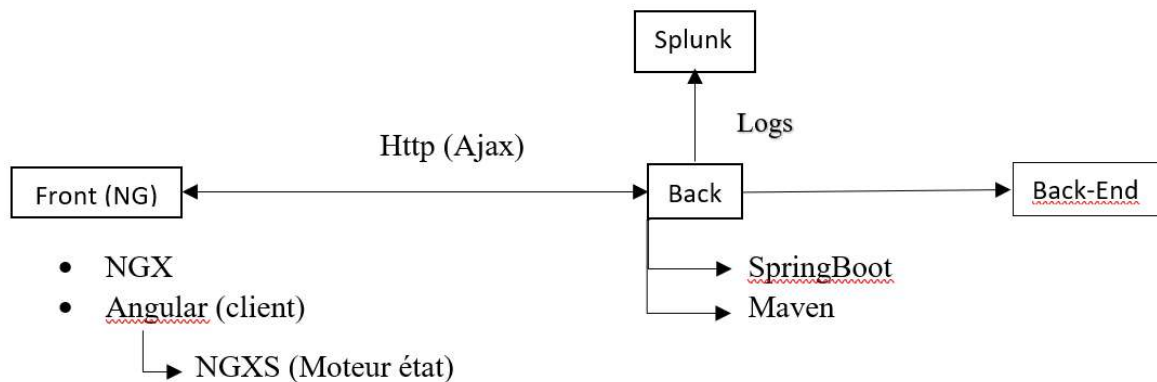
Il s'agit donc d'un ensemble d'outils (comme l'éditeur de texte, un compilateur, le débogueur, etc) destiné à vous "prendre par la main" pendant l'ensemble du processus de développement, du lancement du projet ("je veux créer un nouveau projet") à l'obtention du "produit fini" (mon projet fonctionne, je l'ai testé, je peux le diffuser).

Un **Framework** est un ensemble de classes et de structures écrit dans un langage précis pour un langage précis et destiné à faciliter le développement dans ce langage.

Il regroupe un ensemble d'implémentations de différentes fonctionnalités "clé en main", que vous pourrez — dans une certaine mesure — adapter à vos besoins, et que l'on pourrait comparer — en première approximation — à l'ensemble des pièces qui devraient composer une voiture.

AppDiag

Plusieurs environnements : - INTG pour Intégration
- RECT pour Recette
- PEXP pour PreProd
- EXPL pour Prod



Le Ngx-Bootstrap est une bibliothèque très populaire pour utiliser des composants bootstrap dans des projets basés sur Angular. Il contient presque tous les composants de base de Bootstrap. Les composants ngx-bootstrap sont par conception modulaires, extensibles et adaptables.

NGXS est un modèle de gestion d'état + une bibliothèque pour Angular. Il agit comme une source unique de vérité pour l'état de votre application, fournissant des règles simples pour les mutations d'état prévisibles.

NGXS est calqué sur le modèle CQRS couramment implémenté dans des bibliothèques telles que Redux et NgRx, mais réduit le passe-partout en utilisant des fonctionnalités TypeScript modernes telles que des classes et des décorateurs.

AJAX (*Asynchronous JavaScript* + *XML*) n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes, dont : [HTML](#) ou [XHTML](#), [CSS](#), [JavaScript](#), [DOM](#), [XML](#), [XSLT](#), et surtout [l'objet XMLHttpRequest](#). Lorsque ces technologies sont combinées dans le modèle AJAX, les applications web sont capables de réaliser des mises à jour rapides et incrémentales de l'interface utilisateur sans devoir recharger la page entière dans le navigateur. Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisatrice ou de l'utilisateur.

Bien que le X de AJAX signifie XML, c'est le format [JSON](#) qui est le plus souvent utilisé aujourd'hui à la place de XML, du fait de sa proximité avec JavaScript et de sa légèreté par rapport à XML. Autrement dit, tant JSON que XML peut être utilisés comme format afin d'empaqueter des données en AJAX.

Extensible Markup Language (XML) permet de définir et de stocker des données de manière à pouvoir les partager. XML prend en charge l'échange d'informations entre des systèmes informatiques tels que les sites web, les bases de données et les applications tierces. Les règles prédéfinies facilitent la transmission des données sous forme de fichiers XML sur n'importe quel réseau, car le destinataire peut utiliser ces règles pour lire les données avec précision et efficacité.

Apache **Maven** est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. Maven est géré par l'organisation Apache Software Foundation.

Avec la plate-forme **Splunk**, nous pouvons rechercher rapidement et facilement nos logs. Nous pouvons également apporter de légères modifications au code pour améliorer la lisibilité de nos logs dans l'ensemble de l'organisation.

Spring Boot est un framework qui facilite le développement d'applications fondées sur **Spring** en offrant des outils permettant d'obtenir une application packagée en JAR, totalement autonome

Spring est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests.

Angular est un framework Javascript qui permet de réaliser des applications WEB Cross Platform : Web, mobile et desktop. Il est possible de développer sur **Angular** soit en Javascript natif, soit en Dart (un langage développé par Google), soit en TypeScript (un langage développé par Microsoft). C'est un framework pour clients, open source, basé sur TypeScript.

TypeScript est un langage fortement typé qui permet d'avoir un code plus structuré et plus facile à maintenir. Il se repose sur le langage JavaScript en ajoutant la possibilité de typer de manière statiques les variables et les fonctions. Le code peut être ensuite convertit en JavaScript en retirant les déclarations liées aux types.

Mercredi 04/01

9h30 : Recherche et mise à jour de la documentation

12h00 : Réunion avec Adil, Thierry et Etienne pour savoir où nous en étions dans le téléchargement des applications métiers et fin des téléchargements.

13h50 : Autonomie, prise en main des applications (tentative)

Angular est un framework pour créer des Single Pages Application (SPA)

Une Single Page Application (SPA) est une application qui fonctionne dans un navigateur sans que l'utilisateur n'ait besoin de recharger la page.

Approche Composant Angular

Les applications Angular sont faites des composants (components).

Un composant est une combinaison entre un template HTML et une classe composant qui contrôle une portion de l'écran.

app.component.ts

```
import {Component} from '@angular/core';
import {VnSpinnerComponent} from '../_coreModule/components/vn-spinner/vn-spinner.component';
import build from '../build';

@Component({
  selector: 'app-module',
  template: '<router-outlet></router-outlet><ng-http-loader [entryComponent]='vnSpinnerComponent'></ng-http-loader>'
})
export class AppComponent {

  public vnSpinnerComponent = VnSpinnerComponent;

  constructor() {
    console.log(
      `Build Info:\n\n` +
      `Build Version: ${build.version}\n` +
      `Build Timestamp: ${build.timestamp}`
    );
  }
}
```

Chaque composant contient un décorateur **@Component** qui prend un objet contenant des métadonnées. Elles contiennent l'emplacement du template HTML et du style pour le composant. Ici, il n'y a que le template.

La propriété **selector** permet d'indiquer à Angular d'afficher le composant à l'emplacement d'une balise html spécifique, ici **<app-module>** dans le fichier **index.html**.

index.html

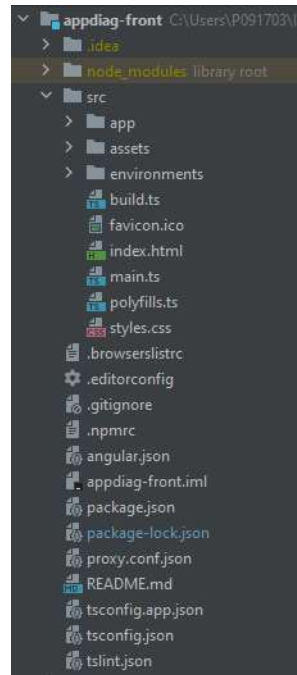
```
<!DOCTYPE html>
<html lang="fr-FR" data-resize-browser="">
<head>
  <title>Espace de signature - Abeille assurances</title>
  <base href="/">
  <link rel="stylesheet" media="all" href="/assets/css/bootstrap.min.css">
  <link rel="icon" type="image/ico" href="/assets/images/icons/abeille-assurances.ico"/>
  <link href="/assets/css/bundle.css" media="all" rel="stylesheet" />

  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
  <meta name="apple-mobile-web-app-capable" content="yes"/>
  <meta name="msapplication-TileImage" content="/assets/images/favicon/aviva-windows-logo.png"/>
  <meta name="msapplication-TileColor" content="#ffd900"/>
  <meta name="title" content="Abeille-assurances"/>
  <meta http-equiv="X-UA-Compatible" content="IE=Edge"/>
</head>
<body class="t3 noSubNav wholePageTabs js no-touch svg">
  <div>
    <div>
      <app-module></app-module>
    </div>
  </div>
</body>
<script src="http://code.jquery.com/jquery-2.1.1.js"></script>
</html>
```

Nous retrouvons bien la balise `<app-module>` dans le fichier `index.html` qui indique où doit être placé le composant.

Comprendre la structure

L'arborescence



Racine ./

.editorconfig : Configuration simple pour l'éditeur pour s'assurer que tous ceux qui utilisent le projet aient la même configuration de base. La plupart des éditeurs prennent en charge ce fichier.

.gitignore : Fichier git pour assurer que les fichiers générés automatiquement ne sont pas pris en compte. Par exemple les dépendances (/node_modules), ainsi que le build du projet (/dist).

.npmrc : Fichiers de configuration npm.

- npm obtient ses paramètres de configuration à partir de la ligne de commande, des variables d'environnement et des npmrcfichiers.
- La npm configcommande peut être utilisée pour mettre à jour et modifier le contenu des fichiers utilisateur et global npmrc.

angular.json : Configuration pour Angular CLI. Vous pouvez par exemple configurer le nom et l'emplacement du folder où sera buildé le projet.

```
"outputPath": "dist/appdiag-front",
```

appdiag-front.iml : Un fichier IML est un fichier de paramètres de module créé par IntelliJ IDEA, un environnement de développement intégré ([IDE](#)) utilisé pour développer des applications Java. Il stocke des informations sur un module de développement, qui est un composant Java, Plugin, Android ou Maven d'une application Java. Les fichiers IML stockent le type, les chemins, les dépendances et d'autres paramètres d'un module.

Les fichiers IML stockent les paramètres au niveau du module pour les projets de développement plus importants. IntelliJ IDEA utilise des fichiers [.IPR](#) ou [.IDEA](#) (selon la version du projet) pour stocker les paramètres généraux du projet. Ces fichiers peuvent référencer plusieurs fichiers IML. Les fichiers IPR et IML utilisent tous deux le formatage XML.

Si vous supprimez accidentellement le fichier IML d'un module, vous pouvez souvent régénérer le fichier en ouvrant et en fermant votre IDE, en reconstruisant votre projet ou en réimportant votre projet.

package.json : Configuration npm listant les paquets tiers que le projet utilise (les dépendances). On peut également ajouter nos propres scripts personnalisés ici.

- **NPM (Node Package Manager)** comme son nom l'indique est le "package manager" officiel de l'univers JavaScript (frontend / backend). Il est installé automatiquement lors de l'installation de **NodeJS**.

package-lock.json : Arbre exacte des dépendances et de leurs propres dépendances, permettant de réinstaller exactement les mêmes versions dans votre équipe.

proxy.conf.json : Un proxy est un logiciel qui se situe entre votre application JavaScript/Angular effectuant la requête Ajax et votre API backend. C'est le choix à faire dans une appli classique.

README.md : Documentation de base pour le projet, pré-remplie d'information de commande CLI. S'assurer de l'améliorer avec la documentation du projet afin que quiconque puisse builder l'application en suivant les instructions contenues dans le fichier !

tsconfig.app.json : Extension de la configuration du compilateur TypeScript pour compiler l'application.

tsconfig.json : Configuration du compilateur TypeScript pour compiler l'application. Ce fichier est également utilisé par l'IDE afin de nous donner des informations utiles (mauvais type d'input par exemple).

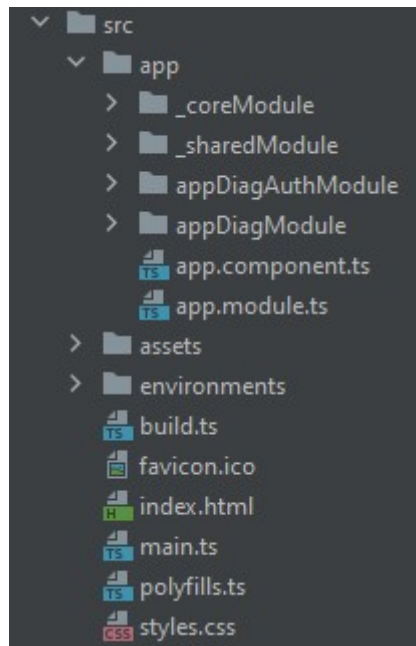
tslint.json : TSLint est un outil d'analyse statique extensible qui vérifie le code [TypeScript](#) pour les erreurs de lisibilité, de maintenabilité et de fonctionnalité. Il est largement pris en charge par les éditeurs et les systèmes de construction modernes et peut être personnalisé avec vos propres règles, configurations et formateurs.

Jeudi 05.01.2023

Node modules

Ce sont les dépendances installées par npm qui correspondent à celles déclarées dans [package.json](#).

Src



app/app.component.(ts, html, scss, spec.ts) : Définit AppComponent avec un template HTML (.html), une feuille de style SCSS ou CSS (.scss ou .css) et un test unitaire (.spec.ts). C'est le composant racine de ce qui deviendra un arbre de composants imbriqués à mesure que l'application évolue.

app/app.module.ts : Définit AppModule, le module racine qui indique à Angular comment assembler l'application. A l'initialisation du projet il déclare uniquement l'AppComponent.

assets/* : Un dossier où l'on peut mettre des images qui seront copiées directement lorsque l'on construit l'application.

favicon.ico : Comme son nom l'indique, il permet de configurer le *favicon* de l'application pour les navigateurs.

index.html : La page HTML principale qui est servie en premier lorsque quelqu'un visite le site. La plupart du temps, on n'aura jamais besoin de le modifier. Le CLI ajoute automatiquement tous les fichiers JS et CSS lors de la création de l'application afin que nous n'ayons jamais besoin d'ajouter manuellement des balises 'script' ou 'link'.

Le CLI d'Angular est un outil d'interface de ligne de commande que vous utilisez pour initialiser, développer, échauder et maintenir des applications angulaires directement à partir d'un shell de commande.

main.ts : Le point d'entrée principal de l'application. Compile l'application avec le compilateur JIT et bootstrap le module racine de l'application (AppModule) pour s'exécuter dans le navigateur.

Le compilateur JIT (Just-In-Time) est un composant de l'environnement d'exécution qui améliore les performances des applications Java™ en repoussant au dernier moment (celui de l'exécution) la compilation des bytecodes en code machine natif.



Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

styles.css : Les styles globaux vont ici. La plupart du temps, les styles locaux sont directement dans les composants pour faciliter la maintenabilité, mais les styles qui affectent l'ensemble de l'application doivent être dans ce fichier.

Les cycles de vie Angular (lifecycle hook)

Les crochets de cycle de vie sont une fonctionnalité spéciale dans Angular qui nous permet de "se connecter" et d'exécuter du code à un événement de cycle de vie spécifique d'un composant ou d'une directive.

Angular gère les composants et les directives pour nous lorsqu'il les crée, les met à jour ou les détruit. Avec les hooks de cycle de vie, nous pouvons mieux contrôler notre application.

Pour ce faire, nous ajoutons des méthodes de hook spécifiques préfixées par `ng` à notre composant ou directive. Ces crochets sont divisés en deux types : les crochets pour les composants ou les directives et les crochets pour les composants enfants.

Voici les crochets pour les composants ou les directives, dans l'ordre d'appel :

1. `constructor()`
2. `OnChanges`
3. `OnInit`
4. `DoCheck`
5. `OnDestroy`

Et voici les crochets pour les composants enfants d'un composant :

1. `AfterContentInit`
2. `AfterContentChecked`
3. `AfterViewInit`
4. `AfterViewChecked`

<code>OnChanges</code>	Appelé lorsque les propriétés d'entrée ont changé
<code>OnInit</code>	Appelé à l'initialisation
<code>DoCheck</code>	Détection de changement personnalisée du développeur
<code>OnDestroy</code>	Appelé avant que le composant ne soit détruit
<code>AfterContentInit</code>	Appelé lorsque le contenu du composant <code>ngContent</code> est initialisé
<code>AfterContentChecked</code>	Appelé lorsque le contenu du composant est mis à jour ou vérifié pour les mises à jour
<code>AfterViewInit</code>	Appelé lorsque la vue projetée du composant a été initialisée
<code>AfterViewChecked</code>	Appelé après la vérification de la vue projetée

Les directives sont techniquement des composants, mais elles fonctionnent sur la partie non-UI du DOM. Leur travail consiste à ajouter des fonctionnalités supplémentaires au DOM.

Le Document Object Model (DOM) est une interface de programmation normalisée par le W3C, qui permet à des scripts d'examiner et de modifier le contenu du navigateur web.

Le DOM est une interface de programmation pour des documents HTML ou XML qui représente le document (la page web actuelle) sous une forme qui permet aux langages de script comme le JavaScript d'y accéder et d'en manipuler le contenu et les styles.

Constructor()

C'est le constructeur que nous connaissons dans les classes JS. Il est inclus car nos composants/directives sont des classes JavaScript avec un décorateur `@Component` ou `@Directive`. Ainsi, ce constructeur est appelé lorsque notre instance de composant/directive est créée par Angular à l'aide du mot- `new` clé.

OnChanges

`OnChanges` est un hook de cycle de vie qui est appelé lorsqu'une propriété liée aux données d'une directive change. Une méthode doit être définie pour gérer les modifications `.ngOnChanges()`.

Mission 1 :

recherche.component.html :

```
<li [ngbNavItem]="5">
  <a ngbNavLink
    style="color: black; border-top-left-radius: 1.5rem; border-top-right-radius: 1.5rem; text-decoration: none">Anti-Fraude</a>
    <ng-template ngbNavContent>
      <div class="pod podContent">
        <div>
          <h2>Recherche d'un client</h2>
        </div>
        <form [formGroup]="rechercheAFForm" action="#">
          <div class="pod podContent text-center">
            <div class="form-group row">
              <label for="mvtCliId" class="col-sm-4">Numéro
client : </label>
              <div class="col-sm-5">
                <input class="form-control" id="mvtCliId"
type="text" name="mvtCliId" formControlName="mvtCliId" vnNumber>
              </div>
            </div>
            <div class="form-group row">
              <label for="mvtVal" class="col-sm-4">Mail client :
</label>
              <div class="col-sm-5">
                <input class="form-control"
type="email" id="mvtVal" name="mvtVal" formControlName="mvtVal" >
              </div>
            </div>
            <div class="form-group row">
              <label for="mvtVal" class="col-sm-4 ">Numéro mobile
: </label>
              <div class="col-sm-5">
                <input class="form-control" type="tel"
id="mvtVal" name="mvtVal" formControlName="mvtVal" vnNumber>
              </div>
            </div>
            <vn-control-message id="errorFields"
[control]="rechercheAFForm"></vn-control-message>
          </div>
          <button class="standard" style="float: right; font-size:
150%;" type="submit" (keydown.enter,keydown.a)="doSearchAF()"
(click)="doSearchAF()">Lancer la recherche</button>
        </form>
      </div>
      <div class="pod podContent">
        <ng-container *ngIf="mvtdata">
          <table class="tableRes">
            <thead>
              <tr>
                <th scope="col">Numéro client</th>
                <th scope="col">Mail client</th>
                <th scope="col">Tel client</th>
              </tr>
            </thead>
            <tbody>
              <tr class="active-row"
                *ngFor="let mvt of mvtdata">
                <td>{{mvt.feMvtClimodel.mvtCliId}}</td>
```

```

        <!-- <td *ngIf="mvt.feMvtClimodel.mvtMvtC ==
'email'">{{mvt.feMvtClimodel.mvtVal}}</td>
        <td *ngIf="mvt.feMvtClimodel.mvtMvtC ==
'tel'">>{{mvt.feMvtClimodel.mvtVal}}</td>-->
        <td>{{mvt.feMvtClimodel.mvtVal}}</td>
        <td>{{mvt.feMvtClimodel.mvtVal}}</td>
    </tr>
</tbody>
</table>
<table class="tableRes">
    <thead>
        <tr>
            <th scope="col">Numéro client</th>
            <th scope="col">Mail client</th>
            <th scope="col">Tel client</th>
        </tr>
    </thead>
    <tbody>
        <!--<div *ngFor="let mvtOce of mvtOcedata">
            <div *ngIf="mvtOce.feMvtClimodel.mvtMvtC
=='email' || mvtOce.feMvtClimodel.mvtMvtC =='tel_portable'">-->
                <tr class="active-row" *ngFor="let mvtOce of
mvtOcedata">
                    <td>{{mvtOce.feMvtClimodel.mvtCliId}}</td>
                    <!-- <td *ngIf="mvtOce.feMvtClimodel.mvtMvtC ==
'email'">{{mvtOce.feMvtClimodel.mvtVal}}</td>
                    <td *ngIf="mvtOce.feMvtClimodel.mvtMvtC ==
'tel_portable'">{{mvtOce.feMvtClimodel.mvtVal}}</td>-->
                    <td>{{mvtOce.feMvtClimodel.mvtVal}}</td>
                    <td>{{mvtOce.feMvtClimodel.mvtVal}}</td>
                </tr>
            <!-- </div>
        </div>-->
    </tbody>
</table>
<table class="tableRes">
    <thead>
        <tr>
            <th scope="col">Numéro client</th>
            <th scope="col">Mail client</th>
            <th scope="col">Tel client</th>
        </tr>
    </thead>
    <tbody>
        <tr class="active-row"
            *ngFor="let PcBp of PcBpdata">
            <td>{{PcBp.PcBpContactmodel.numPers}}</td>
            <!--<td *ngIf="PcBp.PcBpContactmodel.mvtMvtC ==
'email'">{{PcBp.PcBpContactmodel.nomZone}}</td>
            <td *ngIf="PcBp.PcBpContactmodel.mvtMvtC ==
'tel'">{{PcBp.PcBpContactmodel.nomZone}}</td>-->
            <td>{{PcBp.PcBpContactmodel.nomZone}}</td>
            <td>{{PcBp.PcBpContactmodel.nomZone}}</td>
        </tr>
    </tbody>
</table>
<div class="d-flex justify-content-between p-2">
    <ngb-pagination
        [page]="page"
        [pageSize]="pageSize"
        [collectionSize]="mvtdata.length"

```

```

                [maxSize]="3"
                [ellipses]="true"
            ></ngb-pagination>
        </div>
    </ng-container>
</div>

</ng-template>
</li>

```

recherche.component.ts : (ligne 74)

```

this.rechercheAFForm = this.fb.group({
    mvtCliId: '',
    mvtCliVal: ''
});
this.rechercheAFForm.setValidators(CustomValidator.onlyOneFieldRequiredValidator('Seulement un champ doit être renseigné'));

```

ligne 86

```

rechercheAFForm: FormGroup;

```

ligne 195

```

doSearchAF() {
    if (this.doValidate(this.rechercheAFForm)) {
        const criteria = this.rechercheAFForm.value;
        this.searchService.doSearchMvtCli(criteria.mvtCli).subscribe((res)
=> {
            this.mvtdata = res.resultMvt;
            const currentState = this.store.selectSnapshot(AppDiagState);
            this.store.dispatch(new SaveSearch({resMvt: res}));
        });
    }
}

```

app.diag.service

```

doSearchMvtCli(MvtCli): Observable<any> {
    const currentState = this.store.selectSnapshot(AppDiagAuthState);
    this.auth = currentState.token;
    const headers = new HttpHeaders({
        'Content-Type': 'application/json',
        'Authorization': this.auth
    });
    return this.http.post<any>('/ma/appdiag/api/v1/MvtCli' , MvtCli,
{headers: headers});
}

```

Vendredi 06/01/2023

```
spring.datasource.url=${CONFIG_DB_DATASOURCE_URL:jdbc:oracle:thin:@pensee:2571:cccCC}
spring.datasource.username=${CONFIG_DB_DATASOURCE_USERNAME:
spring.datasource.password=${CONFIG_DB_DATASOURCE_PASSWORD:
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
```

Config BDD

BACK : repository -> entity (ctrl + click)

FRONT : services -> core-routing.module.ts

NumCli : 7397403

UtilRect_01

Debugg avec la partie inspecter sur le site (Front)

14h00 : Installation BDD Oracle sur IntelliJ.

Lundi 09/01/2023

9h00 : Reprise du projet

Création d'une classe dans le back ; FeMvtcli.java

```
package com.aviva.appdiag.beans.entity;

import javax.persistence.*;
import java.util.Date;

@Entity
@Table(name = "FE_MVTCLI", schema = "CRM")
public class FeMvtCli {
    private Long mvtId;

    private Date mvtTrtD;

    private Date mvtEnrD;

    private String mvtUsrOri;
    private Long mvtCliId;

    private String mvtMvtC;
    private String mvtVal;

    @Id
    @Column(name = "MVT_ID")
    public long getMvtId() { return mvtId; }

    public void setMvtId(Long mvtId) { this.mvtId = mvtId; }

    @Basic
    @Column(name = "MVT_TRT_D")
    public Date getMvtTrtD() { return mvtTrtD; }

    public void setMvtTrtD(Date mvtTrtD) { this.mvtTrtD = mvtTrtD; }
```

```

@Basic
@Column(name = "MVT_Enr_D")
public Date getMvtEnrD() { return mvtEnrD;}

public void setMvtEnrD(Date mvtEnrD) { this.mvtEnrD = mvtEnrD;}

@Basic
@Column(name = "MVT_CLI_ID")
public long getMvtCliId() { return mvtCliId;}

public void setMvtCliId(Long mvtCliId) { this.mvtCliId = mvtCliId;}

@Basic
@Column(name = "MVT_Usr_Ori")
public String getMvtUsrOri() { return mvtUsrOri;}

public void setMvtUsrOri(String mvtUsrOri){
this.mvtUsrOri = mvtUsrOri;}

@Basic
@Column(name = "MVT_MVT_C")
public String getMvtMvtC() { return mvtMvtC;}

public void setMvtMvtC(String mvtMvtC) { this.mvtMvtC = mvtMvtC;}

@Basic
@Column(name = "MVT_VAL")
public String getMvtVal() { return mvtVal;}

public void setMvtVal(String mvtVal) { this.mvtVal = mvtVal;}
}

```

Création Interface FeMvtCliRepository.java dans com.aviva.appdiag.repository

```

package com.aviva.appdiag.repository;

import com.aviva.appdiag.beans.entity.FeMvtCli;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import javax.transaction.Transactional;
import java.util.List;

@Repository
@Transactional
public interface FeMvtCliRepository extends CrudRepository<FeMvtCli,
Long>{

    @Query (value ="select mvtId, mvtTrtD, mvtEnrD, mvtUsrOri, mvtCliId,
mvtMvtC, mvtVal from FeMvtCli where mvtCliId = 10112666 and rownum < 100
order by mvtEnrD desc")
    List<FeMvtCli> findAllByCliId(Long cliId);

    @Query (value ="select mvtId, mvtTrtD, mvtEnrD, mvtUsrOri, mvtCliId,
mvtMvtC, mvtVal from FeMvtCli where mvtMvtC = 'email' and mvtVal like

```



```
'%abeille@aviva.com' and rownum < 100 order by mvtEnrD desc ")
    List<FeMvtCli> findAllByMail(String mvtVal);

    @Query (value ="select mvtId, mvtTrtD, mvtEnrD, mvtUsrOri, mvtCliId,
mvtMvtC, mvtVal from FeMvtCli where mvtMvtC = 'tel_portable' and mvtVal
like '%0607080910%' and rownum < 100 order by mvtEnrD desc ")
    List<FeMvtCli> findAllByTel(String mvtVal);

}
```

Besoin d'avoir une List<...> Je mets quoi dedans ? Un fichier que j'ai crée ou un fichier déjà existant ?

Ex :

```
List<FePerson> findAllByMail(String mailcle,String mailDom);
```

FindAllByMail -> RechercherPersonServicesImpl.java

RechercherPersonServicesImpl.java :

```
package com.aviva.appdiag.services;

import com.aviva.appdiag.beans.criterias.FePersonCriteria;
import com.aviva.appdiag.beans.entity.FePerson;
import com.aviva.appdiag.controllers.RechercheController;
import com.aviva.appdiag.repository.FePersonRepository;
import com.aviva.appdiag.services.interfaces.RecherchePersonService;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class RecherchePersonServicesImpl implements RecherchePersonService
{

    private static final Log LOGGER =
LogFactory.getLog(RechercheController.class);
    @Autowired
    FePersonRepository fePersonRepository;

    @Override
    public List<FePerson> searchByCriteria(FePersonCriteria criteria) {
        List<FePerson> result = null;
        if(criteria.getCliId() != null){
            result =
fePersonRepository.findAllByCliId(criteria.getCliId());
        }
        else if(!criteria.getMail().isEmpty() &&
criteria.getMail().contains("@"))
        {
            String[] mail = criteria.getMail().split("@");
            result = fePersonRepository.findAllByMail(mail[0], mail[1]);
        }
    }
}
```

```

        else if(criteria.getPriId() != null)
            result =
fePersonRepository.findAllByPriId(criteria.getPriId());
        else if(criteria.getTamId() != null)
            result =
fePersonRepository.findAllByTamId(criteria.getTamId());
        else if(criteria.getUsername() != null &&
!criteria.getUsername().isEmpty())
            result =
fePersonRepository.findAllByUsername(criteria.getUsername());
        else if(!criteria.getNumContrat().isEmpty())
            result =
fePersonRepository.findAllByContratNum(criteria.getNumContrat());

        //FIXME deconnecter l'entity
        return result;
    }
}

```

J'ai donc créé un ServicesImpl aussi pour pouvoir mettre une List mais je me demande si je dois en créer une ou bien reprendre ce qui existe déjà et ajouter ce dont j'ai besoin ?

Création Class RechercheMvtCliServiceImpl dans com.aviva.appdiag.services

```

package com.aviva.appdiag.services;

import com.aviva.appdiag.beans.criterias.FeMvtCliCriteria;
import com.aviva.appdiag.beans.entity.FeMvtCli;
import com.aviva.appdiag.controllers.RechercheController;
import com.aviva.appdiag.repository.FeMvtCliRepository;
import com.aviva.appdiag.services.interfaces.RechercheMvtCliService;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class RechercheMvtCliServiceImpl implements RechercheMvtCliService {

    private static final Log LOGGER =
LogFactory.getLog(RechercheController.class);

    @Autowired
    FeMvtCliRepository feMvtCliRepository;

    @Override
    public List<FeMvtCli> searchByMvtCliCriteria(FeMvtCliCriteria criteria)
    {
        List<FeMvtCli> result = null;
        if(criteria.getMvtCliId() != null){
            result =
feMvtCliRepository.findAllByCliId(criteria.getMvtCliId());
        }
        else if(!criteria.getMvtVal().isEmpty() &&
criteria.getMvtVal().contains("@"))
        {

```

```

        String[] mail = criteria.getMvtVal().split("@");
        result =
feMvtCliRepository.findAllByMail(criteria.getMvtVal());
    }
    else if (!criteria.getMvtVal().isEmpty() )
    {
        result = feMvtCliRepository.findAllByTel(criteria.getMvtVal());
    }

    //FIXME deconnecter l'entity
    return result;
}
}

```

Document auxquels je dois toucher ? Ou créer en me basant sur ces modèles

.criterias.FePersonCriteria **Fait**

.controllers.RechercheController **Fait**

.services.interfaces.RecherchePersonService **Fait**

com.aviva.appdiag.beans.dtos.DtoResult.java **Fait**

Création du fichier .criterias.FeMvtCliCriteria :

```

package com.aviva.appdiag.beans.criterias;

public class FeMvtCliCriteria {
    private Long mvtCliId;
    private String mvtMvtC;
    private String mvtVal;

    public FeMvtCliCriteria(Long mvtCliId, String mvtMvtC, String mvtVal) {
        this.mvtCliId = mvtCliId;
        this.mvtMvtC = mvtMvtC;
        this.mvtVal = mvtVal;
    }

    public Long getMvtCliId() { return mvtCliId; }

    public void setMvtCliId(long mvtCliId) {
        this.mvtCliId = mvtCliId;
    }

    public String getMvtMvtC() {
        return mvtMvtC;
    }

    public void setMvtMvtC(String mvtMvtC) {
        this.mvtMvtC = mvtMvtC;
    }

    public String getMvtVal() {
        return mvtVal;
    }

    public void setMvtVal(String mvtVal) {
        this.mvtVal = mvtVal;
    }
}

```

Création du fichier RechercheMvtCliService.java dans .services.interfaces

```
package com.aviva.appdiag.services.interfaces;

import com.aviva.appdiag.beans.criterias.FeMvtCliCriteria;
import com.aviva.appdiag.beans.entity.FeMvtCli;

import java.util.List;

public interface RechercheMvtCliService {
    List<FeMvtCli> searchByMvtCliCriteria(FeMvtCliCriteria criteria);
}
```

Mardi 10/01/2023

RechercheContrôller est liée à DtoResult qui dans les exemples est liée à des models.

Mais je ne sais pas quoi mettre dans les modèles ? Le fichier controller est un enfer, tout comme le fichier DtoResult...

FePurgePerson ?? Je n'ai pas de table PurgeMvtCli ? Je fais comment du coup ? Je crée une table ? Ou je crée quand même le fichier ??

Autre souci ! Il n'y a pas de table pour les DCCE et surtout la table *dcce.pc_bp_contact* ne figure même pas dans la bdd... Et puis je ne comprends pas pourquoi il faudrait une table front et back ? La table CRM_MVTCLI je l'ai mise dans le back alors qu'apparemment il devait être dans le front, c'est pas good ? Sinon je fais quiiiiiii ??? Kyyyya trop complexe !

Finalement je n'ai peut-être pas besoin de refaire le controller, je peux le créer moi-même à partir d'un autre fichier.

MvtCliController.java

```
package com.aviva.appdiag.controllers;

import com.aviva.appdiag.services.interfaces.MvtCliService;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.Parameter;
import io.swagger.v3.oas.annotations.tags.Tag;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
```

```

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.util.HtmlUtils;

import static io.swagger.v3.oas.annotations.enums.ParameterIn.HEADER;

@Controller
@RequestMapping("/v1")
@Tag(name = "MvtCli", description = "MvtCliController")
public class MvtCliController {

    private static final Log LOGGER =
LogFactory.getLog(MvtCliController.class);

    @Autowired
    private MvtCliService MvtCliService;

    @Operation(summary = "Permet de recuperer le mvtId")
    @Parameter(name = "Authorization", description = "Username:password",
required = true, allowEmptyValue = true, in = HEADER)
    @PostMapping(value = "/getMvtId/{mvtId}", produces =
MediaType.TEXT_PLAIN_VALUE)
    @ResponseBody
    public ResponseEntity<String> getMvtId(@PathVariable String mvtId) {
        String xml = MvtCliService.getMvtId(mvtId);
        return new ResponseEntity<>(xml, HttpStatus.OK);
    }

    @Operation(summary = "Permet de recuperer le mvtCliId")
    @Parameter(name = "Authorization", description = "Username:password",
required = true, allowEmptyValue = true, in = HEADER)
    @PostMapping(value = "/getMvtId/{mvtId}", produces =
MediaType.TEXT_PLAIN_VALUE)
    @ResponseBody
    public ResponseEntity<String> getMvtCliId(@PathVariable String
mvtCliId) {
        String xml = MvtCliService.getMvtCliId(mvtCliId);
        return new ResponseEntity<>(xml, HttpStatus.OK);
    }

    @Operation(summary = "Permet de recuperer le mvtVal")
    @Parameter(name = "Authorization", description = "Username:password",
required = true, allowEmptyValue = true, in = HEADER)
    @PostMapping(value = "/getMvtId/{mvtId}", produces =
MediaType.TEXT_PLAIN_VALUE)
    @ResponseBody
    public ResponseEntity<String> getMvtVal(@PathVariable String mvtVal) {
        String xml = MvtCliService.getMvtVal(mvtVal);
        return new ResponseEntity<>(xml, HttpStatus.OK);
    }
}

```

J'ai dû pour cela créer une autre interface du nom de *MvtCliService.java*

```

package com.aviva.appdiag.services.interfaces;

public interface MvtCliService {

```

```
String getMvtId (String mvtId);
String getMvtCliId (String mvtCliId);
String getMvtVal (String mvtVal);
}
```

Faire le Dto correspondant à DtoResultActe, le liée à mon controller ? ou à un autre fichier correspondant au modèle SuiviActecvController.

Faire le model avec des gets et des sets des champs utiles ?

Remplacer * dans la requête par tous les champs séparés par une virgule.

Recherche :

Débogueur et débogage

Le *débogage* de terme peut signifier beaucoup de choses différentes, mais la plus littéralement, cela signifie supprimer des bogues de votre code. Maintenant, il y a beaucoup de façons de le faire. Par exemple, vous pouvez déboguer en analysant votre code à la recherche de typos ou à l'aide d'un analyseur de code. Vous pouvez déboguer du code à l'aide d'un profileur de performances. Vous pouvez également déboguer à l'aide d'un *débogueur*.

Un débogueur est un outil de développement très spécialisé qui s'attache à votre application en cours d'exécution et vous permet d'inspecter votre code.

Le débogueur est un outil essentiel pour rechercher et corriger les bogues dans vos applications. Toutefois, le contexte est roi et il est important de tirer parti de tous les outils à votre jetable pour vous aider à éliminer rapidement les bogues ou les erreurs. Parfois, le bon « outil » peut être une meilleure pratique de codage. En apprenant à utiliser le débogueur par rapport à un autre outil, vous allez également apprendre à utiliser le débogueur plus efficacement.

DtoResultMvtCli.java

```
package com.aviva.appdiag.beans.dtos;

import com.aviva.appdiag.beans.entity.FeMvtCli;

import java.util.ArrayList;
import java.util.List;

public class DtoResultMvtCli {

    private List<Long> ResultMvtCli = new ArrayList<>();
}
```

```

    public DtoResultMvtCli(List<FeMvtCli> feMvtCli) {
        for (FeMvtCli mvt: feMvtCli) {
            ResultMvtCli.add(mvt.getMvtId());
        }
    }

    public List<Long> getResultMvtCli() {
        return ResultMvtCli;
    }

    public void setResultMvtCli(List<Long> ResultMvtCli) {
        this.ResultMvtCli = ResultMvtCli;
    }
}

```

Pour que tout fonctionne correctement, j'ai également dû toucher à des fichiers déjà créés tel que :

FailureResponse :

```

public class FailureResponse extends Response{

    private String errorCode;
    private String errorMessage;

    public FailureResponse(List<DtoResult> res, List<DtoResultDetails>
resultDetails, List<DtoProduit> resultProduit, List<DtoResultArt83>
resultArt83, List<DtoResultMvtCli> resultMvtCli, String errorCode, String
errorMessage) {
        super(res, resultDetails, resultArt83, resultProduit,
resultMvtCli);
        this.errorCode = errorCode;
        this.errorMessage = errorMessage;
        this.status = "FAILURE";
    }
}

```

Response :

```

public class Response {

    protected String status = "SUCCESS";
    protected List<DtoResult> res;
    protected List<DtoResultDetails> resDetails;
    protected List<DtoResultArt83> resultArt83;
    protected List<DtoProduit> resultProduit;
    protected List<DtoResultMvtCli> resultMvtCli;

    public Response(List<DtoResult> res, List<DtoResultDetails>
resDetails, List<DtoResultArt83> resultArt83, List<DtoProduit>
resProduit, List<DtoResultMvtCli> resMvtCli ) {
        this.res = res;
    }
}

```

```

        this.resDetails = resDetails;
        this.resultArt83 = resultArt83;
        this.resultProduit = resProduit;
        this.resultMvtCli = resMvtCli;
    }

    public static Response addDtoResult(List<DtoResult> res) {
        return new Response(res,null,null,null, null);
    }

    public static Response addDtoResultDetails(List<DtoResultDetails> res)
{
        return new Response(null,res,null,null, null);
    }

    public static Response addDtoResultArt83(List<DtoResultArt83> res) {
        return new Response(null,null,res,null, null);
    }

    public static Response addDtoResultProduit(List<DtoProduit> res) {
        return new Response(null,null,null,res, null);
    }

    public static Response addDtoResultMvtCli(List<DtoResultMvtCli> res) {
        return new Response(null,null, null,null, res);
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public List<DtoResult> getRes() {
        return res;
    }

    public void setRes(List<DtoResult> res) {
        this.res = res;
    }

    public List<DtoResultDetails> getResDetails() {
        return resDetails;
    }

    public void setResDetails(List<DtoResultDetails> resDetails) {
        this.resDetails = resDetails;
    }

    public List<DtoResultArt83> getResultArt83() {
        return resultArt83;
    }

    public void setResultArt83(List<DtoResultArt83> resultArt83) {
        this.resultArt83 = resultArt83;
    }

    public List<DtoProduit> getResultProduit() {
        return resultProduit;
    }

```



```

    }

    public void setResultProduit(List<DtoProduit> resultProduit) {
        this.resultProduit = resultProduit;
    }

    public List<DtoResultMvtCli> getResultMvtCli() {
        return resultMvtCli;
    }

    public void setResultMvtCli(List<DtoResultMvtCli> resultMvtCli) {
        this.resultMvtCli = resultMvtCli;
    }
}

```

BACK : *Com.aviva.appdiag*

.beans.criteria.FeMvtCliCriteria : Ce sont les critères qui nous seront utiles pour la mission.

.beans.dtos.DtoResultMvtCli : Permet de renvoyer un résultat à l'aide du model et de l'entité.

.beans.entity.FeMvtCli : Ce sont tous les champs de la table **MVTCLI**.

.beans.model.FeMvtCliModel : Ce sont les champs utiles pour la mission (les mêmes que Criteria).

.beans.response.FailureResponse : Permet de gérer les erreurs de résultat.

.beans.dtos.response.Response : Permet de gérer l'affichage du résultat.

.controllers.RechercheController : Permet de contrôler la recherche, il est le pont entre tous les fichiers.

.repository.FeMvtRepository : C'est là que se trouve les requêtes qui permettent de donner un rôle aux champs.

.services.interfaces.RechercheMvtCliService : Permet de créer un pont vers l'implémentation.

.services.RechercheMvtCliServiceImpl : Permet de remplir 1 champs parmi les 3, il est liée au repository qui donne les rôles tandis que l'implémentation donne les conditions.

FRONT : *src.app.appDiagModule*

.components.MvtCli.detail.DetailMvtCli.component.html :

.components.MvtCli.detail.DetailMvtCli.component.ts :

.components.MvtCli.MvtCli.component.html :

.components.MvtCli.MvtCli.component.ts :

.services.app.diag.service.ts :

.app.diag.module.ts :

.app.diag.routing.module.ts :

Apprendre le JavaScript

Pourquoi apprendre le JavaScript ?

- Principal langage de script côté navigateur
- Langage polyvalent
- Ecosystème & Communauté

JavaScript n'est pas Java !!!

JavaScript = Langage

ECMA = Organisme de standardisation

Pour se documenter : MDN

Variable :

Const → ne pourra pas être réassigner (la valeur ne pourra plus changer)

Let → la valeur peut changer

Var → équivaut à écrire let (variable par défaut) ; à ne pas retenir car souvent considéré comme une faute.

Const n'a pas de limite de type, on peut tout mettre dans un const.

`a == b` // a égale à b

`a === b` // a == b et a est de même "type" que b

`'4' == 4` → true

`'4' === 4` → false

`[] == []` → false

Je n'ai pas l'impression que mes tableaux marchent, pourquoi ?

Hé bien, quand je tape une valeur présente dans la 2^e table, ça me l'affiche correctement dans la 2^{ème} mais également dans la 1^{ère}.

J'ai également l'impression qu'il y a des valeurs dans la 1^{ère} table qui ne font pas partis des tables dans lesquels doivent chercher les requêtes, il y a des valeurs je ne sais pas elles sortent d'où.

Par exemple, **6536513** apparaît 3 fois dans la 2ème table et 0 fois dans la 1ere or dans les résultats elle est affichée 4 fois.

Faire marcher la 3^e table

Faire en sorte que l'email n'accepte que des emails et le tel que le tel

On est sûr que tout est good niveau tableau ?

Tableau 1 = table 1 ; Tableau 2 = table 2 ; Tableau 3 = table 3 ???

Code Finale

Recherche.component.html

```
<li [ngbNavItem]="6">
  <a ngbNavLink
    style="color: black; border-top-left-radius: 1.5rem; border-top-
right-radius: 1.5rem; text-decoration: none">Anti-Fraude</a>
    <ng-template ngbNavContent>
      <div class="pod podContent">
        <div>
          <h2>Recherche d'un client</h2>
        </div>
        <form [formGroup]="rechercheAFForm" action="#">
          <div class="pod podContent text-center">
            <div class="form-group row">
              <label for="mvtCliId" class="col-sm-4">Numéro
client : </label>
              <div class="col-sm-5">
                <input class="form-control" id="mvtCliId"
type="text" name="mvtCliId" formControlName="mvtCliId" vnNumber>
              </div>
            </div>
            <div class="form-group row">
              <label for="mvtValEmail" class="col-sm-4">Mail
client : </label>
              <div class="col-sm-5">
                <input class="form-control" type="email"
email="true" id="mvtValEmail" name="mvtVal" formControlName="mvtValEmail"
maxlength="50" >
              </div>
            </div>
            <div class="form-group row">
              <label for="mvtValTel" class="col-sm-4 ">Numéro
mobile : </label>
              <div class="col-sm-5">
                <input class="form-control" id="mvtValTel"
name="mvtVal" formControlName="mvtValTel" maxlength="10" vnNumber>
              </div>
            </div>
            <vn-control-message id="errorFields"
[control]="rechercheAFForm"></vn-control-message>
          </div>
          <button class="standard" style="float: right; font-size:
150%;" type="submit" (keydown.enter,keydown.a)="doSearchAF()"
(click)="doSearchAF()">Lancer la recherche</button>
        </form>
      </div>
      <div class="pod podContent">
        <ng-container *ngIf="mvtdata">
          <table class="tableRes">
            <thead>
              <tr>
```

```

        <th scope="col">Numéro client</th>
        <th scope="col">Mail client</th>
        <th scope="col">Tel client</th>
    </tr>
</thead>
<tbody>
<tr class="active-row"
    *ngFor="let mvt of mvtdata | slice: (page-1) *
pageSize : page * pageSize">
    <td>{{mvt.feMvtClimodel.mvtCliId}}</td>
    <!--<td>{{mvt.feMvtClimodel.mvtMvtC == 'email' ?
mvt.feMvtClimodel.mvtVal : '' }}</td>
    <td>{{mvt.feMvtClimodel.mvtMvtC == 'tel_portable' ?
mvt.feMvtClimodel.mvtVal : ''}}</td>-->
    <td>{{mvt.feMvtClimodel.mvtVal}}</td>
    <td>{{mvt.feMvtClimodel.mvtVal}}</td>
</tr>
</tbody>
</table>

<div class="d-flex justify-content-between p-2" >
    <ngb-pagination

        [ (page) ]="page"
        [pageSize]="pageSize"
        [collectionSize]="mvtdata.length"
        [maxSize]="5"
        [ellipses]="true">
    </ngb-pagination>
</div>
<table class="tableRes">
    <thead>
    <tr>
        <th scope="col">Numéro client</th>
        <th scope="col">Mail client</th>
        <th scope="col">Tel client</th>
    </tr>
    </thead>
    <tbody>
    <!--<div *ngFor="let mvtOce of mvtOcedata">
        <div *ngIf ="mvtOce.feMvtClimodel.mvtMvtC
=='email' || mvtOce.feMvtClimodel.mvtMvtC =='tel_portable'">-->
        <tr class="active-row" *ngFor="let mvtOce of
mvtOcedata | slice: (page2-1) * pageSize : page2 * pageSize">
            <td>{{mvtOce.feMvtClimodel.mvtCliId}}</td>
            <!--<td>{{mvtOce.feMvtClimodel.mvtMvtC == 'email' ?
mvtOce.feMvtClimodel.mvtVal : '' }}</td>
            <td>{{mvtOce.feMvtClimodel.mvtMvtC ==
'tel_portable' ? mvtOce.feMvtClimodel.mvtVal : ''}}</td>-->
            <td>{{mvtOce.feMvtClimodel.mvtVal}}</td>
            <td>{{mvtOce.feMvtClimodel.mvtVal}}</td>
        </tr>
    </tbody>
</table>
<div class="d-flex justify-content-between p-2">
    <ngb-pagination

        [ (page) ]="page2"
        [pageSize]="pageSize"
        [collectionSize]="mvtOcedata.length"
        [maxSize]="5"
        [ellipses]="true"

```

```

        >
        </ngb-pagination>
    </div>
    <table class="tableRes">
        <thead>
            <tr>
                <th scope="col">Numéro client</th>
                <th scope="col">Mail client</th>
                <th scope="col">Tel client</th>
            </tr>
        </thead>
        <tbody>
            <tr class="active-row"
                *ngFor="let PcBp of PcBpdata | slice: (page3-1) *
                pageSize : page3 * pageSize">
                <td>{{PcBp.pcBpContactModel.numPers}}</td>
                <!--<td>{{PcBp.pcBpContactModel.nomZone == 'email'
                ? PcBp.pcBpContactmodel.valeurZone : '' }}</td>
                <td> {{PcBp.pcBpContactModel.nomZone ==
                'tel_portable' ? PcBp.pcBpContactmodel.valeurZone : '' }}</td>-->
                <td>{{PcBp.pcBpContactModel.nomZone}}</td>
                <td>{{PcBp.pcBpContactModel.nomZone}}</td>
            </tr>
        </tbody>
    </table>
    <div class="d-flex justify-content-between p-2">
        <ngb-pagination
            [ (page) ]="page3"
            [pageSize]="pageSize"
            [collectionSize]="PcBpdata.length"
            [maxSize]="5"
            [ellipses]="true"
        >
        </ngb-pagination>
    </div>
</ng-container>
</div>

</ng-template>
</li>

```

Recherche.component.ts

```

this.rechercheAForm = this.fb.group({
    mvtCliId: '',
    mvtValEmail: '',
    mvtValTel: ''
});

this.rechercheAForm.setValidators(CustomValidator.onlyOneFieldRequiredValidator('UN champ doit être renseigné'));

}

rechercheForm: FormGroup;
rechercheTForm: FormGroup;
rechercheArtForm: FormGroup;
rechercheGdprForm: FormGroup;
modifyArtForm: FormGroup;

```

```

rechercheProdForm: FormGroup;
rechercheAFForm: FormGroup;

data: AppDiagData = new AppDiagData();
artdata: any;
proddata: any;
mvtdata: any;
mvtOcedata: any;
PcBpdata: any;
date: { year: number, month: number };
allProduit: any;
model: NgbDateStruct;
page = 1;
page2 = 1;
page3 = 1;
pageSize = 4;
resultServiceCall: String;
urlSuivi: String;
active = 1;
suivi: any;
filename: string;
downloadUrl: any;

```

```

doSearchAF() {
  if (this.doValidate(this.rechercheAFForm)) {
    const criteria = this.rechercheAFForm.value;
    console.log('Hi ! Je passe par là ! hihi !:)|');
    this.searchService.doSearchMvtCli(criteria).subscribe((res) => {
      this.mvtdata = res.resultMvtCli;
      this.mvtOcedata = res.resultMvtCliOce;
      this.PcBpdata = res.resultPcBpContact;
      console.log(res);
      const currentState = this.store.selectSnapshot(AppDiagState);
      this.store.dispatch(new SaveSearch({resMvt: res}));
    });
  }
}

```

Criteria ne correspond pas aux critere de la requête dans repository, en faite elle reprend les valeurs utilisé dans le front.

FeMvtCliCriteria

```

package com.aviva.appdiag.beans.criterias;

import java.util.Date;

public class FeMvtCliCriteria {
  private Long mvtCliId;
  private String mvtValEmail;
  private String mvtValTel;

  public FeMvtCliCriteria(Number mvtCliId, String mvtValEmail, String mvtValTel) {
    this.mvtCliId = mvtCliId.longValue();
    this.mvtValEmail = mvtValEmail;
    this.mvtValTel = mvtValTel;
  }
}

```

```

    public Long getMvtCliId() {
        return mvtCliId; }

    public void setMvtId(Long mvtCliId) {
        this.mvtCliId = mvtCliId;
    }

    public String getMvtValEmail() {
        return mvtValEmail;
    }

    public void setMvtValEmail(String mvtValEmail) {
        this.mvtValEmail = mvtValEmail;
    }

    public String getMvtValTel() {
        return mvtValTel;
    }

    public void setMvtValTel(String mvtValTel) {
        this.mvtValTel = mvtValTel;
    }

    public FeMvtCliCriteria() {
    }
}

```

DtoResultMvtCli, DtoResultMvtCliOce, DtoResultPcBpContact

```

package com.aviva.appdiag.beans.dtos;

import com.aviva.appdiag.beans.entity.FeMvtCli;
import com.aviva.appdiag.beans.model.FeMvtCliModel;
import org.modelmapper.ModelMapper;

import java.text.SimpleDateFormat;

public class DtoResultMvtCli {

    private FeMvtCliModel feMvtCliModel;

    public DtoResultMvtCli(FeMvtCli feMvtCli) {
        ModelMapper modelMapper = new ModelMapper();
        SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
        this.feMvtCliModel = modelMapper.map(feMvtCli,
FeMvtCliModel.class);
    }

    public FeMvtCliModel getFeMvtCliModel() {
        return feMvtCliModel;
    }

    public void setFeMvtCliModel(FeMvtCliModel feMvtCliModel) {
        this.feMvtCliModel = feMvtCliModel;
    }
}

```



```

package com.aviva.appdiag.beans.dtos;

import com.aviva.appdiag.beans.entity.DccMvtCliOce;
import com.aviva.appdiag.beans.model.FeMvtCliModel;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.modelmapper.ModelMapper;

import java.text.SimpleDateFormat;
import java.time.LocalDate;

public class DtoResultMvtCliOce {

    private static final Log LOGGER =
LogFactory.getLog(DtoResultPcBpContact.class);
    private FeMvtCliModel feMvtCliModel;

    public DtoResultMvtCliOce(DccMvtCliOce dccMvtCliOce) {
        ModelMapper modelMapper = new ModelMapper();
        LocalDate date = null;
        if(dccMvtCliOce.getMvtEnrD() != null){
            int annee =
Integer.parseInt(dccMvtCliOce.getMvtEnrD().substring(0, 4));
            int mois =
Integer.parseInt(dccMvtCliOce.getMvtEnrD().substring(4,6));
            LOGGER.info(mois);
            int jour =
Integer.parseInt(dccMvtCliOce.getMvtEnrD().substring(6,8));
            LOGGER.info(jour);
            date = LocalDate.of(annee,mois,jour);
            LOGGER.info(date);}

        LocalDate date2 = null;
        if(dccMvtCliOce.getMvtTrtD() != null){
            int annee =
Integer.parseInt(dccMvtCliOce.getMvtTrtD().substring(0, 4));
            int mois =
Integer.parseInt(dccMvtCliOce.getMvtTrtD().substring(4,6));
            LOGGER.info(mois);
            int jour =
Integer.parseInt(dccMvtCliOce.getMvtTrtD().substring(6,8));
            LOGGER.info(jour);
            date2 = LocalDate.of(annee,mois,jour);
            LOGGER.info(date2);}
        FeMvtCliModel feMvtCli = modelMapper.map(dccMvtCliOce,
FeMvtCliModel.class);
        feMvtCli.setMvtEnrD(date.toString());
        feMvtCli.setMvtTrtD(date2.toString());
        this.feMvtCliModel = feMvtCli;
    }

    public FeMvtCliModel getFeMvtCliModel() {
        return feMvtCliModel;
    }

    public void setFeMvtCliModel(FeMvtCliModel feMvtCliModel) {
        this.feMvtCliModel = feMvtCliModel;
    }
}

```

```

package com.aviva.appdiag.beans.dtos;

import com.aviva.appdiag.beans.entity.PcBpContact;
import com.aviva.appdiag.beans.model.PcBpContactModel;
import com.aviva.appdiag.controllers.RechercheController;
import com.fasterxml.jackson.databind.ObjectMapper;
import org.apache.commons.lang3.time.DateFormatUtils;
import org.apache.commons.lang3.time.DateUtils;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.modelmapper.ModelMapper;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Date;
import java.util.logging.Logger;

public class DtoResultPcBpContact {
    private PcBpContactModel pcBpContactModel;

    private static final Log LOGGER =
LogFactory.getLog(DtoResultPcBpContact.class);

    public DtoResultPcBpContact(PcBpContact pcBpContact) throws
ParseException {
        ObjectMapper modelMapper = new ObjectMapper();
        LocalDate dateExtraction = null;
        if(pcBpContact.getExtraction() != null){
            int annee =
Integer.parseInt(pcBpContact.getExtraction().substring(0, 4));
            int mois =
Integer.parseInt(pcBpContact.getExtraction().substring(4,6));
            LOGGER.info(mois);
            int jour =
Integer.parseInt(pcBpContact.getExtraction().substring(6,8));
            LOGGER.info(jour);
            dateExtraction = LocalDate.of(annee,mois,jour);
            LOGGER.info(dateExtraction);}
        LocalDate ld = LocalDate.of(1800,1,1);
        ld = ld.plusDays(pcBpContact.getDebut()-1);
        this.pcBpContactModel = new
PcBpContactModel(pcBpContact.getValeurZone(), pcBpContact.getNumPers(),
ld.toString(), pcBpContact.getNomZone(),
dateExtraction!=null?dateExtraction.toString():pcBpContact.getExtraction());
    }

    public PcBpContactModel getPcBpContactModel() {
        return pcBpContactModel;
    }

    public void setPcBpContactModel(PcBpContactModel pcBpContactModel) {
        this.pcBpContactModel = pcBpContactModel;
    }
}

```

FeMvtCli

```
package com.aviva.appdiag.beans.entity;

import javax.persistence.*;
import java.util.Date;

@Entity
@Table(name = "FE_MVTCLI", schema = "CRM")
public class FeMvtCli {
    private Long mvtId;

    private Date mvtTrtD;

    private Date mvtEnrD;

    private String mvtUsrOri;
    private Long mvtCliId;

    private String mvtMvtC;
    private String mvtVal;

    @Id
    @Column (name = "MVT_ID")
    public Long getMvtId() { return mvtId; }

    public void setMvtId(Long mvtId) { this.mvtId = mvtId; }

    @Basic
    @Column (name = "MVT_TRT_D")
    public Date getMvtTrtD() { return mvtTrtD; }

    public void setMvtTrtD(Date mvtTrtD) { this.mvtTrtD = mvtTrtD; }

    @Basic
    @Column (name = "MVT_Enr_D")
    public Date getMvtEnrD() { return mvtEnrD; }

    public void setMvtEnrD(Date mvtEnrD) { this.mvtEnrD = mvtEnrD; }

    @Basic
    @Column(name = "MVT_CLI_ID")
    public Long getMvtCliId() { return mvtCliId; }

    public void setMvtCliId(Long mvtCliId) { this.mvtCliId = mvtCliId; }

    @Basic
    @Column(name = "MVT_Usr_Ori")
    public String getMvtUsrOri() { return mvtUsrOri; }

    public void setMvtUsrOri(String mvtUsrOri) { this.mvtUsrOri =
mvtUsrOri; }

    @Basic
    @Column(name = "MVT_MVT_C")
    public String getMvtMvtC() { return mvtMvtC; }

    public void setMvtMvtC(String mvtMvtC) { this.mvtMvtC = mvtMvtC; }
```

```

    @Basic
    @Column(name = "MVT_VAL")
    public String getMvtVal() { return mvtVal; }

    public void setMvtVal(String mvtVal) { this.mvtVal = mvtVal; }
}

```

DccMvtCliOce

```

package com.aviva.appdiag.beans.entity;

import javax.persistence.*;

@Entity
@Table(name = "DCC_MVTCLIOCE", schema = "DCCE")
public class DccMvtCliOce {

    private Long mvtId;

    private String mvtTrtD;

    private String mvtEnrD;

    private String mvtUsrOri;
    private Long mvtCliId;

    private String mvtMvtC;
    private String mvtVal;
    private String mvtIntAbvC;

    @Id
    @Column(name = "MVT_ID")
    public Long getMvtId() { return mvtId; }

    public void setMvtId(Long mvtId) { this.mvtId = mvtId; }

    @Basic
    @Column(name = "MVT_TRT_D")
    public String getMvtTrtD() { return mvtTrtD; }

    public void setMvtTrtD(String mvtTrtD) { this.mvtTrtD = mvtTrtD; }

    @Basic
    @Column(name = "MVT_ENR_D")
    public String getMvtEnrD() { return mvtEnrD; }

    public void setMvtEnrD(String mvtEnrD) { this.mvtEnrD = mvtEnrD; }

    @Basic
    @Column(name = "MVT_CLI_ID")
    public Long getMvtCliId() { return mvtCliId; }

    public void setMvtCliId(Long mvtCliId) { this.mvtCliId = mvtCliId; }
}

```

```

        @Basic
        @Column(name = "MVT_UTI_ORI")
        public String getMvtUsrOri() { return mvtUsrOri; }

        public void setMvtUsrOri(String mvtUsrOri) { this.mvtUsrOri =
mvtUsrOri; }

        @Basic
        @Column(name = "MVT_MVT_C")
        public String getMvtMvtC() { return mvtMvtC; }

        public void setMvtMvtC(String mvtMvtC) { this.mvtMvtC = mvtMvtC; }

        @Basic
        @Column(name = "MVT_VAL")
        public String getMvtVal() { return mvtVal; }

        public void setMvtVal(String mvtVal) { this.mvtVal = mvtVal; }

        @Basic
        @Column(name = "MVT_INT_ABV_C")
        public String getMvtIntAbvC() { return mvtIntAbvC; }

        public void setMvtIntAbvC(String mvtIntAbvC) { this.mvtIntAbvC =
mvtIntAbvC; }

    }

```

PcBpContact

```

package com.aviva.appdiag.beans.entity;

import javax.persistence.*;

@Entity
@Table(name = "PC_BP_CONTACT", schema = "DCCE")
public class PcBpContact {
    private String oNum;

    private Long oSt;

    private String valeurZone;

    private Long debut;
    private Long numPers;

    private Long numMvt;
    private String nomZone;
    private Long fin;
    private String extraction;
    private String codeInter;

    @Id
    @Column(name = "O__NUM")
    public String getONum() { return oNum; }

    public void setONum(String oNum) { this.oNum = oNum; }
}

```

```

@Basic
@Column (name = "O__ST")
public Long getOSt() { return oSt; }

public void setOSt(Long oSt) { this.oSt = oSt; }

@Basic
@Column (name = "VALEUR_ZONE")
public String getValeurZone() { return valeurZone; }

public void setValeurZone(String valeurZone) { this.valeurZone =
valeurZone; }

@Basic
@Column (name = "DEBUT")
public Long getDebut() { return debut; }

public void setDebut(Long debut) { this.debut = debut; }

@Basic
@Column (name = "NUMERO_PERS")
public Long getNumPers() { return numPers; }

public void setNumPers(Long numPers) { this.numPers = numPers; }

@Basic
@Column (name = "NUMERO_MVT")
public Long getNumMvt() { return numMvt; }

public void setNumMvt(Long numMvt) { this.numMvt = numMvt; }

@Basic
@Column (name = "NOM_ZONE")
public String getNomZone() { return nomZone; }

public void setNomZone(String nomZone) { this.nomZone = nomZone; }

@Basic
@Column (name = "FIN")
public Long getFin() {return fin;}

public void setFin(Long fin) {this.fin = fin;}

@Basic
@Column (name = "DATE_EXTRACTION")
public String getExtraction() {return extraction;}

public void setExtraction(String extraction) {this.extraction =
extraction;}

@Basic
@Column (name = "CODE_INTER")
public String getCodeInter() {return codeInter;}

public void setCodeInter(String codeInter) {this.codeInter =
codeInter;}
}

```

FeMvtCliModel

```
package com.aviva.appdiag.beans.model;

public class FeMvtCliModel {
    private Long mvtCliId;
    private String mvtMvtC;
    private String mvtVal;

    public Long getMvtCliId() { return mvtCliId; }

    public void setMvtCliId(Long mvtCliId) {
        this.mvtCliId = mvtCliId;
    }

    public String getMvtMvtC() {
        return mvtMvtC;
    }

    public void setMvtMvtC(String mvtMvtC) {
        this.mvtMvtC = mvtMvtC;
    }

    public String getMvtVal() {
        return mvtVal;
    }

    public void setMvtVal(String mvtVal) {
        this.mvtVal = mvtVal;
    }
}
```

PcBpContactModel

```
package com.aviva.appdiag.beans.model;

import javax.persistence.Basic;
import javax.persistence.Column;

public class PcBpContactModel {
    private String valeurZone;

    private Long numPers;

    private Long numMvt;
    private String nomZone;

    @Basic
    @Column(name = "VALEUR_ZONE")
    public String getValeurZone() { return valeurZone; }

    public void setValeurZone(String valeurZone) { this.valeurZone =
valeurZone; }
```

```

@Basic
@Column(name = "NUM_PERS")
public Long getNumPers() { return numPers; }

public void setNumPers(Long numPers) { this.numPers = numPers; }

@Basic
@Column(name = "NUM_MVT")
public Long getNumMvt() { return numMvt; }

public void setNumMvt(Long numMvt) { this.numMvt = numMvt; }

@Basic
@Column(name = "Nom_Zone")
public String getNomZone() { return nomZone; }

public void setNomZone(String nomZone) { this.nomZone = nomZone; }
}

```

FailureResponse

```

package com.aviva.appdiag.beans.response;

import com.aviva.appdiag.beans.dtos.*;
import java.util.List;

public class FailureResponse extends Response{

    private String errorCode;
    private String errorMessage;

    public FailureResponse(List<DtoResult> res, List<DtoResultDetails>
resultDetails, List<DtoResultArt83> resultArt83, List<DtoProduit>
resultProduit, List<DtoDLProduit> resAllProduit, List<DtoResultMvtCli>
resultMvtCli, List<DtoResultMvtCliOce>
resultMvtCliOce, List<DtoResultPcBpContact> resultMvtContact, String
errorCode, String errorMessage) {
        super(res, resultDetails, resultArt83, resultProduit, resAllProduit,
resultMvtCli, resultMvtCliOce, resultMvtContact);
        this.errorCode = errorCode;
        this.errorMessage = errorMessage;
        this.status = "FAILURE";
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getErrorCode() {
        return errorCode;
    }

    public void setErrorCode(String errorCode) {

```



```

        this.errorCode = errorCode;
    }

    public String getErrorMessage() {
        return errorMessage;
    }

    public void setErrorMessage(String errorMessage) {
        this.errorMessage = errorMessage;
    }
}

```

Response

```

package com.aviva.appdiag.beans.response;

import com.aviva.appdiag.beans.dtos.*;

import java.util.List;

public class Response {

    protected String status = "SUCCESS";
    protected List<DtoResult> res;
    protected List<DtoResultDetails> resDetails;
    protected List<DtoResultArt83> resultArt83;
    protected List<DtoProduit> resultProduit;
    protected List<DtoDLProduit> resultAllProduit;
    protected List<DtoResultMvtCli> resultMvtCli;
    protected List<DtoResultMvtCliOce> resultMvtCliOce;
    protected List<DtoResultPcBpContact> resultMvtContact;

    public Response(List<DtoResult> res, List<DtoResultDetails>
resDetails, List<DtoResultArt83> resultArt83, List<DtoProduit> resProduit,
List<DtoDLProduit> resAllProduit, List<DtoResultMvtCli> resMvtCli,
List<DtoResultMvtCliOce> resMvtCliOce, List<DtoResultPcBpContact>
resContact ) {
        this.res = res;
        this.resDetails = resDetails;
        this.resultArt83 = resultArt83;
        this.resultProduit = resProduit;
        this.resultMvtCli = resMvtCli;
        this.resultAllProduit = resAllProduit;
        this.resultMvtCliOce = resMvtCliOce;
        this.resultMvtContact = resContact;
    }

    public static Response addDtoResult(List<DtoResult> res) {
        return new Response(res,null,null,null, null, null, null, null);
    }

    public static Response addDtoResultDetails(List<DtoResultDetails> res)
{
        return new Response(null,res,null,null, null,null, null, null);
    }

    public static Response addDtoResultArt83(List<DtoResultArt83> res) {
        return new Response(null,null,res,null, null, null, null, null);
    }
}

```

```

    }

    public static Response addDtoResultProduit(List<DtoProduit> res) {
        return new Response(null,null,null,res, null,null, null, null);
    }

    public static Response addDtoResultAllProduit(List<DtoDLProduit> res){
        return new Response(null, null, null, null , res, null, null,
null);
    }

    public static Response addDtoResultMvtCli(List<DtoResultMvtCli> res,
List<DtoResultMvtCliOce> resMvtCliOce, List<DtoResultPcBpContact>
resContact) {
        return new Response(null,null, null,null,null, res, resMvtCliOce,
resContact);
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public List<DtoResult> getRes() {
        return res;
    }

    public void setRes(List<DtoResult> res) {
        this.res = res;
    }

    public List<DtoResultDetails> getResDetails() {
        return resDetails;
    }

    public void setResDetails(List<DtoResultDetails> resDetails) {
        this.resDetails = resDetails;
    }

    public List<DtoResultArt83> getResultArt83() {
        return resultArt83;
    }

    public void setResultArt83(List<DtoResultArt83> resultArt83) {
        this.resultArt83 = resultArt83;
    }

    public List<DtoProduit> getResultProduit() {
        return resultProduit;
    }

    public void setResultProduit(List<DtoProduit> resultProduit) {
        this.resultProduit = resultProduit;
    }

    public List<DtoDLProduit> getResultAllProduit() {
        return resultAllProduit;
    }

```

```

    public void setResultAllProduitt(List<DtoDLProduit> resultAllProduit) {
        this.resultAllProduit = resultAllProduit;
    }
    public List<DtoResultMvtCli> getResultMvtCli() {
        return resultMvtCli;
    }

    public void setResultMvtCli(List<DtoResultMvtCli> resultMvtCli) {
        this.resultMvtCli = resultMvtCli;
    }

    public List<DtoResultMvtCliOce> getResultMvtCliOce() {
        return resultMvtCliOce;
    }

    public void setResultMvtCliOce(List<DtoResultMvtCliOce>
resultMvtCliOce) {
        this.resultMvtCliOce = resultMvtCliOce;
    }

    public List<DtoResultPcBpContact> getResultPcBpContact() {
        return resultMvtContact;
    }

    public void setResultPcBpContact(List<DtoResultPcBpContact>
resultMvtContact) {
        this.resultMvtContact = resultMvtContact;
    }
}

```

RechercheController

```

@Operation(summary = "Permet de recuperer un mvt")
@Parameter(name = "Authorization", description = "Username:password",
required = true, allowEmptyValue = true, in = HEADER)
@PostMapping(value = "/MvtCli", produces =
MediaType.APPLICATION_JSON_VALUE, consumes =
MediaType.APPLICATION_JSON_VALUE)
@ResponseBody
public ResponseEntity<Response> getMvtCliByCriteria(@RequestBody
FeMvtCliCriteria MvtCliCriteria) {
    List<FeMvtCli> result =
rechercheMvtCliService.searchByMvtCliCriteria(MvtCliCriteria);
    LOGGER.info(result);
    List<DccMvtCliOce> resultOce =
rechercheMvtCliService.searchByMvtCliOceCriteria(MvtCliCriteria);
    List<PcBpContact> resultPcBp =
rechercheMvtCliService.searchByPcBpContactCriteria(MvtCliCriteria);
    List<DtoResultMvtCli> resultMvtCli = new ArrayList<>();
    List<DtoResultMvtCliOce> resultMvtCliOce = new ArrayList<>();
    List<DtoResultPcBpContact> resultContact = new ArrayList<>();
    for (FeMvtCli singleResult : result) {
        resultMvtCli.add(new DtoResultMvtCli(singleResult));
    }
    for (DccMvtCliOce singleResult : resultOce){
        resultMvtCliOce.add(new DtoResultMvtCliOce(singleResult));
    }
}

```

```

    for (PcBpContact singleResult : resultPcBp) {
        resultContact.add(new DtoResultPcBpContact (singleResult));
    }

    return new ResponseEntity<> (Response.addDtoResultMvtCli (resultMvtCli,
resultMvtCliOce,resultContact), HttpStatus.OK);
}

```

FeMvtCliRepository

```

package com.aviva.appdiag.repository;

import com.aviva.appdiag.beans.entity.DccMvtCliOce;
import com.aviva.appdiag.beans.entity.FeMvtCli;
import com.aviva.appdiag.beans.entity.PcBpContact;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import javax.transaction.Transactional;
import java.util.List;

@Repository
@Transactional
public interface FeMvtCliRepository extends CrudRepository<FeMvtCli,
Long>{

    @Query (value ="select fe " + " from FeMvtCli fe " + " where
fe.mvtCliId = ?1 " + " order by fe.mvtEnrD desc")
    List<FeMvtCli> findAllByMvtCliId (Number mvtCliId, Pageable pageable);

    @Query (value ="select fe " + " from FeMvtCli fe " + " where fe.mvtMvtC
= 'email' " + " and fe.mvtVal like ?1 " + " order by fe.mvtEnrD desc ")
    List<FeMvtCli> findAllByMvtCliMail (String mvtVal, Pageable pageable);

    @Query (value ="select fe " + " from FeMvtCli fe " + " where fe.mvtMvtC
= 'tel_portable' " + " and fe.mvtVal like ?1 " + " order by fe.mvtEnrD desc
")
    List<FeMvtCli> findAllByMvtCliTel (String mvtVal, Pageable pageable);

    @Query (value ="select dcc " + " from DccMvtCliOce dcc " + " where
dcc.mvtCliId = ?1 " + " order by dcc.mvtEnrD desc")
    List<DccMvtCliOce> findAllByMvtCliIdOce (Long mvtCliId, Pageable
pageable);

    @Query (value ="select dcc " + " from DccMvtCliOce dcc" + " where
dcc.mvtMvtC = 'email' " + " and dcc.mvtVal like ?1 " + " order by
dcc.mvtEnrD desc ")
    List<DccMvtCliOce> findAllByMvtCliOceMail (String mvtVal, Pageable
pageable);

    @Query (value ="select dcc " + " from DccMvtCliOce dcc" + " where
dcc.mvtMvtC = 'tel_portable' " + " and dcc.mvtVal like ?1 " + " order by
dcc.mvtEnrD desc ")
    List<DccMvtCliOce> findAllByMvtCliOceTel (String mvtVal, Pageable
pageable);
}

```

```

        @Query (value ="select pc " + " from PcBpContact pc " + " where
pc.numPers = ?1 " + " order by pc.numMvt desc")
        List<PcBpContact> findAllByContactId(Long numPers, Pageable pageable);

        @Query (value ="select pc " + " from PcBpContact pc" + " where
pc.nomZone = 'email' " + " and pc.valeurZone like ?1 " + " order by
pc.numMvt desc ")
        List<PcBpContact> findAllByContactMail(String nomZone , Pageable
pageable);

        @Query (value ="select pc " + " from PcBpContact pc" + " where
pc.nomZone = 'tel_portable' " + " and pc.valeurZone like ?1 " + " order by
pc.numMvt desc ")
        List<PcBpContact> findAllByContactTel(String nomZone , Pageable
pageable);
    }

```

RechercheMvtCliServiceImpl

```

package com.aviva.appdiag.services;

import com.aviva.appdiag.beans.criterias.FeMvtCliCriteria;
import com.aviva.appdiag.beans.entity.DccMvtCliOce;
import com.aviva.appdiag.beans.entity.FeMvtCli;
import com.aviva.appdiag.beans.entity.PcBpContact;
import com.aviva.appdiag.controllers.RechercheController;
import com.aviva.appdiag.repository.FeMvtCliRepository;
import com.aviva.appdiag.services.interfaces.RechercheMvtCliService;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.PageRequest;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class RechercheMvtCliServiceImpl implements RechercheMvtCliService {

    private static final Log LOGGER =
LogFactory.getLog(RechercheController.class);
    @Autowired
    FeMvtCliRepository feMvtCliRepository;

    @Override
    public List<FeMvtCli> searchByMvtCliCriteria(FeMvtCliCriteria criteria)
    {
        List<FeMvtCli> resultMvtCli = null;
        if (criteria.getMvtCliId() != null) {
            LOGGER.info(criteria.getMvtCliId());
            resultMvtCli =
feMvtCliRepository.findAllByMvtCliId(criteria.getMvtCliId(),
PageRequest.of(0, 99));
        } else if (!criteria.getMvtValEmail().isEmpty() &&

```

```

criteria.getMvtValEmail().contains("@")) {
    resultMvtCli =
feMvtCliRepository.findAllByMvtCliMail(criteria.getMvtValEmail(),
PageRequest.of(0, 99));
    } else if (!criteria.getMvtValTel().isEmpty()) {
        resultMvtCli =
feMvtCliRepository.findAllByMvtCliTel(criteria.getMvtValTel(),
PageRequest.of(0, 99));
    }

    //FIXME deconnecter l'entity
    return resultMvtCli;
}

public List<DccMvtCliOce> searchByMvtCliOceCriteria(FeMvtCliCriteria
criteria) {
    List<DccMvtCliOce> resultMvtCliOce = null;
    if(criteria.getMvtCliId() != null){

        resultMvtCliOce =
feMvtCliRepository.findAllByMvtCliIdOce(criteria.getMvtCliId(),
PageRequest.of(0, 99));
    }
    else if(!criteria.getMvtValEmail().isEmpty() &&
criteria.getMvtValEmail().contains("@"))
    {
        resultMvtCliOce =
feMvtCliRepository.findAllByMvtCliOceMail(criteria.getMvtValEmail(),
PageRequest.of(0, 99));
        LOGGER.info("Yoooooooo ! c'est muah" + resultMvtCliOce);
    }
    else if (!criteria.getMvtValTel().isEmpty() )
    {
        resultMvtCliOce =
feMvtCliRepository.findAllByMvtCliOceTel(criteria.getMvtValTel(),
PageRequest.of(0, 99));
    }
    //FIXME deconnecter l'entity
    return resultMvtCliOce;
}

public List<PcBpContact> searchByPcBpContactCriteria(FeMvtCliCriteria
criteria) {
    List<PcBpContact> resultBpContact = null;
    if(criteria.getMvtCliId() != null){
        resultBpContact =
feMvtCliRepository.findAllByContactId(criteria.getMvtCliId(),
PageRequest.of(0, 99));
    }
    else if(!criteria.getMvtValEmail().isEmpty() &&
criteria.getMvtValEmail().contains("@"))
    {
        resultBpContact =
feMvtCliRepository.findAllByContactMail(criteria.getMvtValEmail(),
PageRequest.of(0, 99));
    }
    else if (!criteria.getMvtValTel().isEmpty() )
    {
        resultBpContact =

```

```
feMvtCliRepository.findAllByContactTel(criteria.getMvtValTel(),
PageRequest.of(0,99));
    }
    //FIXME deconnecter l'entity
    return resultBpContact;
}
}
```

Recherche d'un client

Numéro client :

Mail client :

Numéro mobile :

UN champ doit être renseigné

Lancer la recherche

Compte Client

Url Démat

Art83

GDPR

Ref Produit

Anti-Fraude

Recherche d'un client

Numéro client :

6132369

Mail client :

Numéro mobile :

Lancer la recherche

Numéro client	Mail client	Tel client
6132369	64	64
6132369		
6132369	2	2
6132369	MA	MA

1 (current)

2

3

4

5

25

Numéro client	Mail client	Tel client
6132369	64	64
6132369		
6132369	2	2
6132369	MA	MA

1 (current)

2

3

4

5

25

Numéro client	Mail client	Tel client
---------------	-------------	------------

Recherche d'un client

Numéro client :

6132369

Mail client :

Numéro mobile :

Lancer la recherche

Numéro client	Mail client	Tel client
6132369	01	01
6132369	0	0
6132369		
6132369		

1 2 3 (current) 4 5 25 2

Recherche d'un client

Numéro client :

Mail client :

zozo@toto.fr

Numéro mobile :

Lancer la recherche

Numéro client	Mail client	Tel client
6502916	zozo@toto.fr	zozo@toto.fr
7479756	zozo@toto.fr	zozo@toto.fr
101130	zozo@toto.fr	zozo@toto.fr
5726498	zozo@toto.fr	zozo@toto.fr

1 2 3 (current) 4 5 25 2

Numéro client	Mail client	Tel client
7651993	zozo@toto.fr	zozo@toto.fr

1 (current) 2

Numéro client	Mail client	Tel client
---------------	-------------	------------

2

Votre espace de recherche

Compte Client

Url Démat

Art83

GDPR

Ref Produit

Anti-Fraude

Recherche d'un client

Numéro client :

Mail client :

Numéro mobile :

0512345678

Lancer la recherche

Numéro client	Mail client	Tel client
6531466	0512345678	0512345678

1 (current)

Numéro client	Mail client	Tel client
6531466	0512345678	0512345678

1 (current)

Numéro client	Mail client	Tel client
---------------	-------------	------------

N° client	Clé	Valeur	Modifié par	Date de traitement	Date d'envoi
2894921	tranche_patrim_fin	1	Mr TRIXXXXX YVXXXXXX	2022-10-27	2022-10-27
2894921	profession	DIRECTEUR SERVICE TECHNIQ	TRIXXXXX YVXXXXXX	2022-10-27	2022-10-27
2894921	type_profession	1	TRIXXXXX YVXXXXXX	2022-10-27	2022-10-27
2894921	secteur_activite	18	TRIXXXXX YVXXXXXX	2022-10-27	2022-10-27

« 1 (current) 2 3 4 5 ... 25 »

Valeur	N° client	Date d'envoi	Clé	Date de traitement
en ligne	2894921	2022-12-06	optim_gestion	2022-12-06
	2894921	2022-11-21	fatca_code	2022-11-21
	2894921	2022-11-21	fatca_identifiant	2022-11-21
Non	2894921	2022-11-21	fatca_statut	2022-11-21

« 1 (current) 2 3 4 5 ... 20 »