

COMPTE-RENDU SITUATION DE STAGE

STAGE 2^{ème} année – 2TSSIOD année scolaire 2021 – 2022 – Lycée
CHEVROLLIER

Du 28 Février au 16 Avril

SOMMAIRE

COMPTE-RENDU SITUATION DE STAGE.....	1
SOMMAIRE	2
INFORMATIONS SUR L'ENTREPRISE.....	3
Informations.....	3
Contexte de la situation professionnelle	3
INFORMATIONS SUR LE STAGE	3
Informations.....	3
Intitulé de la situation professionnelle	4
REALISATION DU PROJET	4
Travail réalisé lors des 4 premières semaines	4
L'équipe.....	4
Mise en place du projet	4
Le projet	5
Travail réalisé lors des 3 dernières semaines	6
Le projet	6
OUTILS UTILISES.....	10
DIFFICULTES RENCONTREES ET SOLUTIONS APPORTEES.....	10
CONCLUSION	11
Conclusion sur la première partie.....	11
Conclusion sur la dernière partie.....	11

INFORMATIONS SUR L'ENTREPRISE

Informations

Entreprise : PRIOS

Activité principale et activités annexes : Editeur et intégrateur de logiciels de gestion

Nombre de salarié global : 70 collaborateurs

Mon équipe : Andry-Vincent (Tuteur) / Florian (Directeur Technique)

Contexte de la situation professionnelle

Prios est une entreprise éditeur et intégrateur de logiciels de gestion d'entreprise pour les PME/PMI et groupes. Ils conçoivent des solutions logicielles dédiées au secteur de l'agriculture, aux métiers de la mode, à la boisson, à la menuiserie, à la gestion comptable et financière ainsi qu'aux ressources humaines.


L'entreprise possède 30 ans d'expérience avec plus de 1200 entreprises clientes ainsi qu'environ 10 000 utilisateurs. Elle possède également 70 collaborateurs dont 17% en recherche et développement.

INFORMATIONS SUR LE STAGE

Informations

Période de réalisation : du 28 Février au 25 Mars

Nombre de semaines : 7 semaines

	<p>STAGE 2ème année</p> <p>Prios / Editeur et intégrateur de logiciels de gestion</p> <p>Parcours SLAM</p>
---	--

Modalité : Seul

Intitulé de la situation professionnelle

Quand je suis arrivé, l'entreprise m'a confié un projet sur l'ensemble des 7 semaines de mon stage. Je dois rassembler des données provenant d'apis pour ensuite les envoyer au service administratif de l'entreprise. Les données sont les comptes en statut « client » avec les contacts qui sont associés avec ce compte en statut « en activité » et les contrats associés à chaque compte en statut « en cours ». Des réunions ont été mises en place chaque semaine avec Andry afin qu'il puisse suivre l'évolution du projet et ainsi pouvoir m'aider si j'ai des problèmes.

REALISATION DU PROJET

Travail réalisé lors des 4 premières semaines

L'équipe

Pour la réalisation du projet, je suis entouré de Florian et Andry-Vincent. Florian est le directeur technique de l'entreprise. Il fait en quelque sorte l'intermédiaire entre le service administratif et Andry-Vincent et moi. C'est lui qui nous explique au début ce qu'il faudrait que je fasse. Andry-Vincent est mon tuteur. C'est lui qui me suit sur l'ensemble de mon projet. J'ai des réunions avec lui toutes les semaines pour suivre l'avancement du projet. Même si Andry-Vincent me suit, je suis le seul à travailler sur le projet.

Mise en place du projet

Durant la première réunion Andry-Vincent m'explique les logiciels ainsi que le langage informatique que je vais devoir utiliser. Je dois donc me familiariser avec Java ainsi qu'avec Springboot. On installe Spring Tool Suite comme IDE car celui-ci est plus pratique dans la création de projet spring. Puis il me montre le site prios.myportal où sont recensé toutes les données des comptes, contacts et contrats. C'est aussi sur ce site que se trouve la documentation d'api.

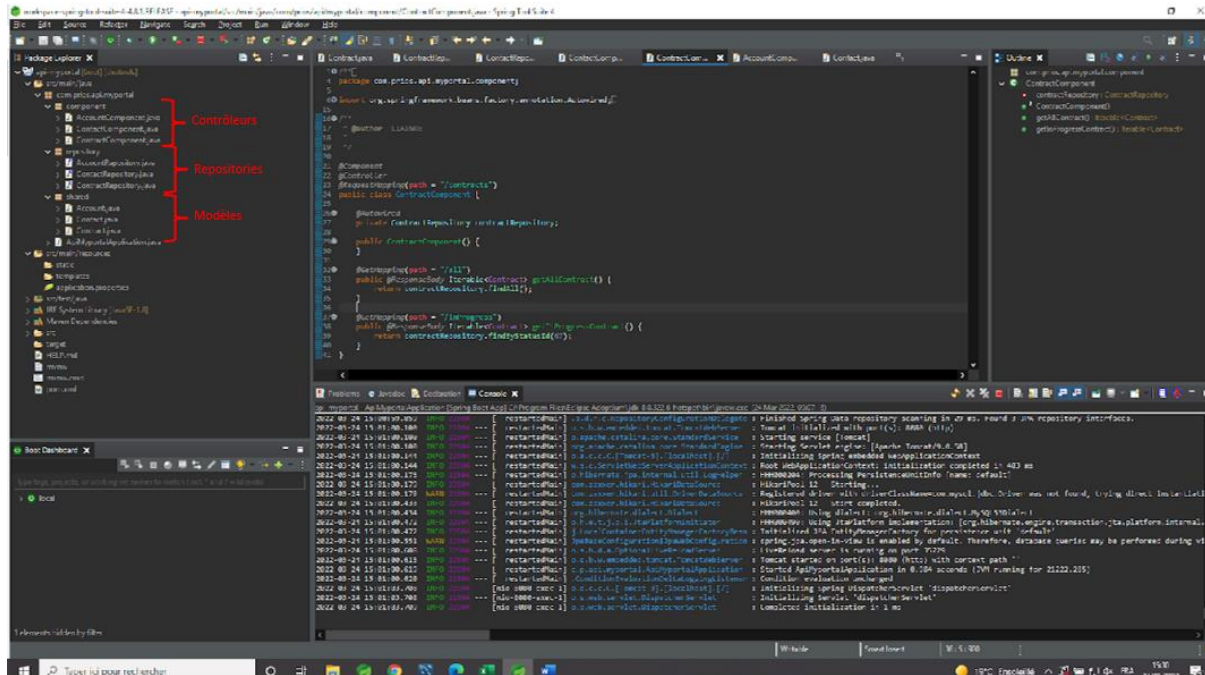
Telephone	Adresse 1	Ville	Statut	Affectation client/RH	Contrat de Support	Contact(s) de facturation	Envoyer automatiquement les factures émises ?	Hébergement	Interlocuteur principal	Action
02 86 52 44 22	49 Route d'Angers	MONTELEU	Inactif	Prior F & R	-	-	NON	-	PONROY Laurent	🔍 ⚙️
09 72 29 97 74	9 Rue James Watt	BRAY-SUR-SEINE	Fournisseur	-	-	-	NON	-	-	🔍 ⚙️
06 72 12 39 19	417 Rue du Valentin	ANGERS	Prospect	-	-	-	NON	-	ULI TIRIL MARCOTTE	🔍 ⚙️
02 40 41 09 28	-	-	Organisme	-	-	-	NON	-	-	🔍 ⚙️
01 40 00 80 27	96 Boulevard Houssin	PARIS	Prospect	-	-	-	NON	-	GOIN Fabrice	🔍 ⚙️
01 39 07 20 75	57 rue des Chantiers	Angers	Prospect	-	-	-	NON	-	DUPANLON Franck	🔍 ⚙️
02 51 80 80 66	10 Boulevard du 30/08	Saint-Florentin	Fournisseur	-	-	-	NON	-	-	🔍 ⚙️
02 41 53 04 00	21, Saurer Chaus	SAUMUR-EN-JUVE	Inactif	Prior F & R	-	-	NON	-	BRUNEAU Charles	🔍 ⚙️
01 89 87 43 24	-	TEIGNES	Prospect	-	-	-	NON	-	LEFFERT	🔍 ⚙️
05 50 48 06 06	1840 Avenue de Cédric	ANGERS	Inactif	Prior F & R	-	-	NON	-	GROBENIN Gilles	🔍 ⚙️
+33 1 52 02 30 39	Deux Quai 182	44 Delf	Fournisseur	-	-	-	NON	-	-	🔍 ⚙️
06 72 24 20 28	12 Place Georges Pompidou	NOUVEAU LARSEN	Fournisseur	-	-	-	NON	-	-	🔍 ⚙️
02 40 92 34 52	72 Rue Robert	LA CHAPELLE-SUR-LOGNON	Fournisseur	00000000	-	-	NON	-	LAURENT Gilbert	🔍 ⚙️

Le projet

L'objectif du projet est de récupérer les données des apis des comptes, des contacts et des contrats afin de les transmettre au service administration de l'entreprise. Florian nous a expliqué qu'il fallait récupérer uniquement les comptes en statut « client » avec les contacts en statut « en activité » et les contrats en statut « en cours ». Or il faut être connecté pour pouvoir avoir accès aux apis. J'ai donc dû gérer l'authentification aux apis. L'authentification est une authentification de base Rest Template.

Ensuite j'ai commencé par la récupération du service restful. J'ai donc commencé par créer un modèle pour chaque type de données (comptes, contacts, contrats) afin de récupérer les données de l'api. J'ai ensuite créé des contrôleurs pour connecter le java avec les apis. Après ceci fini, j'ai commencé à faire le reload automatique pour que le chargement ne dure pas trop longtemps. Cependant, je me suis rendu compte que toutes les informations qu'avait demandées Florian n'étaient pas présentes dans les apis.

Après concertation lors d'une réunion, nous avons décidé de laisser tomber les apis pour récupérer les données directement depuis la base de données. Je devais donc tout reprendre depuis le début. J'ai dû me connecter à la base de données via MySQL Workbench pour voir à quoi elle ressemble et trouver les bonnes tables. Pour accéder aux données MySQL de la base de données via l'application java, j'ai utilisé Spring Data JPA. J'ai donc refait les modèles en fonction des données de la base de données. Ensuite j'ai créé les repositories pour créer les méthodes et ainsi récupérer les comptes uniquement s'ils sont « client », si les contacts sont « en activité » et si les contrats sont « en cours ». Les contrôleurs récupèrent les méthodes des repositories pour afficher les données demandées.



Travail réalisé lors des 3 dernières semaines

Le projet

Afin d'afficher les données récupérer sur la base de données, j'ai utilisé des web services. Les données affichées changeront en fonction du nom du répertoire passée dans l'url. Sans ajout de paramètre à l'url, les comptes, les contacts et contrats ne sont pas triés. Il y a alors les données qui ne correspondent pas avec le statut demandé (Exemple : <http://localhost:8080/count>). Si en répertoire on a count, se sera alors tous les comptes qui s'afficheront, avec contact se sera tous les contacts et avec contract se sera tous les contrats. Si en revanche on passe un paramètre après le répertoire, les données seront triées (Exemple : <http://localhost:8080/count?client>). Le paramètre permet de trier les données en fonction du statut de la donnée, comme ce qui était demandé lors des réunions. Client pour compte, inActivity pour contact et inProgress pour contract.

Une fois ceci terminé, j'ai remis toutes les variables aux normes et commenté tout le code.

```
/**
 * @param inProgress indique les contrats qui sont en cours
 * @return retourne tous les contrats de base, si inProgress est en paramètre on
 *         affiche les contrats en cours
 */
@GetMapping()
public @ResponseBody Iterable<Contract> getContract(
    @RequestParam(name = "inProgress", required = false) Boolean inProgress) {
    if (inProgress != null && inProgress.equals(true)) {
        return getInProgressContract();
    }
    return contractRepository.findAll();
}
```

Lors de la réunion hebdomadaire avec Andry, nous avons vu qu'il me restait à faire les tâches planifiées mais ne pas recharger les comptes déjà envoyés puis l'ajout sur l'annuaire eureka. C'est là que les micro-services communiquent.

Je ne pouvais pas recharger tout le code à chaque fois. J'ai donc décidé de recharger uniquement les méthodes concernées. Les méthodes devaient se recharger une fois par jour. J'ai réalisé les tâches planifiées pour les comptes, les contacts et les contrats.

```
/**
 * @return toutes les 5 secondes les contrats en cours
 */
@Scheduled(fixedDelay = 5000)
public @ResponseBody Iterable<Contract> getInProgressContract() {
    return contractRepository.findById("67");
}
```

Juste après avoir fini les tâches planifiées, Andry et Florian sont venus me voir. Ils m'ont dit que finalement ce serait une meilleure idée si les données étaient dans un fichier CSV. J'ai donc abandonné les web services.

J'ai créé une méthode qui permet de récupérer les données en fonction de la date du jour pour ne pas récupérer toutes les données à chaque fois. Je déclare une date à nulle, si cette date reste à nulle je récupère toutes les données en fonction de leurs statuts respectifs puis je fais passer la date à la date du jour. Sinon on récupère toutes les données en fonction de leurs statuts respectifs qui ont été créés ou modifiés après date.

Je précise que toutes les illustrations qui vont suivre proviennent du code des comptes et que c'est le même principe pour les contacts et contrats.

```
@Autowired
private AccountRepository accountRepository;

/**
 * date déclaré à null
 */
Date date = null;

/**
 * @throws IOException Méthode qui se répète toutes les 10 secondes. Si la date
 * est null alors on récupère tous les comptes en statut
 * client qu'on envoie dans le csv. On fait prendre date à
 * la date du jour. Sinon on récupère tous les compte qui
 * ont été créé et modifié après date ainsi qu'en statut
 * client et on envoie dans le csv. On fait prendre date à
 * la date du jour.
 */
@Scheduled(fixedDelay = 10000)
public void accountInCsv() throws IOException {
    if (date == null) {
        List<Account> accounts = accountRepository.findByStatusId("317");
        csv(accounts);
        date = new Date();
    } else {
        List<Account> accountsUpdate = accountRepository.findByUpdatedAtAfterAndStatusId(date, "317");
        csv(accountsUpdate);
        List<Account> accountsCreate = accountRepository.findByCreatedAtAfterAndStatusId(date, "317");
        csv(accountsCreate);
        date = new Date();
    }
    System.out.println("ok");
}
```

Cette méthode renvoie les données récupérées à une autre méthodes. Cette autre méthode créer un fichier csv avec son numéro et récupère les données pour elle-même les mettre dans le fichier CSV qu'elle vient de créer.

```
/**
 * count déclaré à 0
 */
int count = 0;

/**
 * @param accounts est la liste des comptes passée en paramètre
 * @throws IOException On créer un nouveau fichier csv avec comme nom count. On
 * ajoute les données de chaque compte qu'on aura récupéré
 * en paramètre dans le fichier. Puis on incrémente count
 */
public void csv(List<Account> accounts) throws IOException {

    CSVWriter write = new CSVWriter(new FileWriter("dataAccount/accounts_" + count + ".csv"));

    String title[] = { "Id", "Name", "Code", "Email Adress", "Status Id", "Siret", "Eu Vat", "Accounting Account",
        "Created At", "Updated At" };

    write.writeNext(title);

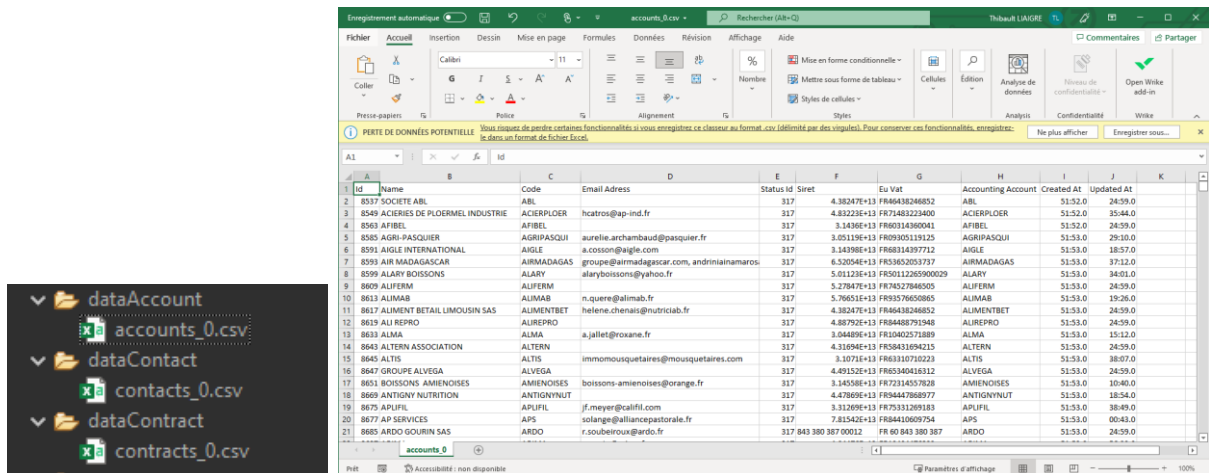
    List<String[]> accountsData = new ArrayList<String[]>();

    for (Account account : accounts) {
        String data[] = { Integer.toString(account.getId()), account.getName(), account.getCode(),
            account.getEmailAddress(), account.getStatusId(), account.getSiret(), account.getEuVat(),
            account.getAccountingAccount(), account.getCreatedAt().toString(),
            account.getUpdatedAt().toString() };

        accountsData.add(data);

        write.writeAll(accountsData);
        write.flush();
    }
    count++;
}
```


Il y a donc un fichier de comptes, contacts et contrats créé tous les jours dans le dossier indiqué dans la méthode.



Id	Name	Code	Email Address	Status Id	Siret	Eu Vat	Accounting Account	Created At	Updated At
2	8537 SOCIETE ABL	ABL		317	4.38247E+13	FR040438246852	ABL	51:52:0	24:59:0
3	8549 ACIERIES DE PLOERMEL INDUSTRIE	ACIERPLOER	hcatros@ap-ind.fr	317	4.83223E+13	FR73483223400	ACIERPLOER	51:52:0	35:44:0
4	8565 AFIREL	AFIREL		317	3.14360E+13	FR06314360041	AFIREL	51:52:0	24:59:0
5	8585 AGRI-PASQUIER	AGRI-PASQUIER	aurelie.archambaud@pasquier.fr	317	3.05119E+13	FR09305119125	AGRI-PASQUIER	51:53:0	29:10:0
6	8591 AIGLE INTERNATIONAL	AIGLE	a.cosson@aigle.com	317	3.14398E+13	FR08314397712	AIGLE	51:53:0	18:57:0
7	8593 AIR MADAGASCAR	AIR-MADAGASCAR	groupe@airmadagascar.com, andriainamamos	317	6.52054E+13	FR33652053737	AIR-MADAGASCAR	51:53:0	37:12:0
8	8599 ALARY BOISSONS	ALARY	alaryboissons@yahoo.fr	317	5.01123E+13	FR0511230590029	ALARY	51:53:0	34:01:0
9	8609 ALIFERM	ALIFERM		317	5.27847E+13	FR78327846505	ALIFERM	51:53:0	24:59:0
10	8613 ALIMAB	ALIMAB	n.querre@alimab.fr	317	5.76651E+13	FR09357665085	ALIMAB	51:53:0	19:26:0
11	8617 ALIMENT BETAIL LIMOUSIN SAS	ALIMENTBET	helene.chenais@nutriciab.fr	317	4.38247E+13	FR040438246852	ALIMENTBET	51:53:0	24:59:0
12	8619 ALI REPRO	ALI-REPRO		317	4.86792E+13	FR040486791948	ALI-REPRO	51:53:0	24:59:0
13	8633 ALMA	ALMA	a.jallet@roxane.fr	317	3.04489E+13	FR10402571889	ALMA	51:53:0	15:12:0
14	8643 ALTERN ASSOCIATION	ALTERN		317	4.31894E+13	FR08431894215	ALTERN	51:53:0	24:59:0
15	8645 ALTIS	ALTIS	immomousquetaires@mousquetaires.com	317	3.10711E+13	FR03310710223	ALTIS	51:53:0	38:07:0
16	8647 GROUPE ALVEGA	ALVEGA		317	4.49152E+13	FR03340416112	ALVEGA	51:53:0	24:59:0
17	8651 BOISSONS AMIENOISES	AMIENOISES	boissons-amienoises@orange.fr	317	1.44558E+13	FR72314557928	AMIENOISES	51:53:0	10:40:0
18	8669 ANTIQNY NUTRITION	ANTIQNYNUT		317	4.47869E+13	FR04447869777	ANTIQNYNUT	51:53:0	18:54:0
19	8675 APIFIL	APIFIL	jf.meyer@califil.com	317	3.31269E+13	FR73331269183	APIFIL	51:53:0	38:49:0
20	8677 AP SERVICES	APS	solange@alliancepastorale.fr	317	7.81542E+13	FR04011809754	APS	51:53:0	00:43:0
21	8665 ARDO GOURIN SAS	ARDO	r.soubeyrou@ardo.fr	317	843 380 387 00012	FR 80 843 380 387	ARDO	51:53:0	24:59:0

Après avoir fini tout ceci, j'ai dû externaliser les variables qui sont susceptibles de changer à l'avenir ainsi que la connexion à la base de données. Pour ceci j'ai créé un nouveau fichier properties à la racine du projet. J'y ai ajouté la connexion à la base de données, ainsi que certaines variables comme le statusId de count, contact et contrat.

```

1 spring.jpa.hibernate.ddl-auto=none
2 spring.datasource.url=jdbc:mysql://db01.myportal.fr/prios
3 spring.datasource.username=prios_s
4 spring.datasource.password=EgGCleSVx
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
7 date=null
8 count=0
9
10 accountStatusId=317
11 accountPath=dataAccount
12
13 contactStatusId=381
14 contactPath=dataContact
15
16 contractStatusId=67
17 contractPath=dataContract

```

Ensuite je suis allé présenter le projet à Florian lors d'une réunion prévue à cet effet.

OUTILS UTILISES

Prios.myportal : pour l'accès aux comptes, contacts et contrats

Prios.myportal/api : pour l'accès à la documentation des apis et l'accès aux apis

MySQL Workbench : pour l'accès à la base de données

Postman : pour tester les web services

Java / Spring boot : pour le langage et le Framework utilisés

Spring Tools Suite : pour l'IDE utilisé

Microsoft Teams / Outlook : pour les conversations inter-entreprise

DIFFICULTES RENCONTREES ET SOLUTIONS APORTEES

Se familiariser avec un nouveau langage et comprendre un nouveau Framework

Le déploiement d'un projet Spring

Comprendre comment sont reliés les comptes avec les contacts et les contrats

Récupérer directement les données à la base de données avec Java

Ajouter les données dans le fichier csv

Pour n'importe quelle difficulté, j'ai demandé de l'aide à mon tuteur qui venait m'aider et m'expliquer comment ça fonctionne. J'essayais de ne pas rester bloqué dessus trop longtemps. C'est souvent lors des réunions hebdomadaires qu'il me montrait un exemple de ce que je devais faire pour que je puisse bien comprendre et ainsi le refaire de mon côté. Il m'envoyait aussi des liens (via Microsoft teams) de site qui pouvait m'aider à la réalisation des tâches.

CONCLUSION

Conclusion sur la première partie

Pour conclure sur ces 4 premières semaines, le début du projet était assez complexe pour moi. Le fait de devoir voir un nouveau langage mais surtout le déploiement de Springboot demandait beaucoup d'attention. Plus les semaines ont avancé et plus je me suis senti à l'aise sur ce Framework ce qui est un bon point pour la suite de mon stage.

Il me reste à trier les contacts et les contrats en fonction du compte de référence, à gérer la recharge automatique toutes les 5 secondes de récupération des données pour que cela ne prenne pas trop de temps à chaque fois que l'on charge les données. Il me reste aussi à déployer le logiciel pour qu'il fonctionne correctement.

Conclusion sur la dernière partie

Pour conclure sur les 3 dernières semaines de mon stage, j'ai trouvé ces semaines très intéressantes car j'ai pu utiliser plusieurs approches avec l'utilisation des données. Je suis passé d'utilisation d'API à web services ou encore utiliser un fichier csv. Le projet étant terminé, il ne me reste plus rien à faire dessus.

Le point fort de ce projet c'est qu'il m'a permis de me rendre compte que la réalisation d'un projet peut totalement changer en fonction des difficultés rencontrées. L'objectif reste le même mais la démarche pour y arriver varie. C'est aussi dans ce genre de projet que je me suis rendu compte de l'importance des réunions hebdomadaires avec l'équipe. Cela nous permet d'échanger sur les problèmes rencontrés pour trouver une solution adéquate.