



Annexe du rapport de stage : RESTLET

APISpark

Etudiante : Marina BLIN
Tutrice : Solen QUINIOU
Maître de stage : Manuel BOILLOD



LaTeX

Tableau de bord

Lundi 13 avril

Matin

- Réunion sur ce qui a été fait le vendredi.
- Installation de **Linux mint** en dual boot avec **windows 7**. (Avec Constant)
- Lecture du tutoriel **AngularJS** sur le site officiel (<https://docs.angularjs.org>) :
 - **0- Bootstrapping**
 - **1- Static Template**
 - **2- Angular Template**
 - **3- Filtering Repeater**

Après-midi

- Installation de **NodeJS**, **sublime text** et de **git**.
- Lecture et apprentissage des 7 premiers chapitres du tutoriel d'angularJS :
 - **0- Bootstrapping**
 - **1- Static Template**
 - **2- Angular Template**
 - **3- Filtering Repeater**
 - **4- Two way Data Binding**
 - **5- XHRS & Dependency Injection**
 - **6- Templating Links & images**

Mardi 14 avril

Matin

- Petite réunion de 15 minutes pour faire le point sur ce qui a été fait la veille.
- Suite de l'apprentissage d'AngularJS sur le même site que lundi 13 avril.
 - **7- Routing & Multiple Views**
 - **8- More Templating**
 - **9- Filters**
 - **10- Event Handlers**
 - **11- Rest and Custom Services**

Après-midi

- Suite et fin du tutoriel AngularJS :
 - **12- Applying Animations**
- Retour sur le onzième point du tutoriel pour approfondir les points non compris.
- Quelques essais en plus sur angular à partir du code donné dans le tuto.

Mercredi 15 avril

Matin

- Petite réunion de 15 minutes pour faire le point sur ce qui a été fait la veille.
- Réinstallation de Linux Mint en dual boot avec windows 7 sur le nouvel ordinateur portable.
- Réinstallation de Git, Sublime text et NodeJS.
- Découverte d'APIspark (avec Chong)

Après-midi

- Apprentissage d'APIspark à partir du tutoriel se trouvant à l'adresse suivante :
<http://restlet.com/technical-resources/apispark/tutorials>
 - **First API from scratch**
- Installation de **mysql**.
 - **SQL database to API** (Quelques problèmes pour trouver le nom de l'host. Aide de Jonathan)

Jeudi 16 avril

Matin

- Petite réunion de 15 minutes pour faire le point sur ce qui a été fait la veille.
- Continuation du tutoriel d'APIspark :
 - **Google and invoke a web API**
 - **Management of an existing API**
- Installation de **maven**.
 - **Documentation for an existing API** (Quelques problèmes avec maven. Aide de Guillaume)
 - **Parse backend to API**

Après-midi

- Continuation du tutoriel d'APIspark :
 - **Parse backend to API**
 - **Firestore backend to API**
 - **GitHub repository to API**
 - **AWS S3 bucket to API**

- Retour sur quelques points mals compris d'APIspark.
 - **Management on an existing API**
- Clonnage du git Restlet (Droit par Thierry)
- Continuation d'apprentissage AngularJS le temps des clonnages de fichiers git
- Installation Docker (Avec Constant)

Vendredi 17 avril

Matin

- Petite réunion de 15 minutes pour faire le point sur ce qui a été fait la veille.
- Entrainement et approfondissement d'AngularJS.
- Entrainement et approfondissement d'APIspark.
- Testes de la nouvelle version de APIspark en préprod.

Après-midi

- Entrainement et approfondissement d'AngularJS. (A partir du fichier git d'Angular, ajout de fonctionnalités)
- Mise en relation entre AngularJS et APIspark.

Lundi 20 avril

Matin

- Petite réunion pour savoir ce qui a été fait vendredi.
- Suite de la mise en relation entre AngularJS et APIspark.
- Réunion de mise au point sur l'ensemble de projets (Abyss, parachute, bridge et Engin).

Après-midi

- Démonstration d'APIspark 3.7.
- Suite de la mise en relation entre Angular et APIspark.
- Mise en place des objectifs pour les 10 prochains jours.

Mardi 21 avril

Matin

- Petite réunion de mise au point sur les 10 jours à venir.
- Suite de la mise en relation entre Angular et APIspark. (Récupération du JSon) (Aide de Guillaume)

Après-midi

- Suite de la mise en relation entre Angular et APIspark. (Utilisation des fichiers récupérés).
- Installation d'**Eclipse**. (Avec Constant)
- Installation de l'**environnement de développement**. (Avec Constant)

Mercredi 22 avril

Matin

- Petite réunion sur ce qui a été fait la veille.
- Suite de la mise en relation entre Angular et APIspark. (Aide de Guillaume et Pascal)
- Mise à disposition d'une adresse mail de la société (Par Manuel)
- Recherche d'informations sur les promise d'AngularJS. (Conseil de Guillaume)

Après-midi

- Suite de la recherche d'informations sur les promise.
- Réglage du problème entre APIspark et AngularJS. (Aide de Manuel)
- Explications pour lancer l'environnement de développement. (Avec Constant et Manuel)
- Explications de l'utilisation de l'environnement de développement. (Avec Constant et Manuel)

Jeudi 23 avril

Matin

- Petite réunion sur ce qui a été fait la veille.
- Première mission :
 - **Utilisation de l'environnement de développement**
 - **Se renseigner sur l'issue #3331**
 - **Créer une branche git**
 - **Corriger l'issue**
 - **Faire une pull request**
- Utilisation de l'environnement de développement
- Renseignement sur l'issue #3331 (Faire en sorte que les listes déroulantes s'affichent pareil sur firefox et sur chrome)
- Création d'une branche git
- Commencement de la correction de l'issue
- Recherche d'informations sur la compatibilité entre chrome et firefox avec le framework Bootstrap.

Après-midi

- Continuation recherche de solutions pour l'issue #3331.
- Changement des listes déroulantes en "boutons" déroulants.

Vendredi 24 avril

Matin

- Petite réunion sur ce qui a été fait la veille.
- Suite de la mission : renseignement sur normalize.css (Déjà présent)
- Discussion avec Pascal à propos de différentes pistes pour régler le problème d'item impossible à sélectionner.
- Discussion avec Nicolas à propos de Bootstrap ui.

Après-midi

- Suite de la résolution du problème du bouton déroulant. (Aide Nicolas)
- Redirection vers ui-select. (Aide Nicolas)
- Fonctionnement des listes déroulantes avec les ui-select.

Lundi 27 avril

Matin

- Petite réunion de 15 minutes sur ce qui a été fait vendredi.
- Suite des listes déroulantes :
 - Correction de l'input. Lorsque l'utilisateur cherche quelque chose dans la liste déroulante, il peut faire une recherche. Si le mot est présent dans la liste, il est mis en gras.
 - Recherche de solutions pour la correction des listes de création d'API web, d'Entity store et de File store :
 - ⇒ A partir du code HTML/CSS.
 - ⇒ Ajout d'une fonction dans le fichier javascript.
 - Discussion avec Pascal sur le code en lui même.
 - ⇒ Explications de la création du projet.
 - ⇒ Explications sur Yeoman. (grunt et bower)
 - Explication de pourquoi on utilise un ui-select. (Avec Pascal)
 - Il existe trois formes de liste déroulante : celles avec les boutons, celles avec la bague select

Après-midi

- Résolution du problème de l'affichage des options de la liste déroulante pour la création d'une webAPI.
- Résolution du problème d'affichage des listes déroulantes non conforme sous le navigateur Firefox.
- Recherche de solutions pour le problème de "suppression" d'une option de la liste déroulante lors de la création d'une webAPI.
- Solution non gardée car trop coûteuse.
- Suppression de la branche git créée pour l'issue #3331.
- Retour sur la branche 3.8.

Mardi 28 avril

Matin

- Attribution de l'issue #3304.
- Lecture de l'issue : afficher un message d'alerte si le champs "Mapped from" est vide.
- Création d'une branche git.
- Petite réunion sur ce qui a été fait la veille.
- Relecture par Pascal de la solution apportée au problème de l'issue #3331.
- Quelques problèmes avec l'environnement de dev impossible à lancer, du au changement de branche. (Vu avec Constant)
- Réglage du problème du lancement de l'environnement de dev résolue avec Constant.
- Résolution de l'issue #3304 : affichage d'un message d'erreur si le champs obligatoire est vide.
- Installation de **Webstorm**. (IDE web utilisé par les employés de Restlet)
- Commencement du rapport de stage. (Partie "mission")

Après-midi

- Explications de tout le déroulement d'une correction d'issue avec Pascal. (Utilisation du système des branches et des pull request)
- Attribution des issues par Manuel.

Mercredi 29 avril

Matin

- Prise de connaissances des issues pour la mission de stage.
- Listes des issues :
 - #778 : Google Spreadsheet wrapper : problem with Add Entities
 - #853 : Wrappers : review "add, update, delete Entities" feature
 - #1053 : GSheet wrapper : cannot add worksheet/entity after the store has been created
 - #1423 : Bug : with wrapper spreadsheet and multiple spreadsheets
 - #1884 : Entity dependent elements aren't deleted when deleteing entities from imports
 - #2476 : GSheet wrapper : property name is not editable
 - #2639 : NPE on DataStoreWrapperServiceImpl.getGoogleSheetDependencyEntities
 - #3048 : Cannot edit the pk of an entity from a Google Spreadsheet or the pk is not valuated automatically in the browser
 - #3050 : Tutorial 'Expose data from a Google Spreadsheet as an API' : change worksheet Name and screen-shots do not match with current version
 - #3245 : NullPointerException when deploying GSheet
 - #3302 : GSheet wrapper - cannot browse all entities in case of import of several gsheets
 - #3384 : Gsheet Wrapper - Finding a sheet in the drop-down menu is difficult
 - #3449 : GoogleSpreadsheet - Support entity and property with name not in camel case
 - #3457 : Enhance GSheet wrapper to support other property types than just String

→ **#3461** : Google spreadsheet - If two spreadsheet have the same names, apispark take the first one

- Petite réunion pour savoir ce qui a été fait hier.
- Analyse des issues attribuées.
- L'intégralité des tickets à régler porte sur le Googlespread Sheet d'APIspark.

Après-midi

- Commencement par l'issue **#3050** : changement de mots dans le tutoriel. Redirection vers Fabienne.
- Continuation avec l'issue **#3384** : Permettre à l'utilisateur de faire une recherche dans les import de Google-sheet.
- Issue **#1884** fermée.
- Recherche de solutions pour l'issue **#3384** avec AngularJS.
- Application d'un filtre grace à AngularJS.
- Discussion avec Nicolas sur la raison du non fonctionnement du filtre.
- Mise à notre disposition d'un seul objet récupérant toutes les sheets.
- Découvert de lodash.
- Application de la fonction `_.value` pour ne prendre que les valeurs qui nous intéresse.
- Recherche fonctionnelle.

Jeudi 30 avril

Matin

- Réalisation du design de l'issue **#3384**.
- Petite réunion pour savoir ce qui a été fait la veille.
- Suite de la modification du design de la liste déroulante.
- Changement de la liste déroulante du bouton à l'ui-select.
- Fonction de recherche : élément recherché en gras.

Après-midi

- Suite de la fonction de recherche. Objectif : juste afficher l'élément recherché.
- Achevement de la fonction de recherche.
- Gestion du dépassement avec l'élément css text-overflow.
- Impossibilité d'appuyer sur le bouton d'importation si une sheet n'est pas sélectionné.

Vendredi 1^{er} Mai

Féfé

Lundi 4 Mai

Matin

- Petite réunion pour savoir ce qui a été fait jeudi 30 avril.
- PR de l'issue #3384. (Aide de Pascal)
- Explication des "bonnes pratiques" à avoir sur le code et autre explication sur le code en lui-même par Pascal.

Après-midi

- Issue #3384 fermée.
- Issue #3050 fermée.
- Commencement du cahier des charges.

Mardi 5 Mai

Matin

- Commencement de l'issue #3461 : Si deux spreadSheet google ont le même nom, faire un classement par id.
- Apprentissage de la lecture des dossiers/fichiers java avec eclipse.
- Réunion sur ce qui a été fait la veille.
- Participation à la QA : trouver des bugs, améliorations(...) sur la partie google spreadsheet d'APIspark.
- Issue #1053 remplacée par l'issue #3466 : identique.

Après-midi

- Continuation de la QA.
- Fin du cahier des charges.
- Réalisation du diagramme de Gantt.
- Réalisation du diagramme de Pert.

Mercredi 6 Mai

Matin

- Mise à jour des branches git
- Petite réunion pour savoir ce qui a été fait hier.
- Suite de l'issue #3461.
- En attente de plus de précisions sur l'issue #3461, lecture de l'issue #3449.
- Plus de précisions sur l'issue #3461 pas Manuel.

Après-midi

- Suite de l'issue #3461.
- Démo de l'itération.
- Réunion sur l'itération suivante.

Jeudi 7 Mai

Matin

- Suite de #3461.
- Petite réunion sur ce qui doit être fait les 3 prochains jours.
- Suite de #3461.

Après-midi

- Explication sur le back par Jonhatan
- Remplacement des List<String> par des Map<String, String> pour pouvoir récupérer le nom du fichier google et sa clé unique.
- Fonctionnement du back
- Travail sur le front pour pouvoir afficher correctement les données et récupérer celles qui nous intéressent pour le back lorsque l'utilisateur choisit le fichier à importer.

Vendredi 8 Mai

Férié

Lundi 11 Mai

Matin

- Suite de la mission #3461
- Petite réunion pour faire le point sur ce qui a été fait jeudi et le nombre de problèmes à régler d'ici demain soir.
- Issue #1423 traitée par Cyprien.
- Suite de l'issue #3461 : map du côté front.

Après-midi

- Suite de l'issue #3461 : abandon du map du côté front.
- Ajout de la visibilité de la clé lors de la sélection par l'utilisateur.
- Issue #3461 : ok.
- Voir la persistance avec Cyprien.
- Début de l'issue #3449.
- Lecture de l'issue #3449 : problème lorsqu'une entité et/ou une propriété possède deux majuscules.

Mardi 12 Mai

Matin

- Suite de la mission #3461.
- Problème de persistance.
- Petite réunion sur ce qui a été fait la veille.
- Problème avec l'environnement de développement.
- Suite de l'issue #3461.

Après-midi

- Persistance avec Cyprien.
- Suite du réglage des bugs de l'issue #3461.
- Import des spreadsheet et de leurs sheets : ok.
- Problème au deployment.

Mercredi 13 Mai

Matin

- Suite de la mission #3461.
- Petite réunion sur les différents problèmes à traiter pendant la fin de semaine.
- Suite de l'issue #3461 : problème deploy -> récupération de l'ID du google spreadsheet donné par APIspark ? (Voir avec Cyprien)
- Commencement de l'issue #3449.
- Vu problème de deploy de l'issue #3461 avec Cyprien : côté front ok pour l'id. Problème au niveau de l'import du côté back.

Après-midi

- Problème de deploy résolu.
- Problème lorsque l'on clique sur "browser".
- Recherche de la google doc.
- Mise en place de la récupération de la clé dans la persistance.

Jeudi 14 Mai

Férié

Vendredi 15 Mai

Matin

- Suite de l'issue #3461.
- Faire en sorte de prendre l'id google de la spreadsheet et non son nom.
- Petite réunion pour savoir ce qui a été fait mercredi 13 mai et faire le point sur les problèmes qu'il reste encore pour la dernière journée de debug.
- Suite de la mission #3461.
- Commencement de la mission #3633.
- Création d'une branche git.
- Lecture de la mission : rendre possible l'utilisation du caractère "=" lorsque l'utilisateur veut ajouter une valeur.

Après-midi

- Suite de l'issue #3433.
- Recherche de solutions pour changer un caractère dans une String.

Lundi 18 Mai

Matin

- Suite de l'issue #3433.
- Réussi à changer le caractère '=' par '"=' ou '"=".
- Problème de fonctionnement avec le google spreadsheet : le lien ne se fait plus.
- Sprint launch : fin de l'itération 8 et début de l'itération 9.
- Itération de 10 jours.
- Suite de l'issue #3433.
- Correction si un utilisateur entre une valeur commençant par un '"'. Cela lève une erreur 422.
- Correction si un utilisateur ne rentre pas de valeur. Affichage "null" sur l'API et rien sur le google spreadsheet.

Après-midi

- Lecture de l'issue #3449 : faire en sorte que la casse de la valeur rentrée par l'utilisateur reste comme elle est.
- Création d'une nouvelle branche git : "3449-camel-case"
- Lecture de la mission : faire en sorte que les mots rentrés par l'utilisateur fonctionnent et qu'ils s'affichent correctement.

Mardi 19 Mai

Matin

- Suite de la mission **#3449**.
- Lecture de la mission **#3658**.
- Petite réunion pour savoir ce qui a été fait la veille et fixer les objectifs du jour.
- Suite de la lecture de la spécification de l'issue **#3658**.
- Ajouter le système de pagination à la partie google sheet wrapper. Binome avec Thierry B.
- Recherche dans l'API google s'il y a déjà un système de pagination.
- Analyse du code à l'endroit où il y a déjà un système de pagination.
- Création d'une branche pour la pagination après le passage à l'itération 3.9.

Après-midi

- Continuation de l'issue **#3658**.
- Création d'une version 2 de la classe `GoogleSpreadsheetPersistenceProvider`.
- Réalisation de la pull request du ticket **#3633**.
- Changement du code de la classe `GoogleSpreadsheetPersistenceProvider` pour les méthodes `executeQuery`.
- Relecture de la PR **#3633** par Thierry B.
- Changement du code par rapport au commentaire de la relecture.
- Voir pour ne pas afficher "null" dans APIspark si rien est marqué.
- Pas de changement en ce qui concerne l'affichage dans APIspark.
- Attendre validation PR pour l'issue **#3658** par Thierry B.
- Suite de l'issue **#3658**.

Mercredi 20 Mai

Matin

- Réalisation des testes de l'issue **#3633**.
- Petite réunion pour savoir ce qui a été fait la veille.
- Testes de l'issue **#3633** réussissent.
- Suite de la mission **#3658**.

Après-midi

- Explication du changement de version d'un fichier par Thierry. Il faut changer le chemin du fichier dans le fichier `JavaApplicationTemplateWorker.java` qui permet la génération de code.
- Suite de la mission **#3658**.
- Fin de la mission **#3658** côté back.
- Début de la mission **#3662** : mission **#3658** côté front.

Jeudi 21 Mai

Matin

- Suite de la mission #3658 : problème de pagination.
- Petite réunion pour savoir ce qui a été fait la veille.
- Suite de la mission #3658.

Après-midi

- Résolution du problème d'affichages du nombre de ligne par page.
- Problème pour l'affichage du nombre total d'items.

Vendredi 22 Mai

Matin

- Suite de la mission #3658 : problème d'affichage du nombre total d'items présents.
- Petite réunion pour savoir ce qui a été fait la veille.
- Suite de la mission #3658 : génération de code.

Après-midi

- Suite de la mission #3658 : génération de code.
- Résolution de la génération de code. (Aide de Thierry B)
- Résolution de la mission #3658.
- Début de la mise en place de la PLVERSION 2 si nouveau déploiement.

Lundi 1 juin

Matin

- Suite de la mission #3461.
- Petite réunion pour savoir ce qui a été fait la veille.
- Création du script de migration.
- Création de la PR.
- Relecture par Thierry.
- Mise à jour de la partie front.
- Création de la PR.
- Relecture par Pascal.
- PR front ok.
- Réglage des problèmes de back avec Thierry.
- Début de la mission #3831 : Ajout du système de filtre.

Après-midi

- Ecriture du rapport de stage.
- Suite de la mission sur l'ajout du système de filtre : #3831.

Mardi 2 juin

Matin

- Suite de la mission sur l'ajout du système de filtre : #3831.
- Petite réunion pour savoir ce qui a été fait la veille.
- Suite de la mission #3831.
- Solution trouvée avec la class CustomParameter.
- Problème pour la récupération des conditions.

Après-midi

- Suite de l'issue #3831.
- Fin de la mission #3831.
- Début des tests.
- Réalisation de la PR lorsque la branche 3.10 sera créée.
- Recherche de solution pour pouvoir les trier les valeurs à partir de \$sort.
- Première solution trouvée. Récupération de la bonne adresse mais non passé dans postman
- Deuxième solution : utiliser la classe CustomParameter.

Mercredi 3 juin

Matin

- Suite de la mission sur les tris
- Sprint launch de l'itération 3.10.
- Réunion avec Manuel et Mme Quiniou.
- Problème de connexion sur Google.

Après-midi

- Résolution de problème avec Thierry : recréer un cookie et l'ajouter aussi dans cqlsh
- Fin de la mission sur les filtres et sur les tris.
- Lancement des testes.
- Création de la PR.
- Création d'une version de migration mineur.

Jeudi 4 juin

Matin

- Suite de la mise en place de la PL version.
- Petite réunion pour savoir ce qui a été fait la veille.
- Création d'une PR.
- Réalisation des tests.
- Tests passés.
- Demande de relecture de PR à Jonathan.
- Ecriture du rapport de stage : refaire le plan.
- Lecture de la mission **#3466** : lorsque l'utilisateur ajout, renomme ou supprime une feuille, le même comportement est attendu pour l'entité concernée.
- Redirection sur le ticket **#3910**.
- Lecture de la mission **#3910** : amélioration de la recherche du googleSpreadSheet par id

Après-midi

- Correction de l'issue suivant les commentaires de relecture de Jonathan.
- Suppression de la migration de la PLVersion : on reste à la 2 et non à la 2.1.
- Reprise de l'issue **#3910**.

Vendredi 5 juin

Matin

- RDV
- Correction des tickets **#3831** et **#3885**.

Après-midi

- Suite de la correction des tickets **#3831** et **#3885** suite à la PR.
- Problème avec plusieurs filtres : non géré.
- Création d'un ticket github pour faire remonter le problème.
- Réalisation d'une PR.
- Début de la réalisation du ticket **#3910**.
- Solution trouvée.
- Problème pour récupérer les sheets du google spreadsheet voulu.

Lundi 8 juin

Matin

- Suite de la mission #3910.
- Réglage du problème de récupération des entités.
- Réalisation de la PR.
- Réalisation des tests.
- Petite réunion pour faire le point.
- Validation de la PR sur les filtres et les tris par Jonathan.
- Correction du code pour l'amélioration de la recherche de google spreadsheet par clé suite à la relecture de Thierry.
- Validation de la PR par Thierry.
- Attente pour fusionner les deux PR sur la branche origin/3.10.
- Réalisation du rapport de stage.
- Début de la mission #3457.
- Lecture de la mission : permettre à l'utilisateur de changer le type de la valeur.

Après-midi

- Merge des deux PR.
- Reflexion sur le changement des types des propriétés.
- Discussion avec Thierry.
- Réalisation du code côté front.

Mardi 9 juin

Matin

- Suite de la mission #3457 côté front.
- Possibilité de cliquer uniquement sur la liste déroulante.
- Petite réunion pour savoir ce qui a été fait la veille.
- Suite de la mission #3457 côté front.

Après-midi

- Suite de la mission #3457 côté front.
- Réalisation de la PR.
- Relecture par Pascal.

Mercredi 10 juin

Matin

- Début de la mission #3457 côté back.
- Petite réunion pour savoir ce qui à été fait la veille.
- Discussion avec Pascal sur la PR du ticket côté front.
- Réalisation des corrections.

Après-midi

- Fin des corrections.
- Nouvelle PR.
- Relecture par Pascal.
- Continuation de la mission côté back.
- Problème de niveau de la conversion des valeurs. Feature mise de côté.
- Début du ticket #4002.
- Résolution du ticket.
- PR de Cyprien.

Jeudi 11 juin

Matin

- Début du ticket #3996.
- Petite réunion pour savoir ce qui a été fait.
- Suite du ticket #3996.
- Problème trouvé : problème de format du côté de google spreadsheet.
- Retour sur le ticket #3457 du côté back.

Après-midi

- Suite du ticket #3457.
- Reflexion sur les cas ou le changement de type de fonctionne pas. Listage des cas. Message à l'utilisateur ?
- Reprise de la mission #4002.

Vendredi 12 juin

Matin

- Suite de la mission #4002.
- Petite réunion pour savoir ce qui à été fait la veille.
- Suite de la mission #4002.

- Fin de la mission **#4002**.
- Réalisation des tests.
- Réalisation de la PR.
- Réalisation des différents cas pour l'issue **#3996**.
-

Après-midi

- Problème de connection avec le nouveau système de connection google de l'API. (correction temporaire avec Constant et Pascal. Problème de port).
- Suite de la mission **#3996**.

Lundi 15 juin

Matin

- Petite réunion pour voir ce qui a été fait vendredi.
- Suite du ticket **#3996**.

Après-midi

- Suite du ticket **#3996**.
- Correction du ticket **#4002** pour google spreadsheet.
- Problème de connection à solr.
- installation de l'environnement de développement "classique".
- Correction avec Manuel : problème de target platform.
- Correction du ticket **#4046** pour le problème de filtre du full stack.
- Correction du ticket **#4046**.

Mardi 16 juin

Matin

- Suite du ticket **#3996**.
- Petite réunion pour savoir ce qui a été fait la veille.
- Recherche de solution pour le ticket **#3996**.

Après-midi

- Sprint retro
- Sprint demo

Mercredi 17 juin

Matin

- Petite réunion pour savoir ce qui a été fait la veille
- Suite de la mission #3996.
- Réunion avec Manuel.
- Ticket #3457 côté front.

Après-midi

- Suite du ticket #3457 côté front.
- Réalisation d'une PR suite au changement définit.

Jeudi 18 juin

Matin

- Petite réunion pour savoir ce qui a été fait la veille.
- Réunion avec Jonathan et Manuel pour faire le point sur les spécifications et l'avancé du ticket #3457 côté front et back.
- Application des spécifications côté front.

Après-midi

- Suite de l'application des spécifications côté front.
- Réalisation d'une PR pour ce ticket.
- Application des spécifications côté back.

Vendredi 19 juin

Matin

- Petite réunion pour savoir ce qui a été fait la veille.
- Réalisation de tests sur le ticket #3457 côté front.
- Réalisation de la PR pour ce ticket.
- Suite de l'écriture du rapport.

Après-midi

- Fin de l'écriture du rapport.
- Validation de la PR par Pascal.

Cahier des charges

1 Présentation générale du problème

1.1 Projet

1.1.1 Finalités

La finalité des missions du projet est de corriger des problèmes et de créer des fonctionnalités sur la partie feuilles de calcul de type Google Spreadsheet de l'interface de programmation APIspark, en fonction des tickets créés via le gestionnaire de version GitHub.

1.1.2 Espérance de retour sur investissement

Immédiat.

1.2 Contexte

1.2.1 Situation du projet par rapport aux autres projets de l'entreprise

Cette mission s'intègre dans le projet global d'APIspark. APIspark est une "application programming interface" ou API qui permet aux utilisateurs de créer une base de données, d'importer une base de données SQL déjà existante, utiliser une feuille créer sur google ou encore utiliser Firebase ou Parse qui qui sont des "cloud services provider". Les utilisateurs peuvent, dans tous les cas, exporter leurs bases de données en API web.

De plus, cette API permet aussi de créer directement une API web. Il y a trois types d'API web possible : Full stack qui permet la création direct d'une API web. Descriptor qui permet la création d'une documentation en ligne d'une API déjà existante. Et enfin le connector qui permet la mise à jour d'une API déjà existante.

Enfin, il est possible de créer des file store. Il existe aussi trois types possibles : full stack qui permet de mettre ses fichiers directement sur l'API. AWS S3 wrapper qui permet d'accéder aux fichiers à partir d'un compte Amazon S3. Et enfin un github wrapper qui permet d'accéder aux fichiers à partir d'un répertoire github.

1.2.2 Études déjà effectuées

Il existe d'autres fonctionnalités présentes dans APIspark tel que Parse wrapper et Firebase wrapper.

1.2.3 Études menées sur des sujets voisins

Apprentissage du code et des fichiers sur des sujets voisins comme firebase.

Apprentissage d'APIspark sur les différentes fonctionnalités avec le tutoriel lié.

1.2.4 Suites prévues

Les missions évolue en fonction des problèmes trouvés et des fonctionnalités voulues au cours des itérations.

1.2.5 Nature des prestations demandées

Il s'agit de missions à l'intérieur d'une fonctionnalité d'un des produit Restlet.

1.2.6 Parties concernées par le déroulement du projet et ses résultats (demandeurs, utilisateurs)

Demandeur : Product manager.

Utilisateurs : Les utilisateurs d'APIspark et l'équipe de R&D.

1.2.7 Caractère confidentiel s'il y a lieu

Aucun.

1.3 Énoncé du besoin (finalités du produit pour le futur utilisateur tel que prévu par le demandeur)

Amélioration de la partie Google spreadsheet d'APIspark.

1.4 Environnement du produit recherché

1.4.1 Listes exhaustives des éléments (personnes, équipements, matières...) et contraintes (environnement)

Je dispose d'un ordinateur portable possédant Windows 7 et linux mint.

Je travaille avec l'équipe recherche et développement.

1.4.2 Caractéristiques pour chaque élément de l'environnement

Pour réaliser le travail demandé, eclipse, webstorm, git et l'environnement de développement sont installés sur mon poste de travail.

2 Expression fonctionnelle du besoin

2.1 Fonctions de service et de contrainte

2.1.1 Fonctions de service principales (qui sont la raison d'être du produit)

La fonction de service principale est de permettre à l'utilisateur d'importer une feuille google dans APIspark puis d'effectuer des opérations tel que l'ajout de lignes dans la base donnée et l'exporter en API web.

2.1.2 Fonctions de service complémentaires (qui améliorent, facilitent ou complètent le service rendu)

Les fonctions de service complémentaires sont : faciliter l'utilisation de la fonctionnalité à l'utilisateur et améliorer son utilisation.

2.1.3 Contraintes (limitations à la liberté du concepteur-réalisateur)

Aucune.

2.2 Critères d'appréciation (en soulignant ceux qui sont déterminants pour l'évaluation des réponses)

La solution trouvée est adaptée au ticket, avec la meilleure qualité possible.

2.3 Niveaux des critères d'appréciation et ce qui les caractérise

2.3.1 Niveaux dont l'obtention est imposée

La solution trouvée répond au ticket qui lui est lié.

2.3.2 Niveaux souhaités mais révisables

La solution trouvée a la meilleure qualité possible.

3 Cadre de réponse

3.1 Pour chaque fonction

3.1.1 Solution proposée

Voir le diagramme de Gantt.

3.1.2 Niveau atteint pour chaque critère d'appréciation de cette fonction et modalités de contrôle

Le critère d'appréciation de chaque fonction est la fermeture du ticket.

La modalités de contrôle est de passer les testes et qu'il n'y ait pas de nouveau ticket d'ouvert.

3.2 Pour l'ensemble du produit

3.2.1 Options et variantes proposées non retenues au cahier des charges

Il n'y a pas d'options ni de variantes proposées non retenues au cahier des charges puisque la mission porte sur des tickets qui ont été d'abord validés, qu'ils s'agissent de problème ou de la création d'une fonctionnalité.

3.2.2 Mesures prises pour respecter les contraintes et leurs conséquences économiques

Il n'y a pas de contraintes.

3.2.3 Outils d'installation, de maintenance ... à prévoir

L'utilisateur doit au moins être un possession du navigateur Internet explorer 9.

3.2.4 Perspectives d'évolution technologique

Les perspectives d'évolution technologique sont en fonction des besoins futurs des utilisateurs.

Filtres

contactid	firstName	lastName	age
1	Darth	Veder	61
2	Bob	Durant	23
3	Janna	Dunant	53
4	Martin	Dupont	60
5	Lilian	Dupond	47
6	Rachel	Dupont	32
7	Martin	Veder	46
8	Julien	Durand	54
9	Corentin	Jean	32
10	Jean	Corentin	47
11	John	Smith	47
12	Camille	Bidule	47
13	Martin	Bydule	47
14	Christopher	Truc	72
15	Alex	Truk	54

FIGURE 1 – Données dans le tableur google

contactid	firstname	lastname	age
1	Darth	Veder	61
2	Bob	Durant	23
3	Janna	Dunant	53
4	Martin	Dupont	60
5	Lilian	Dupond	47
6	Rachel	Dupont	32
7	Martin	Veder	46
8	Julien	Durand	54
9	Corentin	Jean	32
10	Jean	Corentin	47
11	John	Smith	47
12	Camille	Bidule	47
13	Martin	Bydule	47
14	Christopher	Truc	72
15	Alex	Truk	54

FIGURE 2 – Données récupérées dans le tableur google

<http://googleghangeapi.apispark-dev.net:8182/v1/g1s/>

```
[
  {
    "contactid": "1",
    "firstname": "Darth",
    "age": "61",
    "lastname": "Veder"
  },
  {
    "contactid": "2",
    "firstname": "Bob",
    "age": "23",
    "lastname": "Durant"
  },
  {
    "contactid": "3",
    "firstname": "Janna",
    "age": "53",
    "lastname": "Dunant"
  },
  {
    "contactid": "4",
    "firstname": "Martin",
    "age": "60",
    "lastname": "Dupont"
  },
  {
    "contactid": "5",
    "firstname": "Lilian",
    "age": "47",
    "lastname": "Dupond"
  },
  {
    "contactid": "6",
    "firstname": "Rachel",
    "age": "32",
    "lastname": "Dupont"
  },
  {
    "contactid": "7",
    "firstname": "Martin",
    "age": "46",
    "lastname": "Veder"
  },
  {
    "contactid": "8",
    "firstname": "Julien",
    "age": "54",
    "lastname": "Durand"
  },
  {
    "contactid": "9",
    "firstname": "Corentin",
    "age": "32",
    "lastname": "Jean"
  },
]
```

FIGURE 3 – Filtre avec aucun paramètre dans la requête

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?firstname=Martin`

```
[
  {
    "contactid": "4",
    "firstname": "Martin",
    "age": "60",
    "lastname": "Dupont"
  },
  {
    "contactid": "7",
    "firstname": "Martin",
    "age": "46",
    "lastname": "Veder"
  },
  {
    "contactid": "13",
    "firstname": "Martin",
    "age": "47",
    "lastname": "Bydule"
  }
]
```

FIGURE 4 – Filtre avec un paramètre dans la requête

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?firstname=Martin&lastname=Dupont`

```
[
  {
    "contactid": "4",
    "firstname": "Martin",
    "age": "60",
    "lastname": "Dupont"
  }
]
```

FIGURE 5 – Filtre avec deux paramètres dans la requête

Tris

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=firstname`

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=firstname ASC`

```
[
  {
    "contactid": "15",
    "firstname": "Alex",
    "age": "54",
    "lastname": "Truk"
  },
  {
    "contactid": "2",
    "firstname": "Bob",
    "age": "23",
    "lastname": "Durant"
  },
  {
    "contactid": "12",
    "firstname": "Camille",
    "age": "47",
    "lastname": "Bidule"
  },
  {
    "contactid": "14",
    "firstname": "Christopher",
    "age": "72",
    "lastname": "Truc"
  },
  {
    "contactid": "9",
    "firstname": "Corentin",
    "age": "32",
    "lastname": "Jean"
  },
  {
    "contactid": "1",
    "firstname": "Darth",
    "age": "61",
    "lastname": "Veder"
  },
  {
    "contactid": "3",
    "firstname": "Janna",
    "age": "53",
    "lastname": "Dunant"
  },
]
```

FIGURE 6 – Tri croissant

[http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?\\$sort=firstname DESC](http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=firstname DESC)

```
[
  {
    "contactid": "6",
    "firstname": "Rachel",
    "age": "32",
    "lastname": "Dupont"
  },
  {
    "contactid": "4",
    "firstname": "Martin",
    "age": "60",
    "lastname": "Dupont"
  },
  {
    "contactid": "7",
    "firstname": "Martin",
    "age": "46",
    "lastname": "Veder"
  },
  {
    "contactid": "13",
    "firstname": "Martin",
    "age": "47",
    "lastname": "Bydule"
  },
  {
    "contactid": "5",
    "firstname": "Lilian",
    "age": "47",
    "lastname": "Dupond"
  },
  {
    "contactid": "8",
    "firstname": "Julien",
    "age": "54",
    "lastname": "Durand"
  },
  {
    "contactid": "11",
    "firstname": "John",
    "age": "47",
    "lastname": "Smith"
  },
  {
    "contactid": "10",
    "firstname": "Jean",
    "age": "47",
    "lastname": "Corentin"
  }
].
```

FIGURE 7 – Tri décroissant

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=firstname DESC tropParametres`

```
{
  "code": 400,
  "description": "Too many arguments have been passed to sort",
  "reasonPhrase": "Bad Request"
}
```

FIGURE 8 – Tri avec trop d'arguments

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=firstname mauvaisParametre`

```
{
  "code": 400,
  "description": "Invalid sort firstname tropParametres, tropParametres is not supported",
  "reasonPhrase": "Bad Request"
}
```

FIGURE 9 – Tri avec un mauvais paramètre de tri

`http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?$sort=mauvaisNom DESC`

```
{
  "code": 400,
  "description": "Invalid $sort value: 'mauvaisNom' is not a property name",
  "reasonPhrase": "Bad Request"
}
```

FIGURE 10 – Tri avec un mauvais nom pour le paramètre de la propriété

Filtre et Tri

[http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?lastname=Veder&\\$sort=firstname DESC](http://googlehangeapi.apispark-dev.net:8182/v1/g1s/?lastname=Veder&$sort=firstname DESC)

```
[
  {
    "contactid": "7",
    "firstname": "Martin",
    "age": "46",
    "lastname": "Veder"
  },
  {
    "contactid": "1",
    "firstname": "Darth",
    "age": "61",
    "lastname": "Veder"
  }
]
```

FIGURE 11 – Requête combinant filtre et tri