florian roura – epsi Montpellier I4

2017-2018

Rapport de stage

Création d’une interface web avec angularJS, SYMFONY

Adaptation d’une API envoyant des données en json

supérieur : M. François marchanD

TUTEUR : M. Sebastien rousselle



437 Avenue des Apothicaires 34090 Montpellier

[04 67 04 20 01](https://www.google.fr/search?q=epsi+montpellier+adresse&rlz=1C1CHBF_frFR798FR798&oq=epsi+montpellier+adresse&aqs=chrome..69i57.2830j0j9&sourceid=chrome&ie=UTF-8)

Route de Gallargues

30470 Aimargues

[04 66 73 65 65](https://www.google.fr/search?safe=off&rlz=1C1CHBF_frFR798FR798&ei=Npw0W7XXAcbxUO_bnIAK&q=eminence+adresse+aimargues&oq=eminence+adresse+aimargues&gs_l=psy-ab.3..0i22i30k1l2.7163.8761.0.8872.10.10.0.0.0.0.273.891.2j4j1.7.0....0...1c.1.64.psy-ab..3.7.891....0.kz5HthYWG6g)

Table des matières

[Remerciements 2](#_Toc518350542)

[Introduction 1](#_Toc518350543)

[Présentation de l’entreprise 2](#_Toc518350544)

[Aujourd’hui 2](#_Toc518350545)

[Historique 2](#_Toc518350546)

[Aimargues 4](#_Toc518350547)

[Environnement de travail 5](#_Toc518350548)

[Mission 1 – Portail Négoce 10](#_Toc518350549)

[Contexte 10](#_Toc518350550)

[1. L’ordonnancement 10](#_Toc518350551)

[2. Le service Négoce 11](#_Toc518350552)

[De l’existant à la demande 11](#_Toc518350553)

[1. Workflow 11](#_Toc518350554)

[2. Portail négoce actuel 12](#_Toc518350555)

[3. Demande du service Négoce 14](#_Toc518350556)

[demandes & réalisations 15](#_Toc518350557)

[1. Cahier des charges 15](#_Toc518350558)

[2. Technologies choisies 17](#_Toc518350559)

[3. Processus d’implémentation 24](#_Toc518350560)

[4. Réalisations 28](#_Toc518350561)

# Remerciements

Il n’est jamais aisé pour un étudiant de trouver un stage qui lui permette non seulement de s’épanouir mais également d’appliquer les connaissances qu’il a pu acquérir au cours de sa formation. C’est pourquoi je tiens à remercier l’entreprise EMINENCE de m’avoir accueilli durant ces 9 mois.

Je tiens à remercier tout particulièrement M. François Marchand, qui m’a accordé sa confiance, et inclus au sein de l’équipe. Grâce à lui j’ai pu me voir attribuer des missions valorisantes, et ce encore aujourd’hui.

Je tiens également à remercier M. Sébastien Rousselle, mon tuteur, ainsi que M. Vincent Descreux, ancien étudiant de l’EPSI dont je suis les traces, pour leur aide constante, leur support et leurs conseils. J’ai pu, grâce à eux, assister à des réunions professionnelles mettant en relation plusieurs services, et progresser davantage sur les missions qui m’ont été confiées.

J’adresse également mes remerciements à l’équipe du service informatique, que je côtoie chaque jour. Lorsque je suis arrivé le premier jour, tout mon environnement de travail était déjà préparé par eux. Ils contribuent à l’atmosphère chaleureuse toujours présente au bureau, constituant une présence bienveillante, ceci contribuant largement à un environnement propice au développement, tant au niveau du projet que personnel.

Enfin, je tiens à remercier l’école EPSI Montpellier pour son soutien sans faille et son accompagnement continu des étudiants.

# Introduction

Ce document a pour but de décrire mon séjour au sein de l’entreprise EMINENCE, basée à Aimargues, ainsi que les missions qui m’ont été attribuées lors du stage que j’effectue en tant que membre de son service informatique, encore aujourd’hui.

Après une présentation de l’entreprise, son secteur d’activité et son cœur de métier, ses origines et les locaux de l’entreprise, sera présenté l’environnement de travail établi par le service informatique, à l’intérieur duquel j’ai pris plaisir à développer.

Ensuite, j’aborderai les missions qui m’ont été confiées lors de mon stage, ainsi que leur contexte. Je décrirai le fonctionnement de l’entreprise et les raisons qui ont mené à la création de ces missions.

Enfin, après avoir justifié les choix de technologies que j’ai pris suite à réflexion, sans oublier ma méthode d’implémentation, j’exposerai les réalisations effectuées au cours du temps et au fil des demandes, avant de conclure.

# Présentation de l’entreprise

## Aujourd’hui

**EMINENCE** est une entreprise de fabrication et création de **sous-vêtements**, créée en 1944. Leader sur le marché français, elle propose d’année en année des produits modernes de grande qualité, soucieuse de la satisfaction des clients.

Elle possède plusieurs sites : celui d’Aimargues, et celui de **SAUVE**, dans les Cévennes. Ce dernier réalise toutes les opérations sauf la teinture, qui sera sous-traitée au sud de la France ou au nord de l’Espagne. La marque principale est **Eminence**, mais on pourra également retrouver la marque **Athena** pour les sous-vêtements féminins, et finalement **Liabel**, en italie.

Néanmoins, EMINENCE ne produit pas seulement des slips pour homme. En effet, elle a décliné son secteur d’activité pour pouvoir maintenant produire des boxers, des caleçons, des chaussettes, des T-shirts, des débardeurs, des pyjamas et des sous-vêtements chauds. Cette diversité de production et création, couplée à la minutie continuelle et appliquée de l’entreprise, lui permet d’assurer sa place de **marque** **numéro 1 en France** sur ce marché.

La plupart des produits d’EMINENCE sont confectionnés à Aimargues. On pourra retrouver ses produits en grande surface ou sur son site Internet. Elle propose également des lots d’articles pour chaque produit conçu à prix réduit, ce qui permet aux clients d’obtenir leurs produits à un coût plus favorable.

## Historique

En 1937, **Georges Jonathan**, représentant de commerce à Paris, et **Gilbert** **Sivel**, technicien du textile dans les Cévennes, s’associent pour créer **l’Atelier Artisanal de Bonneterie de Nîmes**. Cependant, la Seconde Guerre mondiale, ayant lieu deux ans après, empêche la société de fonctionner.

Ne s’avouant pas vaincus, les deux créateurs de l’entreprise décident de continuer la production dès 1944, et par la même occasion créant le nom de marque **Eminence**, en honneur au surnom du **cardinal de Richelieu**.

En 1946, les deux créateurs apportent de la nouveauté au milieu du sous-vêtement. En effet, ils parviennent à acquérir **des métiers à tisser suisses** « point tamisé », à ce jour inconnus en France. Grâce à ces outils, ils parviennent à confectionner un tissu à mailles plus aérées et donc plus confortable qu’ils nommeront le « **petit point noué** ».

En 1947, l’un des deux fondateurs de l’entreprise effectue un voyage en Argentine. Lors de ce voyage, il découvre que les « gauchos », peuple de gardiens de troupeaux, portent un slip fendu. S’inspirant de leurs vêtements, la société met au point un nouveau **slip à poche**, le **modèle 100** en « petit point noué ». Par la suite, vers la fin des années 1950, le **modèle** **108 en côtes fines** fera son apparition. Grâce à ce dernier, la société EMINENCE devient une société spécialiste en matière de sous-vêtement masculin.

La même année, l’entreprise tente une nouvelle technique commerciale couronnée de succès, la **vente** de ses produits sous **plastique** **transparent**. Elle est la première à utiliser ce procédé.

Aujourd’hui elle est l’une des dernières entreprises fabriquant des sous-vêtements en grande série. Ces quelques dernières années ont été difficiles pour l’entreprise, mais depuis 2016, sa situation s’est améliorée. S’investissant dans les **marché publics**, tels que des vêtements ignifugés et antimoustiques pour la police ou encore des chemises tactiques pour l’armée, et ayant recours à la **fabrication en sous-traitance** pour des marques telles que Celio, Monoprix, Carrefour ou le Slip Français, elle connaît une remontée phénoménale, prête d’atteindre son record précédent de 130 millions d’euros sur un marché en recul.

## Aimargues

Le site principal d’EMINENCE se situe aujourd’hui à Aimargues, en Camargue. Il est constitué d’une usine, du siège de l’entreprise ainsi que d’un magasin d’usine, dont les produits viennent directement de ladite usine. Les produits y sont proposés à des prix 30% inférieurs comparés au circuit classique. Ces produits peuvent être des invendus, des fins de collections ou des écoulements de surstocks des différentes marques de l’entreprise.

L’entreprise y produit ici la plupart de ses produits, qui sont redistribués en grande surface ou à des particuliers. Le siège social de l’entreprise est donc directement lié à l’usine, et l’on pourra constater de nombreux échanges entre celui-ci et les employés de cette dernière.

**A noter que plusieurs des départements du siège social sont en relation directe avec l’usine, bien qu’ils ne travaillent pas au même endroit. Ainsi, un service nommé l’« ordonnancement » est chargé de vérifier que l’usine dispose, à tout moment des ressources nécessaires à la fabrication des produits.**

# Environnement de travail

Base de données

EMINENCE fonctionne avec la bases de données IBM DB2. De ce fait, l’entreprise utilise des serveurs AS400 pour stocker l’intégralité des fichiers. La base de données est le lieu d’échanges constants, et des rotations de scripts sont organisées afin de garantir son bon fonctionnement. Certains scripts s’exécutent à intervalles définis, d’autres sont exécutés à partir d’une certaine heure. Pour ces derniers, il faut en général qu’une personne veille à leur bon déroulement, en plus d’effectuer d’autres opérations. Ce rôle incombe à une personne spécifique au sein du service informatique, mais les autres membres du service informatique peuvent l’assumer en cas d’absence ou d’indisponibilité.

Frameworks

Plusieurs portails, dont certains existent pour le développement, se situent sur des AS400 différents. Une partie du portail, incluant la production, est réalisée en PHP procédural, tandis que l’autre tourne sur le Framework Symfony, d’ailleurs récemment mis à jour.

Machine

La machine sur laquelle je développe est un ordinateur fixe, doté de Windows 10. L’ensemble est connecté sur le réseau en Ethernet, qui donne un accès à internet (filtré à un certain point, par exemple il est impossible de se connecter sur Outlook ou Gmail) ainsi qu’aux serveurs de l’entreprise. Durant mes missions, j’ai eu l’occasion de travailler sur le serveur de test, l’une demandant la création d’un nouveau script php, l’autre la création de tout un module sur l’application Symfony.

IDE

Durant l’intégralité de stage, puis par la suite sur l’ensemble de mes projets, autant personnels que scolaires, j’ai utilisé le logiciel Visual Studio Code. Il s’agit ici d’une version simple et épurée du logiciel Visual Studio, sans pour autant manquer de fonctionnalités ou de possibilités.

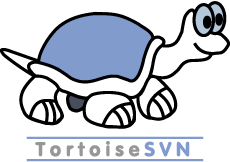
En effet, la modification de cet IDE par le biais de réglages et de plug-ins additionnels lui permet de s’adapter au développeur, en lui proposant des fonctionnalités supplémentaires telles que de l’autocorrection sur un langage spécifique, une possibilité de déboguer des éléments, de se connecter sur une base de données distante, et bien d’autres.



Je m’étais au début tourné vers l’IDE Atom, au début de mon arrivée à l’entreprise. Cependant, j’ai pu constater que bien qu’il dispose de possibilités d’adaptation, comme Visual Studio Code, il est en revanche incapable de fonctionner avec trop de plug-ins installés, surtout sur une machine avec peu de ressources comme celle sur laquelle j’ai pu travailler. Visual Studio code est bien plus stable, et offre une bibliothèque de plug-ins en ligne, ainsi qu’une documentation détaillée, ce qu’Atom ne propose pas. J’ai donc, dès la découverte de Visual Studio Code, décidé d’immédiatement sauter le pas et d’utiliser cet IDE, décision que je ne regrette pas et que j’aurais au contraire préféré prendre plus tôt.

J’ai pu avoir ce choix car le seul autre IDE présent sur ma machine lors de mon arrivée est une distribution Adelia Studio, davantage liée à la programmation des AS400, et donc bien en dehors de mes compétences ainsi que des missions qui allaient m’être proposées.

Gestionnaire de sources

Lorsque je suis arrivé en entreprise, le service informatique se servait du logiciel Tortoise SVN. Après les modifications, il suffisait de faire un clic droit dans le dossier principal et envoyer une notification de changement sur ce logiciel.

Mais récemment, la décision a été prise de passer sous le très célèbre système de versioning décentralisé Git. J’ai accueilli ce changement avec joie, ayant eu l’opportunité de découvrir cette manière de fonctionner au cours de mes années de formation à l’EPSI. Il m’a donc été facile d’intégrer la nouvelle manière de fonctionner, avec laquelle j’étais plus à l’aise.

En parallèle, un serveur GitBucket a été mis en place, constituant une origine commune pour les projets. Le répertoire de base est le projet Symfony, sur lequel chacun des développeurs effectue ses modifications. Chacun possède sa propre branche, synchronisée en l ocal et sur le serveur, sur lesquelles les autres développeurs peuvent « pull » et ainsi récupérer toutes les modifications effectuées. Une branche plus générale nommée « develop » permet de regrouper de temps à autre les modifications effectuées par les différents développeurs du service informatique. Chacun étant sur une partie différente du portail, il n’y eu jusqu’à ce jour aucun problème lors de la fusion. Cependant, les fichiers inclus dans le portail tels que les scripts javascript, les feuilles de style css ou encore les assets utilisés par Symfony restent communs à l’ensemble du portail. Il nous incombe donc de rester vigilants quant à leur modification (qui n’a généralement pas lieu.)

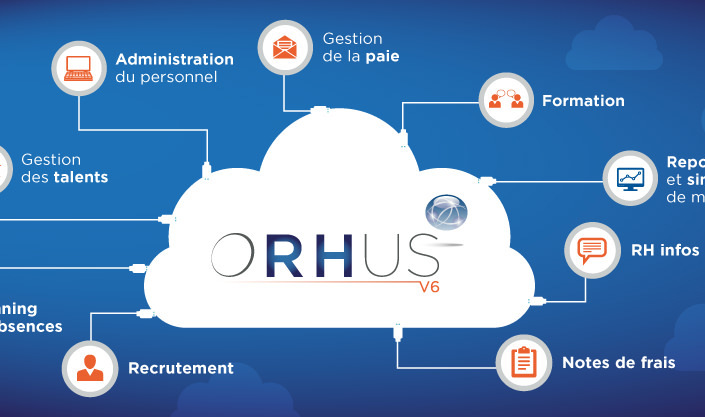
J’ai trouvé cette approche beaucoup plus efficace que celle que j’ai eu l’occasion de suivre l’année dernière en entreprise. En effet, l’année dernière nous ne disposions que d’un serveur central sur lequel existait une branche « master » et une branche « dev ». Nous travaillions sur le même projet en même temps, et devions tous pousser nos modifications sur la branche « dev ».

Ceci impliquait que la moindre erreur de fusion entre les fichiers posait des problèmes qui mettaient parfois la journée à résoudre. De plus, la politique de BYOD (Bring Your Own Device) mise en place l’année dernière, qui consiste à faire travailler chacun des membres sur ses appareils personnels, posait également des problèmes de compatibilité, deux d’entre nous étant sur Windows, un autre sur Mac OS. Des erreurs sur les fichiers se multipliaient et nuisaient grandement à notre productivité, les problèmes ne venant pas de notre capacité à mener le projet mais aux différences entre nos environnements de travail. Par ailleurs, la manière que nous avions de travailler sur nos machines personnelles soulevait également des questions quant à la sécurité du travail que nous étions en train d’accomplir, l’application étant une toute nouvelle idée dont il ne fallait surtout pas dévoiler le succès, et dont le cryptage et autres options de sécurité ne devaient pas dépasser la salle dans laquelle nous développions.

Je suis ravi de constater que la situation est fondamentalement différente au sein d’EMINENCE. Chacun travaille sur des postes prédéfinis, dont l’accès est règlementé. Chacun est identifiable à tout moment sur le réseau et des permissions ont été mises en place pour des raisons de sécurité, autant sur les sessions Windows que sur les différents portails et outils de l’entreprise.

Outils d’entreprise

Afin d’accompagner non seulement les membres du service informatique, mais également l’ensemble de l’entreprise, plusieurs outils ont été mis en place par les Ressources Humaines afin de simplifier quelques procédés.

Le premier outil mis en place s’appelle Orhus. Il s’agit d’une sorte de ressource disponible à tous les membres, accessible sur le réseau local en se connectant avec un identifiant unique, différent de ceux utilisés pour se connecter sur les machines.

Cet outil met à disposition plusieurs ressources utiles aux membres de l’entreprise, telles que (liste non exhaustive) :

* Un annuaire contenant les noms, informations et services du personnel de l’entreprise
* Un blog relatant des dernières nouvelles de l’entreprise, pouvant relater du recrutement d’un nouveau membre, d’une notice générale ou de messages au contenu divers
* Un espace membre individuel mettant à disposition des ressources telles que des règles générales de conduite, un catalogue interne lié au magasin d’usine, des références, et autres
* …

L’autre logiciel est nommé Horoquartz. Il s’agit ici d’un logiciel de gestion de temps et de pointage pour l’entreprise. En effet, chacun (même moi !) possède un badge qu’il doit bipper lorsqu’il entre ou sort de l’entreprise. Ces pointages sont comptabilisés par ce logiciel, qui calcule alors les temps de présence et fait le lien avec les compteurs généraux de l’employé en question (Temps de présence obligatoire par jour, temps dépassé / heures supplémentaires effectuées, et cætera.) Par ailleurs, ce logiciel permet également de réserver ses congés et d’en connaître le montant restant. Il permet de faire ces opérations en toute simplicité, et je me retrouve à l’utiliser souvent pour veiller à ma présence en entreprise.

Le jour suivant mon arrivée, j’ai suivi une formation individuelle d’une heure sur ces deux outils, qui m’a été dispensée par une employée travaillant au secteur des Ressources Humaines. J’ai pu y voir ici la preuve de mon inclusion au sein de l’entreprise et le sentiment de travailler au sein d’une structure à caractère hautement professionnel, n’ayant jusque-là travaillé que dans des start-ups. J’ai d’ailleurs apprécié de voir de tels outils mis à disposition, facilitant les opérations de chacun.

E-mail et planning

J’ai évoqué ci-dessus le fait qu’il était, suite aux mesures de sécurité établies au sein de l’entreprise, impossible d’accéder aux messageries disponibles sur le web, telles qu’Outlook ou Gmail. On pourra donc se demander par quel biais est-il possible de communiquer au sein de l’entreprise.

C’est là qu’intervient le logiciel IBM Notes, installé par défaut sur chacune des machines de l’entreprise. Il s’agit là d’une messagerie locale, synchronisée en internet, permettant des échanges simples et sécurisés entre les employés de l’entreprise. L’ajout de l’annuaire général permet de ne pas éprouver de difficultés à joindre qui que ce soit, les noms et adresses étant automatiquement complétées suite à quelques caractères saisis.

Par ailleurs cette application permet d’envoyer par mail des notices de réunions. Ce sont des e-mails spécifiques au logiciel, qui contiennent des actions pour confirmer ou annuler la réunion, ainsi que d’autres actions relatives à celle-ci. J’ai eu l’occasion de recevoir de tels mails visant à me convier à une réunion, et je dois dire que je trouve cette manière de procéder efficace. On se retrouve bien loin des réunions organisées sur des post-it ou dites à l’oral qui sont oubliées ou reportées par la suite. Il est également possible de spécifier l’objet de la réunion et d’adresser le tout à plusieurs personnes. Un agenda rudimentaire est également mis à disposition au sein de l’outil, permettant une visualisation des réunions plus simple et plus claire.

# Mission 1 – Portail Négoce

## Contexte

### L’ordonnancement

Au sein de l’entreprise EMINENCE existent plusieurs services, comme dans toute entreprise. Ressources humaines, accueil, service informatique, et autres services que l’on peut retrouver partout ailleurs.

Mais il existe également des services propres à EMINENCE, qui se rapprochent de son cœur de métier. De plus, étant donné la proximité immédiate de l’usine, connectée au siège social de l’entreprise, il existe plusieurs services dont la fonction principale est directement liée à l’usine : production, fabrication, et cætera. En particulier, il existe un service lié à la production, dont la fonction est de vérifier si, à tout moment, l’entreprise possède assez de produits pour fabriquer tel ou tel lot ou colis.

Son activité est donc dépendante d’une multitude de facteurs, les commandes effectuées sur internet étant diverses et imprévisibles. A noter que certains produits sont plus prisés lors de certaines saisons, comme par exemple des maillots de bains ou des sous-vêtements plus légers lors de l’été, qui laisseront la place aux vêtements plus chauds en hiver. Ainsi, ce service est constamment à l’écoute de l’équipe chargée de la production, et réalise de manière continue des commandes de produits afin de ne pas en manquer si ceux-ci sont nécessaires à la production de lots ou colis récemment commandés.

Le service de l’ordonnancement, contrairement à ce que l’on pourrait penser, ne se situe pas au niveau de l’usine. Des échanges et des vérifications sont faites avec l’équipe qui travaille à l’usine, mais les commandes, réunions et autres actions du service sont réalisées au sein de l’entreprise, ce qui facilite leur transmission au service suivant.

### Le service Négoce

Après passage de la commande, celle-ci est acheminée dans les bureaux du premier étage. Ceux-ci sont le lieu de deux services aux fonctions similaires : Le service Sous-traitance et le service Négoce.

Le service Sous-traitance a pour fonction de relayer une partie des commandes à des entreprises distantes, souvent présentes dans d’autres pays. Ces dernières sont souvent tirées des commandes internet, qui sont alors produites à l’étranger et ensuite expédiées. On pourra également faire appel à ce service lorsqu’on aura besoin de composants simples nécessaires à la fabrication des produits, comme par exemple du tissu, des élastiques, et cætera.

En revanche, le service Négoce a pour fonction de relayer une partie des produits que produirait normalement EMINENCE à d’autres entreprises réparties dans le monde. Il s’agit surtout ici de faire appel à des fournisseurs extérieurs, en leur fournissant des documents spécifiques décrivant les étapes de confection des produits, afin d’obtenir par la suite suffisamment de produits pouvant ensuite être expédiés par éminence sous forme de lots ou colis, à destination de grandes surfaces ou de particuliers ayant commandé sur Internet.

## De l’existant à la demande

### Workflow

Une fois que la commande est passée par l’ordonnancement, elle prend le statut de « **nouvelle ».** Elle est, après réception, examinée par les membres du service Négoce. L’ordonnancement spécifiant le fournisseur désigné, un employé du service est en général assigné à un fournisseur spécifique. Il faut bien sûr noter qu’en cas de problème il est possible aux autres employés de prendre en charge les opérations concernant un fournisseur, et également qu’un employé est à même de gérer plusieurs fournisseurs externes (le nombre de fournisseurs dépassant les 3500, il est rigoureusement impossible d’avoir un employé assigné à chacun d’entre eux).

Après décisions internes propres au Négoce, regardant les prix, dates d’expéditions demandées et de réception, la commande est « **validée »** par le service Négoce.

Elle est à ce moment-là renseignée par un document technique, qui regroupe l’ensemble des informations nécessaires à la confection des produits, d’une charte qui établit des règles quant à la production, l’envoi, ou toutes autres opérations regardant ce domaine, ainsi que la commande d’achat contenant l’ensemble des produits commandés, leur quantité et leur prix. Peu après, ces documents sont envoyés par e-mail au responsable du fournisseur.

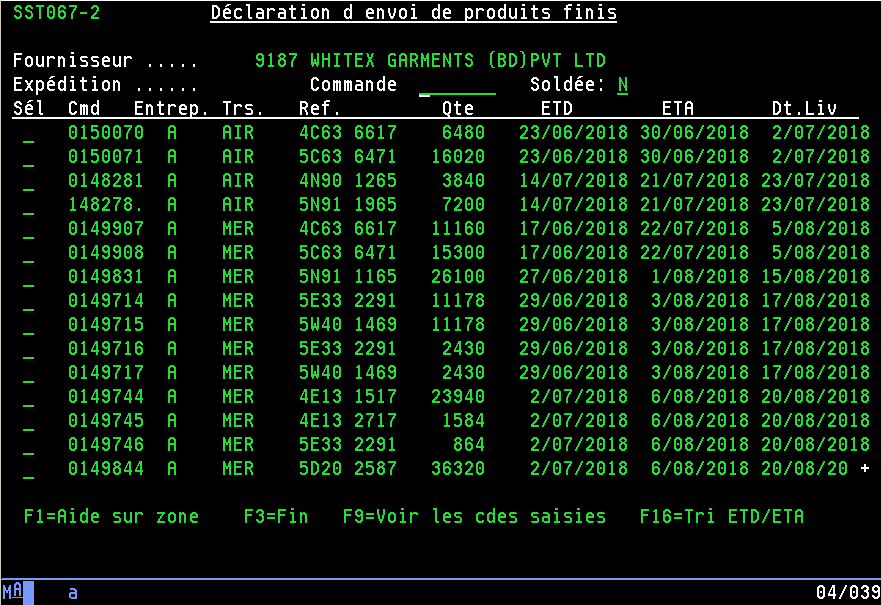
Le fournisseur doit alors accuser réception de la commande par mail, l’examiner, et ensuite indiquer au Négoce qu’il s’apprête à commencer la production, ce qui fait passer le statut de la commande à « **acceptée »**, sous-entendu par le fournisseur. Le fournisseur, par la suite, envoie les produits confectionnés, qui seront reçus sur le port de l’usine, où un traitement de qualité sera effectué sur une petite partie des produits.

Suite à la réception des produits, et en assumant qu’ils passent avec succès le contrôle qualité (dont s’occupait récemment mon collègue Vincent Descreux, au niveau informatique), ils sont enregistrés en base et comptabilisés, jusqu’au moment où la commande est considérée remplie par EMINENCE. Ceci donne le statut final de « **soldée** » à la commande, qui est ensuite payée au fournisseur.

### Portail négoce actuel

Les deux services, Négoce et Sous-traitance, étant similaires, on pourrait penser qu’ils utilisent tous deux les mêmes ressources. Cependant, le service de sous-traitance a fait, il y a quelques années, une demande au service informatique concernant une refonte du portail de sous-traitance.

En effet, avant la création de ce portail (par mon tuteur Sébastien Rousselle), les deux services se reposaient sur un écran AS400.



Ecran typique de l’AS400, pour le service négoce.

Il s’agit d’une sorte de console, certes différente du DOS que l’on peut retrouver sur Windows ou du terminal sur Mac, mais dont le fonctionnement s’en approche.

Des zones de fonctionnement sont définies lors de la programmation des pages, et l’utilisateur utilise les touches fléchées <↑> <↓> <←> <→> afin de se déplacer sur les champs éditables, qui sont repérables par les tirets « underscore » <\_>.

Le service sous-traitance a donc fait appel au service informatique afin qu’il puisse disposer d’une interface web simplifiée permettant de réaliser les opérations de manière plus simple et avec plus de visibilité. Sébastien a donc mis au point un outil qui s’est avéré être plus efficace qui a permis au service de pouvoir opérer de meilleure manière.

Le service Négoce, a donc décidé de faire une demande similaire auprès du service informatique : disposer d’une interface web plus intuitive.

Il faudra cependant souligner que la nature et les tâches assignées des deux services étant différentes, il ne suffisait pas de trivialement adapter l’interface réalisée par Sébastien afin qu’elle fonctionne pour le service Négoce. Le type des produits, les opérations à réaliser sur la commande et le fait que les fournisseurs doivent pouvoir accéder à une partie du nouveau portail sont d’autant de facteurs induisant la nécessité de création d’un tout nouveau portail, exclusivement destiné au service Négoce ainsi qu’aux fournisseurs qui y sont rattachés.

### Demande du service Négoce

Etant donné l’organisation particulière de l’entreprise, et compte tenu des problèmes rencontrés lors de l’émission de la demande, et de manière générale à la communication inter et intra services, il semble nécessaire de clarifier quelques points.

Tout d’abord, le service Négoce est supervisé par M. Frédéric Beaumer. Il est à l’origine de la toute première demande effectuée auprès du service. Cette demande était au début une idée, qui s’est ensuite transformée en véritable demande. Il a donc spécifié les éléments qu’il souhaitait voir sur le nouveau portail, ce qui a constitué la toute première opération sur laquelle j’ai dû travailler. Après une première période de réalisation de ma part a eu lieu une première réunion regroupant M. Beaumer ainsi que quelques employés du service. La réunion a été concluante, avec quelques points à améliorer.

Cependant, M. Jean-Michel Lacassagne, directeur général/industriel d’EMINENCE, étant en charge de plusieurs secteurs y compris celui du Négoce, a effectuée des modifications sur la demande initiale lors de la seconde réunion, à laquelle il était cette fois-ci présent. Sur certains points, le projet a même pris une direction opposée.

Après développement, une troisième réunion a eu lieu, cette fois-ci sans M. Lacassagne. Et à nouveau, M. Beaumer a réitéré qu’un certain point, auparavant modifié lors de la dernière réunion, devait être à nouveau changé, l’aperçu ne correspondant pas à ce qu’il avait demandé au départ.

Il s’agit ici de la dernière réunion ayant eu lieu, la phase courante de développement portant sur les points évoqués lors de cette dernière réunion.

Etant donné le caractère singulier de cette demande, et en tenant compte du fait que certains développements ont été réalisés en vain, je me dois de décrire chaque sous-demande, et, concernant les points différents d’une réunion à l’autre, les considérer comme des réalisations différentes. Je présenterai ainsi la demande de base et préciserai ensuite les modifications apportées à celle-ci au cours des différentes réunions.

## demandes & réalisations

### Cahier des charges

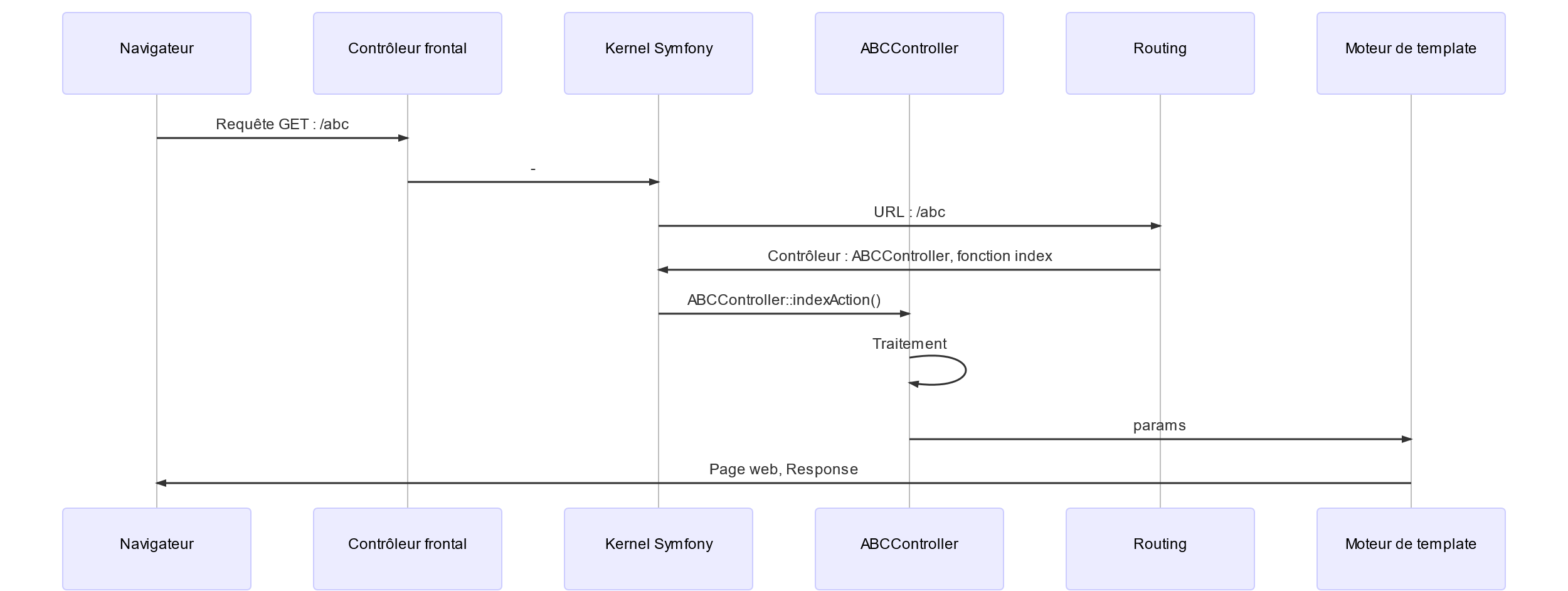
Le cahier des charges se présente comme tel :

* Un onglet « Commandes » dans lequel on pourra retrouver :
  + La liste des commandes passées par l’ordonnancement :
    - Informations de la commande
    - Informations du fournisseur
    - Pouvoir afficher le détail de la commande :
      * Produits avec le prix, libellé, tailles.
      * Quantité des produits et prix total
  + Une action pour chaque commande permettant :
    - D’envoyer les documents nécessaires à la confection du produit (Dossier technique, Commande d’achat)
    - D’envoyer un mail à des responsables que l’on pourra sélectionner, devant en même temps contenir les fichiers mais également les stocker à disposition des fournisseurs sur leur partie du portail
* Un onglet « Expéditions » permettant de :
  + Voir, pour chaque employé du négoce, les expéditions effectuées par les fournisseurs dont il est en charge, consulter leur statut et la quantité reçue.
  + Afficher le détail pour chaque expédition
* Deux onglets accessibles par les fournisseurs :
  + Un onglet « Commandes » qui, à l’instar de celui pour le Négoce, leur montre les commandes qui lui ont été passées, avec :
    - Visualisation des documents envoyés lors du passage de la commande
    - Menu pour accepter ou non la commande
  + Un onglet « Expéditions » qui permet :
    - De consulter les expéditions déjà réalisées
    - De créer une expédition :
      * Pouvoir sélectionner parmi les commandes envoyées
      * Pour chaque commande sélectionnée, de pouvoir saisir une quantité de produits

### Technologies choisies

Backend

En ce qui concerne le backend de l’application, j’ai dû utiliser le support existant qui est Symfony. Les onglets mentionnés dans le cahier des charges allaient donc être présentés sur une page faisant partie d’un nouveau module, ou **Bundle** selon les termes du Framework. L’avantage d’utiliser un tel système est que des mesures de sécurité, notamment concernant les rôles utilisateurs, ont déjà été mis en place pour d’autres modules répartis sur le portail. Ainsi, il sera aisé d’afficher, pour l’utilisateur connecté sur le portail Négoce, les onglets auxquels il a le droit d’accéder.

Rappelons un instant le fonctionnement de Symfony :

Une requête passée par le navigateur est donc interceptée par le contrôleur frontal, qui fait ensuite appel au Kernel. Celui-ci examine le fichier de routing (généralement routing.yml), qui contient toutes les fonctions de tous les contrôleurs destinés à interagir lorsque la route est appelée.

Une fois que la fonction et le contrôleur sont identifiés, ils sont appelés, avec en paramètre la requête. Le traitement normal de la fonction du contrôleur désigné s’effectue, et dans le cas où le contrôleur utilise la fonction « render », qui retourne une page, il fait appel au moteur de template qui se chargera d’afficher la page correctement.

Le moteur de template par défaut, généralement utilisé avec Symfony est Twig. Il permet de référencer des fichiers de manière modulaire. Prenons un fichier html inventé de toutes pièces.



Twig va permettre la décomposition de cette page en plusieurs modules, séparés par leur fonction.

base.html.twig



list.html.twig



Twig permet d’avoir une page simplifiée, définie par des éléments contenus dans des « blocks ».

Front-End

Conseillé par mon collègue Vincent Descreux ayant déjà utilisé le framework sur certaines pages, et également motivé par ma propre inconnaissance de cet outil, j’ai utilisé AngularJS afin d’avoir rapidemment une interface facilement dynamique et adaptable aux besoins de l’utilisateur.

Il est important de différencier AngularJS d’Angular simple. En effet, l’un est un script javascript inclus dans la page, créant une série d’événements en fond qui, couplé à un css personnalisé, permet d’avoir une interface dynamique. En revanche, le second est un framework solide destiné aux webapps.

Afin de simplifier les références futures de ce composant, nous utiliserons le terme « Angular » pour parler d’Angular JS. Lors des comparaisons, les deux termes seront employés clairement.

Un système incluant Angular JS se présentera généralement comme ceci :

*index.html*

**

*script.js*

**

Tout d’abord le fichier html inclut le fichier javascript, qui va créer plusieurs events et éléments qui vont tourner dans le navigateur avant l’affichage de la page. Il faudra indiquer à Angular son domaine de définition avec ng-app, et la zone du contrôleur en question avec ng-controller.

La fonction ng-repeat a pour but de répéter l’élément sur lequel elle est renseignée, en spécifiant un ensemble d’élements, et de lister le contenu de celui-ci. Il s’agira ici d’une liste d’élèves, dont on affichera dynamiquement le prénom avec les doubles accolades {{ }}.

On pourra remarquer qu’Angular et Twig utilisent tous deux les doubles accolades afin d’afficher des informations qui ne doivent pas être interprétées comme simple texte html mais bien comme variables dynamiques. Lorsqu’on utilise les deux dans le même fichier, cela crée donc des conflits.

Il existe deux manières de résoudre ce problème :

* Changer les balises de reconnaissance de Twig pour que les doubles accolades soient exclusivement reconnues par Angular
* Englober le code Angular par des tags Twig {% verbatim %}…{% endverbatim %}, ce qui va forcer Twig à interpréter le code contenu entre ces balises comme du texte.

La solution utilisée par mon entreprise est la seconde. Je ne suis personnellement pas très convaincu par cette décision, ces tags posant souci lorsqu’on doit appeler des variables Twig au sein du code Angular. Cela donne lieu à des ouvertures et fermetures intempestives de balises, qui rend le code peu visible et manque d’efficacité.

Pour consolider l’apparence et obtenir un résultat dynamique à souhait, sans pour autant trop se verser dans l’édition de styles et de scripts farfelus, ce qui en plus de réinventer la roue se révèle être véritablement chronophage, j’ai également pris la décision d’inclure la libraire de google Angular Material.

Cette librairie permet d’inclure encore plus de fonctionnalités compatibles avec Angular. La principale différence avec les directives Angular est qu’il s’agit ici de véritables composants, donc le style et le comportement sont définis par des scripts et feuilles de styles créées par Google. En suivant la documentation, il est possible de créer des composants facilement, toujours compatibles avec Angular.

De la même manière qu’Angular JS et Angular, il existe une version solide de ce composant, à ne pas confondre avec ici encore un ou plusieurs script(s) et feuille(s) de style, qui ne prennent vie qu’au sein du navigateur.

Base de données

Je n’ai pas eu le choix concernant cette partie du projet. Travaillant sur un site existant, il aurait été de toute façon peu judicieux de créer une base extérieure, je n’ai donc pas été déçu lorsque j’ai appris ce détail avant de commencer à développer.

Les AS400 étant conçus par IBM, EMINENCE se sert de leur base de données, IBM DB2, pour fonctionner. La connexion est globale et accessible dans tous les fichiers PHP.

Afin de récupérer les données, il faut se servir de la syntaxe PDO :

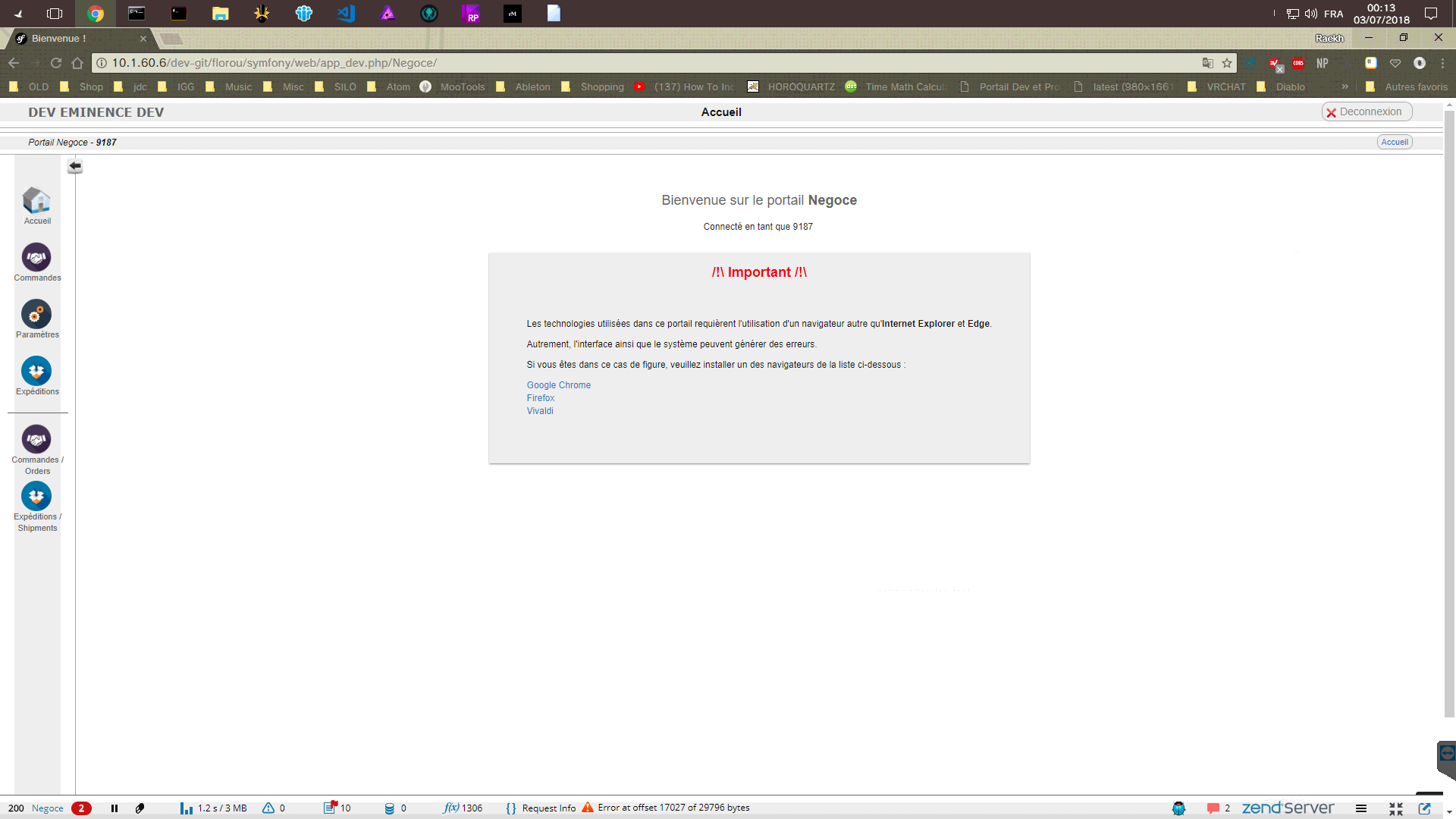


Les données peuvent ensuite être traitées dans la suite de la fonction, ou bien directement renvoyées sous forme JSON, avec la syntaxe :

return new JsonResponse($result).

### Processus d’implémentation

Bundle Symfony

Suivant le fonctionnement de Symfony, j’ai créé un nouveau Bundle, appelé NegoceBundle. Ensuite, j’ai référencé les composants de base présent dans le template twig, afin d’obtenir un fichier simple et lisible, comme démontré dans l’exemple au-dessus. Ceci a permis à chaque page d’afficher le menu horizontal et la sidebar verticale :

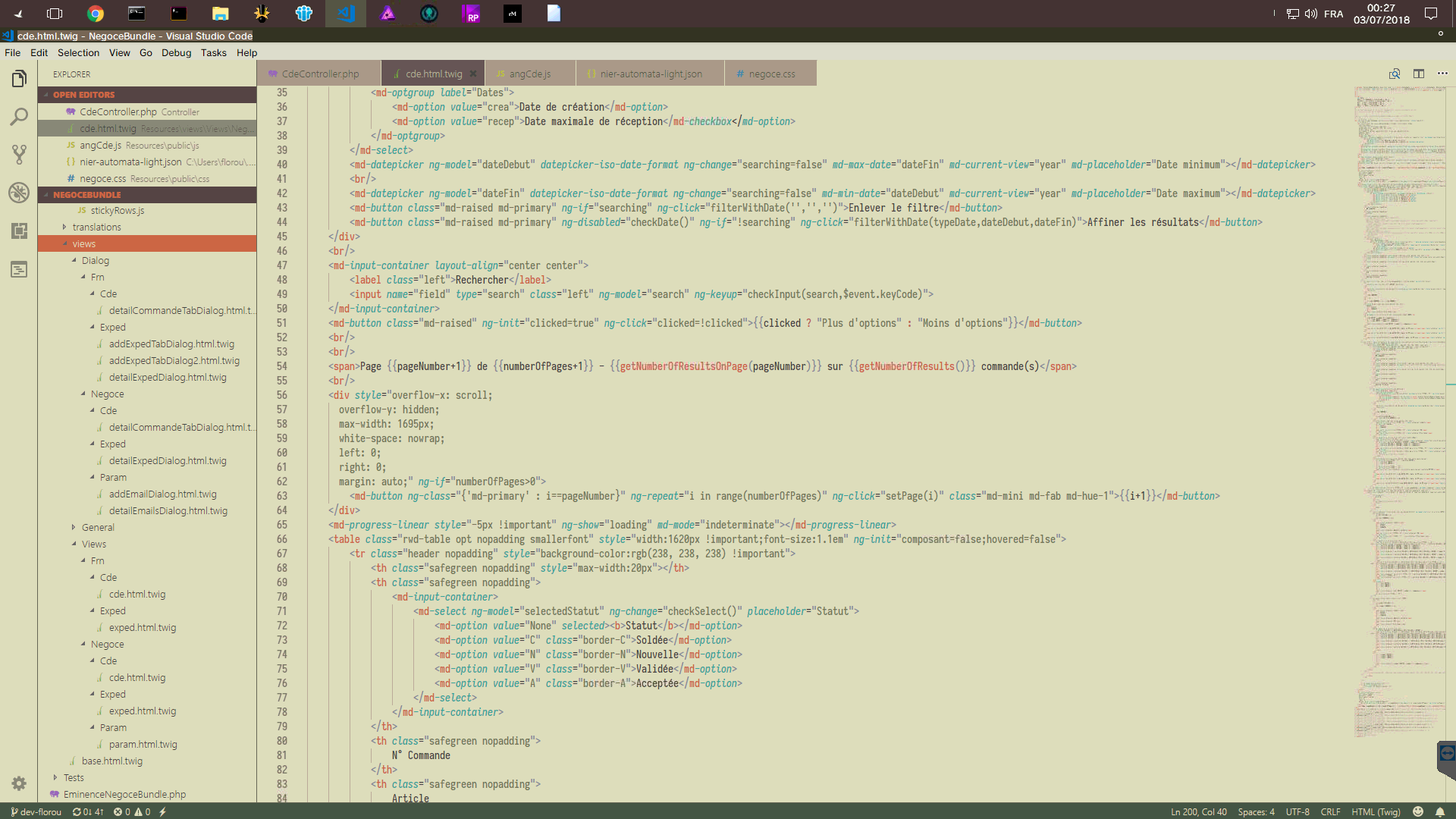
Fichiers

Dès le début du projet, je me doutais bien que j’allais devoir trouver une manière efficace de stocker les fichiers dans l’architecture Symfony.

En effet, j’allais devoir gérer plusieurs types de fichiers, certains destinés au service Négoce, d’autres aux fournisseurs. De plus, chacun de ces fichiers remplit une fonction différente, et certains d’entre eux sont en réalité des template de dialogues, qui ne sont pas vraiment des pages à part entière mais plutôt des composants destinés à être appelés par la fonction $mdDialog d’Angular Material, qui affichera le composant convenablement.

J’ai donc conçu l’architecture suivante :

*Views*



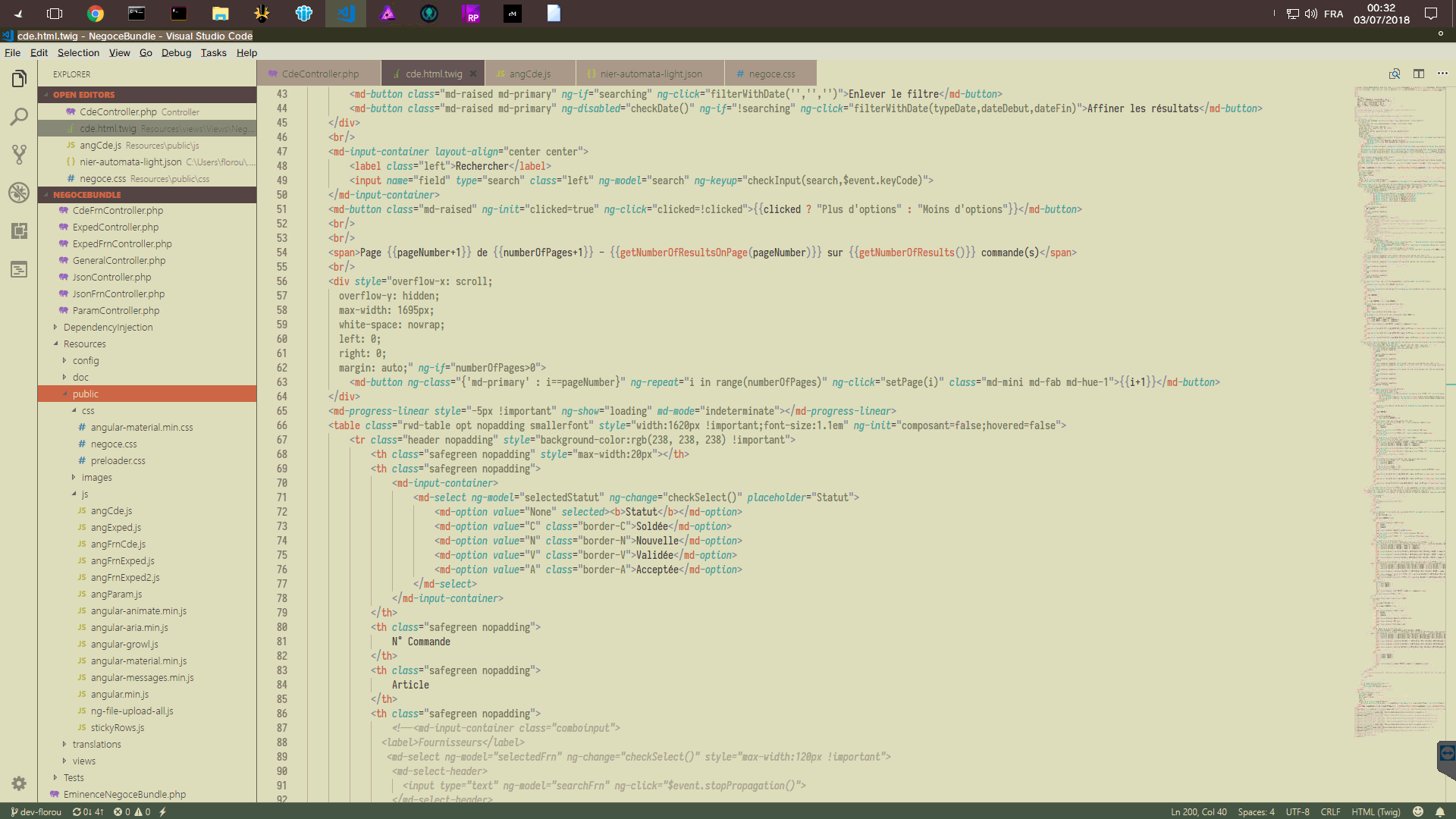
Les vues sont donc découpées en deux dossiers : Dialog et Views.

* *Views* représente les pages telles qu’on pourrait y accéder via un lien.
* *Dialog* représente les templates qui serviront de repère pour Angular Matérial.

Au sein de ces deux dossiers, il existe un dossier Negoce et Frn pour différencier les deux parties.

Pour finir, à l’intérieur de ces sous-dossiers existent des dossiers destinés à séparer les différentes pages.

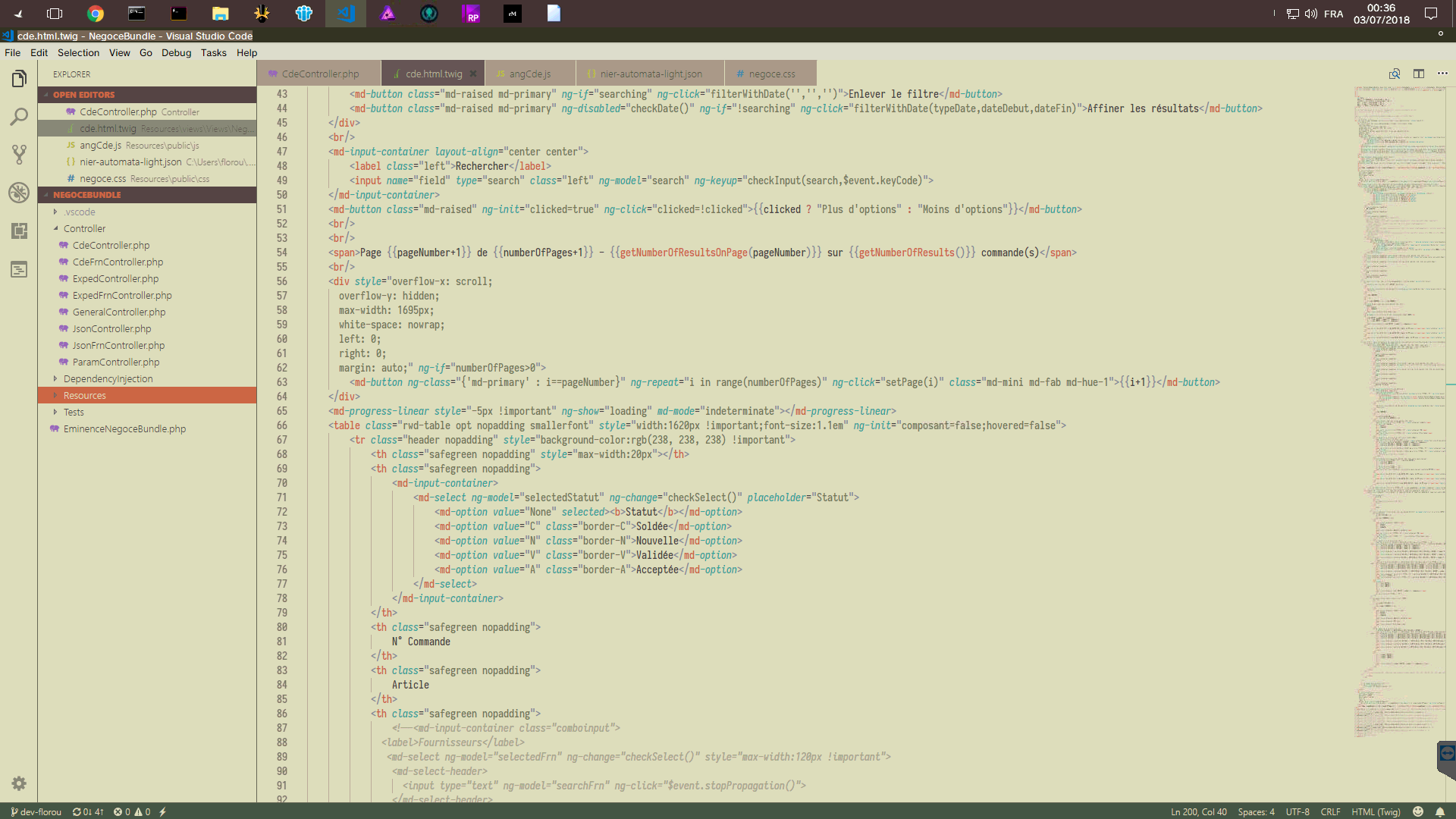
*Scripts et Feuilles de style*



Il existe un fichier javascript pour chaque page. Les fichiers Angular sont présents sur quelques pages pour résoudre des problèmes de versioning sur quelques éléments.

Il n’existe en revanche qu’un CSS pour tout le Bundle. En effet, cette pratique oblige une rédaction simple des règles CSS, de façon à ce qu’elles puissent être réutilisées facilement dans toutes les autres pages du Bundle.

*Contrôleurs*



Il existe un contrôleur par set d’actions différentes. Pour l’instant, chaque page s’occupe d’une seule notion, il y a donc un contrôleur par page. J’ai choisi cette implémentation pour suivre le même code qu’avec les vues.

On pourra observer encore une fois la séparation entre les fonctions destinées au Négoce et celles destinées aux fournisseurs.

Il y a également, dans ce cas-ci, la distinction entre les contrôleurs normaux (par exemple CdeController) et les deux contrôleurs exclusivement destinés au renvoi de données en JSON, et non de pages en soi. Ces deux contrôleurs spéciaux sont également séparés en fonction de s’ils renvoient des données regardant le Négoce ou bien les fournisseurs.

Lorsque je travaille sur une page particulière, tous les noms de fichiers se regroupent autour d’un même terme. Je suis donc moins à même de me tromper de fichier en travaillant ou même de peiner à m’y retrouver dans les différents fichiers.

*Routing*

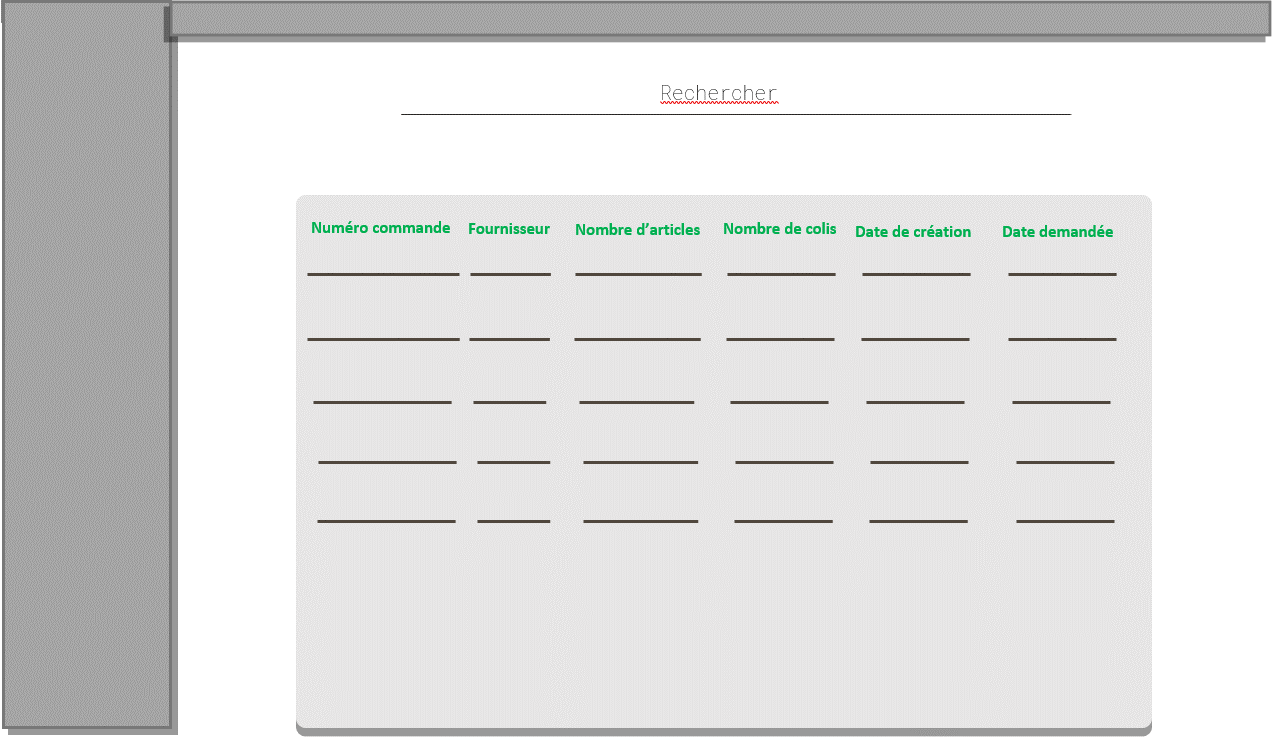
Il n’est pas vraiment nécessaire de montrer ici le fichier de routing. Il s’agit simplement d’un répertoire de routes, appelant telle ou telle méthode en fonction de la route atteinte par le navigateur.

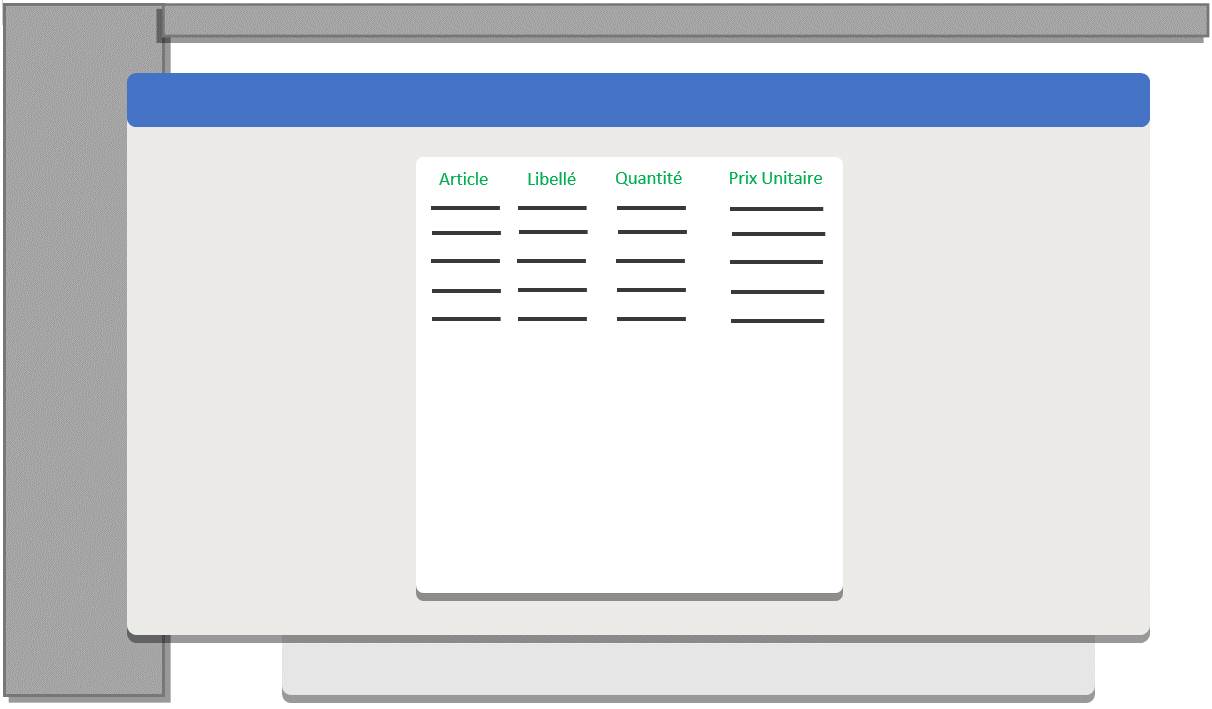
### Réalisations

Etant donné les différents changements de version de git et de Symfony, il est impossible de prendre des captures d’écran de versions trop antérieures de l’application. Dans ces cas-ci je ne pourrai donc que mettre des schémas descriptifs des anciennes versions.

#### Commandes

Lors de la première demande, il avait été spécifié d’afficher les commandes dans une liste contenant des informations relatives à cette commande, avec une possibilité de cliquer pour voir les détails de cette commande.





Cependant, après la première réunion, M. Frédéric Beaumer a expliqué que les articles étaient organisés d’une façon bien précise.

Un produit est constitué, entre autres, d’un SKU (*Stockage Keeping Unit*).

Il est composé comme suit :

0 AAAA BBBB C DD

Où :

* *0* : Produit fini
* *AAAA*: Référence produit
* *BBBB*: Référence Coloris
* *C*: Taille
* *DD*: Conditionnement.

Ainsi, l’article 05C330800631 et l’article 05C330820631 sont les mêmes, à l’exception des coloris.

Par ailleurs, les prix unitaires étant parfois différents en fonction des tailles, il fallait pouvoir montrer cette différence plus facilement. C’est pourquoi la seconde itération de ce menu a été réalisée en colonne plutôt qu’en ligne, pour pouvoir montrer, en fonction de chaque taille, le prix unitaire et la quantité commandée.