

---

# RAPPORT DE PRODUCTION – ADNEOM NEWS

---

## Progression du développement

Ayant eu un problème avec mon serveur mail, j'ai reçu l'énoncé samedi en après-midi. Mon travail a commencé avec une analyse de l'énoncé. J'ai capturé les différents besoins et ai commencé à réfléchir à ma méthode de travail.

Il est bon de savoir que j'ai réalisé ce projet en 24h, soit du samedi soir au dimanche soir (avec interruptions), étant donné que le lundi matin précède notre entretien, j'ai été travailler sur mon lieu de stage au Luxembourg. Je n'ai donc pas pu faire les choses en profondeur comme je l'aurais voulu, néanmoins le projet est terminé et fonctionnel. Le style a été bâclé, et je n'ai pas eu le temps d'effectuer mes tests.

## Chapitre 1 : Introduction

J'ai ensuite installé mon environnement de travail. En effet, ayant revendu mon ancien ordinateur il y a peu, je n'avais pas encore d'environnement correct pour travailler en AngularJs. Auparavant, j'avais été formé à l'utilisation des framework Backbone et Rivets dans le but de mon stage toujours en cours chez CGI Luxembourg, je connaissais donc les principes de bases de ce genre de framework. J'ai recherché un petit cours concernant Angular et l'ai parcouru, en cherchant d'abord les informations sur les différents points techniques du framework (routes, directives,...), je l'ai ensuite laissé de côté et me suis chargé d'installer les différents outils tels que IntelliJ, Yeoman, Bower, Grunt, NPM, Angular, SASS, ...

Une fois l'environnement de travail installé, j'ai démarré les différentes installations pour obtenir le squelette du projet.

## Chapitre 2 : Développement de la page d'accueil

La première tâche a été de naviguer sur les deux sites mentionnés dans l'énoncé pour voir exactement à quoi correspondait le sujet du travail.

Une fois ce sujet apprivoisé, je me suis lancé dans la création de la page d'accueil. Cette page va contenir la liste des news, avec leurs actions associées (vote, lien externe, lien vers les commentaires).

L'application faisant des appels à l'API d'Adneom, il m'a semblé utile de créer un utilitaire de services ajax, nommé `servicesAjax`, qui a été généré à l'aide de Yeoman :

```
'yo angular:services servicesAjax'
```

Cette ligne de commande a lancé des processus qui ont généré le dossier `services` dans `app`, dans lequel un fichier `servicesajax.js` a été créé.

Ce fichier contient les différents appels ajax utilisés dans notre application, et ce fichier sera injecté dans le contrôleur quand nous en aurons besoin. Nos appels ajax se feront sous cette forme dans le fichier servicesajax.js :

```
listNews: function () {  
    return $http.get('http://adneom.herokuapp.com/api/posts');  
}
```

Nous avons ci-dessus créé une fonction listNews qui va faire appel à l'url <http://adneom.herokuapp.com/api/posts> en GET, et qui va retourner le résultat de cette requête.

Cette fonction sera appelée dans notre contrôleur de cette manière :

```
serviceAjax.listNews().success(function(data) {  
    //Traitement en cas de succès  
});
```

Le contrôleur de la page Main contiendra :

- Une fonction permettant de lister l'ensemble des news et de les binder à la variable \$scope.
- Une fonction permettant de voter pour une news. Le vote sera ajouté via un appel ajax, et la liste des news sera rafraîchie

## Chapitre 3 : Développement des détails d'une news

Une fois la page d'accueil terminée, il m'a fallu recréer une route qui permettra l'affichage des détails d'une news.

Le contrôleur va également faire un simple appel ajax en fonction de l'id de la news demandée (id présent dans l'url, accessible grâce à \$routeParams. Cette fonctionnalité va permettre de charger la liste de commentaires de la news en question et l'afficher sur la page grâce au binding via \$scope.

Une fonction de ce contrôleur permet également l'ajout d'un commentaire (et le rafraîchissement de la liste). Si les deux champs sont vides, un message d'erreur avertira l'utilisateur qu'il a mal rempli le formulaire afin d'éviter de surcharger la base de données de news vides.

## Chapitre 4 : Développement de l'ajout d'une news

Rien de bien compliqué sur cette fonctionnalité. Il s'agit du même fonctionnement que le contrôleur de la page de détails d'une news.

Nous voilà arrivés en fin de développement des fonctionnalités. Il est temps de s'attaquer à l'apparence.

## Chapitre 5 : Style de l'application

Une fois le SASS installé correctement et fonctionnel, je me suis mis au boulot. N'ayant pas trouvé d'utilité à utiliser le SASS sur ce projet, j'ai simplement créé trois variables contenant des couleurs... Je ne voyais pas quelle autre fonctionnalité du SASS utiliser sur un projet de ce type.

J'ai tenté de respecter au mieux votre site web d'origine pour le design, dans un délai assez restreint (J'ai réalisé ce projet en 24h). J'ai donc aspiré votre header, et utilisé un de vos backgrounds.

Le style n'est pas parfait, mais je n'avais malheureusement pas le temps de m'y plonger pendant des heures.

## Chapitre 6 : Mise en ligne sur Github

Cette étape peut sembler simple, mais j'ai éprouvé pas mal de soucis sur mon ordinateur au moment de la mise en ligne sur GitHub.

J'avais déjà utilisé des systèmes tels que Bitbucket auparavant, mais jamais de création de repository. Malheureusement les erreurs se sont enchaînées et j'ai perdu pas mal de temps.

Avec persévérance, j'ai réussi à mettre mon application en ligne. Je n'ai pas publié les dépendances, le package.json fera son job !

Lien du repository : <https://github.com/Davidczz/newsAdneomDupuis/>