

# Web service client consumer



Auteur	Brandon Mérillat
Date de début de projet	19/12/16
Date de fin de projet	06/02/17
Date de fin du rapport	06/02/17
Version	Rapport.docx

## Table des matières

1	Introduction .....	3
2	Documentation de développement .....	3
2.1	Cahier des charges .....	3
2.2	Définition des conventions applicables .....	3
2.3	Planning initial.....	3
3	La phase de développement .....	4
3.1	C# Méthode « RecupereMeteo » .....	4
3.2	Explication choix de code.....	5
3.3	La phase de tests .....	5
4	Mode d'emploi utilisateur.....	7
5	Conclusions .....	9
6	Annexes .....	10
6.1	Journal de bord .....	10
6.2	Cahier des charges détaillé .....	11
6.3	Références .....	11

# 1 Introduction

J'ai été mandaté par M. Sacchetti pour réaliser un projet de météo en PHP ainsi qu'en C# ou en Android.

Le projet consiste à afficher la météo pour une ville se trouvant dans une liste déroulante. Quatre villes doivent se trouver dans la liste par défaut. On peut ajouter des villes qui seront conservées.

Le projet à commencer le 19/12/2016 et doit être rendu le 6/02/2017 à 16h20.

## 2 Documentation de développement

### 2.1 Cahier des charges

Il faut impérativement développer 2 applications. Une en PHP et une en C# ou Android. L'application permet d'afficher la météo des prochains 5 jours. Des images doivent illustrer chacun de ces jours. Le choix de la localité doit se faire par une liste déroulante. Par défaut, les villes présentes sont Neuchâtel, La Chaux-de-fonds, Berne et Lausanne. Il doit être possible d'ajouter des localités supplémentaires qui seront conservée au prochain lancement de l'application.

### 2.2 Définition des conventions applicables

Utilisation des conventions de nommage du CPLN.

### 2.3 Planning initial

	Planning initial					
	19.12.2016	09.01.2017	16.01.2017	23.01.2017	30.01.2017	06.02.2017
Création du Trello/Croquis						
Afficher Météo Neuchatel PHP						
Afficher Météo liste déroulante PHP						
Choix du nombres de jours PHP						
Ajouter localité PHP						
Conserver localité PHP						
Afficher Météo Neuchatel C#						
Afficher Météo liste déroulante C#						
Choix du nombres de jours C#						
Ajouter localité C#						
Conserver localité C#						
Ecriture du rapport						

## 3 La phase de développement

### 3.1 C# Méthode « RecupereMeteo »

```
private void RecupererMeteo(string strVille)
{
    Meteo MeteoReponse = new Meteo();

    try
    {
        Connexion au site.
        string endPoint = @"http://www.prevision-meteo.ch/services/json/" + strVille;
        var client = new RestClient(endPoint);
        var json = client.MakeRequest();
        object objResponse = JsonConvert.DeserializeObject(json, typeof(Meteo));

        //Converti dans le type requis
        MeteoReponse = (Meteo)objResponse;

        int iJours = Convert.ToInt16(cbxPrevision.Selected.Value);
        EffacerTout();
        //La boucle for permet de savoir combien de jours on affiche.
        for(int i = 0; i <= iJours; i++)
        {
            //Le switch affiche en fonction de i le nombre de jours. Ecriture des
            //prévisions dans les textboxes et affichage des images.
            switch (i)
            {
                case 0: //Le 0 est là car lorsqu'on lance l'application il se peut que
                //le programme vienne ici. Dans ce cas, comme nous n'avons pas défini de jour, il
                //pourrait générer une erreur sans ce « case ».
                    break;
                case 1:
                    tbxMeteoJour0.Text = AfficherPrevision_Jour_0(MeteoReponse);
                    img_day_0.Source = AfficherIcône(MeteoReponse.fcst_day_0.icon);
                    break;
                case 2: tbxMeteoJour1.Text = AfficherPrevision_Jour_1(MeteoReponse);
                    img_day_1.Source = AfficherIcône(MeteoReponse.fcst_day_1.icon);
                    break;
                case 3:
                    tbxMeteoJour2.Text = AfficherPrevision_Jour_2(MeteoReponse);
                    img_day_2.Source = AfficherIcône(MeteoReponse.fcst_day_2.icon);
                    break;
                case 4:
                    tbxMeteoJour3.Text = AfficherPrevision_Jour_3(MeteoReponse);
                    img_day_3.Source = AfficherIcône(MeteoReponse.fcst_day_3.icon);
                    break;
                case 5:
                    tbxMeteoJour4.Text = AfficherPrevision_Jour_4(MeteoReponse);
                    img_day_4.Source = AfficherIcône(MeteoReponse.fcst_day_4.icon);
                    break;
                default:
                    break;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
```

Explication :

Connexion au site -> Cette partie permet de lancer la requête sur le site. Et de récupérer le résultat sous format Json.

Converti dans le type requis -> Il faut convertir le format Json fraîchement reçu dans un format que l'on peut utiliser. C'est pour cela qu'il est converti en objet de type « Meteo ».

Boucle For / switch -> La boucle for fera des tours en fonction du nombre de jours initialement choisis. Le nombre de jours est récupéré dans la variable « iJours ». Le switch quant à lui, se base sur la variable 'i'. En fonction du nombre de jours, on aura différents affichages qui seront ainsi visible à l'écran.

### 3.2 Explication choix de code

#### PHP

- Switch : Le switch a été utilisé pour ne pas prendre trop de temps sur la partie PHP. Il est également utile car permet de gérer l'intrusion d'un jour supérieur ou inférieur dans l'URL (7 jours / -2 jours). Ce qui ajoute une petite vérification d'erreur.
- Cookie : Les cookies sont utiles car ils permettent de sauvegarder d'une session à l'autre les choix de la ville. Qui plus est, on peut choisir la durée de vie de ces derniers et ils sont enregistrés en local. Ce qui permet à deux utilisateurs distincts de ne pas voir les mêmes villes.

#### C#

- Fichier Texte : Le fichier texte permet de garder une trace en locale des villes que l'on a ajoutées.
- Méthode : J'ai créé une méthode pour chacun des jours séparément. J'ai fait cela pour que chaque jour soit différent. Cela permet également de pouvoir modifier plus facilement chaque jour indépendamment des autres.

### 3.3 La phase de tests

Test à effectuer		Résultat escompté	Résultat obtenu	Constatation
Généralités				
C#	Afficher les prévisions pour Neuchâtel sur 5 jours.	Affichage des 5 jours	Affichage des 5 jours	OK
	Afficher les prévisions pour une ville se trouvant dans la liste déroulante. 5 jours	Affichage des 5 jours pour la bonne ville.	Affichage des 5 jours pour la bonne ville	OK
	Vérifier que Neuchâtel, Berne, La-Chaud-de-fonds et Lausanne sont présent	4 villes	4 villes	OK
	Ajout de Genève	Genève est ajouté.	Geneve est ajouté	OK
	Relancer le programme pour vérifier que l'ajout de Genève est sauvegardé	Genève est sauvegardé.	Geneve est sauvegardé.	OK
PHP	Afficher les prévisions pour Neuchâtel sur 5 jours.	Affichage des 5 jours	Affichage des 5 jours	OK
	Afficher les prévisions pour une ville se trouvant dans la liste déroulante. 5 jours	Affichage des 5 jours pour la bonne ville.	Affichage des 5 jours pour la bonne ville	OK
	Vérifier que Neuchâtel, Berne, La-Chaud-de-fonds et Lausanne sont présent	4 villes	4 villes	OK
	Ajout de Genève	Genève est ajouté	Geneve est ajouté	OK

	Relancer le programme pour vérifier que l'ajout de Genève est sauvegardé	Genève est sauvegardé.	Geneve est sauvegardé	OK
	Changer la valeur du jour dans l'URL	Le programme ne plante pas. Et affiche un message disant de choisir le jour.	Le programme ne plante pas. Et affiche un message disant de choisir le jour.	OK

## 4 Mode d'emploi utilisateur

Utilisation de l'application C# :

The screenshot shows a web application window titled "Web service client consumer". It features two dropdown menus at the top: "Localité :" with "Neuchâtel" selected, and "Prévision pour :" with "5" selected. Below these are five panels for the days of the week: Lundi, Mardi, Mercredi, Jeudi, and Vendredi. Each panel displays the weather condition, temperature range, and a corresponding icon. To the right of the panels is a section for adding a new city, with a text input field and a "Valider" button.

Jour	Météo	Temp. minimal	Temp. maximal
Lundi	Eclaircies	0°	4°
Mardi	Eclaircies	0°	4°
Mercredi	Fortement nuageux	1°	5°
Jeudi	Eclaircies	0°	4°
Vendredi	Faiblement nuageux	0°	6°

Par défaut, le programme affiche la météo des 5 prochains jours à Neuchâtel.

Pour changer la localité, cliquez sur la liste déroulante où « Neuchâtel » est marqué (Se référer à la capture d'écran.). Pour changer le nombre de prévision souhaiter, cliquez sur la liste déroulant où « 5 » est affiché (Se référer à la capture d'écran.).

Pour ajouter une ville, inscrivez votre ville en bas à droite, puis cliquez sur « Valider ». Un message apparaîtra à l'écran pour vous indiquer que la ville a été ajoutée.

La ville ainsi ajoutée apparaîtra tout en bas de la liste et sera sauvegardée pour votre prochaine connexion.

Utilisation de l'application PHP :

The screenshot shows a web application interface with a dark header bar. At the top left, the text 'Localité et nombre de jours' is followed by two dropdown menus: 'Neuchâtel' and 'Aujourd'hui'. Below these is a 'Valider' button. A large horizontal separator line with curved ends divides the interface. Below the separator, the heading 'Choisissez une ville' is displayed in a large, bold font. Underneath, the instruction 'Choisissez le nombre de jour à afficher.' is shown. Another horizontal separator line with curved ends follows. Below this, the text 'Ville à ajouter' is positioned above a text input field. At the bottom left, there is an 'Ajouter' button.

Voici l'affichage par défaut de l'application PHP. Pour choisir la ville, cliquez sur le « Neuchâtel » (Référence à la capture d'écran) puis choisissez quel villes vous souhaitez se voir. Sélectionnez ensuite le nombre de jours. Pour ce faire, cliquez sur « Aujourd'hui » puis sélectionnez le nombre de jour désiré. Cliquez ensuite sur « Valider » pour voir apparaître la météo.

Pour ajouter une ville, entrez la ville désirée dans le champ texte, puis cliquez sur « Ajouter ». ATTENTION les villes doivent être écrites dans la langue de la ville. (Bâle -> Basel car situé en Suisse allemande).



## 5 Conclusions

Les deux applications devant être rendues sont opérationnelles. Elles permettent toutes deux d'afficher les prévisions météorologiques sur 5 jours. Toutes les fonctionnalités devant être ajoutées sont fonctionnelles, on peut ajouter une ville, la choisir dans la liste, choisir le nombre de jours que l'on souhaite voir s'afficher et les villes ajoutées restent lors du second lancement des applications. Des images illustrent chaque jour. En revanche, les 2 applications ne gèrent pas encore les erreurs.

Amélioration potentielle: Ajout des sécurités relatives à l'entrée de la ville. Vérification de la ville pour savoir si elle existe ou non. Optimisation de certaines parties du code. Permettre d'ajouter une ville dans la bonne écriture sans générer d'erreurs (Ex : Écrire Bâle au lieu de Basel).

Ressenti : Le projet était très intéressant à réaliser. Le fait de devoir développer 2 applications utilisant un langage de programmation différent est un type de projet qu'il me plairait à refaire. En revanche, j'aurais dû travailler un peu plus sur la partie PHP pour rendre le code un peu plus « propre ». En effet, l'utilisation du switch rend le code plutôt vulgaire. Si je suis amené à travailler sur ce projet à nouveau ou à pouvoir l'améliorer, c'est la partie que je changerais en priorité. Sinon, le fait de devoir se mettre dans les conditions d'un TPI a rendu le projet beaucoup plus attractif et a permis un engagement plus important.

## 6 Annexes

### 6.1 Journal de bord

Nom : Mérillat	Prénom : Brandon	Classe : 3INF
Semaine N° : 17	Du : 19.12.2016	Au : 23.12.2016
Genre de travail		Durée
<u>Lundi</u> : (9 périodes)		
Travail sur le nouveau projet reçu. <ul style="list-style-type: none"><li>Analyse du cahier des charges</li><li>Réalisation du Trello / Planning</li><li>Réalisations de maquette</li></ul>		5

Nom : Mérillat	Prénom : Brandon	Classe : 3INF
Semaine N° : 20	Du : 23.01.2017	Au : 27.01.2017
Genre de travail		Durée
<u>Lundi</u> : (9 périodes)		
Malade		
<u>Mardi</u> : (9 périodes)		
Mis en place du site PHP avec Bootstrap.		2p
Déplacement des tablaris du rack idéfix		7p
<u>Mercredi</u> : (4 périodes)		
Début du code de l'application PHP		2p

Nom : Mérillat	Prénom : Brandon	Classe : 3INF
Semaine N° : 21	Du : 30.01.2017	Au : 03.02.2017
Genre de travail		Durée
<u>Lundi</u> : (9 périodes)		
Travail sur le projet « Web service client consumer ». Mise en place du service d'affichage météo en PHP avec le format Json.		6
Création du projet C#. Mise en place de l'interface et de la classe « Meteo ». Début du codage		3

<i>Mardi: (9 périodes)</i>	
Avancée dans le projet C# de la météo. Finalisation de l'interface. Finalisation de de l'affichage de la météo. Ajout du changement de jours. Début de l'implémentation de l'ajout de villes	9
<i>Mercredi : (4 périodes)</i>	
Finalisation de l'ajout des villes. Ecriture de commentaire. Début du rapport	4

## 6.2 Cahier des charges détaillé

Le client veut une application permettant d'afficher la météo des prochains 5 jours. Des images doivent illustrer chacun de ces jours.

Le choix de la localité doit se faire par une liste déroulante. Par défaut les villes présentes sont Neuchâtel, La Chaux-de-fonds, Berne et Lausanne.

Vous devez impérativement développer 2 applications :

- La première en PHP, en tenant compte de la responsivité et donc le Framework Bootstrap.
- Une deuxième application : soit en C# soit en Android.

L'utilisateur doit pouvoir ajouter des localités supplémentaires, ces localités seront conservées au prochain lancement de l'application. Proposez des solutions différentes et choisissez celles-qui vous convient le mieux.

## 6.3 Références

Météo :

<https://pixabay.com/fr/pr%C3%A9visions-m%C3%A9t%C3%A9o-m%C3%A9t%C3%A9o-soleil-nuage-146472/> Image de la météo.

<http://www.prevision-meteo.ch> Site pour la météo.

C# PHP :

<https://www.youtube.com/watch?v=3uuZhHXS1Y> L'image provient de la vidéo.

Trello :

<https://trello.com/b/XbRnH5Wb/brandon-meteo>