# Météo 2019 - 2020

## Objectif

Créer une commande unix qui va permettre de consulter automatiquement le site de météo france pour y collecter la météo réelle ainsi que les prévisions météo pour une ville donnée. Par la suite, il sera possible de comparer les prévisions aux différents jours avec la météo réelle du jour.

## Déroulement

Pour vous simplifier le développement, cette commande aura donc plusieurs phases :

* Initialisation du programme
* Effacement des téléchargements précédents
* Téléchargement de la météo pour une ville donnée dans un répertoire temporaire
* Extraction des informations exposées par le site de météo France et stocké dans un répertoire temporaire
* Tricher sur le contenu de la base
* Création d'une page HTML avec les données acquises

Dans ce TP, la notion de phase signifie que j'exécute une commande unix avec des options et la commande fait le travail demandé. Si je souhaite exécuter une nouvelle phase j'exécute une nouvelle commande unix. Ainsi les phases sont totalement disjointes, il est possible d'effacer, puis de télécharger un site, puis télécharger un deuxième puis procéder à l'analyse (des deux sites téléchargés) pour finir par générer une page html décrivant les contenus extraits des deux sites.

Remarque : il est inutile de converser avec l'utilisateur, il n'y aura personne devant le clavier. Il est inutile d'être excessivement bavard sur la console ou la sortie d'erreur, les flux sont neutralisés et vont vers des fichiers qui seront vérifiés en fin de script de test. Vous pouvez tout au plus émettre un message quand une grande fonction est exécutée ou une grosse erreur est rencontrée, mais en principe rien pour un fonctionnement normal. Une commande bavarde n'est pas bon signe.

## Commande

meteo option\*

Option :

-i Initialisation

-c Effacement des téléchargements précédents.

-e Effacement des analyses précédentes.

-m ville Téléchargement de la page web de météo France qui décrit la météo pour la ville donnée en paramètre. Une ville peut être décrite par son nom ou son code postal.

-s ville Cette option marche comme m mais remplace l'URL officielle pour atteindre un site local qui héberge les pages précédemment téléchargées. Cette option simule un téléchargement réel.

-a Analyse l'ensemble des fichiers préalablement téléchargés. Si vous avez rien téléchargé… il ne se passe rien, si vous n'avez utilisez l'option -m qu'une fois vous ne devrez analyser que ce fichier par contre si vous avez utilisez -m 4 fois je peux supposer que vous avez 4 fichiers temporaires qu'il faudra donc analyser. De plus, si je fais deux fois de suite l'option -a j'aurais une double extraction des données… C'est la vie.

-w ville jour Fabrication de la page web qui portera le nom <<ville>>-<<jour>>.html. Cette page extrait de la base des données météo l'ensemble des entrées qui concerne la ville au jour donné sous une forme tabulaire.

-t n triche

-h Affichage de cet aide

## Détails concernant l’initialisation

L'initialisation ne sera effectuée qu'une fois par usage de l'application. Elle pourrait effacer les associations ville code postal que vous avez déjà effectué. Il est évident que l'effacement de ce cache va déclencher de nombreuses requêtes qui pourraient vous rendre indélicat aux yeux du fournisseur avec des conséquences importantes.

## Détails concernant l'effacement des téléchargements précédents

L'effacement des téléchargements sera probablement (mais rien ne permet de la savoir ni de l'imposer) effectué avant une série de téléchargement. L'objectif est d'effacer toutes les informations temporaires **exclusivement** liées **aux téléchargements**. Il est évident que l'effacement de ce cache va déclencher de nouvelles requêtes qui pourraient vous rendre indélicat aux yeux du fournisseur avec des conséquences importantes.

## Détails concernant l'effacement des analyses précédentes

L'effacement des analyses précédentes permet de supprimer les données déjà extraites (erronées ou anciennes personne ne sait). On peut imaginer qu'elle est préalable à l'analyse mais pas obligatoire. L'objectif est d'effacer toutes les informations temporaires **exclusivement** liées **à l'analyse**.

## Détails concernant le téléchargement

Le téléchargement consiste à utiliser curl (et uniquement curl) pour chercher la page de la prévision météo pour une ville donnée (et une seule) et la stocker dans un répertoire temporaire.

Attention, si la page existe déjà elle n'est pas téléchargée à nouveau. Cette fonction 'ne pas télécharger si la page existe déjà' DOIT être codé au plus tôt. Légalement, si vous aspirez massivement du contenu sur la toile vous pouvez être assimilé à un pirate et l'entreprise pourrait porter plainte. Comme je viens de vous prévenir et de vous donner la protection à utiliser vous devenez responsable et coupable !

Pour imiter le fonctionnement normal d'un navigateur, vous mettrez dans la requête un USER-AGENT adapté (vous cherchez sur internet la chaine en vigueur pour le navigateur qui vous est assigné). En fin de sujet se trouve votre affectation du USER-AGENT. Cette pratique est double, imiter un navigateur officiel pour obtenir une page exploitable et masquer curl, mais si on vous détecte, l'aspect malicieux de votre action ne fera plus aucun doute.

Il est évident que si ces deux pratiques ne sont pas respectées, avant la première heure du TP, la correction de votre travail va échouer.

L'url à utiliser pour obtenir les prévisions de la météo pour une ville précise est la suivante :

http://www.meteofrance.com/previsions-meteo-france/<<ville>>/<<postal>>

Dans cette url on constate qu'il faut deux données différentes <<ville>> et <<postal>> pour décrire la même ville. Dans notre commande nous donnerons indifféremment le nom de la ville ou le code postal, il faut donc un service complémentaire pour trouver la deuxième information.

Grande chance, le gouvernement avec la poste nous offre ce service (offre n'est carrément pas certain si vous abusez de son usage …). Aussi, il faudra avant l'utilisation de cette url vérifier que vous n'avez pas déjà téléchargé l'association correspondante. Je ne renouvelle pas ma consigne précédente.

https://datanova.legroupe.laposte.fr/api/records/1.0/search/?dataset=laposte\_hexasmal&q=<<param>>&lang=fr&rows=10&facet=nom\_de\_la\_commune&facet=code\_postal

Il suffit de remplacer <<param>> par soit le code postal soit le nom de la ville pour obtenir un fichier json dont il faudra faire une petite analyse pour extraire la bonne entrée. Et oui, si vous cherchez par le nom il peut avoir plusieurs code postaux (comme à Mulhouse les 68100 et 68200) et la recherche par un nombre peut correspondre à un numéro insee ou au code postal (exemple 68100 désigne pour insee la ville de Fulleren et Colmar pour le code postal).

Pour le service météo, n'importe quel code postal est valide s'il pointe bien sur la ville souhaitée et inversement il faut sélectionner le bon code postal de la bonne ville si la recherche est sur un numérique. Une fois que l'association est connue il faut la stocker pour impérativement la réutiliser pour les recherches suivantes.

Dans tous les cas (notre commande et les requêtes sur le web), si la ville a un nom composé, il le sera non pas avec des espaces mais avec des tirets.

## Détails concernant la simulation du téléchargement

En remplaçant le début de l'URL (http:// ou https://) par http://localhost/ l'url pointe sur un site qui peut être hébergé localement (il le sera, peut-être, sur mon PC). Ainsi la requête va aboutir sur un PC accueillant et non pas sur le site officiel qui pourrait se sentir agressé et l'entreprise pourrait porter plainte. Pour vous il n'y a aucune différence sur la façon d'acquérir les données, seule la source réelle est masquée.

## Détails concernant l’analyse

Une fois la page (les pages) téléchargée il faut faire une extraction des données pertinentes. Première donnée, la date d'extraction… facile : la date du jour dans un format qui vous arrangera (n'oubliez pas réfléchir à la totalité du sujet avant de prendre une décision -w, -h et -t). Oups, n'oubliez pas qu'il peut se passer un temps non négligeable entre le ou les -m et un -a. Les autres données proviendront du fichier proprement dit, il faudra donc procéder à une extraction automatique. Que faut-il extraire :

Des données pour le jour courant J et les 7 prochains jours de J+1 à J+7. Donc 8 jeux de données composés de :

* L'état du ciel sous un beau format texte (pas besoin de son image)
* La température mini
* La température maxi

Ces trois données sont faciles à attraper, il faut trouver la bonne division (notepad++ et/ou firebug sont vos amis) après ce ne sont que des blocs li/ul etc. qu'il faut analyser.

Puis je souhaite y ajouter :

* La vitesse du vent (pas sa direction) vous comprendrez pourquoi.
* L'indice de confiance (pas toujours présent) absent il est à 6 sur 5

Là c'est plus complexe, car les données ne sont pas accessibles sous la forme d'une synthèse mais détaillées dans la journée. Il faudra donc faire la moyenne… des informations disponibles. Par contre, la page fournie l'information dans un bloc ul classé division-horaire, l'ordre correspond à l'ordre des jours (le premier ul correspond à J, le suivant à J+1 etc.). Comme l'affichage prévoit que la plage horaire déborde sur le lendemain… il serait sain de retirer le lendemain de votre moyenne. Bref c'est chaud.

## Détails concernant la page web d'une ville à un jour donné

Je rappelle que la génération ne peut utiliser que les données extraites lors de la phase analyse. Là c'est facile, vous consultez votre base de données et vous extrayez ce qui concerne la ville spécifiée (nom ou code postal) au jour donnée. Attention, le jour peut correspondre à la date de capture ou à une date de capture antérieure + le décalage correspondant. La date est JJ (le jour dans le mois et l'année courante) ou JJ:MM (le jour dans l'année courante) ou JJ:MM:AA ou JJ:MM:AAAA. On peut décemment espérer que l'utilisateur saisie des dates cohérentes.

La page doit rappeler le nom de la ville et la date concernée puis construire une vue tabulaire sur les données extraites lors de l'analyse. L'ordre de l'affichage est l'ordre dans la base… On pourrait imaginer une valeur spéciale pour le nom de la ville et la date qui serait le tiret (-) elle signifierait toutes les valeurs. Mais je laisse cette option aux enthousiastes qui ont codé le projet en 7h30, ne savent plus quoi faire pendant les dernières demi-heures et souhaitent un bonus

## Détails concernant la triche

Tricher consiste à prendre la totalité de la base et d'effectuer un décalage temporel de n jours (n étant le paramètre de la commande). Ainsi, une capture effectuée aujourd'hui (J0) passe la veille si la triche est de 1. Et une nouvelle analyse fabrique des données (identiques) pour aujourd'hui. Cette solution permet de simuler plusieurs extractions et vérifier la génération des pages html sans devoir attendre le passage au lendemain.

## Déroulements types

Ci-dessous une fraction représentative des nombreux exemples qui seront utilisés pour tester automatiquement votre script. Si vous voulez vous-même tester le fonctionnement il suffit d'utiliser ces lignes (dans un fichier par exemple).

La première fois pour les 20 années à venir.

**meteo –i**

Puis un cas simple juste sur une ville le 13 ème jour du mois de janvier 2020

**meteo –c**

**meteo –e**

**meteo –m mulhouse**

**meteo –a**

**meteo –w mulhouse 13**

ou un autre cas simple sur le code postal le 13 ème jour du mois de janvier 2020

**meteo –c**

**meteo –e**

**meteo –m 67000**

**meteo –a**

**meteo –w strasbourg 13:01**

ou la version compacte

**meteo –c -e –m Mulhouse –a –w Mulhouse 13:01:20**

Puis un cas plus riche que je montre en version compacte pour être plus dense dans le document

**meteo –c -e –m Mulhouse -m 68000 –a –w Colmar 13:01:2020 –w Mulhouse 13**

Il est bien sûr possible de remplacer -m par -s.

Si je fais le 1er janvier 2020 : **meteo –c**

Puis les jours suivant : **meteo -e -m Mulhouse -m 67000 -m 68000 –a**

Puis je fais un **meteo -w mulhouse 06**

J'obtiens une page, qui donne les prévisions météo concernant Mulhouse le 06 janvier 2020. Dans le tableau, je devrais donc voir 6 entrées (du 1er au 6).

Bien sûr, je peux simuler les jours suivant en déclenchant **meteo -t**

## User-Agent

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Noms** | **Prénoms** | **User agent** |  | **Noms** | **Prénoms** | **User agent** |
| CHICHAOUI | OMAR | **Chrome** |  | HERR | NICOLAS | **Opera** |
| DAGLI | BUSRA | **Internet explorer** |  | HEYSER | MANON | **Safari** |
| DIJOUX | QUENTIN | **Edge** |  | KEBOURI | HAMZA | **Chrome** |
| DURAND | KYLLIAN | **Firefox** |  | KURZ | FABIEN | **Edge** |
| GASSER | ARTHUR | **Opera** |  | LONG | LAURA | **Firefox** |
| HERBADJI | ROBIN | **Safari** |  | PRETESAC | BATISTE | **Chrome** |
| IHSANE | MOHAMED AMINE | **Chrome** |  | QUANG | BENJAMIN | **Firefox** |
| JIMENEZ | RAPHAËL | **Edge** |  | SCHUMMER | HUGO | **Chrome** |
| JUMEAUX | MAXIME | **Firefox** |  | BAHRALLIL | KHADIJA | **Chrome** |
| MANTEGAZZA | PIERRE | **Opera** |  | BRUGGER | MARIE-CAMILLE | **Internet explorer** |
| PAVUE | LOUIS | **Safari** |  | CARIO | JULIETTE | **Edge** |
| PERRET | LOUIS | **Chrome** |  | CAZAUBON | COLIN | **Firefox** |
| RHIMI | RAYEN | **Edge** |  | CHAIEB | ADAM | **Opera** |
| SAMEH | YASSINE | **Firefox** |  | CHAMPION | MELVIN | **Safari** |
| SUTTER | VALENTIN | **Chrome** |  | DE LA BROISE | JEAN-BAPTISTE | **Chrome** |
| WEBER | ARTHUR | **Firefox** |  | FENART | LUCAS | **Edge** |
| ABRAHAM | OSCAR | **Chrome** |  | GASSER | XAVIER | **Firefox** |
| BALANCHE | JEAN-NOËL | **Internet explorer** |  | KRAUTH | THOMAS | **Opera** |
| BEN SIDHOM | WISSEM | **Edge** |  | LOISON | GUILLEM | **Safari** |
| BENOMAR | YASSINE | **Firefox** |  | Marian Jude  Selvachandran | ANTONY | **Chrome** |
| CARON | LESLIE | **Opera** |  | MOULIS | LISA | **Edge** |
| CHAPOULIE | BENJAMIN | **Safari** |  | SCHLOESSLIN | LUCAS | **Firefox** |
| COSCOY | ULYSSE | **Chrome** |  | SCHULTZ | THOMAS | **Chrome** |
| EL HARBIL | SAMI | **Edge** |  | SOTTANI | GAËLLE | **Firefox** |
| FANFAN | JERÔME | **Firefox** |  |  |  |  |