Livrable 2: contexte

Dans le premier livrable, vous avez conçu une application web "front-end" qui permettait d'afficher certaines statistiques météo historiques. Dans ce livrable, vous aurez à implémenter des fonctionnalités supplémentaires:
1) Des informations météo historiques pour une journée donnée ("voyage dans le temps")
2) Les prévisions météo pour aujourd'hui et les prochains jours

Pour ces deux items, une nouveauté est que vous aurez à obtenir les données à partir d'APIs publics fournis par Environnement Canada. Or, il ne sera pas possible d'accéder à ces APIs uniquement au moyen de votre application "front-end" en raison de la *same-origin policy* (que nous verrons plus tard en cours). Dans ce livrable, vous aurez à implémenter une première version d'une application dorsale ("back-end") qui permettra d'accéder aux APIs externes et qui pourra fournir les données demandées à la couche "front-end".

Important: Dans le livrable 1, vous avez considéré qu'il est possible d'obtenir des données sur toutes les stations du pays - bien que dans les faits nous ne disposions de données que pour 16 stations. Dans ce livrable et le suivant, nous limiterons notre application qu'aux stations disposant d'un code d'aéroport. Vous devez modifier votre livrable 1 pour ne tenir compte que des stations disposant d'un code d'aéroport. Le livrable 3 adoptera aussi cette exigence. Ceci permettra de limiter les requêtes effectuées sur les API externes.

APIs externes

Les deux APIs externes suivantes d'Environnement et changement climatique Canada vous sont fournies pour ce livrable.

Important: vous devez *consommer* les données des APIs de façon responsable (c'est-à-dire en effectuant un nombre de requêtes modéré, en fonction des exigences demandées), comme c'est le cas pour toute API que vous utiliseriez en tant que développeur.

Données météo horaires pour une journée donnée

Cette API vous donne accès aux données météo heure par heure pour une journée donnée (passée).

URL: `https://climate.weather.gc.ca/climate_data/bulk_data_e.html?format=csv&stationID=${stationId}&Year=${year}Month=${month}&Day=${day}&timeframe=1&submit=%20Download+Data`

Paramètres:

- `${stationId}`: ID de la station météo (voir plus bas)
- `${year}, ${month}, ${day}` : année, mois et jour demandés

Retour: l'API vous retournera un fichier CSV avec des données pour les 24 heures de la journée. Si des données ne sont pas disponibles pour la journée demandée, le fichier contiendra tout de même les 24 heures, mais l'ensemble des données seront vides.

Exemple: `https://climate.weather.gc.ca/climate_data/bulk_data_e.html?format=csv&stationID=51157&Year=2020&Month=01&Day=07&timeframe=1&submit=%20Download+Data`

Pour plus d'informations, accédez au Google Drive suivant fourni par Environnement Canada: `https://drive.google.com/drive/folders/1WJCDEU34c60IfOnG4rv5EPZ4IhhW9vZH`. Notez que ce dossier contient des instructions permettant de télécharger en lot les données -- **ce n'est pas ce qui est demandé dans ce livrable**. Vous devez plutôt accéder aux données "à la demande".

Prévisions météo pour la journée actuelle et les jours à venir

Cette API vous donne accès aux données météo pour la journée actuelle ainsi qu'aux données pour les jours à venir.

URL: `https://meteo.gc.ca/rss/city/${code}_f.xml` (voir le champ `rss_url` dans le fichier `station_mapping.json`)

Paramètre: `${code}` (code de la station pour les flux RSS du site d'environnement Canada)

Retour: l'API vous retournera un fichier XML (RSS Atom) qui contient données météo, principalement au format texte intercalées entre les balises pertinentes. Vous devrez le *parser* et afficher les données textuelles présentes (plus de détails à la section T3.*)

Exemple: `https://meteo.gc.ca/rss/city/ab-50_f.xml`

Note: pour des raisons de compatibilité, nous recommandons l'utilisation de l'API fournie. Toutefois, si vous souhaitez utiliser des données météo provenant d'une autre source, c'est permis. Veuillez l'indiquer au chargé de laboratoire lors de la correction ainsi que dans votre rapport. Le cas échéant, assurez-vous que vos données contiennent un jeu similaire de données (prévisions pour la journée actuelle et les jours suivants) afin de répondre aux exigences du livrable.

Fichier `station_mapping.json`

Le fichier `station_mapping.json` contient un tableau associatif. Pour chaque code d'aéroport (clé -- ex., `YUL`), le fichier contient un ou plusieurs codes de stations (`station_ids`) pour récupérer les données historiques journalières, ainsi que l'URL du flux RSS/XML pour obtenir les prévisions météo pour cette station.

Ce fichier devra être chargé par votre "back-end".

Pourquoi peut-il y avoir plus d'un code de station pour un aéroport donné? Je ne suis pas exactement sûr de cette réponse, mais il semblerait que Environnement Canada ait parfois abandonné un ancien code utilisé pour une certaine période, puis utilisé un nouveau code. Ma théorie est qu'ils ont possiblement déployé une nouvelle station météo à cet emplacement, puis discontinué l'ancienne.

Tâches à réaliser

Les tâches à réaliser sont les suivantes:

T0: Filtrer les données de la liste de stations

T0.1 Vous devez modifier votre premier livrable pour que, les stations qui apparaissent soient uniquement celles associées à un aéroport.

T1: Application dorsale ("back-end")

T1.1 Vous aurez à implémenter une application dorsale qui acceptera des requêtes de type *AJAX* de l'usager pour satisfaire aux requis T2.* et T3.*. Votre "back-end" devra donc démarrer un serveur web qui acceptera des requêtes AJAX en provenance du code côté-client. Lorsque nécessaire, le "back-end" devra accéder aux données demandées à partir des APIs externes d'Environnement Canada et les servir en réponse au client.

Notes:

- Plus tard dans le cours, nous aurons un cours sur Node.JS. Vous êtes toutefois libres d'utiliser le langage et/ou cadriciel de votre choix pour le back-end.
- Pour ce livrable, il n'y a pas d'exigences spécifiques au niveau du back-end autres que celles visant à fournir les fonctionnalités demandées. Toutefois, au livrable 3, votre back-end devra exposer les fonctionnalités sous forme d'APIs REST. Cette information peut donc possiblement aider à orienter votre conception, bien que cet aspect ne soit pas évalué au livrable 2.
- Il n'est pas nécessaire d'intégrer l'accès aux fichiers `*.csv.js` du livrable 1 au "back-end"; cela sera toutefois requis du livrable 3 (vous pouvez donc prendre de l'avance si vous le désirez). Il est donc acceptable pour ce livrable d'accéder à ces données directement à partir du front-end comme au livrable 1 (cet aspect ne fait pas partie de l'évaluation).

T1.2 Il devra être possible d'accéder à l'application web au moyen de l'application dorsale, c'est-à-dire via une URL servie par votre serveur web (par exemple, `http://localhost:8080/`). Cela sera nécessaire également en raison de la *same-origin policy*. Votre serveur web devra servir l'ensemble des ressources demandées (ex., HTML, CSS, JS, images, etc.).

T2: Informations météo pour une journée donnée passée

T2.1 Une nouvelle section "Informations journalières historiques" doit être ajoutée (nouvel onglet) au contenu de la page principale (il est permis d'abréger le titre).

T2.2 Puisque la section "plage de dates" n'est pas requise pour cette nouvelle section du site, vous devez restructurer ("réfactorer") légèrement votre ancien site afin que la section plage de dates fasse partie plutôt des sections "Données" et "Statistiques" du livrable précédent uniquement.

Conseil: vous pourriez mettre la section "plage de dates" dans un `<DIV>` que vous ferez afficher comme premier élément des sections "Données" et "Statistiques" (ou toute autre façon de procéder à votre choix).

Note: faites attention à ne pas "briser" des fonctionnalités du dernier livrable.

T2.3 Vous devez offrir la possibilité à l'utilisateur de choisir une journée spécifique (année-mois-date), plutôt qu'une plage de dates. Cela peut se faire notamment au moyen de noeuds `<SELECT>` ou d'un `<input type='date'>` . Lorsque la date change et que tous les champs contiennent une valeur valide (année, mois et jour), vous devez rafraîchir les informations journalières (**T2.4**).

T2.4 Les informations journalières doivent être chargées via une requête AJAX au back-end qui va consommer l'API appropriée. Si les informations sont disponibles, elles doivent être affichées sous forme de tableau contenant les colonnes suivantes. Les rangées contiennent les heures de la journée (24h). Vous pouvez abréger les titres des colonnes. Omettez les champs qui ne sont pas disponibles pour la date choisie.

- Température réelle
- Température ressentie
- Météo
- Humidité
- Direction du vent
- Vitesse du vent (km/h)
- Pression atmosphérique

Notes:

- Si la station demandée a plusieurs codes de stations (ex., `YUL`: `5415,51157`), il se peut que vous ayez besoin de tenter d'obtenir l'information au moyen des différents codes fournis (au niveau du back-end). Exemple: le client demande la journée du 1er mars 2020 pour la station `YUL` via un appel AJAX -- le "back-end" tente d'obtenir les données pour le code de station `5415`. Les résultats retournés en CSV ne contiennent pas de données pour la journée demandée; le "back-end" essaie alors `51157`. Les résultats sont disponibles -- ils sont alors retournés en réponse à la requête AJAX.
- Si les résultats ne sont pas disponibles pour la station demandée (après avoir essayé tous les codes de stations), une erreur devrait être retournée au client, qui devrait afficher un message à l'effet que les données ne sont pas disponibles et invitant le client à choisir une autre journée, plutôt que le tableau de résultats.

T2.5 La vue choisie doit demeurer active lorsqu'on sélectionne une autre station (il s'agit de l'extension d'une exigence du précédent livrable). Dans cette section, vous devez donc rafraîchir les données à la date choisie pour la nouvelle station. (Cette exigence s'applique à toutes les vues.)

T3: Prévisions météo (pour aujourd'hui et les prochains jours)

T3.1 Une nouvelle section "Prévisions météo" doit être ajoutée (nouvel onglet) au contenu de la page principale. Pour cette section, aucun sélecteur de date ne doit être proposé.

T3.2 Les prévisions météo doivent être obtenues via une requête AJAX au "back-end" qui va consommer l'API appropriée. Si les informations sont disponibles, les éléments suivants doivent être affichés.

- Nom de la station (incluez un lien vers la page web d'Environnement Canada des prévisions pour cette station)
- Date de la dernière mise à jour
- Veilles et avertissements météo
- Conditions actuelles
- Prévisions pour les prochains jours (sommaires (`<title>`) et détaillés (`<summary>`) dans le flux RSS)

T3.3 Il n'y a pas d'exigences spécifiques au sujet de la présentation, mais vous devez faire en sorte que le visuel soit bien présenté au moyen de noeuds logiquement appropriés (ex., titres, listes, etc.) et au moyen de styles CSS.

Utilisation de cadriciels

Bien que les livrables puissent être totalement accomplis avec les technologies web standard (HTML/CSS/VanillaJS), nous autorisons et encourageons l'utilisation des cadriciels web si vous le désirez. Dans ce cas, il sera de votre responsabilité de vous documenter sur le cadriciel choisi et de l'utiliser selon les bonnes pratiques, et de résoudre les problèmes éventuels rencontrés. Le chargé de laboratoire tentera de vous aider au meilleur de ses connaissances, mais il est possible qu'il ne connaisse pas précisément tous les cadriciels web.

Rapport

Un court rapport est demandé. Vous devez répondre aux questions suivantes (maximum 5 pages). Une pénalité sera appliquée pour les fautes de français (voir le barème) et une mise en page incorrecte ou un manque de rigueur dans la présentation.

R1: Décrivez l'architecture logicielle utilisée pour la partie "front-end" de ce livrable et comment elle a évolué par rapport au livrable 1, ainsi que pour la partie "back-end". Décrivez brièvement l'organisation et le rôle des différents éléments (classes, fonctions) de votre code JavaScript. Si vous avez utilisé des cadriciels, mentionnez-les et indiquez de quelle façon ils sont utilisés.

R2: Décrivez de quelle façon vous vous y êtes pris pour traiter ("parser") et extraire les différentes données pertinentes des APIs externes (formats CSV et XML/RSS). Si vous avez utilisé des librairies, mentionnez-les et décrivez de quelle façon elles sont utilisées. Décrivez également comment vous gérez (du point de vue interface et de votre code) les données demandées par l'utilisateur qui sont manquantes dans les sources externes (par exemple, les informations météo historiques qui ne sont pas disponibles pour une station et journée donnée).

R3: Comment gérez-vous les conditions d'erreurs des différentes requêtes (du "front-end" au "back-end" ainsi que du "back-end" aux APIs externes), tant du point de vue de votre code que dans l'interface de votre site?

R4: De quelques façon avez-vous subdivisé les tâches en équipe pour ce livrable? Décrivez le rôle et les tâches assignées à chacun des membres.

R5: Notez qu'une brève introduction et conclusion est également demandée.

Code source

Bien que le code ne soit pas évalué, nous nous réservons quand même le droit de le vérifier, et **le code complet devra être soumis avec la remise**. Toutefois, nous nous attendons à ce que vous mettiez des bonnes pratiques de conception et d'implémentation en oeuvre. L'évaluation de cet aspect sera faite de manière indirecte par l'évaluation de vos réponses aux questions du rapport qui traitent de ces aspects. De plus, le livrable 3 vous demandera d'ajouter encore une fois un nouvel ensemble de fonctionnalités et de *refactorer* votre code, ce qui sera plus facile à réaliser avec une bonne conception/implémentation.

Utilisation d'un entrepôt Git

Pour le projet de session, vous devez utiliser un entrepôt Git (GitHub ou GitLab disponible à l'ÉTS) pour votre projet. Tous les membres de l'équipe devraient "pousser" du code pour étayer la contribution de tous et chacun. Advenant le cas où nous recevions une plainte concernant un membre de l'équipe ne fournissant pas sa juste part de travail, nous pourrions vous demander de nous donner accès à votre entrepôt ou de nous fournir une copie des "logs" (bien entendu, nous espérons fortement que cela ne se sera pas nécessaire!).

Barème de correction

Tâche	Description	Points max.
T0	**Filtrer les données de la liste de stations**	**Total 5**
T1	**Application dorsale ("back-end")**	**Total 10**

T1.1	Backend qui accepte et fournit des réponses aux requêtes	7
T1.2	Accès à l'application via l'URL servie par le serveur web	3
T2	**Informations météo pour une journée donnée passée**	**Total 20**
T2.1	Nouvelle section informations journalières historiques	2
T2.2	La sélection de la plage de dates fait partie uniquement des sections "données" et "statistiques"	3
T2.3	Choix d'une journée spécifique pour l'historique et rafraîchissement	2
T2.4	Obtention et affichage des informations journalières via le back-end (+ gestion des erreurs si les infos sont non-disponibles)	10
T2.5	La vue actuelle demeure active lors du choix d'une autre station et les données sont rafraîchies	3
T3	**Prévisions météo (pour aujourd'hui et les prochains jours)**	**Total 15**
T3.1	Nouvelle section pour les prévisions météo (aucun sélecteur de dates)	2
T3.2	Obtention et affichage des prévisions météo via le back-end	10
T3.3	L'affichage est adéquat et bien structuré pour l'affichage des prévisions	3
Rapport		**Total 30**
R1	(voir la description)	6
R2	(voir la description)	6
R3	(voir la description)	6
R4	(voir la description)	6
R5	Introduction et conclusion	6
Pénalité	Code JS mal structuré (-10)	
Pénalité	Code JS dans fichier HTML (-5)	
	Pratiques non-recommandées (ex., insertion de code brut HTML dans le	

Pénalité	DOM) (-5)	
Pénalité	Mise en page HTML/CSS mal structurée (ex., utilisation de tableaux) (-5)	
Pénalité	Des fonctionnalités du livrable précédent ne fonctionnent plus (-10)	
Pénalité	Fautes de français dans le rapport (-0.5% par faute, jusqu'à -10%)	
Total		**Max. 80**

Politique sur le plagiat

Tel qu'énoncé dans le plan de cours:

Les clauses du « Règlement sur les infractions de nature académique de l'ÉTS » s'appliquent dans ce cours ainsi que dans tous les cours du département. Les étudiants doivent consulter le Règlement sur les infractions de nature académique (https://www.etsmtl.ca/A-propos/Direction/Politiques-reglements/Infractions_nature_academique.pdf) pour identifier les actes considérés comme étant des infractions de nature académique ainsi que prendre connaissance des sanctions prévues à cet effet. À l'ÉTS, le respect de la propriété intellectuelle est une valeur essentielle et les étudiants sont invités à consulter la page Citer, pas plagier ! (https://www.etsmtl.ca/Etudiants-actuels/Baccalaureat/Citer-pas-plagier).

Nous attirons votre attention sur le fait qu'il est **interdit de réutiliser, en tout ou en partie**, le travail d'une autre équipe ou réalisé à une session antérieure. Nous conservons l'ensemble des travaux des sessions antérieures et avons accès au code source soumis, et nous nous réservons le droit d'utiliser des outils de comparaison du code source et de détection de plagiat. Tout plagiat détecté sera sanctionné selon les politiques en vigueur à l'ÉTS.

Note: si vous reprenez le cours GTI525, vous devez également refaire les laboratoires.

Procédure et date de remise

Les instructions de remise sont sur Moodle (ENA).