



IFT3225 (Techno. Web) — Projet 3
MySQL / REST / D3 / DataTables
dans une application Simple Page
(Single Page Application)

Contact :
Philippe Langlais
RALI/DIRO
Université de Montréal

+1 514 343 61 11 ext: 47494
felipe@iro.umontreal.ca
<http://www.iro.umontreal.ca/~felipe/>

■ dernière compilation : 5 avril 2022 (09:05)

Conseil

Ne commencez pas à la dernière minute ! Vous pouvez travailler en groupe de 2 personnes, profitez-en pour vous diviser adéquatement les tâches.

Contexte

Dans ce projet, vous allez réunir un ensemble de techniques vues en cours afin d'interagir avec une base de données contenant des informations sur les [micro-brasseries au Québec](#). Vous allez donc développer du code côté serveur (pour la gestion de MySQL et pour le code d'un service REST) et côté client (pour la gestion d'une application simple page). Vous devez exposer deux choses : une URL indiquant une page HTML donnant accès à l'application.

À priori, votre application Web devrait contenir une unique page HTML. Vous pouvez utiliser pour cela la librairie [Sammy.js](#) vue en cours, plus maintenue mais disponible sur Github.

À faire

1. Création de la base **projet3** abritée au DIRO. Cette base contiendra au moins les deux tables suivantes :
 - Considérez le fichier [microbrasseries.csv](#) qui contient une liste de micro-brasseries québécoises. Transformez-le en une table MySQL **brasseries**. Vous pouvez pour cela créer un fichier `.sql` ou utiliser du code PHP pour lire ce fichier et créer la base. Dans le premier cas, remettez le fichier **brasseries.sql** que vous avez créée.
 - Vous utiliserez également une table d'utilisateurs (avec leur mot de passe) ayant droit à consulter votre application : **account**. Cette table contiendra au moins les utilisateurs suivants : **ift3225** (qui peut consulter mais pas modifier les bases) et **admin** (qui peut en plus modifier les bases) avec leur mot de passe respectif : **sésame** et **ouvre-toi**. Vous pouvez (c'est même une recommandation) utiliser le code du [tutoriel](#) vu en cours pour gérer le login d'utilisateurs autorisés. Remettre le fichier **account.sql** permettant de créer cette base avec ces deux utilisateurs et leur mot de passe.
2. Création d'un service REST pour manipuler la table **brasseries**. Vous devriez offrir au moins les services suivants :
 - **GET <micro>/names** retourne une structure json avec tous les noms
 - **GET <micro>/names/<prefix>** retourne une structure json avec tous les noms qui commencent par le préfixe **<prefix>**

- GET `<micro>/permis/<permis>` retourne sous forme json toutes les informations correspondant à l’item dont le numéro de permis est celui spécifié (`<permis>`)
- POST `<micro>/add` qui permet d’ajouter une entrée dans la table `brasseries` dont les valeurs sont passée dans le corps de la requête. Vous prendrez soin d’expliquer la syntaxe du corps de la requête.

où `<micro>` est la partie du path dans votre espace Web du DIRO correspondant au service REST. Vous prendrez soin dans le rapport de lister les patrons d’URIs de votre service REST et de spécifier le cas échéant la syntaxe du corps de la requête ainsi que les éventuels paramètres de chaque route.

Le routage des URIs vers des ressources PHP a été abordé en cours. Je vous recommande l’usage de la librairie minimaliste [Kunststube-Router](#) qui est facile à utiliser. Nous avons vu en cours comment le faire avec la librairie encore plus minimaliste [phprouter](#) que vous pouvez également utiliser. J’ai ajouté dans les transparents [rest.pdf](#) des transparents montrant comment effectuer le routage avec `Kunststube-Router`.

Si vous ne parvenez pas à faire de routage, alors indiquez les URLs (ressources PHP) à solliciter pour un service REST équivalent (à la façon du premier [tutoriel](#) vu dans le cours sur REST). Prenez note que le service REST offert sera testé par nos soins à l’aide de requêtes `curl` directement en ligne de commande. Il est donc important que votre description suffisamment claire pour le faire.

3. Application Web

Voici une description fonctionnelle de la logique de l’application découpée en routes. Idéalement, votre application devrait impliquer une seule URL, désignée ici par `<path>`. Si vous ne parvenez pas à faire tout cela en une seule page, vous devez identifier les URLs correspondant dans votre application à chacune de ces routes :

- `<path>#/login` qui offre un formulaire permettant à un utilisateur de s’identifier. Vous pouvez gérer une session permettant à l’utilisateur identifié d’utiliser les différentes routes sans avoir à se reconnecter.
- `<path>#/table` qui offre une vue tabulaire de la table `brasseries`. Je vous recommande pour cela l’usage de la librairie [DataTables](#) qui est d’un usage relativement facile. Les données affichées dans la table devront venir d’une requête Ajax à la table `brasseries`.

Voici une vue possible où lorsqu'on clique sur le signe +, de l'information complémentaire est affichée, ce qui évite à l'utilisateur de scroller horizontalement :

Show

10

entries

Search:

Nom	Adresse	Ville	CP	#Permis	Courriel
<div> <div></div> <div>À L'ABORDAGE MICROBRASSERIE</div> </div>	10, RUE PRINCIPALE SUD	SUTTON	J0E 2K0	AB076	info@brasserialabordage.com
<div> <div>Appellation</div> <div>À L'ABORDAGE MICROBRASSERIE INC.</div> </div> <div> <div>Création</div> <div>2015</div> </div> <div> <div>Web</div> <div>http://www.brasserialabordage.com/</div> </div> <div> <div>Courriel</div> <div>info@brasserialabordage.com</div> </div> <div> <div>Latitude/Longitude</div> <div>45.11263/-72.61737</div> </div>					
<div> <div></div> <div>À L'ABRI DE LA TEMPÊTE</div> </div>	286, CHEMIN COULOMBE	LES ÎLES-DE-LA-MADELEINE	G4T 3V5	BR041	info@alabridelatempete.com
<div> <div></div> <div>À LA FÛT</div> </div>	670, RUE NOTRE-DAME	SAINT-TITE	G0X 3H0	BR058	info@alafut.qc.ca
<div> <div></div> <div>ALBION BRASSERIE ARTISANALE</div> </div>	408, BOULEVARD MANSEAU	JOLIETTE	J6E 3E1	AB049	brasserialbion@gmail.com
<div> <div></div> <div>ARCHIBALD MICROBRASSERIE</div> </div>	1021, BOULEVARD DU L'AC, LOCAL 200	LAC-BEAUPORT	G0A 2C0	BR046	info@archibaldmicrobrasserie.ca
<div> <div></div> <div>AUBERGE SUTTON BROUËRIE</div> </div>	27, RUE PRINCIPALE SUD	SUTTON	J0E 2K0	BR118	info@aubergesuttonbrouerie.com
<div> <div></div> <div>AVANT-GARDE BRASSEURS</div> </div>	2350, RUE DICKSON, LOCAL 1400	MONTREAL	H1N 3T1		renaud@brasseursavantgarde.com
<div> <div></div> <div>BARBE BROUE PUB</div> </div>	69, RUE SAINTE-ANNE	VILLE-MARIE	J9V 2B6	AB078	barbe.broue.pub@gmail.com
<div> <div></div> <div>BEDONDAINE ET BEDONS RONDS</div> </div>	255, RUE OSTIGUY	CHAMBLY	J3L 2Z7	AB028	info@bedondaine.com
<div> <div></div> <div>BELGH BRASSE</div> </div>	8, RUE DE LA BRASSERIE	AMOS	J9T 3A2	BR043	info@belghbrasse.com
Nom	Adresse	Ville	CP	#Permis	Courriel

Showing 1 to 10 of 176 entries

Previous

1

2

3

4

5

...

18

Next

- `<path>#/add` permet à un utilisateur d'ajouter une micro-brasserie dans la table `brasseries`. Notez que pour ce faire, vous devez vous être identifié comme administrateur et non comme simple utilisateur. Vous pouvez utiliser des librairies comme [Editor](#) pour faire cela, mais cette librairie est payante (vous avez accès à 15 jours d'essai gratuit). Le plus simple est probablement d'offrir un formulaire de saisie permettant de spécifier les champs d'une nouvelle entrée.
- `<path>#/delete` permet à un utilisateur de supprimer une micro-brasserie dans la table `brasseries`. Notez que pour ce faire, vous devez vous être identifié comme administrateur et non comme simple utilisateur.

- `<path>#/plot` offre une vue graphique (latitude / longitude) des micro-brasseries. Vous utiliserez pour cela la librairie [D3](#) vue en cours.¹ La vue offerte peut par exemple ressembler à [celle-ci](#) où le nom de chaque micro-brasserie peut apparaître (au lieu de marques de voitures dans l'exemple). Il faut être un utilisateur reconnu de la base (admin ou pas) pour accéder à cette route. Vous tenterez de rendre votre vue la plus informative possible. Vous pourriez par exemple afficher un pop-up avec certaines informations lorsque la souris passe sur un point particulier de votre graphe.
- `<path>#/doc` offre une page HTML accessible à tous décrivant le service REST disponible ainsi que les routes de l'application client (où les différentes ressources HTML si vous ne parvenez pas à faire une application en une page).

À remettre

- Un rapport au format pdf `readme.pdf` d'au plus 4 pages qui indique :
 - a) l'identité et le rôle des personnes du groupe dans le projet, b) l'url de la page HTML abritée au DIRO donnant accès à l'application (vous prendrez soin tout comme dans les projets précédents de ne pas exposer cette url afin de minimiser les risques de plagiat), et c) la liste de toutes les ressources externes que vous avez utilisées (ex : JQuery + Bootstrap + Sammy.js + D3 + DataTables).
- Une archive (zip, tar, etc.) de toutes les ressources de votre projet.

La remise est à faire sur Studium sous le libellé **projet3**. Vous devez remettre votre code et votre rapport (format pdf) dans une archive (gzip, tar, tar.gz) dont le nom est préfixé de **projet3-noms**, où **noms** est à remplacer par l'identité des personnes (**prénom_nom**) impliquées dans le projet. Donc si j'avais à remettre seul ma solution au projet3, je le ferais sous le nom **projet3-philippe_langlais.tar.gz**. Le projet est à remettre en groupe d'au plus deux personnes **au plus tard vendredi 22 avril à 23h59**.

1. Cours de la semaine 10 reporté en raison de la grève.

Notation

Ce projet compte pour 25 points répartis selon différents critères comme :

- la bonne gestion de la base de données
- la qualité et l'organisation du code
- l'utilisation de librairies
- l'ergonomie générale de l'application
- gestion du routage
- tests
- qualité du rapport

Questions

Posez vos questions via le slack du cours sur le canal **demo**.