COLLÈGE DE MAISONNEUVE

582-640 - Programmation et veille technologique

Épreuve finale volet #2 /15

Chaque question correspond à une étape de projet. Chaque commit devra contenir un message « -m » indiquant le numéro de l'étape.
Exemple:
git commit -m "Étape #1 lecture du fichier"
git commit -m "Étape #1 terminée"

Vous devriez avoir un minimum de 10 ***commit***

L'épreuve consiste à développer une petite application Web utilisant le langage Node.js avec les modules Express, MongoDB et EJS.
Il s'agira de définir une route pour chacune des étapes du développement.

Une structure de base est founie. Il s'agira essentiellement d'ajouter les éléments manquant pour faire fonctionner l'application.

Étape #1: Lire un fichier texte représentant des données sur les provinces canadiennes. Ce fichier contient un tableau d'objet JSON en format texte. Il s'agit d'afficher le contenu du fichier.


```
[
        {
         "code": "NF" ,
         "nom" : "Terre-Neuve",
         "capital" : "St-john"
        },
        {
          "code": "IPE",
          "nom" : "Ile du Prince-Édouard ",
          "capital" : "Charlottetown"
        },
        {
          "code": "NS",
          "nom" : "Nouvelle Écosse",
          "capital" : "Halifax"
        },
        {
          "code": "NB",
          "nom" : "Nouveau-Brunswick",
          "capital" : "Fredericton"
        },
        {
        "code": "QC",
        "nom" : "Québec",
        "capital" : "Québec"
        },
        {
        "code": "ON",
        "nom" : "Ontario",
        "capital" : "Toronto"
        },
        {
        "code": "MA",
        "nom" : "Manitoba",
        "capital" : "Winipeg"
        },
        {
        "code": "SK",
        "nom" : "Saskatshewan",
        "capital" : "Regina"
        },
        {
        "code": "AL",
        "nom" : "Alberta",
        "capital" : "Edmonton"
        },
        {
          "code": "BC",
          "nom" : "Colombie Britannique",
          "capital" : "Victoria"
        },
        {
          "code": "NU",
          "nom" : "Nunavut",
          "capital" : "Igaluit"
        },
        {
          "code": "YT",
          "nom" : "Yukon",
          "capital" : "Whitehorse"
        },
        {
          "code": "NT",
          "nom" : "Territoire du Nord-Ouest",
          "capital" : "Yellowknife"
        }
]
```

Étape #2: En utilisant le gabarit ***index.ejs***, faire afficher les données lues dans un tableau HTML

Épreuve finale volet II - Programmation et veille technologique

Node.js, Express, MongoDB, EJS

1-Fichier | 2-Provinces | 3-Collection | 4-Ajouter un document | 5-Détruire | 6-Ajouter plusieurs

NF	Terre-Neuve	St-john	
IPE	Ile du Prince-Édouard	Charlottetown	
NS	Nouvelle Écosse	Halifax	
NB	Nouveau-Brunswick	Fredericton	
QC	Québec	Québec	
ON	Ontario	Toronto	
MA	Manitoba	Winipeg	
SK	Saskatshewan	Regina	
AL	Alberta	Edmonton	
BC	Colombie Britannique	Victoria	
NU	Nunavut	Igaluit	
YT	Yukon	Whitehorse	
NT	Territoire du Nord-Ouest	Yellowknife	

Epreuve finale volet II - Programmation et veille technologique - TIM - Collège de Maisonneuve

Étape #3: En utilisant MongoDB, ajouter la collection «provinces» à la base de données «carnet-adresse».

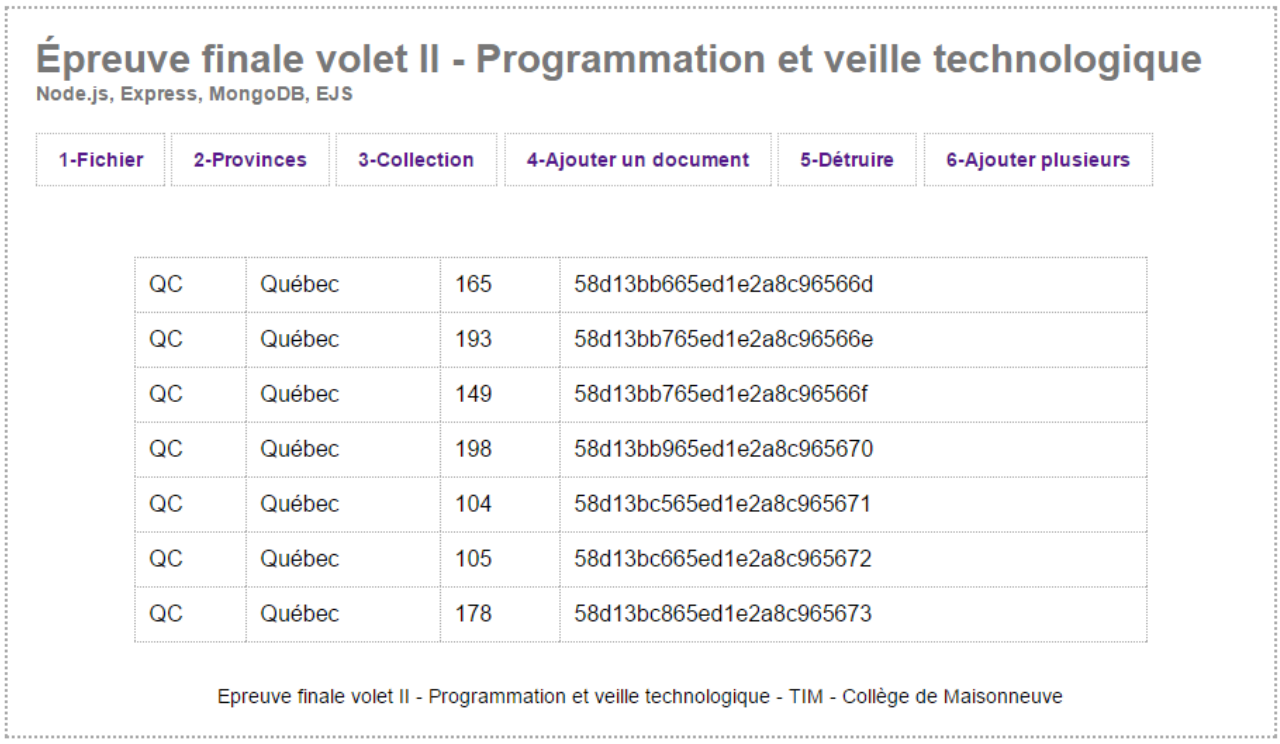
Voici un exemple d'un document de la collection «provinces» :

{code: "QC", nom: "Québec", capital : "Québec"}

Pour cette étape il s'agira d'afficher la collection «provinces» en utilisant le gabarit ***index.ejs***

les champs suivants devront être affichés:

{code: "QC", nom: "Québec", capital : "Québec", _id: 7753590546787083 80}



Épreuve finale volet II - Programmation et veille technologique

Node.js, Express, MongoDB, EJS

1-Fichier | 2-Provinces | 3-Collection | 4-Ajouter un document | 5-Détruire | 6-Ajouter plusieurs

QC	Québec	165	58d13bb665ed1e2a8c96566d
QC	Québec	193	58d13bb765ed1e2a8c96566e
QC	Québec	149	58d13bb765ed1e2a8c96566f
QC	Québec	198	58d13bb965ed1e2a8c965670
QC	Québec	104	58d13bc565ed1e2a8c965671
QC	Québec	105	58d13bc665ed1e2a8c965672
QC	Québec	178	58d13bc865ed1e2a8c965673

Epreuve finale volet II - Programmation et veille technologique - TIM - Collège de Maisonneuve

Étape #4

Créer la route «ajouter» pour ajouter un nouveau document à la collection province.

le document ajouté aura la forme suivante:

{

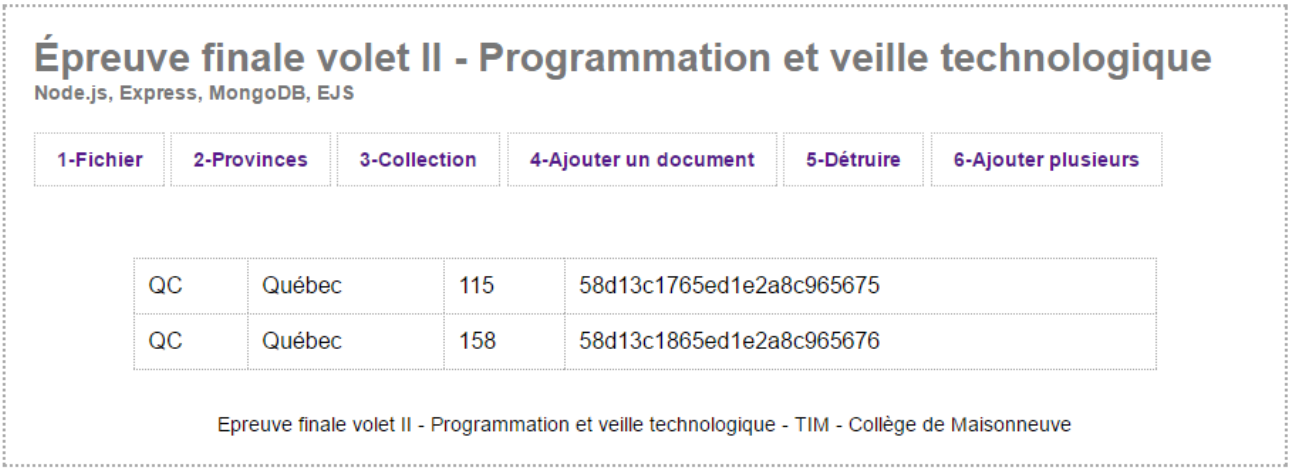
code : "QC",

nom : "Québec",

capital: // un nombre alaétoire situé entre 100 et 200

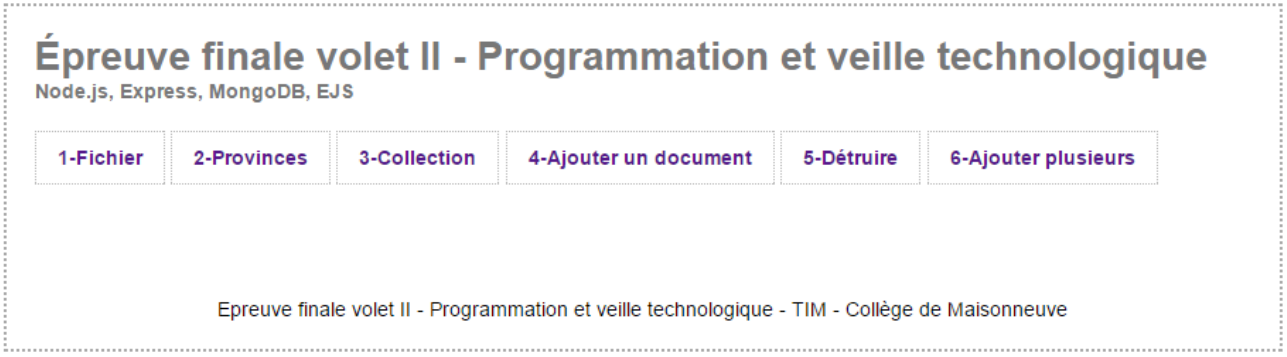
}

une fois qu'un document est ajouté la collection «provinces» est réaffichée. Le nouveau document apparaît à la fin du tableau



Étape #5

Créer la route «**detruire**» qui permettra de détruire le contenu complet de la collection «**provinces**»



Étape #6

Créer la route « ajouter-plusieurs » qui permettra d'ajouter le contenu complet du fichier « collection-provinces » dans la collection « provinces »

Épreuve finale volet II - Programmation et veille technologique

Node.js, Express, MongoDB, EJS

1-Fichier | 2-Provinces | 3-Collection | 4-Ajouter un document | 5-Détruire | 6-Ajouter plusieurs

NF	Terre-Neuve	St-john	58d13cbd65ed1e2a8c965677
IPE	Ile du Prince-Édouard	Charlottetown	58d13cbd65ed1e2a8c965678
NS	Nouvelle Écosse	Halifax	58d13cbd65ed1e2a8c965679
NB	Nouveau-Brunswick	Fredericton	58d13cbd65ed1e2a8c96567a
QC	Québec	Québec	58d13cbd65ed1e2a8c96567b
ON	Ontario	Toronto	58d13cbd65ed1e2a8c96567c
MA	Manitoba	Winipeg	58d13cbd65ed1e2a8c96567d
SK	Saskatshewan	Regina	58d13cbd65ed1e2a8c96567e
AL	Alberta	Edmonton	58d13cbd65ed1e2a8c96567f
BC	Colombie Britannique	Victoria	58d13cbd65ed1e2a8c965680
NU	Nunavut	Igaluit	58d13cbd65ed1e2a8c965681
YT	Yukon	Whitehorse	58d13cbd65ed1e2a8c965682
NT	Territoire du Nord-Ouest	Yellowknife	58d13cbd65ed1e2a8c965683

Epreuve finale volet II - Programmation et veille technologique - TIM - Collège de Maisonneuve

Barème de correction:

Utilisation de git avec un commit minimum pour chaque question et un minimum de 10 commit au total pour la réalisation de l'épreuve. (2 points)

- Dix commit bien commenté (1 point)
- Remise sur GitHub (1 point)

Étape #1 - Lecture du fichier: (2 points)

- Codification : (1 point)
- Fonctionnement: (1 point)

Étape #2 - Afficher le fichier en tableau: (2 points)

- Codification : (1 point)
- Fonctionnement: (1 point)

Étape #3 - Afficher la collection «provinces» : (2 points)

- Codification : (1 point)
- Fonctionnement: (1 point)

Étape #4 - Ajouter un document dans la collection « provinces » : (2 points)

- Codification : (1 point)
- Fonctionnement: (1 point)

Étape #5 - Détruire le contenu de la collection « province »: (2 points)

- Codification : (1 point)
- Fonctionnement: (1 point)

Étape #6 - Ajouter l'ensemble des provinces dans la collection « provinces » : (3 points)

- Codification : (1.5 point)
- Fonctionnement: (1.5 point)