

TP "devise-api-routes"

Consignes du TP

À faire, **si ce n'est pas déjà fait** (au sein d'un TD ou TP antérieur)

- Créer dans C:\tp-js ou ailleurs un nouveau répertoire myNodeServer
- Se placer dans le répertoire myNodeServer (dans une console "CMD")
- Créer un début de package.json via npm init
- ajouter "type" : "module" , dans **package.json**
- Installer le "package" Express via npm install -s express
- Modifier ou créer le fichier server.js avec le **code minimal** suivant :

```
//var express = require('express');
import express from 'express';
//var deviseApiRoutes = require('./devise-api-routes');
import deviseApiRoutes from './devise-api-routes.js';
var app = express();
//...
app.use(deviseApiRoutes.apiRouter); //delegate REST API routes to apiRouter(s)

app.listen(8282 , function () {
  console.log("http://localhost:8282");
});
```

- Ajouter un fichier devise-api-routes.js avec le code initial suivant :

```
//var express = require('express');
import express from 'express';
const apiRouter = express.Router();

var allDevises = [];
allDevises.push({ code : 'EUR' , nom : 'Euro' , change : 1.0 });
allDevises.push({ code : 'USD' , nom : 'Dollar' , change : 1.1 });

//exemple URL: http://localhost:8282/devise-api/public/devise
//(returning all devises)
apiRouter.route('/devise-api/public/devise')
.get( function(req , res , next ) {
  res.send(allDevises);
});

//exports.apiRouter = apiRouter;
export default { apiRouter };
```

- Lancer ensuite server.js avec node ou nodemon
- Tester via l'URL <http://localhost:8282/devise-api/public/devise>

Chercher en mode GET

- Au sein de devise-api-routes.js , coder une nouvelle route en mode GET permettant de récupérer une seule devise en fonction de son code

Exemple d'URL : <http://localhost:8282/devise-api/public/devise/EUR>

Exemple de fin de route : `/devise-api/public/devise/:code`

Récupération du code en fin d'URL via `req.params.code`

- Effectuer un test après avoir, si besoin, redémarré le serveur
 - Ajuster, si besoin, le code de `server.js` de façon à rediriger par défaut sur `html/index.html` (fichier statique à gérer via Express)
 - Ajouter des liens hypertextes de type ` ... ` dans `index.html` de façon à déclencher les 2 appels précédents en mode GET (toutes devises et devise EUR).
 - Au sein de `devise-api-routes.js`, améliorer le code de la route `/devise-api/public/devise` (toutes devises) en :
 - Tenant compte d'un paramètre facultatif `?changeMini` en fin d'URL
 - Utilisant la syntaxe `req.query.changeMini` pour récupérer la valeur du paramètre HTTP
- Exemple d'URL : <http://localhost:8282/devise-api/public/devise?changeMini=1.05>
- Tester via différentes fins d'URL.

Ajout en mode POST (et test avec Postman)

- Ajouter, **si nécessaire**, le code suivant dans le haut du fichier `server.js` pour que Express puisse convertir les données JSON envoyées en mode POST (`req.body`) vers JavaScript :

```
var jsonParser = express.json({ extended: true });
app.use(jsonParser);
```

- Ajouter cette nouvelle route dans `devise-api-routes.js` :

```
// ./devise-api/private/role-admin/devise en mode POST
// avec { "code": "mxy", "nom": "monnaieXy", "change": 123 }
// dans req.body
apiRouter.route('/devise-api/private/role-admin/devise')
.post( function(req, res, next) {
    var nouvelleDevise = req.body;
    console.log("POST,nouvelleDevise="
        +JSON.stringify(nouvelleDevise));
    allDevises.push(nouvelleDevise);
    res.send(nouvelleDevise);
});
```

- Redémarrer `server.js` si nécessaire (selon utilisation de `node` ou `nodemon`)
- Test avec Postman :
 - Nouvel onglet (requête) en mode POST
 - URL = <http://localhost:8282/devise-api/private/role-admin/devise>
 - Dans sous-onglet Headers , Key = Content-Type et Value = `application/json`
 - Dans sous-onglet Body , cliquer sur `raw` et placer :

```
{ "code" : "mxy" , "nom" : "monnaieXy" , "change" : 123 }
```
 - Cliquer ensuite sur `Send` . On devrait normalement voir les mêmes données en retour.

- Re-déclencher l'URL <http://localhost:8282/devise-api/public/devise> dans un navigateur (ou un onglet Postman en GET) pour bien vérifier l'ajout

Suppression en mode DELETE (et test avec Postman)

- Coder dans `devise-api-routes.js` une nouvelle route en mode DELETE *Exemple d'URL =* <http://localhost:8282/devise-api/private/role-admin/devise/EUR>
- Effectuer un test avec POSTMAN et réafficher toutes les devises

Suggestions :

- `tableauQuiVabien.splice(i,1);` pour supprimer l'élément en position `i`
- Retourner une réponse de type `{ deletedDeviseCode : "EUR" }`

Mise à jour en mode PUT (et test avec Postman)

- Coder dans `devise-api-routes.js` une nouvelle route en mode PUT

Exemple de code :

```
apiRouter.route('/devise-api/private/role-admin/devise')
.put( function(req , res , next ) {
    var newValueOfDeviseToUpdate = req.body;
    console.log("PUT,newValueOfDeviseToUpdate="
                +JSON.stringify(newValueOfDeviseToUpdate));
    var deviseToUpdate = findDeviseInArrayByCode(allDevises,
        newValueOfDeviseToUpdate.code);
    if(deviseToUpdate!=null){
        deviseToUpdate.nom = newValueOfDeviseToUpdate.nom;
        deviseToUpdate.change = newValueOfDeviseToUpdate.change;
        res.send(deviseToUpdate);
    }else{
        res.status(404).json({
            error : "no devise to update with code="
                + newValueOfDeviseToUpdate.code });
    }
});
```

- Effectuer un test avec Postman et réafficher toutes les devises