

Tp "devise-api-routes"

Consignes du Tp

A 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**
- 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');
var deviseApiRoutes = require('./devise-api-routes');
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');
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;
```

Lancer ensuite **server.js** avec **node** ou bien **nodemon** et tester via l'url suivante: <http://localhost:8282/devise-api/public/devise> (<http://localhost:8282/devise-api/public/devise>).

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> (<http://localhost:8282/devise-api/public/devise/EUR>).

Fin de route de type `/devise-api/public/devise/:code` Et récupération du code en fin d'url via `req.params.code`

Effectuer un test après avoir si besoin redémarrer le serveur.

Ajuster si besoin de code de ***server.js*** de façon à rediriger par défaut sur ***html/index.html*** (fichier *static* à gérer via express).

Ajouter des liens hypertextes de type ` ... ` dans ***index.html*** de façon à déclencher les 2 appels précédents de WS REST 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 et avec la syntaxe ***req.query.changeMini*** pour récupérer la valeur d'un paramètre http changeMini en fin d'URL (après ?)

Exemple d'url : <http://localhost:8282/devise-api/public/devise?changeMini=1.05> (<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** `var bodyParser = require('body-parser');` et

```
var jsonParser = bodyParser.json() ;  
app.use(jsonParser);
```

Dans le haut du fichier `server.js` de façon à ce que Express puisse convertir de JSON en javascript les données envoyées en mode POST (`req.body`)

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 (requete) , mode POST ,
- url=<http://localhost:8282/devise-api/private/role-admin/devise> (<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> (<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 URL= <http://localhost:8282/devise-api/private/role-admin/devise/EUR> (<http://localhost:8282/devise-api/private/role-admin/devise/EUR>).

Et effectuer un test avec POSTMAN et réafficher tout.

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 });
    }
});
```

Et effectuer un test avec POSTMAN et réafficher tout.