Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

télécom saint-étienne
école d'ingénieurs nouvelles technologies

TD 4 : Simple RESTful service

École associée INSTITUT Mines-Télécom

Université Jean Monnet SAINT-ETIENNE

Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

Objectifs

Le but de ce TD est de créer un web service REST simple avec la méthode GET uniquement.
Pour ce faire, nous utiliserons Eclipse, Java EE et un serveur qui sera GlassFish 4 dans notre cas.

Réponses

Pour les besoins du TD, nous avons créer un projet Java (code fournit avec le compte rendu).
Les réponses ne contiennent pas de code Java mais uniquement des XML.

Avant de commencé, on ajoute au constructeur un méthode qui permettra de remplir la liste vide

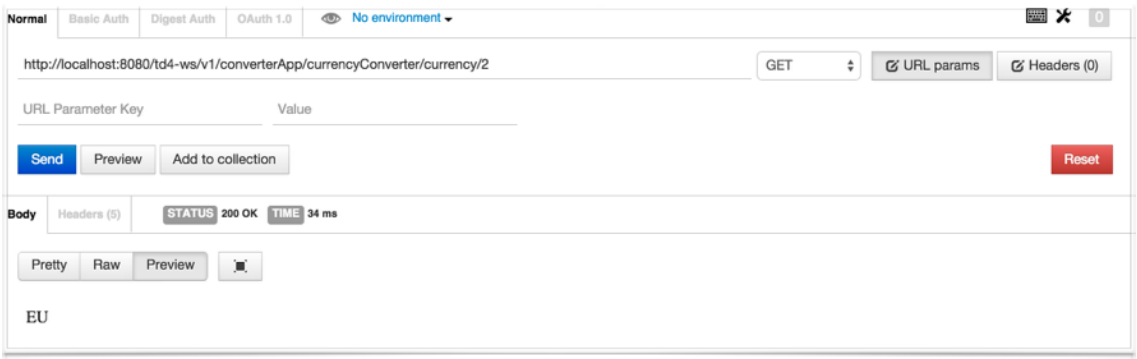
```
/**
* Constructeur de la classe.
* Initialisation de la liste avec les currencies (EURO, YEN, DOLLAR) si elle est vide
*/
public CurrencyConverter() {

	super();

	this.version = "1.0";

	if(currencies.isEmpty()){

		CurrencyConverter.initializeCurrencies();

	}

}



/**
* Constructeur de la classe.
* Initialisation de la liste avec les currencies (EURO, YEN, DOLLAR) si elle est vide
*/
private static void initializeCurrencies() {

	currencies.add(new Currency("USA", "Dollar", 1800, 1));

	currencies.add(new Currency("EU", "Euro", 2000, 2));

	currencies.add(new Currency("Japan", "Yen", 1945, 3));
}
```

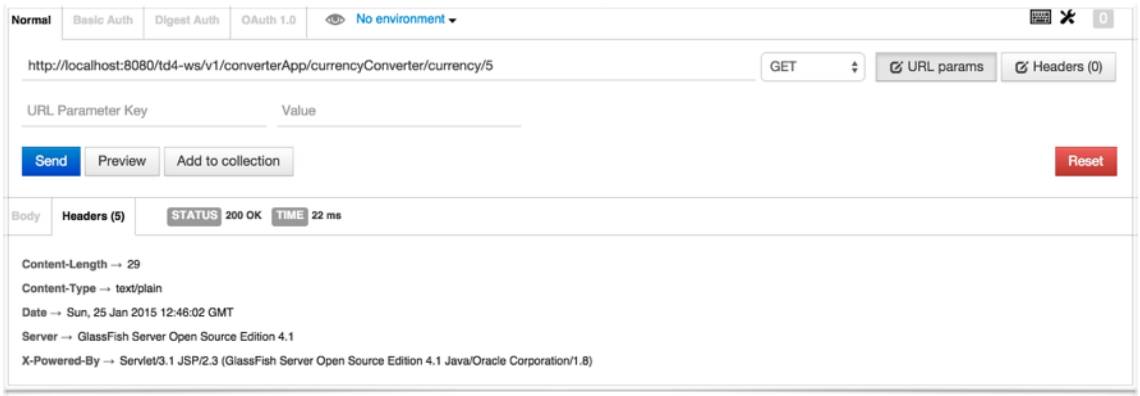
Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

Question 01 :
Write a method that allows to retrieve a currency name when the user provides the ID, e.g : http://localhost:8080/TD4/v1/converterApp/currencyConverter/currency/1

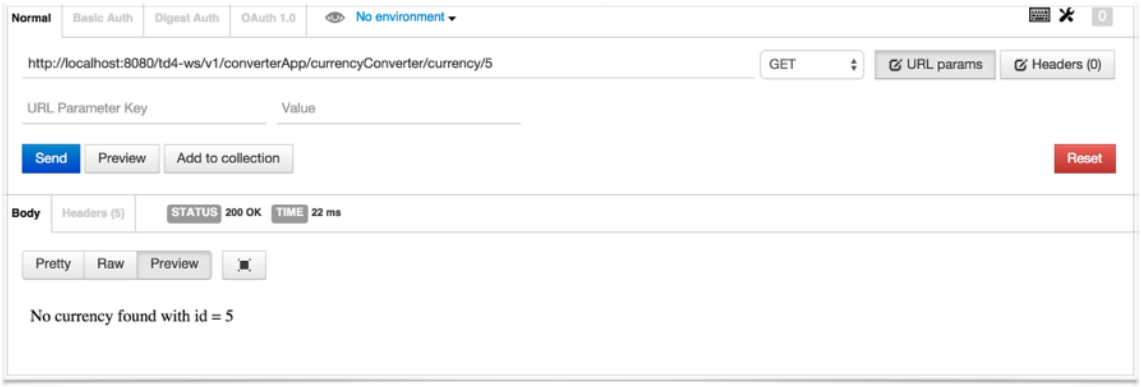
```
/**
* Retourne la currency selon son id (dans notre cas, entre 1-3).
*
* URL :
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currency/:id
*
* @param  Integer id       : id de la currency
* @return String           : version du web serivce REST
*/
@GET
@Path("currency/{identifier: [0-9]*}")
public String currency(@PathParam("identifier") Integer id) {

	String result = null;

	for (Currency c : currencies) {

		if (c.getId() == id) {

			result = c.getName();

		}

	}

	if (result == null) {

		result = "No currency found with id = " + id;

	}

	return result;

}
```

Quand on transmet un « id » qui existe dans la liste, le web service nous répond.





Quand on transmet un « id » qui n'existe pas dans la liste, le web service nous répond quand même.



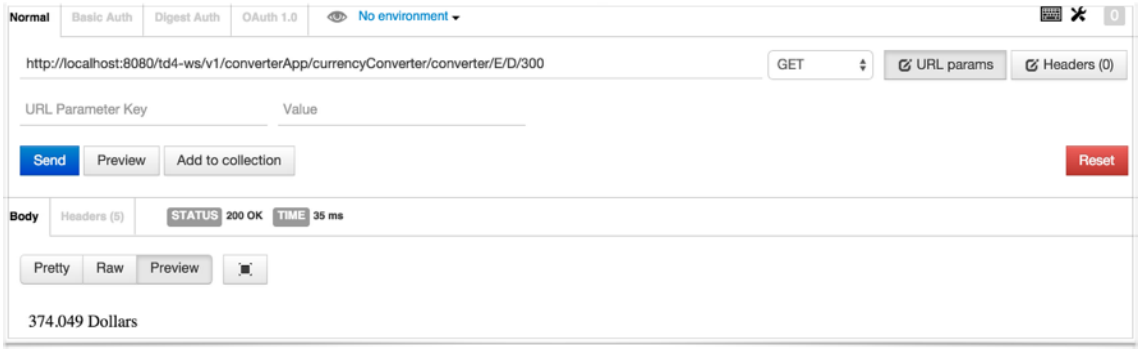
Question 02.1 :
rewrite the method convert of TD2 in REST(Jersey), e.g :
http://localhost:8080/TD4/v1/converterApp/currencyConverter/conversion/D/Y/1000

```
/**
* Convertie un montant entre deux devises (EURO/YEN/DOLLAR)
*
* URL :
*
* Conversion de 300 Euro 'E' en Dollar 'D'
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/converter/E/D/300
*
* Conversion de 4500 Yens 'Y' en Euros 'E'
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/converter/Y/E/4500
*
* @param  String currency_source        : identifiant (E/Y/D) de la currency source
* @param  String currency_destination   : identifiant (E/Y/D) de la currency de destina-
* -tion
* @param  Double amount                 : montant a convertir
* @return String                        : resultat de la conversion
*/
```

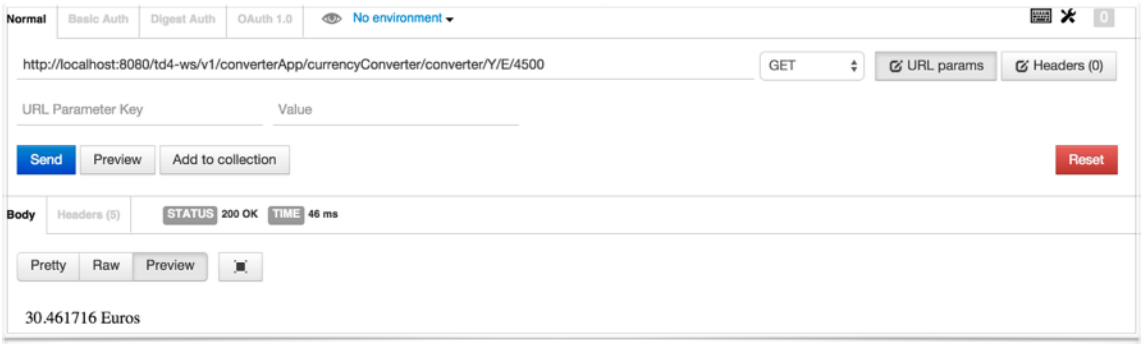
```
@GET
@Path("converter/{currency_source}/{currency_destination}/{amount}")
public String converter(@PathParam("currency_source") String source,
                        @PathParam("currency_destination") String destination,
                        @PathParam("amount") double amount) {

        Double amountConverted = 0d;

        Double device;

        String dest_long = null;

        if (source.equals("D") && destination.equals("Y")) {

                device = 118.481;

                dest_long = "Yens";

                amountConverted = amount * device;

        }

        if (source.equals("D") && destination.equals("E")) {

                device = 0.802032;

                dest_long = "Euros";

                amountConverted = amount * device;

        }

        if (source.equals("E") && destination.equals("D")) {

                device = 1.24683;

                dest_long = "Dollars";

                amountConverted = amount * device;

        }

        if (source.equals("E") && destination.equals("Y")) {

                device = 147.726;

                dest_long = "Yens";

                amountConverted = amount * device;

        }

        if (source.equals("Y") && destination.equals("D")) {

                device = 0.00844069;

                dest_long = "Dollars";

                amountConverted = amount * device;
```

```
        }
        if (source.equals("Y") && destination.equals("E")) {
                device = 0.00676927;
                dest_long = "Euros";
                amountConverted = amount * device;
        }
        return amountConverted.floatValue() + " " + dest_long;
}
```

Quand on test l'URI , la conversion fonctionne.
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/converter/E/D/300



http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/converter/Y/E/4500



Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

Question 02.2 :
compare the method you write, with the method of TD2. Please ex- plain the difference between RESTful Services and SOAP services

Web Service SOAP : (Class : @WebService l Méthode : @WebMethod)

Pour tester nos WebService SOAP nous devions aller dans la console Glassfish, puis dans l'onglet application nous avions accès aux différents WebService. De la nous pouvions tester les méthodes en mettant les paramètres nécessaires.
Ceci nous permettant de générer les SOAP Response et les SOAP Request.

Web Service REST:
(Class : @Path l Méthode : @Get)

Pour tester nos Web Service REST nous utilisions le plugin Postman (ou un navigateur) pour pouvoir lire la réponse d'une requête (GET dans le TD4).

SOAP :

Avantages:
- Conçu pour gérer les environnements informatiques distribués.
- La norme est en vigueur pour les services Web, et a donc un meilleur soutien des autres normes (WSDL, WS-*) et de l'outillage des fournisseurs.
- Built-in de gestion des erreurs (fautes).
- Extensibilité.

Inconvénients:
- Conceptuellement plus difficile, plus "lourd" que le reste.
- Plus bavard.
- Plus difficiles à développer, nécessite des outils.

REST:

Avantages:
- Beaucoup plus simple à développer que SOAP.
- Courbe d'apprentissage petite, moins de dépendance sur les outils.

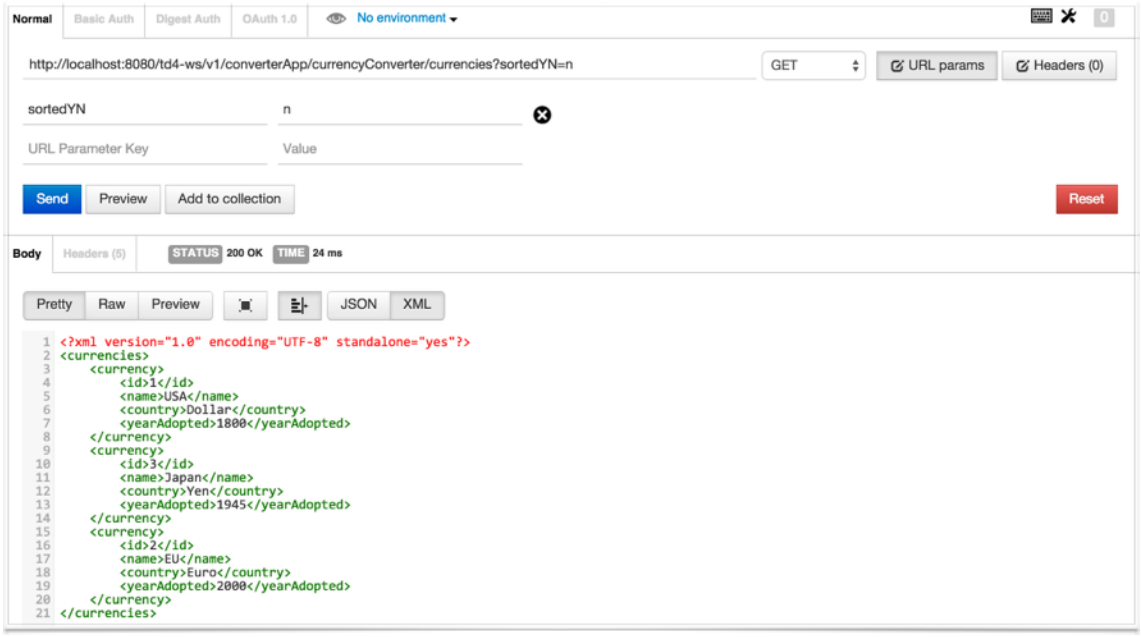
Inconvénients:
- Manque de soutien pour les normes de sécurité, la politique, la messagerie fiable, etc, afin que les services qui ont des exigences plus sophistiquées sont plus difficiles à développer.

Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

Question 03.1 : write a method that returns the list of currencies encoded in XML.

```
/**
* Retourne la liste des currencies en XML.
* En passant le parametre GET 'sortedYN' à 'y' , on demande une liste ordonnée.
* En passant le parametre GET 'sortedYN' à une autre valeur, l'ordre de la liste
* est inversé.
*
* HEADER :  Accept à text/xml
*
* URL :
*
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?
sortedYN=y
*
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies? * *
* sortedYN=n
*
* @param  String sortedYN          : Dans le cas ou c'est 'y', la liste est or
* donnée
* Sinon, la liste est dans l'ordre inverse
* @return String XML               : Liste des currencies en XML
*/
@GET
@Path("/currencies")
@Produces(MediaType.TEXT_XML)
public List<Currency> getCurrenciesXML(@QueryParam("sortedYN") String value){

      // Dans le cas ou on a sortedYN=y
      if(value.equals("y")){

      // on ordonne la liste en spécifiant une fonction pour le  Comparator
            Collections.sort(currencies, new Comparator<Currency>(){

                  public int compare(Currency c1, Currency c2){

                        return c1.getName().compareTo(c2.getName());

                  }

            });

      }else{

            // On inverse la liste
            Collections.reverse(currencies);
      }

      return currencies;

}
```

Quand on test l'URI , le résultat est bien en XML.
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=n

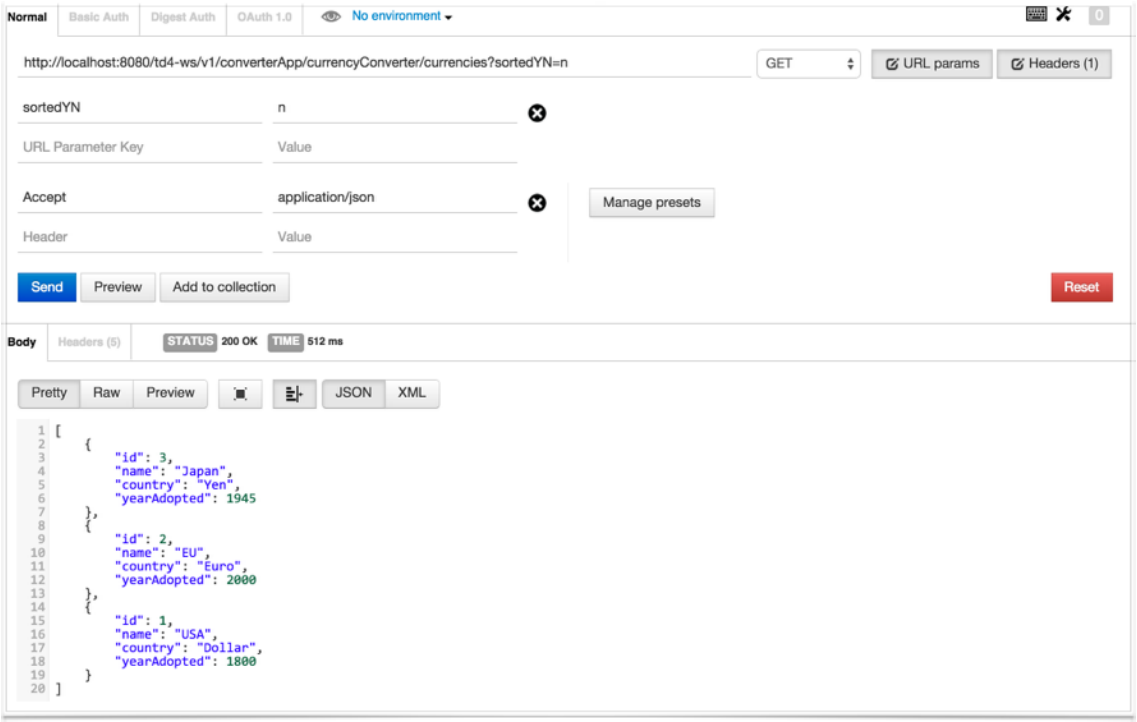


Question 03.2 : write another method that returns the list of currencies encoded in JSON.

```
/**
* Retourne la liste des currencies en JSON.
* En passant le parametre GET 'sortedYN' à 'y' , on demande une liste ordonnée.
*En passant le parametre GET 'sortedYN' à une autre valeur, l'ordre de la liste est in *
*versé
*
* HEADER :  Accept à application/json
*
* URL :
*
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=y
*
* http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=n
*
* @param  String sortedYN        : Dans le cas ou c'est 'y', la liste est ordonnée
* Sinon, la liste est dans l'ordre inverse
* @return String JSON                  : Liste des currencies en XML
*/
@GET
@Path("currencies")
@Produces(MediaType.APPLICATION_JSON)
public List<Currency> getCurrenciesJSON(@QueryParam("sortedYN") String value){

	// Dans le cas ou on a sortedYN=y
	if(value.equals("y")){

		// on ordonne la liste en spécifiant une fonction pour le  Comparator
		Collections.sort(currencies, new Comparator<Currency>(){
```

```
                    public int compare(Currency c1, Currency c2){

                        return c1.getName().compareTo(c2.getName());

                    }

                });

        }else{

                // On inverse la liste
                Collections.reverse(currencies);
        }

        return currencies;

}
```

Quand on test l'URI , le résultat est bien en XML et non pas en JSON.
On ajoute au Header de la requête un paramètre afin de récupérer le résultat au bon format , le JSON , (cf. commentaires de la fonction).
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=n

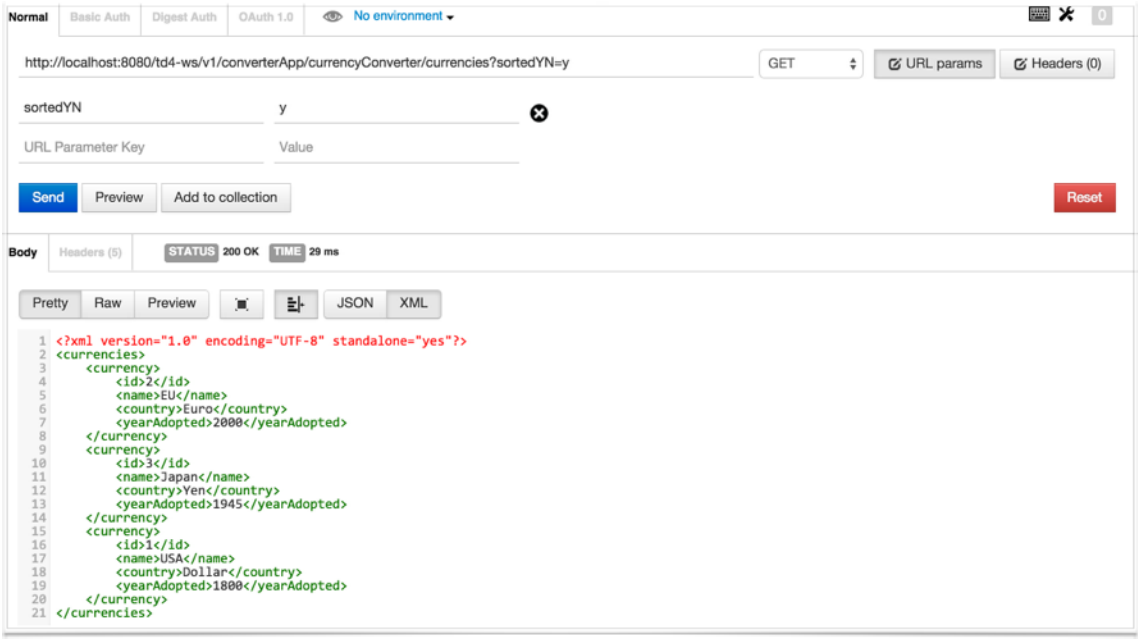


Question 04 : Add a query parameter to the methods getCurrenciesXML() and getCurrenciesJSON() so that the results are sorted

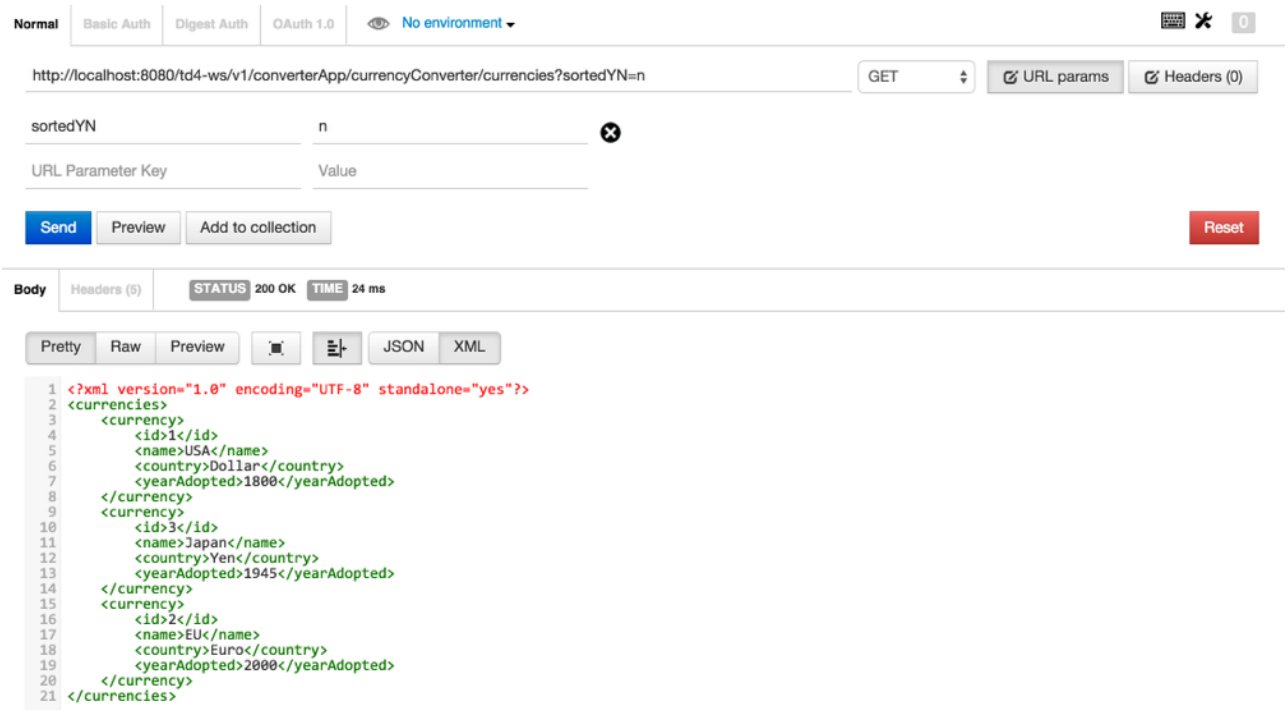
Le code pour ordonner les devises est déjà présent dans la réponse 3 (3.1 , 3.2). Voici les tests sur l'URI avec les deux paramètres.

Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

URI avec paramètre « sortedYN à y » avec une réponse en XML.
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=y

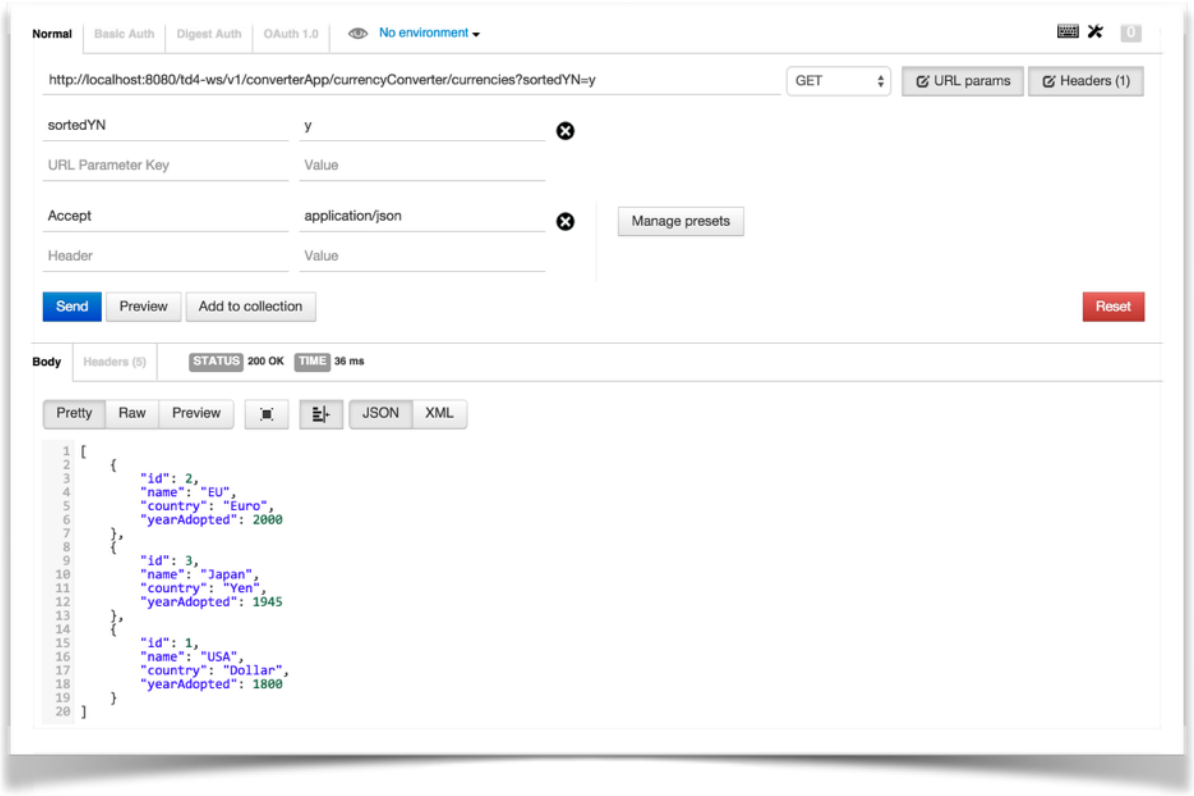


URI avec paramètre « sortedYN à n »
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=n



Abdelkrim **AHMED BACHA** &Nacer **BOUHERROU**

URI avec paramètre « sortedYN à y » avec une réponse en JSON.
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=y



URI avec paramètre « sortedYN à n » avec une réponse en JSON.
http://localhost:8080/td4-ws/v1/converterApp/currencyConverter/currencies?sortedYN=n