TP 4 - Spring Cloud Configuration

- Spring BOOT
- Spring Cloud Configuration

Estimated Time :

- **30** *minutes*

4.0 - Objectifs

Le présent TP se veut utiliser Spring-Cloud Configuration :

- mise en place d'un serveur
- illustration avec resanet-reservation
  - récupération de la datasource

4.1 - Mise en place du serveur

TODO 01

Dans resanet-config

Ajouter les dépendances de Spring-cloud configuration

Changer le starter :

```
    <parent>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-parent</artifactId>
        <version>Brixton.BUILD-SNAPSHOT</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
```

Et ajouter la dépendance de Spring cloud config

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-config-server</artifactId>
</dependency>
```

TODO 02

Dans la configuration de l'application, src/main/java/com.zenika.microservices.resanet.configuration.ConfigurationApplication, configurer le repository de config

```
@EnableConfigServer
```

**Le plus difficile est déjà fait !!! **

TODO 03

Dans /src/main/ressources/application.yml

Donner un nom à sa configuration

```
spring:
    application:
        name: resanet-config
```

Attention ce nom n'est pas anodin, il va servir dans les URL d'interrogation

TODO 04

Configurer un profil

```
spring:
    profiles:
        active: native
```

NB Spring a été défini avant ne pas répéter

TODO 05

Configurer le lieu de la configuration

```
spring:
    cloud:
        config:
            server:
                native:
                    searchLocations: classpath:/config/
```

NB Spring a été défini avant ne pas répéter

TODO 06

Cela veut dire que la configuration sera celle dans /config/<nom-application>.

Pour **SEE** sont possibles :

- native
- dev
- cloud

Les URI possibles sont :

```
/{application}/{profile}[/{label}]
/{application}-{profile}.yml
/{label}/{application}-{profile}.yml
/{application}-{profile}.properties
/{label}/{application}-{profile}.properties
```

démarrer l'application avec la commande

```
mvn spring-boot:run
```

TODO 07

Obtenir les propriétés sur l'URL :

http://localhost:8888/<nom_application>/<nom_profil>

```
curl -X GET 'http://localhost:8888/resanet-catalogue-04/native'
```

4.2 Partie cliente

Importer les bonnes dépendances

TODO 08

Dans resanet-parent/pom.xml

```
    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-dependencies</artifactId>
                <version>Brixton.SR5</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
```

TODO 09

Dans resanet-catalogue/pom.xml ajouter spring-cloud-starter-config

```
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
        </dependency>
```

Récupérer une propriété depuis spring-cloud-config

Dans La classe CatalogueInfoController, il y a un controller qui répond sur

```
GET /infos
```

Par défaut @Value("${catalogue.version:manuelle}"), c'est la valeur de droite qui est prise en compte, c'est à dire **manuelle**

TODO 10

Appeler le endpoint http://localhost:8080/infos/

il doit répondre manuelle.

Configurer la spring-cloud-config

Il faut indiquer le nom de l'application pour répondre comme le fichier property.

dans resanet-catalogue/src/main/ressources/bootstrap.yml

TODO 11

```
spring:
    application:
        name: resanet-catalogue-04
```

Puis configurer l'endoit où l'on va chercher la configuration

```
Spring:
    cloud:
        config:
            uri: http://localhost:8888/
```

TODO 12

Redémarrer l'application et réappeler le endpoint infos/, doit retourner **hiver-2015**

4.3 Comprendre les profils

Comprendre les profils

TODO 13

créer un fichier : **resanet-catalogue-04-dev.properties** dans resanet-config/src/main/ressources/config

ajouter la propriété :

```
catalogue.gamme=dev-gamme
```

Dans resanet-config/src/main/ressources/application.yml

```
Spring:
    profiles:
        active: native,dev
    cloud:
        config:
            server:
                native:
                    searchLocations: classpath:/config/
                dev:
                    searchLocations: classpath:/config/
```

redémarrer le serveur de configuration