

Backend Kata

Notre application a un point d'entrée (Input) et une sortie (Output), elle va nous retourner une chaîne de caractères en se basant sur trois règles. On constate que si une règle est vérifiée :

- Le nombre 3 se convertit en «FOO».
- Le nombre 5 se convertit en « BAR ».
- Le nombre 7 se convertit en « QUIX ».

Dans notre package enums on crée une enum Output qui va permettre de convertir le nombre à une valeur ainsi permettre facilement de rajouter de nouveaux enums si besoin

```
package com.kata.quiz.enums;

public enum Output {
    FOO(number:3),
    BAR(number:5),
    QUIX(number:7);

    public final Integer number;

    private Output(Integer number) {
        this.number = number;
    }

    public static String valueOfNumber(Integer value) {
        for (Output e : values()) {
            if (e.number.equals(value)) {
                return e.toString();
            }
        }
        return "";
    }
}
```

La première vérification, c'est de vérifier si notre input est compris dans l'intervalle [0,...,100] et que notre input doit être un entier valide.

```
private boolean isValueInRange(Integer input) {
    return ValueRange.of(min:0, max:100).isValidIntValue(input);
}
```

Par la suite, la règle de division s'applique sur l'input avec 3 et 5 :

```
private String transformInputWhenDivisible(Integer input) {
    List<Integer> numbers = Arrays.asList(...a:3, 5);
    return numbers.stream()
        .map(number -> getOutputWhenInputDevidedByParam(input, number))
        .collect(Collectors.joining());
}
```

Extraire la chaine de caractères à partir du numéro si le reste de division est égal à zéro.

```
private String getOutputWhenInputDevidedByParam(Integer input, Integer param) {
    return input % param == 0 ? Output.valueOfNumber(param) : "";
}
```

La règle contient consiste à transformer notre input entier en une liste de caractères, ce qui va permettre de vérifier chaque caractère avec « 3, 5, 7 » et d'effectuer l'analyse de gauche à

```
private String transformInputWhenEquals(Integer input) {
    List<String> listParams = Arrays.asList(...a:"3", "5", "7");
    return Arrays.asList(input.toString().split(regex:"")).stream()
        .map(character -> getOutputWhenParamsContainsCharacter(character, listParams))
        .collect(Collectors.joining());
} droite.
```

Extraire la chaine de caractères à partir de l'input, si notre tableau contient l'input.

```
private String getOutputWhenParamsContainsCharacter(String input, List<String> params) {
    return params.contains(input) ? Output.valueOfNumber(Integer.valueOf(input)) : "";
}
```

Crée notre service qui va permettre d'accéder à l'ensemble de nos règles et de les exécuter en ordre selon la priorisation voulue.

```
@Service
public class DataServiceImp implements DataService {

    @Override
    public String convertInputToOutput(Integer input) {
        if (isValueInRange(input)) {
            String result = transformInputWhenDivisible(input) + transformInputWhenEquals(input);
            return result.length() > 0 ? result : input.toString();
        }
        return input.toString();
    }
}
```

Notre controller expose un endpoint qui va permettre de convertir notre input en output. En se basant sur notre service.

```

@RestController
@RequestMapping("/api/")
@CrossOrigin

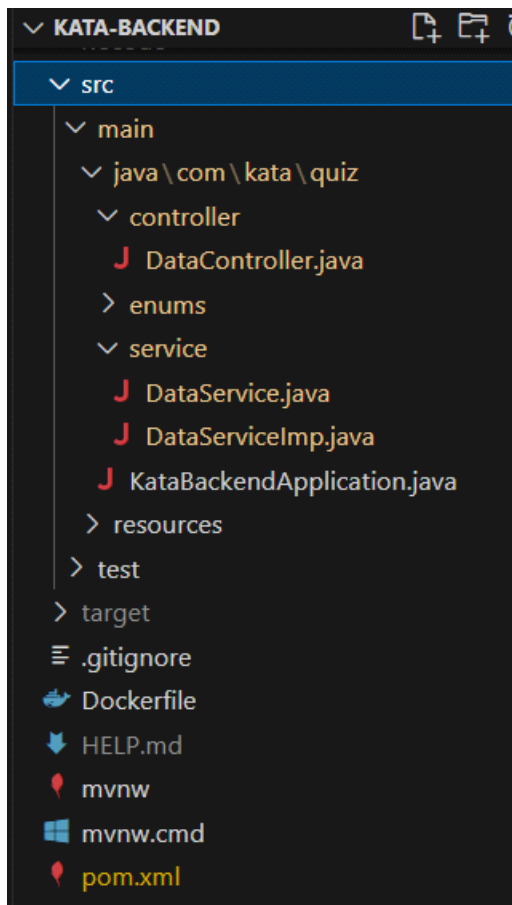
public class DataController {

    @Autowired
    DataService dataService;

    @RequestMapping(value = "convertInputToOutput", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<String> convertInputToOutput(@PathParam("input") Integer input) {
        return ResponseEntity.ok(dataService.convertInputToOutput(input));
    }
}

```

Structure projet



Commande pour lancer l'application :

```

PS C:\Users\Yassine HOUSSAINI\Documents\kata-backend> mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.kata.quiz:kata-backend >-----
[INFO] Building kata-backend 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.0.2:run (default-cli) > test-compile @ kata-backend >>>
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ kata-backend ---
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ kata-backend ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ kata-backend ---
[INFO] skip non existing resourceDirectory C:\Users\Yassine HOUSSAINI\Documents\kata-backend\src\test\resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ kata-backend ---

```

Test effectué sur Postman :

input 33: divisible par 3 et contient deux fois 3

result: "FOOFOOFOO"

<http://localhost:8081/api/convertInputToOutput?input=33>

GET <http://localhost:8081/api/convertInputToOutput?input=33>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
input	33	

Body Cookies Headers (8) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

1 FOOFOOFOO

input 51: divisible par 3 et contient 5

result:"FOOBAR"

<http://localhost:8081/api/convertInputToOutput?input=51>

GET <http://localhost:8081/api/convertInputToOutput?input=51>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
input	51	

Body Cookies Headers (8) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

1 FOOBAR

input 53: contient 3 et contient 5

result: "FOOBAR"

http://localhost:8081/api/convertInputToOutput?input=53

GET http://localhost:8081/api/convertInputToOutput?input=53

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
input	53	

Body Cookies Headers (8) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 BARF00
```

input 11: Aucune règle ne s'applique.

resultat :

http://localhost:8081/api/convertInputToOutput?input=11

GET http://localhost:8081/api/convertInputToOutput?input=11

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
input	11	

Body Cookies Headers (8) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1 11
```

"11"

input test: La méthode n'accepte que des entiers.

resultat: erreur 400

http://localhost:8081/api/convertInputToOutput?input=test

GET http://localhost:8081/api/convertInputToOutput?input=test

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
input	test	

Body Cookies Headers (7) Test Results Status: 400 Bad Request

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-06-02T09:51:50.293+00:00",
3   "status": 400,
4   "error": "Bad Request",
```

Pour tester directement sur le navigateur :

Lors de la première exécution, le navigateur peut prendre du temps pour reprendre, car l'application se

met en veille (serveur gratuit).

<https://kata-backend.onrender.com/api/convertInputToOutPut?input=33>