

Examen 4 – Semaine 12 (20 minutes, 4%) FORMATIF

Étudiant :

Trace

4 points Produisez la trace d'exécution du contrôleur suivant pour les requêtes reçues dans cet ordre. Vous devez préciser les codes et corps des réponses HTTP:

1. GET /api/compteur/inc/5
2. GET /api/compteur/inc/-2

```
1 @Controller
2 class CompteurController {
3     int total = 0;
4
5     @GetMapping("/inc/{valeur}")
6     public int incrementer(@PathVariable int
valeur) {
7         if (valeur > 0) {
8             total += valeur;
9             return total;
10        } else {
11            throw new
IllegalArgumentException("Plop plop plop");
12        }
13    }
14 }
```

```
1 @ControllerAdvice
2 public class Pipo extends
ResponseBodyExceptionHandler {
3
4     @ExceptionHandler(value =
{Exception.class})
5     protected ResponseEntity<Object>
handleConflict(Exception ex, WebRequest request)
{
6         String b =
ex.getClass().getSimpleName();
7         return handleExceptionInternal(ex, b,
HttpStatus.BAD_REQUEST, request);
8     }
9 }
```

No ligne	Effets (variables, retour)	Pile d'appel

Erreur de code

- Le Toast d'erreur ne s'affiche jamais
- même si la réponse du serveur est une erreur avec un code 400
- on souhaite que le Toast s'affiche quand le serveur a répondu mais avec une réponse avec un code 400.

```
1 try{
2     service.enqueue(object : Callback<List<String>> {
3         override fun onResponse(call: Call<List<String>>, response: List<String>) {
4             if (response.isSuccessful()) {
5                 liste = response.body()
6             }
7             //Toast.makeText(context, "Une erreur est survenue", Toast.LENGTH_SHORT).show()
8         }
9
10        override fun onFailure(call: Call<List<String>>, t: Throwable) {
11            Log.e("Service", "Erreur lors de l'appel", t)
12        }
13    })
14 }
15 catch (e: Exception) {
16     Toast.makeText(context, "Une erreur est survenue", Toast.LENGTH_SHORT).show()
17 }
```

3 points Proposez une correction:

Réponse :