

# Examen final - partie papier (1/2)

## 4 points - Code serveur

---

Voici un ensemble des annotations classiques côté serveur :

```
@PathVariable  
@RequestBody  
@GetMapping("/gna/gna")  
@ResponseBody
```

Produis le code d'un contrôleur serveur qui :

1. **2 points** Répond à une requête POST sur l'URL `/truc/n/m` où `n` et `m` sont des entiers et j'ai un autre entier dans le corps.
2. **2 points** Renvoie une réponse HTTP avec comme corps un *long* qui est le produit de `n * m` plus le nombre entier du corps de la requête.

## 6 points - Trace client serveur

Étant donné le code suivant :

```
interface Service {
    @GET("ok/{boomer}")
    fun gogogo(@Path("boomer") boomer: String): Call<String>
}

object retrofit { // Ne pas inclure ce bloc de code dans la trace
    val service: Service = retrofit2.Retrofit.Builder()
        .addConverterFactory(retrofit2.converter.gson.GsonConverterFactory.create())
        .baseUrl("http://10.0.2.2:8080/")
        .build()
        .create(Service::class.java)
}

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        retrofit.service.gogogo("boom").enqueue(object : retrofit2.Callback<String> {
            override fun onResponse(call: retrofit2.Call<String>, response: retrofit2.Response<String>) {

                if (response.isSuccessful) {
                    Toast.makeText(this@MainActivity, response.body(), Toast.LENGTH_LONG).show()
                }
                else{
                    Toast.makeText(this@MainActivity, "Error", Toast.LENGTH_LONG).show()
                }
            }

            override fun onFailure(call: retrofit2.Call<String>, t: Throwable) {
                Log.i("Yo", t.message.toString()) // logcat niveau (i)nfo
            }
        })
    }
}
```

```
@Controller
public class Controleur {

    @GetMapping("/ok/{boomer}")
    public @ResponseBody Character trucs(@PathVariable String boomer) {
        return boomer.charAt(0);
    }
}
```

Produisez la trace d'exécution dans le cas où :

1. on part le serveur Spring Boot localement
2. on démarre l'application Android

**Note** : Vous pouvez utiliser des [...] pour éviter d'écrire les trop longues lignes de code, en autant que c'est clair de quelle ligne il s'agit, par exemple :

```
retrofit.service.gogogo [...]
```

- **3 points** les lignes de code sont mentionnées dans l'ordre précis de leur exécution, il n'y a d'oubli ou de lignes en plus.
- **3 points** les colonnes effet et pile d'appels sont correctement remplies, par exemple les requêtes ou réponses mentionnent tous les éléments HTTP appropriés (méthode, corps etc.)

