

# LICENCE PROFESSIONNELLE / PROGRAMMATION AVANCEE

## ANDROID - PROJET

L'objectif principal du projet est de faire une synthèse des concepts vus jusqu'à maintenant et d'utiliser les appels REST.

L'application à développer permet de lister des « calembours » et de les noter.

## 1 Descriptif de l'application

### 1.1 API « calembour »

L'URL suivante : [https://official-joke-api.appspot.com/random\\_joke](https://official-joke-api.appspot.com/random_joke) ; permet d'interroger une API dans le but d'obtenir une structure de données JSON contenant des jeux de mots.

Plusieurs fonctions de cette API sont à notre disposition :

- /random\_joke
- /random\_ten
- /jokes/random
- /jokes/ten

### 1.2 Notre objectif

Nous souhaitons proposer à l'utilisateur, une application permettant de lui afficher les calembours récupérés depuis l'API. L'utilisateur aura la possibilité de donner une note aux calembours qu'il aura lus.

Aussi, nous permettront à l'utilisateur de saisir et d'enregistrer ses propres calembours.

## 2 Travail à réaliser

### 2.1 Interface de l'application

Pour réaliser ce projet, vous êtes libre de proposer l'interface de votre choix.

Cependant, cette interface doit être adaptée aux différentes fonctionnalités voulues pour notre application ; et elle doit être la plus ergonomique possible.

### 2.2 Lister les calembours

Le but premier de notre application est de présenter à l'utilisateur les calembours obtenus depuis l'API.

Un exemple d'appel à l'API ([https://official-joke-api.appspot.com/random\\_ten](https://official-joke-api.appspot.com/random_ten)) renverra un contenu JSON dont la forme est la suivante :

```
[
  {
    "type": "general",
    "setup": "Why did the A go to the bathroom and come out as an E?",
    "punchline": "Because he had a vowel movement.",
  }
]
```

```

    "id": 319
  },
  {
    "type": "general",
    "setup": "Do you want a brief explanation of what an acorn is?",
    "punchline": "In a nutshell, it's an oak tree.",
    "id": 108
  },
  {
    "type": "general",
    "setup": "Where do you learn to make banana splits?",
    "punchline": "At sundae school.",
    "id": 293
  },
  {
    "type": "general",
    "setup": "What do you call a sheep with no legs?",
    "punchline": "A cloud.",
    "id": 218
  },
  {
    "type": "general",
    "setup": "What do you call a duck that gets all A's?",
    "punchline": "A wise quacker.",
    "id": 205
  },
  {
    "type": "general",
    "setup": "How many seconds are in a year?",
    "punchline": "12. January 2nd, February 2nd, March 2nd, April 2nd.... etc",
    "id": 143
  },
  {
    "type": "general",
    "setup": "What kind of dog lives in a particle accelerator?",
    "punchline": "A Fermilabrador Retriever.",
    "id": 258
  },
  {
    "type": "general",
    "setup": "Why do valley girls hang out in odd numbered groups?",
    "punchline": "Because they can't even.",
    "id": 359
  },
  {
    "type": "general",
    "setup": "Why do scuba divers fall backwards into the water?",
    "punchline": "Because if they fell forwards they'd still be in the boat.",
    "id": 357
  },
  {
    "type": "programming",
    "setup": "Knock-knock.",
    "punchline": "A race condition. Who is there?",
    "id": 368
  }
]

```

Nous constatons que les calembours sont de la forme « Question / Réponse ». Pour le suspense, nous n'afficherons que les questions des calembours à l'utilisateur. La réponse ne sera affichée que si l'utilisateur clique sur une question.

## 2.3 Type des calembours

Dans le contenu JSON, nous constatons que les calembours sont typés. Nous souhaitons ajouter un filtre à notre liste des calembours pour ne proposer à l'utilisateur que les calembours du type que ce dernier aura choisi

## 2.4 Pagination

Nous souhaitons afficher des boutons « précédent » et « suivant » ; permettant de se déplacer dans la liste des calembours.

Un appui sur le bouton suivant affichera les calembours suivants (en appelant éventuellement l'API pour obtenir de nouveaux calembours).

Un appui sur le bouton précédent affichera les calembours déjà listés.

Aussi, une zone au niveau de ces boutons affichera la page en cours pour que l'utilisateur sache à quel endroit il se trouve dans la liste des calembours.

## 2.5 Notation des calembours

Après avoir consulté la réponse d'un calembour, nous permettrons à l'utilisateur de noter le calembour.

Une note peut être changée lorsque l'utilisateur réaffiche la réponse d'un calembour déjà noté.

## 2.6 Saisie des calembours

L'utilisateur aura à sa disposition une interface lui permettant de saisir ses propres calembours. Ils seront de la même forme que les calembours de l'API, c'est-à-dire qu'il auront :

- Un type
- Une question (setup)
- Une réponse (punchline)

L'utilisateur lors de la saisie pourra donner lui-même une note à son propre calembour.

## 2.7 Base de données

Nous gèrerons une base de données pour enregistrer de manière persistante :

- Les notes que l'utilisateur aura données aux calembours
- Les types de calembours (car l'utilisateur peut saisir des types de calembours qui n'existent pas dans l'API)
- Les calembours déjà vus, selon le besoin (voir § Paramètres)

## 2.8 Paramètres

Nous proposerons un écran de paramétrage de l'application. Cet écran aura un paramètre qui permettra à l'utilisateur de ne pas afficher à nouveau un calembour qu'il a déjà vu. Ce paramètre entrera en ligne de compte lorsque nous obtiendrons des calembours en provenance de l'API. Dans ce cas-là, il faudra peut-être « rejeter » des calembours déjà vus pour en obtenir d'autres non vus.

## 2.9 Statistiques

Un écran permettra d'afficher par type de calembours :

- Le nombre de réponses aux calembours vues
- La moyenne des notes obtenues par ces calembours

## 2.10 Affichage des calembours les mieux notés

Nous souhaitons pouvoir afficher les calembours triés par ordre décroissant des notes.

## 2.11 Gestion de la base de données

Nous souhaitons pouvoir gérer les enregistrements de notre base de données. Les opérations sur la base de données seront :

- Modifier le nom d'un type de calembour
- Modifier un calembour saisi
- Supprimer un calembour saisi
- Supprimer tous les calembours d'un type
- Supprimer un type de calembour

## 2.12 Autres fonctionnalités

Selon votre perception du sujet et selon votre propre initiative, proposez et développez un ou plusieurs fonctionnalités qui pourraient apporter un plus à l'application.

J'ai fais un historique des calembours déjà vu pour modifier la note donnée par exemple  
Bouton réinitialiser lors du filtrage pour les recherches

RESTE A FAIRE :  
- Clean code