# Annexe 9 – Révision Examen 2

# Exercice 1- Utilisez Volley seulement ( *40 min* )

En utilisant les données JSON sur le serveur suivant :

<https://api.jsonbin.io/v3/b/6733b233ad19ca34f8c9149a?meta=false>

Répondez aux questions suivantes en utilisant une requête GET de la librairie Volley dans une activité d’un nouveau projet Android :

1. La racine du fichier JSON est-elle un objet ou un tableau ? ( *rép. : Objet* )
2. Affichez le contenu du header du menu ( *rép : SVG Viewer* )
3. Affichez le nombre d’éléments du tableau Items ( *rép. : 22* )
4. Afficher le nombre d’éléments du tableau Items n’ayant pas d’attributs ( seulement null ) *( rép. : 4 )*
5. Afficher le nombre d’éléments du tableau Items n’ayant pas d’attribut «label »

*( rép. : 6 )*

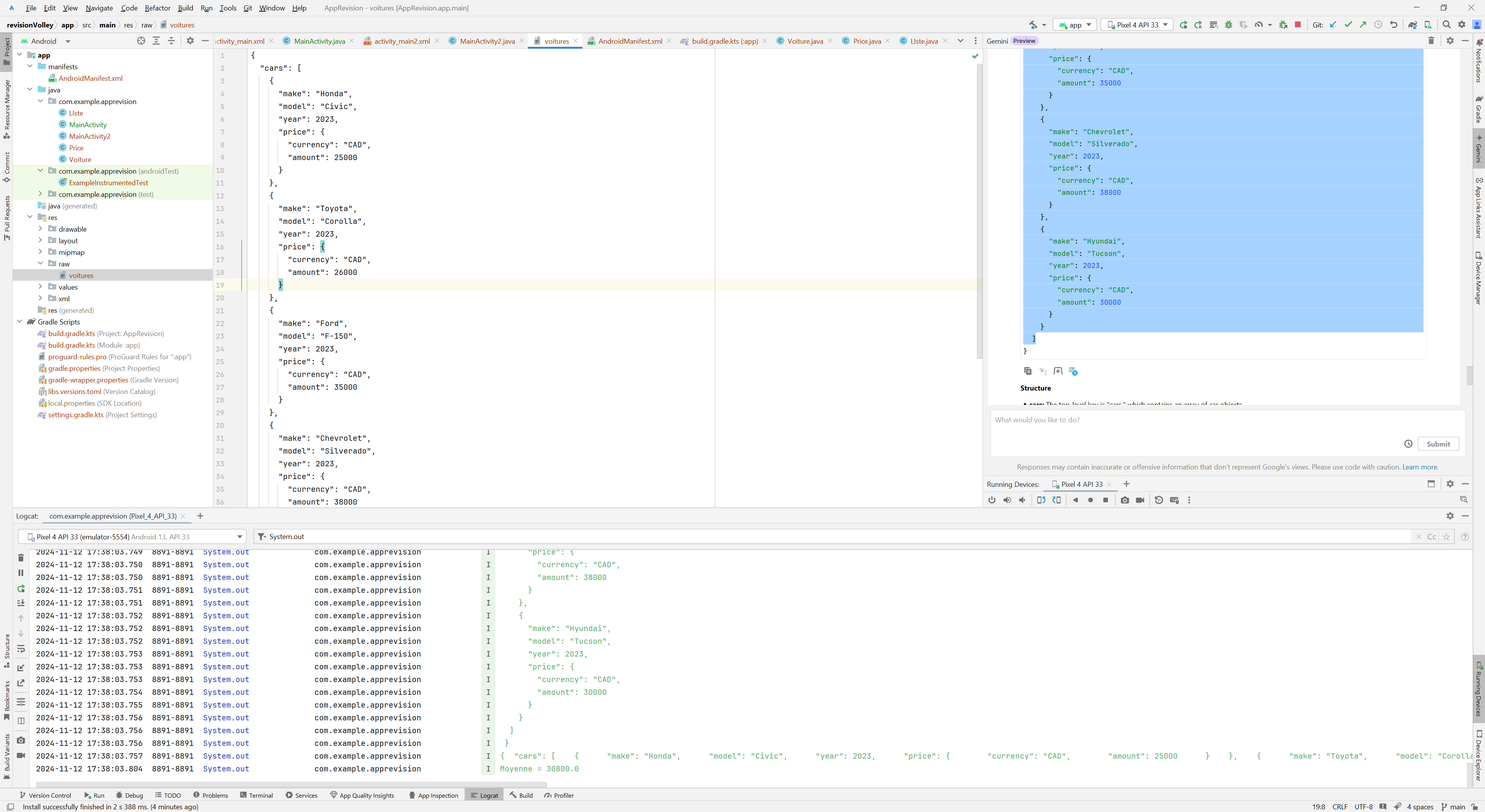
1. Quel type d’exception lance l’appel de getString(« label ») de la classe JSONObject ? ( *rép. JSONException* )

Javadoc de la classe JSONObject :

<https://stleary.github.io/JSON-java/org/json/JSONObject.html>

# Exercice 2 – Utilisez GSON et des flux de données ( *30 min* )

1. Servez-vous de votre AI générative préférée ( Gemini est accessible dans Android Studio ) afin de générer un fichier JSON représentant des voitures et leurs prix au Canada.
2. Nous n’utiliserons pas Volley cette fois car notre fichier JSON ne sera pas sur un serveur externe mais dans un fichier voitures stocké dans un dossier raw que vous créez dans le dossier res de votre projet :



1. Ajoutez des classes représentant le contenu du fichier JSON de la manière vue dans l’atelier 2B ( GSON ) . Ajoutez la libraire GSON dans vos « dependencies »
2. Nous devons maintenant lire le JSON local dans une nouvelle activité de votre projet. Pour ce faire, on va d’abord connecter notre ressource locale à un InputStream :

InputStream is = getResources().openRawResource(R.raw.*voitures*);

Avec vos connaissances sur les flux de données, utilisez d’autres flux afin de pouvoir lire chaque ligne du fichier JSON et les concaténer dans une longue String ( rappelez-vous, la librairie GSON utilise des Strings pour convertir le JSON en objets Java / Kotlin.

1. Sur un objet Gson, utilisez la méthode fromJson de la manière faite dans l’atelier et dans le tp afin d’obtenir une liste d’objets Voiture, Vous pouvez ensuite calculer la moyenne des prix de ces voitures.