

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Projet d'application Android de réservation pour R3st0.fr

Table des matières

Projet d'application Android de réservation pour R3st0.fr	1
Itération 1 - Affichage de la liste des restaurants	2
Création du projet :.....	2
Partie SQL :.....	6
Partie GITLAB :.....	11
Partie Android et API :.....	12
Test fonctionnel :.....	23
Bilan de l'itération :	25

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Itération 1 - Affichage de la liste des restaurants

User story 1 : « En tant qu'utilisateur, je souhaite pouvoir consulter la liste des restaurants (nom et ville) afin de pouvoir en sélectionner un ensuite. »

Date exigibilité : 17/11/2025 à 14H

Pour installer projet voir le *README* du dépôt ci-dessous (branche main)

Rappel du dépôt (les tickets se trouvent sur celui-ci uniquement) :
https://gitlab.com/AnaisPrt/p2_g7_2slam_ap_projet_android

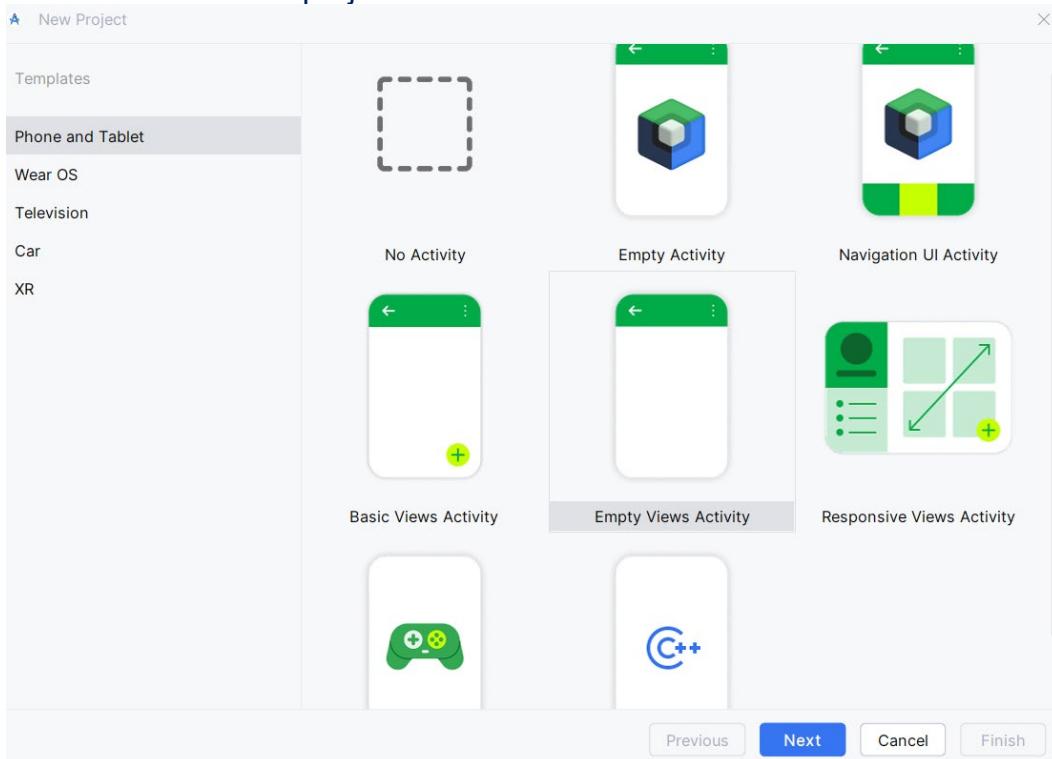
Lien de la branche Itération 1 : https://gitlab.com/AnaisPrt/p2_g7_2slam_ap_projet_android-/tree/Iteration1?ref_type=heads

Lien du dépôt GITLAB de l'API :

https://gitlab.com/Tom_Charbonneau/p2_g7_2slam_ap_projet_android_api-/tree/iteration1?ref_type=heads

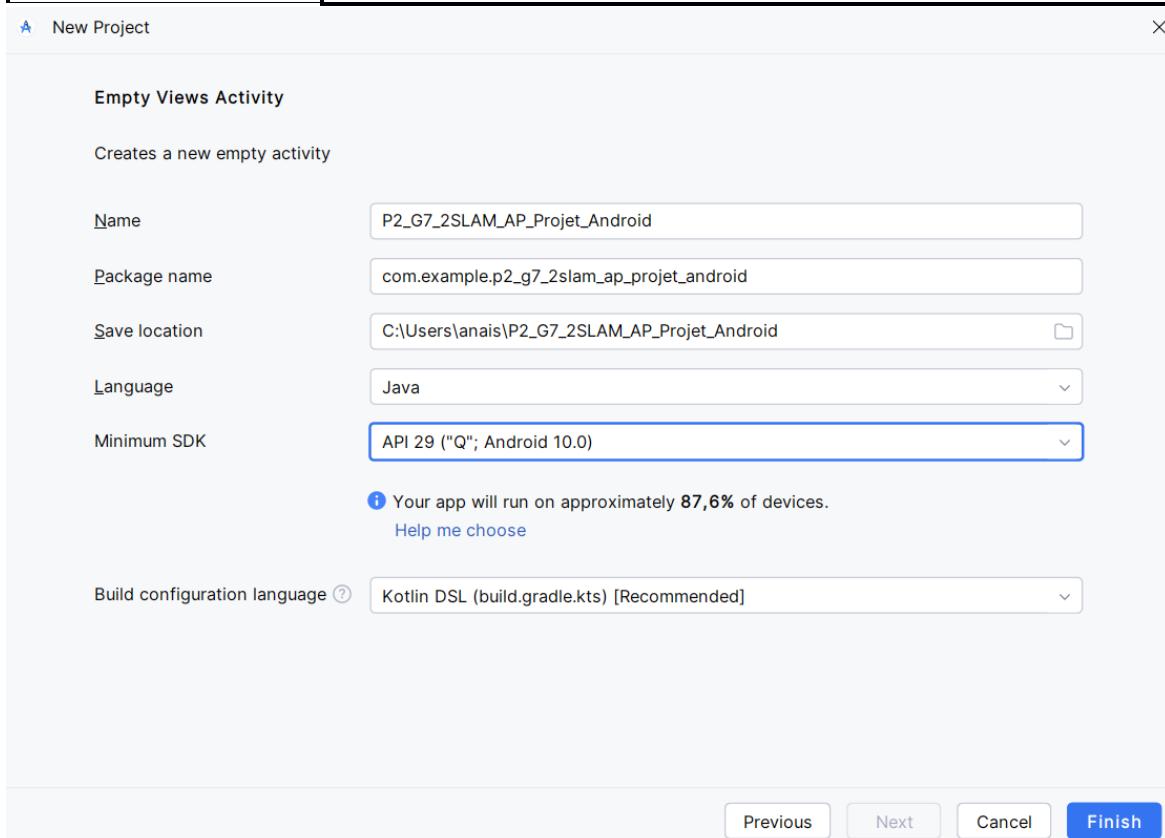
Création du projet :

On a d'abord créé le projet Android :

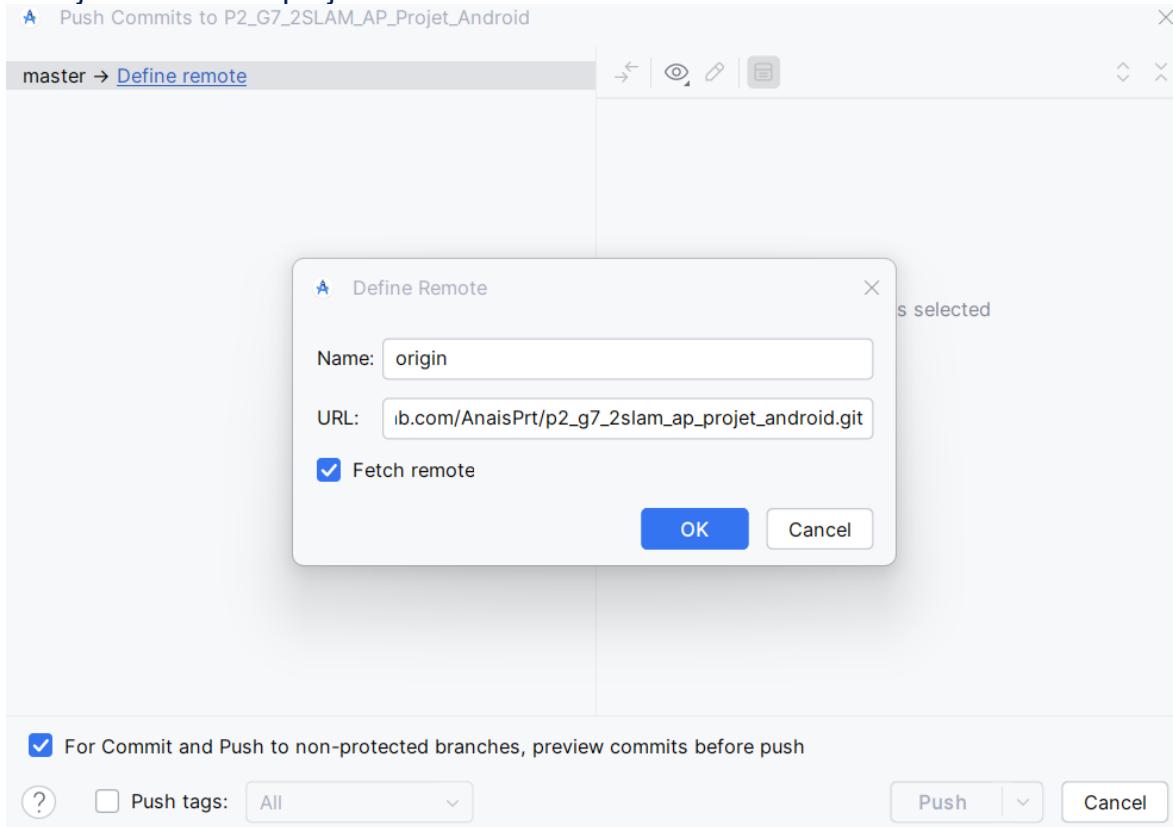


Puis j'ai nommé le projet et j'ai choisi la bonne version comme demandé :

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

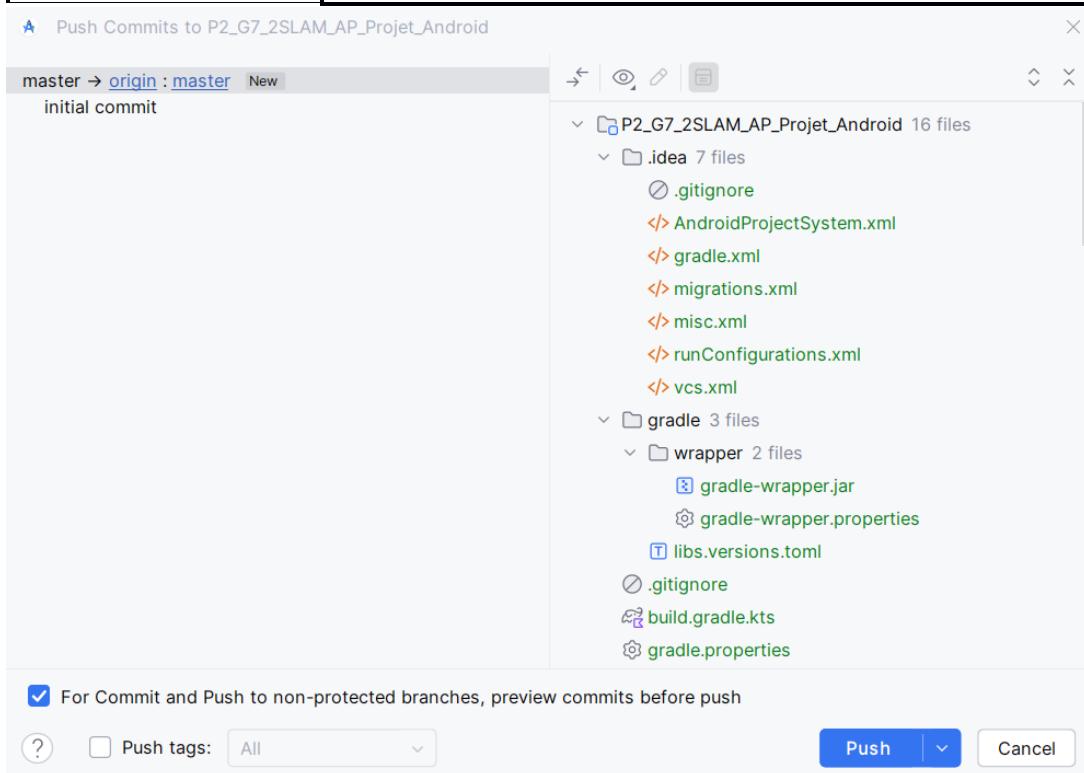


Puis j'ai initialisé le projet :

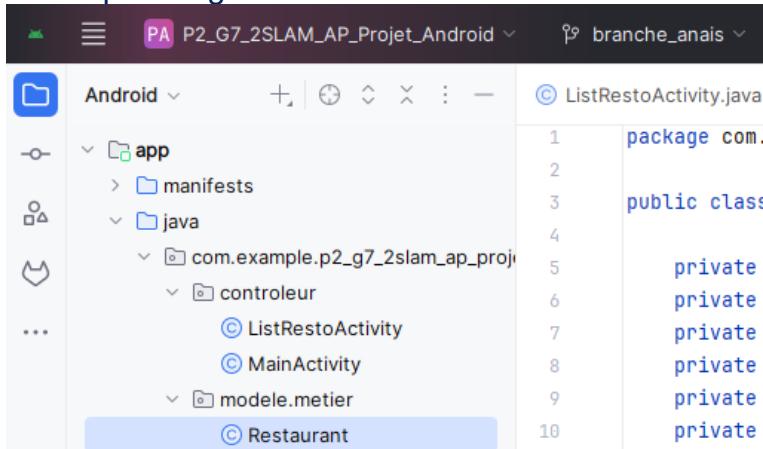


Puis j'ai poussé :

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025



On respecte également le Modèle Vue Contrôleur :



On peut voir que cela a bien réussi :

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Anaïs Portolleau / P2_G7_2SLAM_AP_Projet_Android / Dépôt

5217cb97a9e44451b2... ▾ p2_g7_2slam_ap_projet_android

⌚ p2_g7_2slam_ap_projet_android



initial commit

Anaïs Portolleau authored Il y a 17 heures

Nom	Dernière validation
📁 .idea	initial commit
📁 app	initial commit
📁 gradle	initial commit
❖ .gitignore	initial commit
⚡ build.gradle.kts	initial commit
⚡ gradle.properties	initial commit
⚡ gradlew	initial commit
☒ gradlew.bat	initial commit
⚡ settings.gradle.kts	initial commit

Et la création du dépôt GITLAB de l'API :

P **P2_G7_2SLAM_AP_Projet_Android_API** ⌚

main ▾ p2_g7_2slam_ap_projet_android_api

+ Rechercher un fichier Code :

⌚ Ajout d'un TODO Benjamin-DG44 authored Il y a 13 heures

c7deec1a Historique

Nom	Dernière validation	Dernière mise à jour
📁 .idea	Amélioration du code existant et du REA...	Il y a 16 heures
⚡ README.md	Amélioration du code existant et du REA...	Il y a 16 heures
📄 config.php	Amélioration du code existant et du REA...	Il y a 16 heures
📄 getAllResto.php	Ajout d'un TODO	Il y a 13 heures

Partie SQL :

On a repris la bdd du projet précédent :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
aimer	Parcourir Structure Rechercher Insérer Vider Supprimer	16	InnoDB utf8mb4_general_ci	32,0 kio	-	
critiquer	Parcourir Structure Rechercher Insérer Vider Supprimer	19	InnoDB utf8mb4_general_ci	32,0 kio	-	
photo	Parcourir Structure Rechercher Insérer Vider Supprimer	14	InnoDB utf8mb4_general_ci	32,0 kio	-	
proposer	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB utf8mb4_general_ci	32,0 kio	-	
resto	Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB utf8mb4_general_ci	16,0 kio	-	
typescuisines	Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB utf8mb4_general_ci	16,0 kio	-	
utilisateur	Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB utf8mb4_general_ci	32,0 kio	-	
7 tables	Somme	78	InnoDB utf8mb4_general_ci	192,0 kio	0 o	

Par la suite, j'ai créé une nouvelle table « RESERVATION » qui contient toutes les informations demandées par l'utilisateur dans l'itération 3 :

```
CREATE TABLE reservation (
    idReservation BIGINT(20) AUTO_INCREMENT,
    nomClient VARCHAR(50) NOT NULL,
    telClient VARCHAR(10) NOT NULL,
    dateReservation DATE NOT NULL,
    heureReservation TIME NOT NULL,
    nbPersonnes INT NOT NULL,
    codeReservation VARCHAR(12) UNIQUE,
    idU BIGINT(20) NOT NULL,
    idR BIGINT(20) NOT NULL,
    PRIMARY KEY (idReservation),
    CONSTRAINT FK_reservation_utilisateur FOREIGN KEY (idU) REFERENCES utilisateur(idU),
    CONSTRAINT FK_reservation_resto FOREIGN KEY (idR) REFERENCES resto(idR)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Serveur : MySQL:3306 » Base de données : resto2 » Table : reservation

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations Déclencheur

Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	idReservation 🔑	bigint			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	nomClient	varchar(50)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	telClient	varchar(10)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
4	dateReservation	date			Non	Aucun(e)			Modifier Supprimer Plus
5	heureReservation	time			Non	Aucun(e)			Modifier Supprimer Plus
6	nbPersonnes	int			Non	Aucun(e)			Modifier Supprimer Plus
7	codeReservation 🔑	varchar(12)	utf8mb4_0900_ai_ci		Non	Aucun(e)			Modifier Supprimer Plus
8	idU 🔑	bigint			Non	Aucun(e)			Modifier Supprimer Plus
9	idR 🔑	bigint			Non	Aucun(e)			Modifier Supprimer Plus

Tout cocher Avec la sélection : Parcourir Modifier Supprimer Primaire Unique Index Spatial Texte

Imprimer Déplacer des colonnes Normaliser

Ajouter 1 colonne(s) après idR Exécuter

Index

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer Renommer Supprimer	PRIMARY	BTREE	Oui	Non	idReservation	7	A	Non	
Éditer Renommer Supprimer	codeReservation	BTREE	Oui	Non	codeReservation	7	A	Non	
Éditer Renommer Supprimer	FK_reservation_utilisateur	BTREE	Non	Non	idU	3	A	Non	
Éditer Renommer Supprimer	FK_reservation_resto	BTREE	Non	Non	idR	7	A	Non	

De plus, je me suis rendu compte que la structure des tables PROPOSER et TYPESCUISINES n'étaient pas cohérentes avec le reste des tables de la BDD. En effet, le champ « id » est de type INT au lieu du type BIGINT. Pour ce faire, j'ai deux solutions : soit je peux supprimer la clé étrangère qui lie les deux « id » puis changer les types et recréer la clé étrangère. Soit, j'exporte la BDD, change les types des champs voulu dans le script SQL obtenu, puis je réimporte le fichier de la BDD. De ce fait, comme la BDD est légère et qu'il est plus rapide de l'exporter que de faire ce que j'ai énoncé plus tôt, je vais choisir la seconde option.

Après cela, j'ajoute les données manquante dans la table PROPOSER avec la requête SQL suivante :

```
INSERT INTO `proposer` (`idR`, `idTC`) VALUES
(1, 1), (2, 1), (3, 3), (4, 1), (4, 8), (4, 11), (5, 3), (6, 10), (7, 6), (8, 11), (9, 10), (10, 1), (11, 1), (11, 10);
```

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

✓ Affichage des lignes 0 - 13 (total de 14, traitement en 0,0009 seconde(s).)

SELECT * FROM `proposer`

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source]

Tout afficher

Nombre de lignes : 25

Filtrer les lignes:

Options supplémentaires

	Éditer	Copier	Supprimer	idR	idTC
<input type="checkbox"/>				1	1
<input type="checkbox"/>				2	1
<input type="checkbox"/>				4	1
<input type="checkbox"/>				10	1
<input type="checkbox"/>				11	1
<input type="checkbox"/>				3	3
<input type="checkbox"/>				5	3
<input type="checkbox"/>				7	6
<input type="checkbox"/>				4	8
<input type="checkbox"/>				6	10
<input type="checkbox"/>				9	10
<input type="checkbox"/>				11	10
<input type="checkbox"/>				4	11
<input type="checkbox"/>				8	11

Ensuite, j'ajoute quelques réservations fictives dont le code de réservation est formée de la façon suivante « RES-XXXXXXX » où X est un caractère alphanumérique.

```
INSERT INTO `reservation`
(`nomU`, `telU`, `dateReservation`, `heureReservation`, `nbPersonnes`, `codeReservation`,
`idU`, `idR`)
VALUES
-- Réservations pour l'utilisateur 3 (Mathieu Capliez / "pich")
('Mathieu', '0612345678', '2025-11-15', '19:30:00', 2, 'RES-A7F3C1D2', 3, 1),
('Mathieu', '0612345678', '2025-11-20', '12:30:00', 4, 'RES-A84MC1D7', 3, 4),

-- Réservations pour l'utilisateur 5 (Nicolas Harispe / "Nico40")
('Nicolas', '0676543210', '2025-11-18', '20:00:00', 3, 'RES-97F5KLK2', 5, 6),
('Nicolas', '0676543210', '2025-12-01', '13:00:00', 2, 'RES-Q0CB14D2', 5, 2),
('Nicolas', '0676543210', '2025-12-05', '19:45:00', 5, 'RES-A468GEF5', 5, 10),
```

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

-- Réservations pour l'utilisateur 6 (Testeur SIO)
('Testeur', '0659871234', '2025-11-14', '19:00:00', 1, 'RES-Z4FPI1D2', 6, 8),
('Testeur', '0659871234', '2025-11-21', '20:15:00', 2, 'RES-WHA3F40S', 6, 9);

✓ Affichage des lignes 0 - 6 (total de 7, traitement en 0,0004 seconde(s).)

`SELECT * FROM `reservation`;`

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

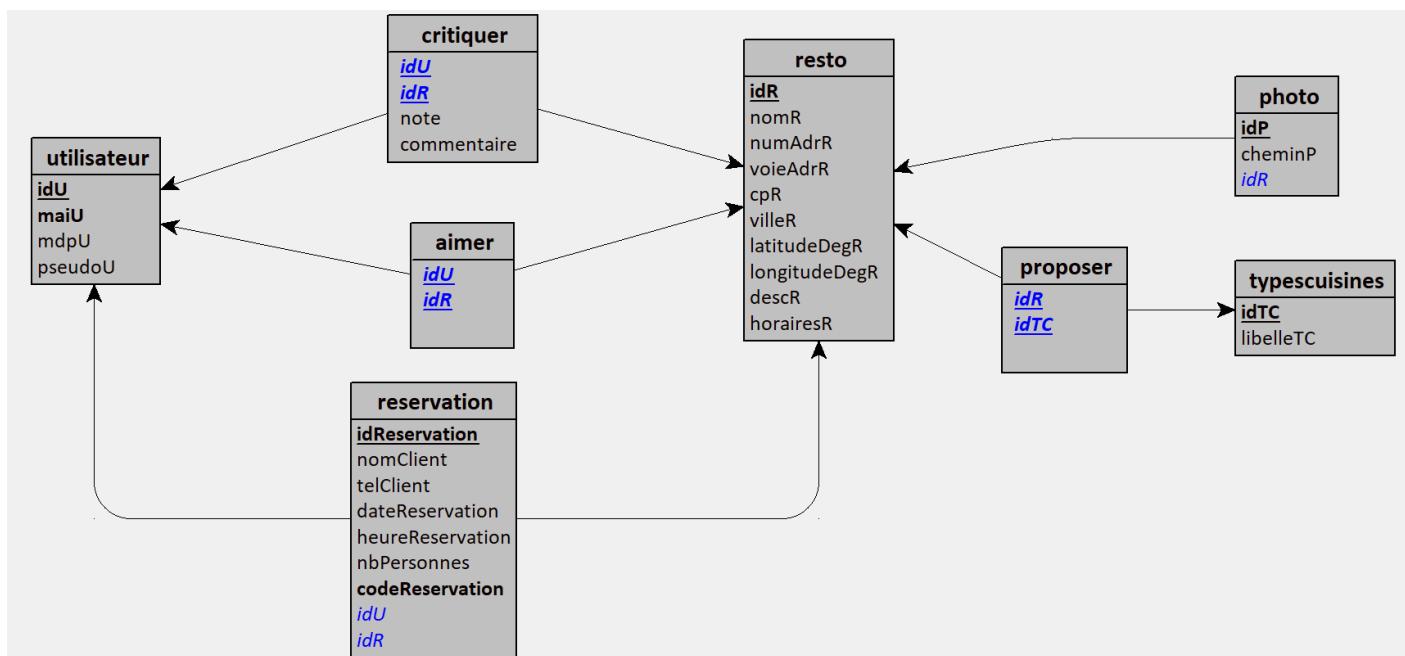
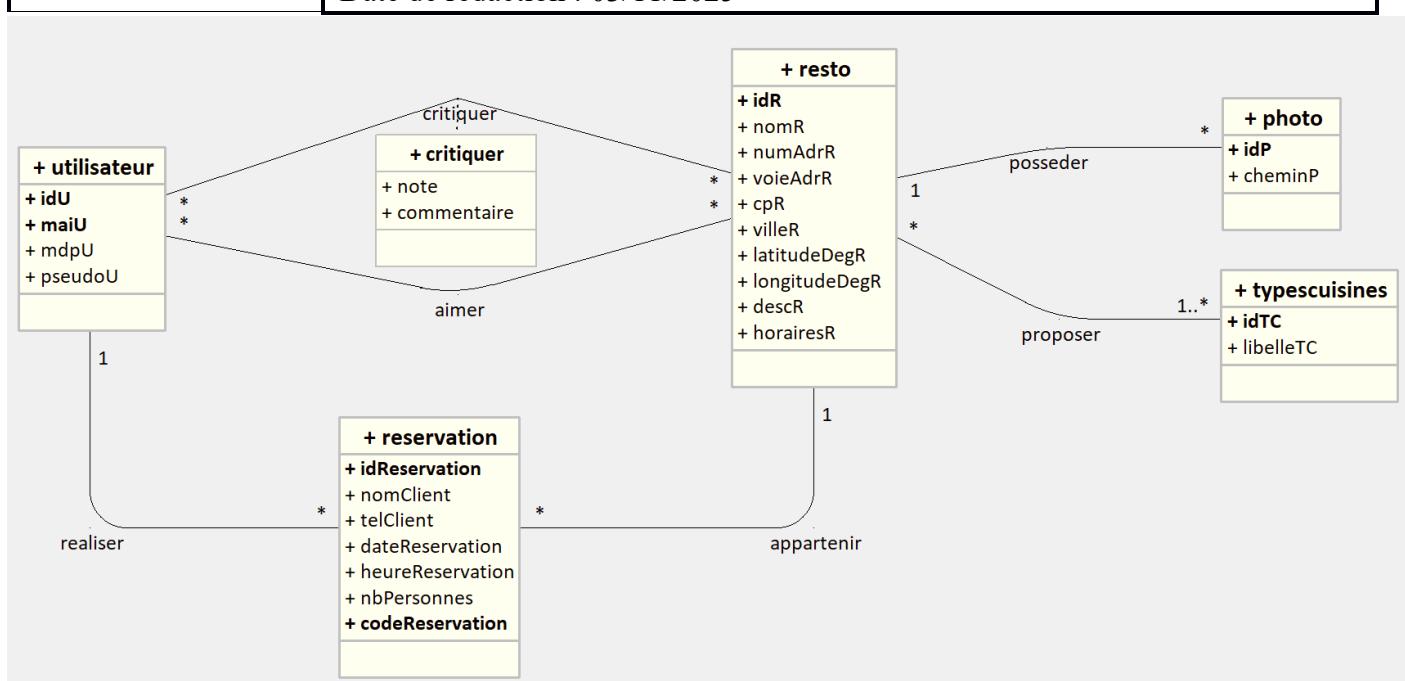
Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	Éditer	Copier	Supprimer	idReservation	nomU	telU	dateReservation	heureReservation	nbPersonnes	codeReservation	idUser	idR
<input type="checkbox"/>				1	Mathieu	0612345678	2025-11-15	19:30:00		2 RES-A7F3C1D2	3	1
<input type="checkbox"/>				2	Mathieu	0612345678	2025-11-20	12:30:00		4 RES-A84MC1D7	3	4
<input type="checkbox"/>				3	Nicolas	0676543210	2025-11-18	20:00:00		3 RES-97F5KLK2	5	6
<input type="checkbox"/>				4	Nicolas	0676543210	2025-12-01	13:00:00		2 RES-Q0CB14D2	5	2
<input type="checkbox"/>				5	Nicolas	0676543210	2025-12-05	19:45:00		5 RES-A468GEF5	5	10
<input type="checkbox"/>				6	Testeur	0659871234	2025-11-14	19:00:00		1 RES-Z4FPI1D2	6	8
<input type="checkbox"/>				7	Testeur	0659871234	2025-11-21	20:15:00		2 RES-WHA3F40S	6	9

```

    erDiagram
        utilisateur ||--o{ resto : "critiquer"
        utilisateur ||--o{ resto : "aimer"
        utilisateur ||--o{ reservation : "realiser"
        resto {
            o--o{ photo : "posseder"
            o--o{ typescuisines : "proposer"
        }
        resto }|--o{ reservation : "appartenir"
    }
  
```



2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Partie GITLAB :

Puis la création des Jalons pour les 3 itérations :

Anais Portolleau / P2_G7_2SLAM_AP_Projet_Android / Jalons

Ouvert 3 Fermé 0 Tous 3

Filtrer par nom de jalon Échéance proche Nouveau jalon

Itération	Créé par	Début	Fin	Statut	Progression	Actions
Itération 1	Anais Portolleau / P2_G7_2SLAM_AP_Projet_Android	nov. 3, 2025	nov. 17, 2025	Ouvert	0/0 terminé 0%	⋮
Itération 2	Anais Portolleau / P2_G7_2SLAM_AP_Projet_Android	nov. 17, 2025	déc. 1, 2025	À venir	0/0 terminé 0%	⋮
Itération 3	Anais Portolleau / P2_G7_2SLAM_AP_Projet_Android	déc. 1, 2025	déc. 15, 2025	À venir	0/0 terminé 0%	⋮

Ainsi que la création des tickets et de l'attribution à des jalons:

Tickets

Ouvert 3 Fermé 0 Tous 3

Modification en masse New item ⋮

Rechercher ou filtrer les résultats... ⚡ Date de création ⌂

- Creation des vues · AnaisPrt/p2_g7_2slam_ap_projet_android#3 · créé à l'instant par Anais Portolleau · Itération 1 · en cours
- Modification de la BDD existante · AnaisPrt/p2_g7_2slam_ap_projet_android#2 · créé Il y a 3 minutes par Anais Portolleau · Itération 1 · en cours
- Initialisation du projet · AnaisPrt/p2_g7_2slam_ap_projet_android#1 · créé Il y a 4 minutes par Anais Portolleau · Itération 1 · achevées

Et la création des tickets pour l'itération 1 :

Itération 1 - Ticket 8 - Tests Fonctionnels	AnaisPrt/p2_g7_2slam_ap_projet_android#23 · créé Il y a 1 jour par Anais Portolleau	Fermé	fermé à l'instant
Itération 1 - Ticket 1 - API : Récupération des données	AnaisPrt/p2_g7_2slam_ap_projet_android#12 · créé Il y a 6 jours par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 2 — Modèle : Création de la classe Restaurant	AnaisPrt/p2_g7_2slam_ap_projet_android#11 · créé Il y a 6 jours par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 3 — UI : Affichage des données la liste	AnaisPrt/p2_g7_2slam_ap_projet_android#10 · créé Il y a 6 jours par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 4 — Interaction : Sélection d'un restaurant	AnaisPrt/p2_g7_2slam_ap_projet_android#9 · créé Il y a 6 jours par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 5 — Configuration : Sécurité réseau	AnaisPrt/p2_g7_2slam_ap_projet_android#5 · créé Il y a 1 semaine par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 7 - Création - Diagramme UML	AnaisPrt/p2_g7_2slam_ap_projet_android#4 · créé Il y a 1 semaine par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 2-2 - Crédit des vues	AnaisPrt/p2_g7_2slam_ap_projet_android#3 · créé Il y a 1 semaine par Anais Portolleau	Fermé	fermé Il y a 11 heures
Itération 1 - Ticket 6 - Modification de la BDD existante	AnaisPrt/p2_g7_2slam_ap_projet_android#2 · créé Il y a 1 semaine par Anais Portolleau	Fermé	fermé Il y a 23 minutes
Itération 1 - Ticket 0 - Initialisation du projet	AnaisPrt/p2_g7_2slam_ap_projet_android#1 · créé Il y a 1 semaine par Anais Portolleau	Fermé	fermé Il y a 11 heures

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Partie Android et API :

On a créé un dépôt GITLAB qui permet de stocker les fichiers de l'API, puis on le clone dans le www (git clone https://gitlab.com/Tom_Charbonneau/p2_g7_2slam_ap_projet_android_api.git) :

Tom Charbonneau / projet_android_G7_API / Dépôt

main ▾ projet_android_g7_api

Projet Android G7 API

éviter d'avoir un sous dossier

Anais Portolleau authored à l'instant

+ Rechercher un fichier Code :

Nom	Dernière validation	Dernière mise à jour
README.md	Initial commit	Il y a 20 heures
config.php	éviter d'avoir un sous dossier	Il y a 1 minute
getAllResto.php	éviter d'avoir un sous dossier second fichier	Il y a 51 secondes

Et la commande a effectué dans le terminal :

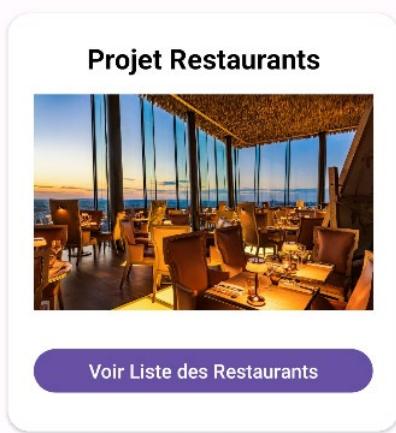
`git clone https://gitlab.com/Tom_Charbonneau/p2_g7_2slam_ap_projet_android_api.git`

```
anaïs@Port-Anaïs MINGW64 /c/wamp64/www
$ git clone https://gitlab.com/Tom_Charbonneau/projet_android_g7_api.git
Cloning into 'projet_android_g7_api'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 15 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 4.14 KiB | 605.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.

anaïs@Port-Anaïs MINGW64 /c/wamp64/www
$
```

PS : entre temps, l'URL à changer, donc veuillez prendre en compte l'url du dépôt présent dans la commande plutôt que dans la capture d'écran.

'abord on a commencé à faire les vues. J'ai voulu faire une page principale qui puisse pouvoir choisir et également une autre page pour voir la liste des restaurants :



Il fallait rajouter aussi dans le Manifest le ListResto:

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

```

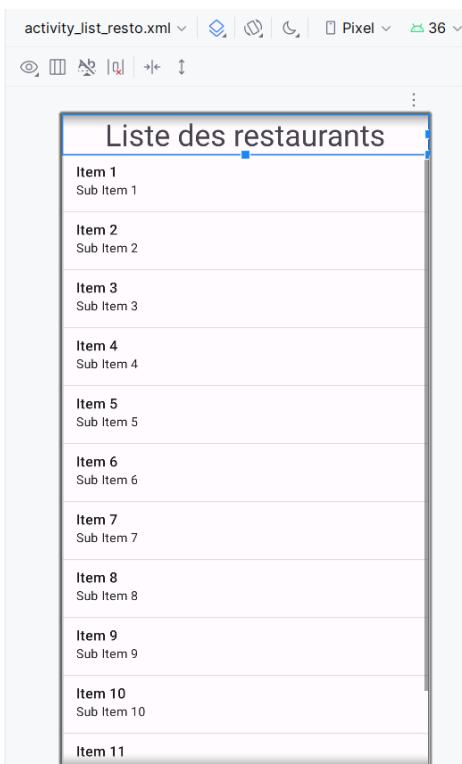
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity android:name=".ListRestoActivity" />

```

Puis j'ai fait le second affichage pour afficher la liste des restaurants :



Ainsi la création de la classe Restaurant :

```

ListRestoActivity.java   activity_list_resto.xml   Restaurant.java ×
1 package com.example.p2_g7_2slam_ap_projet_android.modele.met
2
3 import java.util.ArrayList;
4
5 public class Restaurant { 5 usages
6
7     private int idR; 3 usages
8     private String nomR; 5 usages
9     private String numAddrR; 3 usages
10    private String voieAddrR; 3 usages
11    private String cpR; 3 usages
12    private String villeR; 5 usages
13    private float latitudeDegR; 3 usages
14    private float longitudeDegR; 3 usages
15    private String descR; 3 usages
16    private String horairesR; 3 usages
17    private ArrayList<String> typesCuisines; 4 usages
18

```

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Le constructeur ci-dessous :

```

 /**
 * Constructeur de la classe Restaurant
 * @param idR
 * @param nomR
 * @param numAddr
 * @param voieAddrR
 * @param cpR
 * @param villeR
 * @param latitudeDegR
 * @param longitudeDegR
 * @param descR
 * @param horairesR
 * @param typesCuisines
 */

public Restaurant(int idR, String nomR, String numAddr, String voieAddrR, String cpR, String villeR, float latitudeDegR, float longitudeDegR, String descR, String horairesR, ArrayList<String> typesCuisines) {
    this.idR = idR;
    this.nomR = nomR;
    this.numAddr = numAddr;
    this.voieAddrR = voieAddrR;
    this.cpR = cpR;
    this.villeR = villeR;
    this.latitudeDegR = latitudeDegR;
    this.longitudeDegR = longitudeDegR;
    this.descR = descR;
    this.horairesR = horairesR;
    this.typesCuisines = typesCuisines;
}


```

Avec les accesseurs et les mutateurs de la classe Restaurant (une partie mais tous les attributs ont un accesseur et mutateur:

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

```

public String getNomR() { no usages
    return nomR;
}

public void setNomR(String nomR) { no usages
    this.nomR = nomR;
}

public String getNumAddrR() { no usages
    return numAddrR;
}

public void setNumAddrR(String numAddrR) { no usages
    this.numAddrR = numAddrR;
}

public String getVoieAddrR() { no usages
    return voieAddrR;
}

public void setVoieAddrR(String voieAddrR) { no usages
    this.voieAddrR = voieAddrR;
}

public String getCpR() { no usages
    return cpR;
}

```

Ainsi que le `toString()` pour afficher l'état :

```

@Override
public String toString() {
    return nomR + " - " + villeR; // ce qui s'affichera dans la ListView
}

```

J'ai également ajouter ensuite le lien vers `ListRestoActivity`. Ce code définit une activité Android qui affiche une liste de restaurants. `ListRestoActivity` étend `AppCompatActivity`, déclare une `ListView` pour l'affichage et un `ArrayList<Restaurant>` pour stocker les données. Un client HTTP (`OkHttpClient`) est préparé pour récupérer le JSON depuis le serveur. Dans `onCreate`, la vue est initialisée avec `setContentView`, la `ListView` est liée à son identifiant et la méthode `chargerLesRestaurants()` est appelée pour lancer la récupération et l'affichage des restaurants.

Puis ensuite on a fait une méthode `chargerLesRestaurants()` qui va récupérer directement le dépôt distant qui est sur gitlab (https://gitlab.com/Tom_Charbonneau/projet_android_g7_api.git).

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom Date de rédaction : 03/11/2025

```
1 package com.example.p2_g7_2slam_ap_projet_android.controleur;
2
3 > import ...
4
5 ><public class ListRestoActivity extends AppCompatActivity {
6
7     private ListView listResto; 3 usages
8     private ArrayList<Restaurant> lesRestos = new ArrayList<>(); 4 usages
9     private OkHttpClient httpClient = new OkHttpClient(); 1 usage
10    private static final String TAG = "ListRestoActivity"; 4 usages
11
12    @Override
13    protected void onCreate(Bundle savedInstanceState) {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.activity_list_resto);
16        listResto = findViewById(R.id.listResto);
17        chargerLesRestaurants();
18    }
19}
```

Cette partie définit la méthode chargerLesRestaurants() qui envoie une requête HTTP pour récupérer la liste des restaurants depuis le serveur. Un objet Request est créé avec l'URL de l'API getAllResto.php. Ensuite, httpClient.newCall(request).enqueue(new Callback()) lance la requête de manière asynchrone pour ne pas bloquer le thread principal. La méthode onFailure gère les erreurs de connexion en affichant un message et un log, tandis que onResponse gère la réponse du serveur. Si la réponse n'est pas réussie (!response.isSuccessful()), elle affiche un message d'erreur. Sinon, le corps de la réponse est lu sous forme de chaîne (response.body().string()) et loggé pour le débogage.

```
private void chargerLesRestaurants() { 1 usage
41     Request request = new Request.Builder()
42         .url("http://10.0.2.2/p2_g7_2slam_ap_projet_android_api/getAllResto.php")
43         .build();
44
45     httpClient.newCall(request).enqueue(new Callback() {
46         @Override
47         public void onFailure(@NotNull Call call, @NotNull IOException e) {
48             Log.e(TAG, msg: "Échec de la requête getAllResto.php", e);
49             runOnUiThread(() ->
50                 Toast.makeText(context: ListRestoActivity.this, text: "Erreur connexion au serveur",
51                               Toast.LENGTH_LONG).show()
52             );
53         }
54
55         @Override
56         public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {
57             if (!response.isSuccessful()) {
58                 Log.e(TAG, msg: "Réponse serveur non réussie : " + response.code());
59                 runOnUiThread(() ->
60                     Toast.makeText(context: ListRestoActivity.this, text: "Réponse serveur échouée : " +
61                                   response.code(), Toast.LENGTH_LONG).show()
62                 );
63             }
64             return;
65         }
66
67         final String myResponse = response.body().string();
68         Log.d(TAG, msg: "JSON reçu : " + myResponse);
69     });
70 }
```

Cette partie s'exécute sur le thread principal grâce à runOnUiThread() pour pouvoir mettre à jour l'interface. Elle transforme la réponse JSON (myResponse) en objet JSONObject et récupère le tableau "data" contenant les restaurants. La liste lesRestos est vidée pour recevoir les nouveaux

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

éléments. Pour chaque restaurant, on récupère le nom, la ville et le tableau "typesCuisines", qu'on parcourt pour stocker les types de cuisine dans typesList. Ensuite, un objet Restaurant est créé avec ces données et ajouté à lesRestos. Enfin, un ArrayAdapter est créé pour afficher la liste dans le ListView. En cas d'erreur de parsing JSON, un log et un Toast signalent le problème.

```

69         runOnUiThread(() -> {
70             try {
71                 JSONObject jsonObject = new JSONObject(myResponse);
72                 JSONArray jsonArray = jsonObject.getJSONArray("data");
73                 lesRestos.clear();
74
75                 for (int i = 0; i < jsonArray.length(); i++) {
76                     JSONObject jsonResto = jsonArray.getJSONObject(i);
77                     String nomR = jsonResto.getString("nomR");
78                     String villeR = jsonResto.getString("villeR");
79
80                     // Récupérer les types de cuisines
81                     ArrayList<String> typesList = new ArrayList<>();
82                     JSONArray jsonTypes = jsonResto.getJSONArray("typesCuisines");
83                     for (int j = 0; j < jsonTypes.length(); j++) {
84                         typesList.add(jsonTypes.getJSONObject(j).getString("libelleTC"));
85                     }
86
87                     lesRestos.add(new Restaurant(nomR, villeR, typesList));
88                 }
89
90                 ArrayAdapter<Restaurant> adapter = new ArrayAdapter<>(
91                     context: ListRestoActivity.this,
92                     android.R.layout.simple_list_item_1,
93                     lesRestos
94                 );
95                 listResto.setAdapter(adapter);
96
97             } catch (JSONException e) {
98                 Log.e(TAG, msg: "Erreur parsing JSON", e);
99                 Toast.makeText(context: ListRestoActivity.this, text: "Erreur parsing JSON",
100                             Toast.LENGTH_LONG).show();
101             }
102         });
    
```

Cette partie configure le clic sur un élément de la liste. Quand l'utilisateur touche un restaurant, l'objet Restaurant correspondant est récupéré depuis lesRestos grâce à sa position. Un Toast s'affiche alors pour montrer le nom du restaurant et la liste de ses types de cuisines. Cela permet de tester rapidement que les données sont correctement récupérées et associées à chaque restaurant.

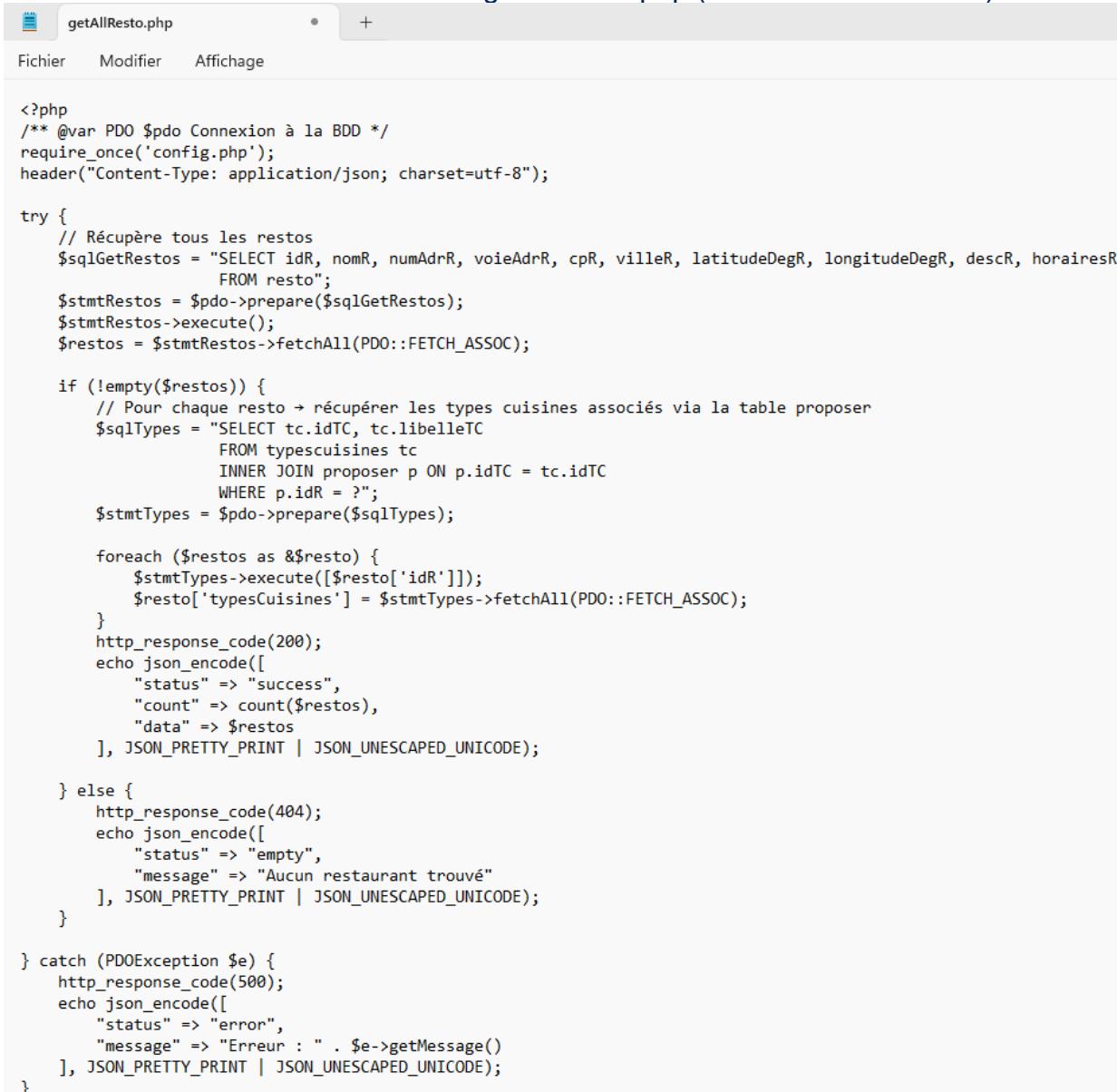
2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

```

103
104         listResto.setOnItemClickListener(( AdapterView<?> parent, View view, int position, long id) -> {
105             Restaurant resto = lesRestos.get(position);
106             Toast.makeText( context: ListRestoActivity.this,
107                 text: "Sélectionné : " + resto.getNomR() + "\nTypes : " + resto.getTypesCuisines(),
108                 Toast.LENGTH_SHORT).show();
109         });
110     });
111 }
112 }
113 }
114

```

Il va chercher ce fichier PHP nommé getAllResto.php (à mettre dans le www) :



```

getAllResto.php

Fichier Modifier Affichage

<?php
/** @var PDO $pdo Connexion à la BDD */
require_once('config.php');
header("Content-Type: application/json; charset=utf-8");

try {
    // Récupère tous les restos
    $sqlGetRestos = "SELECT idR, nomR, numAddr, voieAddr, cpR, villeR, latitudeDegR, longitudeDegR, descR, horairesR
                    FROM resto";
    $stmtRestos = $pdo->prepare($sqlGetRestos);
    $stmtRestos->execute();
    $restos = $stmtRestos->fetchAll(PDO::FETCH_ASSOC);

    if (!empty($restos)) {
        // Pour chaque resto → récupérer les types cuisines associés via la table proposer
        $sqlTypes = "SELECT tc.idTC, tc.libelleTC
                     FROM typescuisines tc
                     INNER JOIN proposer p ON p.idTC = tc.idTC
                     WHERE p.idR = ?";
        $stmtTypes = $pdo->prepare($sqlTypes);

        foreach ($restos as &$resto) {
            $stmtTypes->execute([$resto['idR']]);
            $resto['typesCuisines'] = $stmtTypes->fetchAll(PDO::FETCH_ASSOC);
        }
        http_response_code(200);
        echo json_encode([
            "status" => "success",
            "count" => count($restos),
            "data" => $restos
        ], JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
    } else {
        http_response_code(404);
        echo json_encode([
            "status" => "empty",
            "message" => "Aucun restaurant trouvé"
        ], JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
    }
} catch (PDOException $e) {
    http_response_code(500);
    echo json_encode([
        "status" => "error",
        "message" => "Erreur : " . $e->getMessage()
    ], JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
}

```

Le fichier config.php est inclus pour récupérer la connexion PDO. Le header indique que la réponse sera du JSON UTF-8. Le bloc try protège l'exécution pour capturer les erreurs SQL. Une requête récupère tous les restaurants, puis elle est préparée et exécutée. Les résultats sont stockés sous forme de tableau associatif. Si des restaurants existent, une seconde requête est préparée pour récupérer les types de cuisine liés à chaque restaurant via la table proposer. Pour chaque

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

restaurant, cette requête est exécutée avec son idR et le résultat est ajouté dans un tableau typesCuisines. Ensuite, un code HTTP 200 est renvoyé avec un JSON contenant le statut, le nombre de restaurants et les données complètes. Si aucun restaurant n'est trouvé, un code 404 est renvoyé avec un message JSON indiquant que la liste est vide. Enfin, si une erreur PDO survient, un code 500 est renvoyé avec le message d'erreur en JSON.

Et quand on teste dans POSTMAN :

```

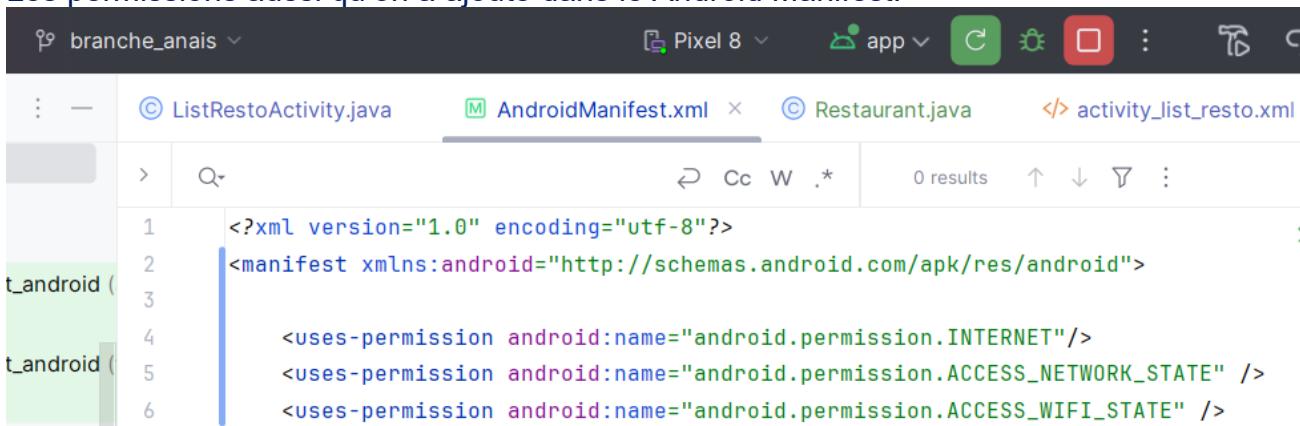
{
  "data": [
    {
      "idR": 1,
      "nomR": "l'entrepose",
      "numAdrR": "2",
      "voieAdrR": "rue Maurice Ravel",
      "cpR": "33000",
      "villeR": "Bordeaux",
      "latitudeDegR": 44.7948,
      "longitudeDegR": -0.58754,
      "descR": "description",
      "horairesR": "<table>\n      <thead>\n        <tr>\n          <th>Ouverture</th><th>Semaine</th><th>Week-end</th>\n        </tr>\n      </thead>\n      <tbody>\n        <tr>\n          <td>08h00-22h00</td>\n          <td>Du lundi au vendredi de 08h00 à 22h00</td>\n          <td>Le samedi de 09h00 à 21h00</td>\n        </tr>\n      </tbody>\n    </table>",
      "typesCuisines": [
        {
          "idTC": 1,
          "libelleTC": "sud ouest"
        }
      ]
    },
    {
      "idR": 2,
      "nomR": "le bar du charcutier",
      "numAdrR": "30",
      "voieAdrR": "rue Parlement Sainte-Catherine",
      "cpR": "33000",
      "villeR": "Bordeaux"
    }
  ]
}
  
```

Puis il fallait ajouter les dépendances dépendance dans le build.gradle.kts :

```

implementation("com.squareup.okhttp3:okhttp:4.12.0")
implementation(libs.constraintlayout)
  
```

Les permissions aussi qu'on a ajouté dans le Android Manifest:



2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

```

2   <manifest xmlns:android="http://schemas.android.com/apk/res/android">
5     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
6     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
7
8     <application
9       android:allowBackup="true"
10      android:usesCleartextTraffic="true"
11      android:dataExtractionRules="@xml/data_extraction_rules"
12      android:fullBackupContent="@xml/backup_rules"
13      android:icon="@mipmap/ic_launcher"
14      android:roundIcon="@mipmap/ic_launcher_round"
15      android:supportsRtl="true"
16      android:theme="@style/Theme.P2_G7_2SLAM_AP_Projet_Android">
17
18       <activity
19         android:name=".MainActivity"
20         android:exported="true">
21         <intent-filter>
22           <action android:name="android.intent.action.MAIN" />
23           <category android:name="android.intent.category.LAUNCHER" />
24         </intent-filter>
25       </activity>
26
27       <activity android:name=".ListRestoActivity" />
28
29     </application>
30   </manifest>

```

Et j'ai ajouté cette ligne de code pour autoriser les connexions http (android:usesCleartextTraffic="true") comme on peut le voir ci-dessous :

```

<application
    android:allowBackup="true"
    android:usesCleartextTraffic="true"

```

Egalement on devait rajouter okhttp pour les requêtes http:

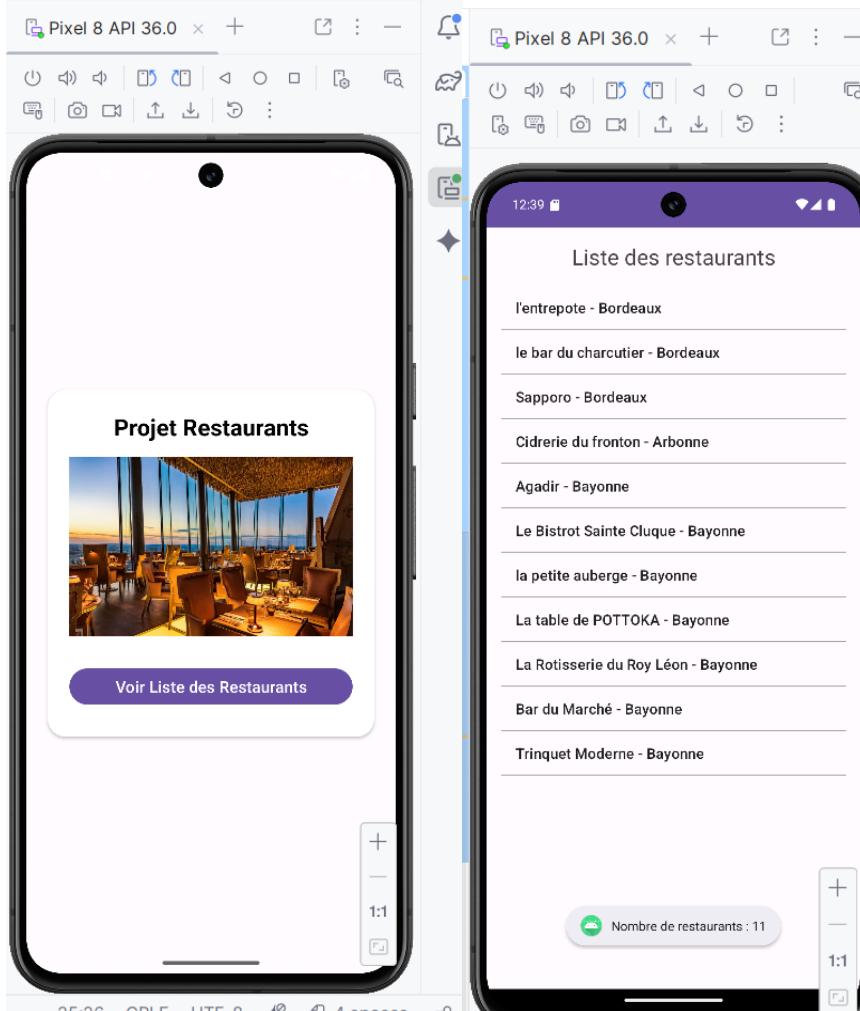
```

dependencies {
    // OkHttp (librairie pour requêtes HTTP)
    implementation("com.squareup.okhttp3:okhttp:5.3.0")
    // AndroidX / UI
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    // Tests
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}

```

Puis lorsqu'on teste cela marche bien, on arrive bien sur la première page puis au clic sur " Voir Liste des Restaurants" cela affiche bien le nom et la ville comme on peut le voir ci-dessous :

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025



Puis ajout du clic mais cela n'est pas fonctionnelle car on ne nous demandait pas d'afficher les détails du restaurant pour l'instant (j'ai juste mis un Toast pour vérifier qu'il arrivait bien à récupérer le nom du restaurant sélectionné ainsi que le type de cuisine) :

```
listResto.setOnItemClickListener(( AdapterView<?> parent, View view, int position, long id) -> {
    Restaurant resto = lesRestos.get(position);
    Toast.makeText(context: ListRestoActivity.this,
        text: "Sélectionné : " + resto.getNomR() + "\nTypes : " + resto.getTypesCuisines(),
        Toast.LENGTH_SHORT).show();
});
```

Affichage pour vérifier marche bien comme on peut le voir ci-dessous :



Ici on peut voir que le type de cuisine est bien affiché pour le restaurant La rotisserie du Roy Léon et si on vérifie dans la bdd

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Serveur : MySQL:3306 » Base de données : resto2 » Table : type

Parcourir Structure SQL Rechercher Insérer

Exécuter une ou des requêtes SQL sur la table « resto2.typescuisines »:

```

1 SELECT r.nomR, tc.libelleTC
2 FROM resto r
3 INNER JOIN proposer p ON r.idR = p.idR
4 INNER JOIN typescuisines tc ON p.idTC = tc.idTC
5 WHERE r.nomR = 'La Rotisserie du Roy Léon';
6

```

Et quand on exécute :

Affichage des lignes 0 - 0 (total de 1, traitement en cours)

```
SELECT r.nomR, tc.libelleTC FROM resto r INNER JOIN proposer p ON r.idR = p.idR INNER JOIN typescuisines tc ON p.idTC = tc.idTC WHERE r.nomR = 'La Rotisserie du Roy Léon';
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL]

Tout afficher | Nombre de lignes : 25

Options supplémentaires

nomR	libelleTC
La Rotisserie du Roy Léon	viande

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

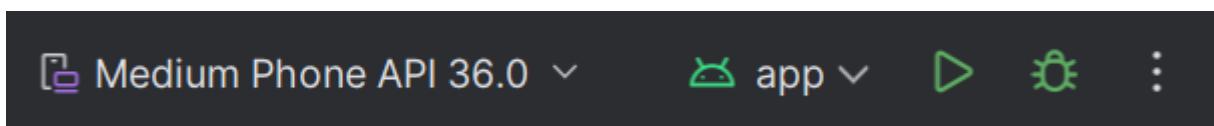
Test fonctionnel :

1) Affichage de la liste des restaurants

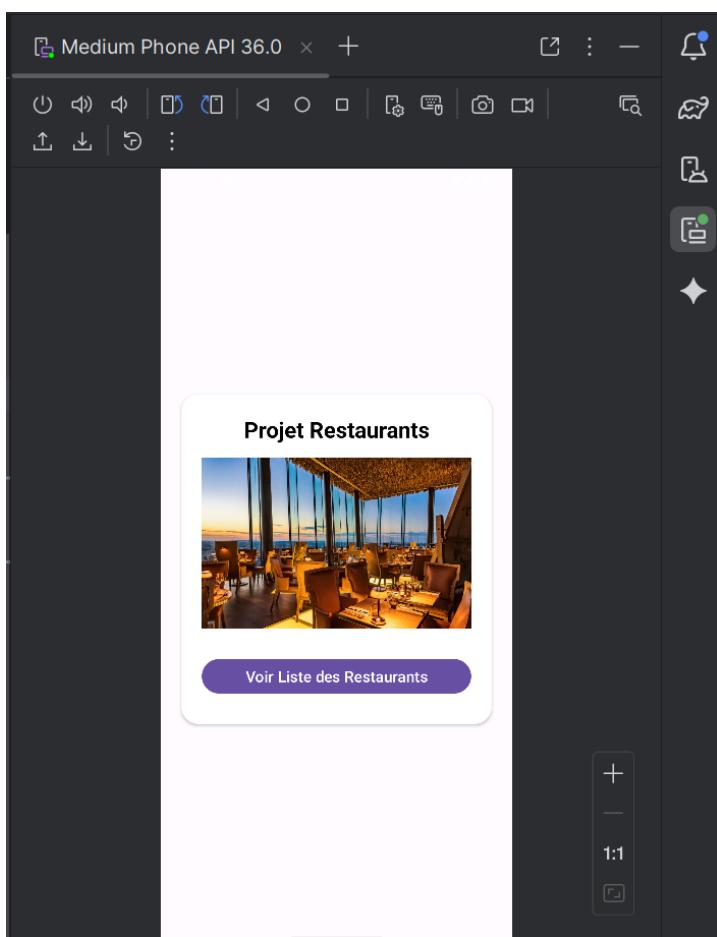
Objectif : Vérifier la liaison entre l'API et l'application.

Procédure :

- a) Lancer le projet sur Android Studio en cliquant sur le bouton pause



- b) Attendre que l'application se build et qu'elle s'affiche par elle-même sur votre écran. Si elle ne s'affiche pas mais que le projet est bien démarré, appuyer sur le bouton « Running Devices » à droite. Vous êtes censé obtenir le résultat suivant :



2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

- c) Cliquer sur le bouton « Voir Liste des Restaurants ». Normalement, vous arrivez sur l'écran suivant qui affiche tous les restaurants envoyés par l'API et vous voyez un message en bas de l'écran avec le nombre de restaurants affichés.



- d) En cas d'échec, merci de vous référez au README présent sur le GITLAB du projet Android. Il explique comment installer le projet proprement.

2025/2026	AP – Projet 2 - Android
BTS SIO 2SLAM	Auteur : PORTOLLEAU Anaïs & DELAUNAY-GUITTON Benjamin & CHARBONNEAU Tom
	Date de rédaction : 03/11/2025

Bilan de l'itération :

Nous avons rencontré des problèmes au niveau de l'API et de la liaison entre la base de données et l'appli Android. Mais aussi en termes de code notamment lors de la création de la BDD lorsque la table RESERVATION avait été créé, on était tous d'accord sur la structure. Puis quelques temps plus tard on se rendait compte qu'il fallait la modifier pour plus de cohérence.

Rappel du dépôt : https://gitlab.com/AnaisPrt/p2_g7_2slam_ap_projet_android

Lien de la branche Itération 1 : https://gitlab.com/AnaisPrt/p2_g7_2slam_ap_projet_android-/tree/d055522ee0795667b86b6e439a5cf12fedd538a0/

Lien du dépôt GITLAB de l'API : https://gitlab.com/Tom_Charbonneau/projet_android_g7_api-/tree/main?ref_type=heads