

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Compte-rendu itération 1

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

Pour réaliser ce projet, Davy est en charge la partie SQL et PHP, Lisa de la partie interface utilisateur de l'application, et Pierre s'occupe de la partie Android Studio.

Notre objectif est de finir l'itération 1 en 2 séances de 2 heures.

Dans cette première séance, l'objectif est de réaliser l'API, de commencer les interfaces utilisateur et d'afficher les restaurants dans une liste.

Dans la deuxième séance l'objectif est de finir l'affichage des restaurants dans la liste.

Partie PHP (Davy) :

Dans un premier temps, il faut importer la base de données comportant les données nécessaires à l'application ou utiliser une base de données précédemment créée.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> aimer	★ Parcourir Structure Rechercher Insérer Vider Supprimer	16	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> critiquer	★ Parcourir Structure Rechercher Insérer Vider Supprimer	19	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> photo	★ Parcourir Structure Rechercher Insérer Vider Supprimer	14	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> resto	★ Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> restotype	★ Parcourir Structure Rechercher Insérer Vider Supprimer	14	InnoDB	utf8mb4_unicode_ci	32,0 kio	-
<input type="checkbox"/> typecuisine	★ Parcourir Structure Rechercher Insérer Vider Supprimer	11	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/> utilisateur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8mb4_general_ci	32,0 kio	-

Pour faire l'API utilisé par l'application, il faut faire un programme qui permet de récupérer les informations de la base de donnée, à l'aide de requêtes SQL.

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

```
<?php
try {
// Connexion à la base de données MySQL
$db="resto2";
$dbhost="localhost";
$dbport=3306;
$dbuser="root";
$dbpasswd="joliverie";

$connexion = new PDO('mysql:host='.$dbhost.';port='.$dbport.';dbname='.$db.'',
$dbuser, $dbpasswd);
$connexion->exec("SET CHARACTER SET utf8");

    $reponse=$connexion->prepare("SELECT nomR, villeR FROM resto;");
    $reponse->execute();
    $datas = array();

    while($res=$reponse->fetch(PDO::FETCH_ASSOC)) {
        $datas['restos'][]=$res;
    }

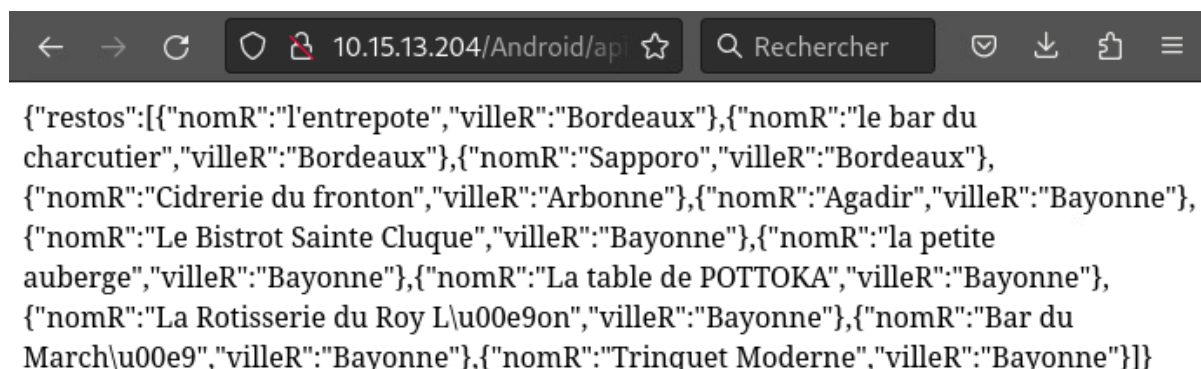
    echo json_encode($datas);
}
catch (Exception $e) {
die('Erreur : ' . $e->getMessage());
}
?>
```

Pour que l'API puisse être utilisé, il faut l'ajouter au fichier *www/Android/apiAppResto*

Enfin, pour vérifier que l'application fonctionne, il faut chercher sur un navigateur de recherche le lien de l'API

<http://10.15.13.204:80/Android/apiAppResto/getAllRestoJSON.php>

Nous obtenons ce résultat, ce qui signifie que l'API fonctionne correctement.



```
{
  "restos": [
    {
      "nomR": "l'entrepote",
      "villeR": "Bordeaux"
    },
    {
      "nomR": "le bar du charcutier",
      "villeR": "Bordeaux"
    },
    {
      "nomR": "Sapporo",
      "villeR": "Bordeaux"
    },
    {
      "nomR": "Cidrerie du fronton",
      "villeR": "Arbonne"
    },
    {
      "nomR": "Agadir",
      "villeR": "Bayonne"
    },
    {
      "nomR": "Le Bistrot Sainte Cluque",
      "villeR": "Bayonne"
    },
    {
      "nomR": "la petite auberge",
      "villeR": "Bayonne"
    },
    {
      "nomR": "La table de POTTOKA",
      "villeR": "Bayonne"
    },
    {
      "nomR": "La Rotisserie du Roy L\u00e9on",
      "villeR": "Bayonne"
    },
    {
      "nomR": "Bar du March\u00e9",
      "villeR": "Bayonne"
    },
    {
      "nomR": "Trinquet Moderne",
      "villeR": "Bayonne"
    }
  ]
}
```

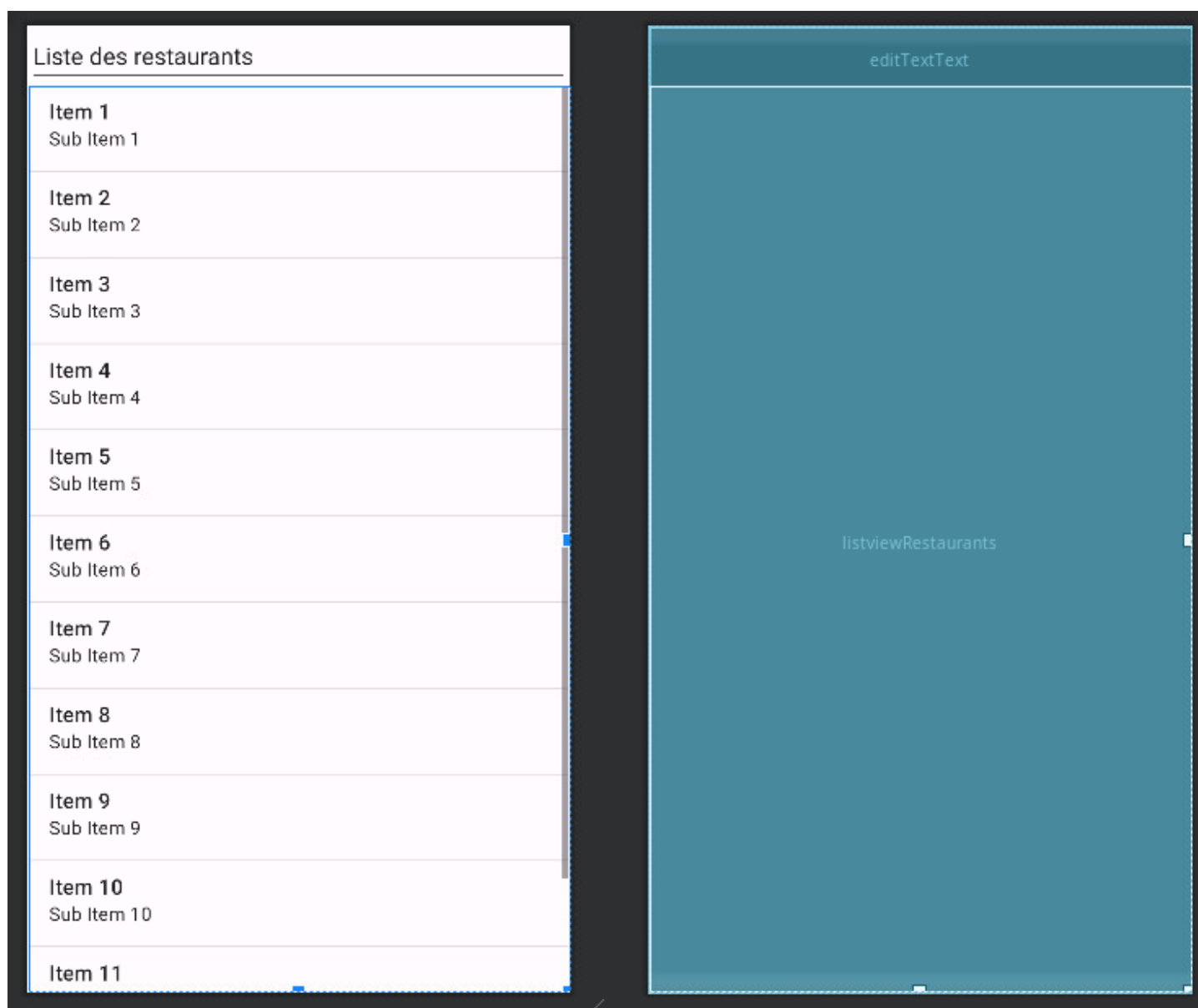
Nous n'avons eu aucun problème pour cette partie.

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Partie Android Studio (Pierre) :

Pour cette partie nous avons besoin de créer une interface graphique avec une liste qui affichera la listes des restaurants que nous récupéreront à l'aide de l'API précédemment crée.

Pour la partie interface graphique nous allons dans le fichier activity_main.xml, et nous créons un interface simple avec une liste.



2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Puis nous devons créer la classe Restaurant

```
public class DetailResto extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_resto);
        // Récupérer les informations depuis l'indent
        Intent intent = getIntent();
        if (intent != null){

            String nomResto = intent.getStringExtra( name: "nomResto");
            String villeResto = intent.getStringExtra( name: "villeResto");
            String numResto = intent.getStringExtra( name: "numResto");
            String voieResto = intent.getStringExtra( name: "voieResto");
            String cpResto = intent.getStringExtra( name: "cpResto");
            String descResto = intent.getStringExtra( name: "descResto");
            String horaireResto = intent.getStringExtra( name: "horaireResto");

            // Affichage des informations
            TextView textViewNom = (TextView)findViewById(R.id.TextViewNom);
            textViewNom.setText(nomResto);

            TextView textViewVille = (TextView)findViewById(R.id.TextViewVille);
            textViewVille.setText(villeResto);

            TextView textViewNum = (TextView)findViewById(R.id.TextViewNum);
            textViewNum.setText(numResto);

            TextView textViewVoie = (TextView)findViewById(R.id.TextViewVoie);
            textViewVoie.setText(voieResto);

            TextView textViewCp = (TextView)findViewById(R.id.TextViewCp);
            textViewCp.setText(cpResto);

            TextView textViewDesc = (TextView)findViewById(R.id.TextViewDesc);
            textViewDesc.setText(descResto);

            TextView textViewHoraire = (TextView)findViewById(R.id.TextViewHoraire);
            textViewHoraire.setText(horaireResto);

        }
    }
}
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Ensuite pour afficher les données dans la liste il faut dans un premier temps initialiser les variables

```
public class MainActivity extends AppCompatActivity {
    2 usages
    ListView listViewRestaurants;
    3 usages
    ArrayList<Restaurant> lesRestos = new ArrayList();
    Pierre Kerlau
```

Puis nous faisons en sorte de récupérer les informations depuis l'API

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Référence à la ListView définie dans le fichier XML
    listViewRestaurants = findViewById(R.id.listViewRestaurants);

    // Création de la requête pour obtenir les restaurants depuis le serveur
    Request requestRestos = new Request.Builder().url("http://10.15.13.197/Android/apiAppResto/getAllRestoJSON.php").build();

    // Exécution de la requête de manière asynchrone
    OkHttpClient httpClient = new OkHttpClient();
    Pierre Kerlau
    httpClient.newCall(requestRestos).enqueue(new Callback() {
```

Nous faisons des exceptions,

Si la requête échoue :

```
@Override
public void onFailure(@NonNull Call call, @NonNull IOException e) {
    // En cas d'échec de la requête, affichez l'erreur dans les logs
    e.printStackTrace();
    Log.i("erreur1", "erreur requête getAllRestoJSON.php");
}
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Si nous récupérons les informations :

```
@Override
public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {
    // Lorsque la réponse est reçue avec succès
    final String myResponse = response.body().string();
    // Pierre Kerlau
    MainActivity.this.runOnUiThread(new Runnable() {
        // Pierre Kerlau
        @Override
        public void run() {
            // Traitement de la réponse JSON sur le thread principal de l'UI
            JSONObject jsonObjectLesRestos = null;
            try {
                // Convertit la réponse en objet JSON
                jsonObjectLesRestos = new JSONObject(myResponse);
            } catch (JSONException e) {
                // En cas d'erreur lors de la conversion, lancez une RuntimeException
                throw new RuntimeException(e);
            }
            // Récupération du tableau JSON "Restos"
            JSONArray jsonArray = jsonObjectLesRestos.optJSONArray( name: "restos");
            // Nettoyage de la liste actuelle des restaurants
            lesRestos.clear();
```

Puis nous parcourons le tableau pour récupérer les informations

```
for (int i = 0; i < jsonArray.length(); i++) {
    JSONObject jsonObject = null;
    try {
        // Récupération des données pour chaque restaurant
        jsonObject = jsonArray.getJSONObject(i);
        String nomResto = jsonObject.getString( name: "nomR");
        String villeResto = jsonObject.getString( name: "villeR");
        Restaurant unResto = new Restaurant(nomResto, villeResto);

        // Ajout du restaurant à la liste
        lesRestos.add(unResto);
    } catch (JSONException e) {
        // En cas d'erreur lors de la récupération des données, lancez une RuntimeException
        throw new RuntimeException(e);
    }
}
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Et nous les affichons dans notre liste

```
// Création d'un adaptateur pour afficher la liste des restaurants dans la ListView
ArrayAdapter<Restaurant> dataAdapter = new ArrayAdapter<>( context: MainActivity.this, android.R.layout.simple_list_item_1, lesRestos);
// Attribution de l'adaptateur à la ListView
listviewRestaurants.setAdapter(dataAdapter);
```

Le résultat est le suivant :



2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Compte-rendu itération 2

User story : En tant qu'utilisateur, je souhaite, à partir de la liste des restaurants, pouvoir consulter la fiche détaillée d'un restaurant.

Dépôt GIT : https://gitlab.com/2slam_ap_android/app_android.git

Pour les deux prochaines séances nous avons pour objectif de finir l'itération 2 et avancer sur les affichages utilisateur des prochaines itérations.

Comme l'API de l'itération 2 ne nous donne que les noms et les villes des restaurants, il faut légèrement changer le code.

```
<?php
try {
    // Connexion à la base de données MySQL
    $db="resto2";
    $dbhost="localhost";
    $dbport=3306;
    $dbuser="root";
    $dbpasswd="joliverie";

    $connexion = new PDO('mysql:host='.$dbhost.';port='.$dbport.';dbname='.$db.'',
    $dbuser, $dbpasswd);
    $connexion->exec("SET CHARACTER SET utf8");

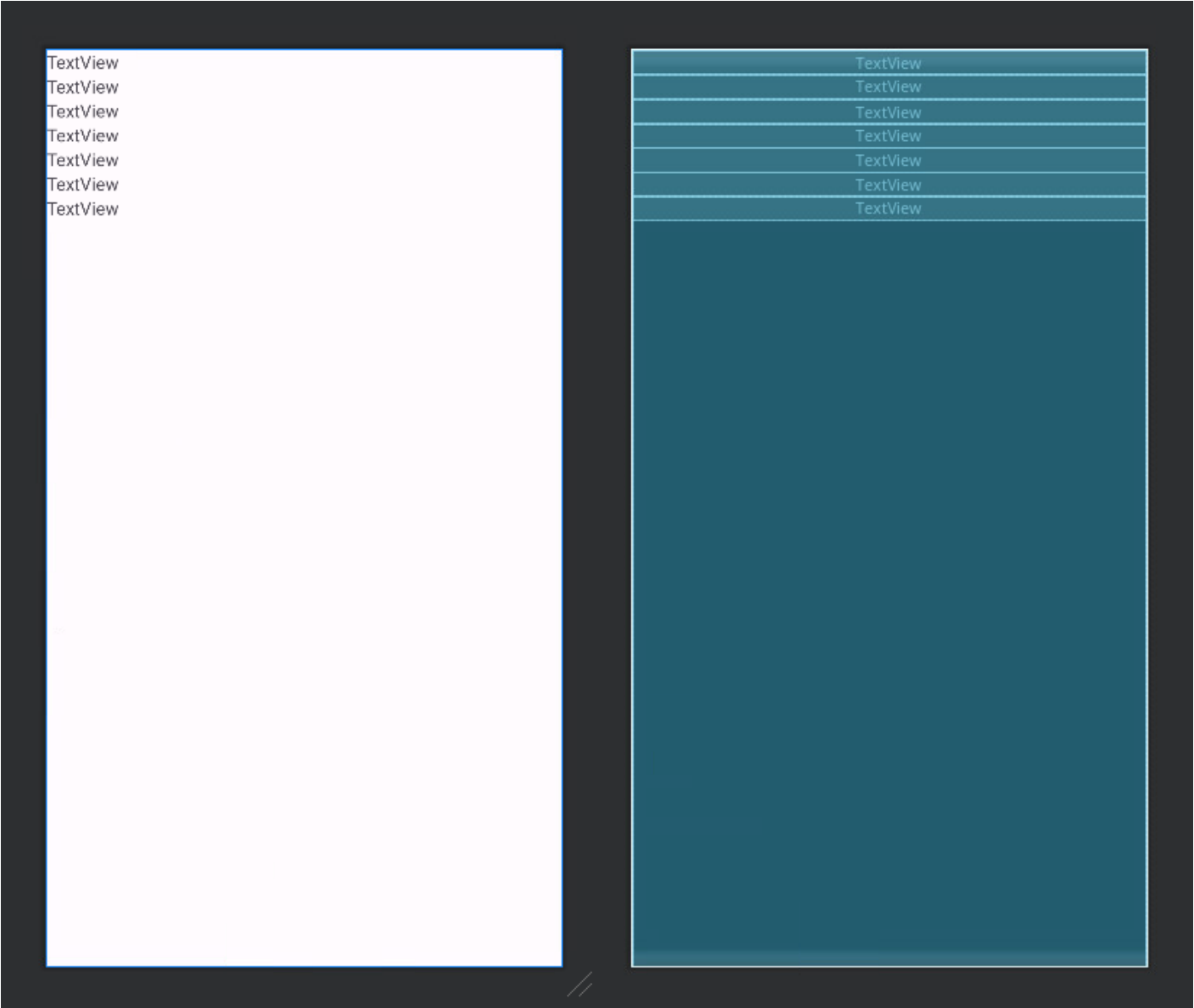
    $reponse=$connexion->prepare("SELECT nomR, numAdrR, voieAdrR, cpR, villeR,
    descR, horairesR FROM resto;");
    $reponse->execute();
    $datas = array();

    while($res=$reponse->fetch(PDO::FETCH_ASSOC)) {
        $datas['restos'][]=$res;
    }

    echo json_encode($datas);
}
catch (Exception $e) {
    die('Erreur : ' . $e->getMessage());
}
?>
```


2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Pour afficher les informations détaillés des restaurants nous faisons une nouvelle vue qui accueillera le texte



2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Pour faire l'itération 2, nous devons dans un premier temps rendre la liste cliquable à l'aide d'un écouteur

```
listviewRestaurants.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // Récupérer le restaurant sélectionné
        Restaurant selectedRestaurant = lesRestos.get(position);
```

Ensuite nous envoyons les informations à la nouvelle vue

```
// Récupérer le restaurant sélectionné
Restaurant selectedRestaurant = lesRestos.get(position);

// Passer à l'activité de détails avec des données supplémentaires
Intent intent = new Intent( packageContext: MainActivity.this, DetailResto.class);
intent.putExtra( name: "nomResto", selectedRestaurant.getNomResto().toString());
intent.putExtra( name: "villeResto", selectedRestaurant.getVilleResto().toString());
intent.putExtra( name: "numResto", selectedRestaurant.getNumResto().toString());
intent.putExtra( name: "voieResto", selectedRestaurant.getVoieResto().toString());
intent.putExtra( name: "cpResto", selectedRestaurant.getCpResto().toString());
intent.putExtra( name: "descResto", selectedRestaurant.getDescResto().toString());
intent.putExtra( name: "horaireResto", selectedRestaurant.getHoraireResto().toString());
startActivity(intent);
```

Enfin, la nouvelle vue récupère les informations

```
// Récupérer les informations depuis l'indent
Intent intent = getIntent();
if (intent != null){

    String nomResto = intent.getStringExtra( name: "nomResto");
    String villeResto = intent.getStringExtra( name: "villeResto");
    String numResto = intent.getStringExtra( name: "numResto");
    String voieResto = intent.getStringExtra( name: "voieResto");
    String cpResto = intent.getStringExtra( name: "cpResto");
    String descResto = intent.getStringExtra( name: "descResto");
    String horaireResto = intent.getStringExtra( name: "horaireResto");
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Puis nous les affichons dans les textview

```
// Affichage des informations
TextView textViewNom = (TextView)findViewById(R.id.TextViewNom);
textViewNom.setText(nomResto);

TextView textViewVille = (TextView)findViewById(R.id.TextViewVille);
textViewVille.setText(villeResto);

TextView textViewNum = (TextView)findViewById(R.id.TextViewNum);
textViewNum.setText(numResto);

TextView textViewVoie = (TextView)findViewById(R.id.TextViewVoie);
textViewVoie.setText(voieResto);

TextView textViewCp = (TextView)findViewById(R.id.TextViewCp);
textViewCp.setText(cpResto);

TextView textViewDesc = (TextView)findViewById(R.id.TextViewDesc);
textViewDesc.setText(descResto);

TextView textViewHoraire = (TextView)findViewById(R.id.TextViewHoraire);
textViewHoraire.setText(horaireResto);
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Enfin nous obtenons ceci :



2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Séance du 11 Décembre 2023

Pour cette séance, notre objectif est de mettre en forme l’affichage des horaires et afficher les images.

Davy et Pierre sont en charge de la partie API, PhpMyAdmin et s’occupent de mettre en forme l’affichage des horaires et des images, Lisa s’occupe de créer les interfaces utilisateur.

Dépôt GIT : https://gitlab.com/2slam_ap_android/app_android.git

Pour les horaires nous changeons dans un premier notre TextView par un WebView, ensuite nous affichons les informations à l’aide de ce code :

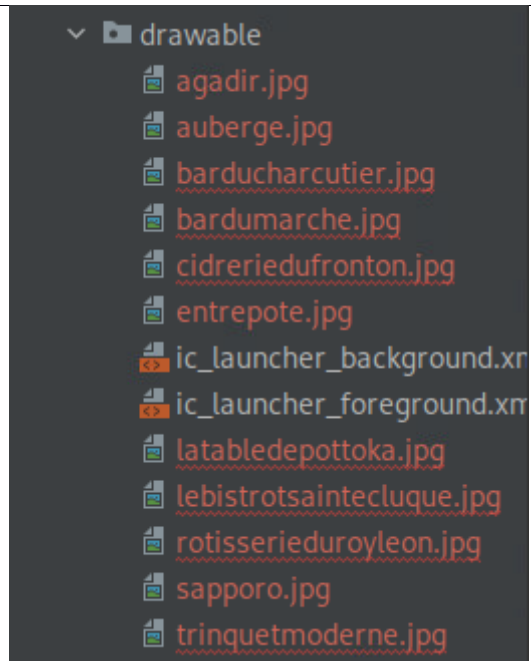
```
WebView WebViewHoraire = (WebView)findViewById(R.id.WebViewHoraire);
WebViewHoraire.loadData(horaireResto, mimeType: "text/html", encoding: "UTF-8");
```

Nous obtenons ceci :

Ouverture	Semaine	Week-end
Midi	de 11h45 à 14h30	de 11h45 à 15h00
Soir	de 18h45 à 22h30	de 18h45 à 1h
À emporter	de 11h30 à 23h	de 11h30 à 2h

Enfin pour les images, nous devons dans un premier temps les importer dans notre fichier drawable en veillant à ce qu’il n’y ai pas de majuscule dans le nom des fichiers :

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023



2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git Date de rédaction : Lundi 4 Décembre 2023

Puis nous devons récupérer le cheminR depuis la base de donnée à l'aide de l'API que nous mettons à jour :

```
<?php
try {
    // Connexion à la base de données MySQL
    $db = "resto2";
    $dbhost = "localhost";
    $dbport = 3306;
    $dbuser = "root";
    $dbpasswd = "joliverie";

    $connexion = new PDO('mysql:host=' . $dbhost . ';port=' . $dbport . ';dbname=' .
$db . '', $dbuser, $dbpasswd);
    $connexion->exec("SET CHARACTER SET utf8");

    $reponse = $connexion->prepare("SELECT r.nomR, r.numAdrR, r.voieAdrR, r.cpR,
r.villeR, r.descR, r.horairesR, GROUP_CONCAT(p.cheminP) AS cheminP FROM resto r JOIN
photo p ON r.idR = p.idR GROUP BY r.idR;");
    $reponse->execute();
    $datas = array();

    while ($res = $reponse->fetch(PDO::FETCH_ASSOC)) {
        $datas['restos'][] = $res;
    }

    echo json_encode($datas);
} catch (Exception $e) {
    die('Erreur : ' . $e->getMessage());
}
?>
```


2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Nous devons ajouter cheminR à notre constructeur en veillant à ajouter les accesseurs et mutateurs

```

3 usages
private String imageResto;

1 usage  👤 Pierre Kerlau *
public Restaurant(String nomResto, String villeResto, String numR
    this.nomResto = nomResto;
    this.villeResto = villeResto;
    this.numResto = numResto;
    this.voieResto = voieResto;
    this.cpResto = cpResto;
    this.descResto = descResto;
    this.horaireResto = horaireResto;
    this.imageResto = imageResto;
}

1 usage  new *
public String getImageResto() {
    return imageResto;
}

new *
public void setImageResto(String imageResto) {
    this.imageResto = imageResto;
}

```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Nous ajoutons cheminR à la boucle qui permet de créer de nouveaux objets restaurants

```
for (int i = 0; i < jsonArray.length(); i++) {
    JSONObject jsonObject = null;
    try {
        // Récupération des données pour chaque restaurant
        jsonObject = jsonArray.getJSONObject(i);
        String nomResto = jsonObject.getString( name: "nomR");
        String villeResto = jsonObject.getString( name: "villeR");
        String numResto = jsonObject.getString( name: "numAdrR");
        String voieResto = jsonObject.getString( name: "voieAdrR");
        String cpResto = jsonObject.getString( name: "cpR");
        String descResto = jsonObject.getString( name: "descR");
        String horaireResto = jsonObject.getString( name: "horairesR");
        String imageResto = jsonObject.getString( name: "cheminP");

        Restaurant unResto = new Restaurant(nomResto, villeResto, numResto,
```

Puis nous passons l'information à la vue DetailResto

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    // Récupérer le restaurant sélectionné
    Restaurant selectedRestaurant = lesRestos.get(position);

    // Passer à l'activité de détails avec des données supplémentaires
    Intent intent = new Intent( packageContext: MainActivity.this, DetailResto.class);
    intent.putExtra( name: "nomResto", selectedRestaurant.getNomResto().toString());
    intent.putExtra( name: "villeResto", selectedRestaurant.getVilleResto().toString());
    intent.putExtra( name: "numResto", selectedRestaurant.getNumResto().toString());
    intent.putExtra( name: "voieResto", selectedRestaurant.getVoieResto().toString());
    intent.putExtra( name: "cpResto", selectedRestaurant.getCpResto().toString());
    intent.putExtra( name: "descResto", selectedRestaurant.getDescResto().toString());
    intent.putExtra( name: "horaireResto", selectedRestaurant.getHoraireResto().toString());
    intent.putExtra( name: "imageResto", selectedRestaurant.getImageResto().toString());
    startActivity(intent);
}
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Enfin, dans DetailResto, nous récupérons l'information

```
String nomResto = intent.getStringExtra( name: "nomResto");
String villeResto = intent.getStringExtra( name: "villeResto");
String numResto = intent.getStringExtra( name: "numResto");
String voieResto = intent.getStringExtra( name: "voieResto");
String cpResto = intent.getStringExtra( name: "cpResto");
String descResto = intent.getStringExtra( name: "descResto");
String horaireResto = intent.getStringExtra( name: "horaireResto");
String imageResto = intent.getStringExtra( name: "imageResto");
```

Puis nous comparons les noms des images pour trouver l'image correspondante au restaurant

```
ImageView imageViewResto = findViewById(R.id.imageView);
if (imageResto.equals("agadir.jpg")){
    imageViewResto.setImageResource(R.drawable.agadir);
}
if (imageResto.equals("auberge.jpg")){
    imageViewResto.setImageResource(R.drawable.auberge);
}
if (imageResto.equals("barDuCharcutier.jpg")){
    imageViewResto.setImageResource(R.drawable.barducharcutier);
}
if (imageResto.equals("barDuMarche.jpg")){
    imageViewResto.setImageResource(R.drawable.bardumarche);
}
if (imageResto.equals("entrepote.jpg")){
    imageViewResto.setImageResource(R.drawable.entrepote);
}
if (imageResto.equals("laTableDePottoka.jpg")){
    imageViewResto.setImageResource(R.drawable.latabledepottoka);
}
if (imageResto.equals("leBistrotSainteCluque.jpg")){
    imageViewResto.setImageResource(R.drawable.lebistrotsaintecluque);
}
if (imageResto.equals("rotisserieDuRoyLeon.jpg")){
    imageViewResto.setImageResource(R.drawable.rotisseriesduroyleon);
}
if (imageResto.equals("sapporo.jpg")){
    imageViewResto.setImageResource(R.drawable.sapporo);
}
```

2023/2024	Module D3-1 – Programmation orienté objet – TP n°xx
BTS SIO 2SLAM	Auteur : Groupe 3 (Belny Davy, Rodriguez Lisa, Kerlau Pierre) https://gitlab.com/2slam_ap_android/app_android.git
	Date de rédaction : Lundi 4 Décembre 2023

Enfin nous obtenons ceci :

