

**Correction TP03****Matière : ATELIER DEVELOPPEMENT MOBILE****Classes : SEM31****Solution1 : Application Web**

- 1- Lancer le navigateur (Browser) du Smartphone
 - 2- Consulter la page <http://192.100.30.12:80/bib/index.html>
- Aucune ligne de code
Fonctionne sur tous les types de smartphone

Solution1 : Application Hybride

- 1- Créer une Application android
 - 2- Ajouter un WebView à l'activité
 - 3- Ecrire le code de la classe MainActivity
- Fonctionne seulement sur les terminaux android
package com.bibhybride;

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
```

```
public class MainActivity extends Activity {
    private WebView webV1;
    String url = "http://192.100.30.12 :80/bib/index.html";

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

    private void init() {
        webV1 = (WebView) findViewById(R.id.webView1);
        webV1.getSettings().setLoadsImagesAutomatically(true);
        webV1.getSettings().setJavaScriptEnabled(true);
        webV1.loadUrl(url);
    }
}
```

Solution1 : Application Native

package bib.com.gestionbibjson;

```
public class Config {  
    public static final String URL = "http://192.168.1.8:80/JSON_PHP_Bib/bib/";  
}
```

Ajout.java

```
package com.bib;  
public class Ajout extends Activity {  
    private EditText edTitre;  
    private EditText edNbPage;  
    private Button btnAjouter;  
    private Button btnRetour;  
  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.ajout);  
        init();  
    }  
  
    private void init() {  
        edTitre = (EditText) findViewById(R.id.edTitre1);  
        edNbPage = (EditText) findViewById(R.id.edNbPage1);  
        btnAjouter = (Button) findViewById(R.id.btnAjouter);  
        btnRetour = (Button) findViewById(R.id.btnRetour);  
        ajouterEcouteur();  
    }  
  
    private void ajouterEcouteur() {  
        btnAjouter.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                ajouterLivre();  
            }  
        });  
        btnRetour.setOnClickListener(new OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                retourner();  
            }  
        });  
    }  
  
    protected void retourner() {  
        finish();  
    }  
  
    protected void ajouterLivre() {  
        RequestQueue queue = Volley.newRequestQueue(this);  
        String url = Config.URL + "ajout/Ajout.php";  
        StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new  
Response.Listener<String>() {  
    @Override  
    public void onResponse(String response) {  
        try {  
            JSONObject r = new JSONObject(response);  
            String etat = r.getString("ETAT");  
            if (etat.equals("SUCCES")) {  
                edTitre.setText("");  
            }  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
    }  
});  
queue.add(stringRequest);  
    }  
}
```

```

        edNbPage.setText("");
        edTitre.requestFocus();
    } else {
        Toast t = Toast.makeText(Ajout.this, "Echec de l'ajout", Toast.LENGTH_LONG);
        t.show();
    }
} catch (JSONException error) {
    Toast t = Toast.makeText(Ajout.this, "Problème d'analyse JSON: " + error.getMessage(),
        Toast.LENGTH_LONG);
    t.show();
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError e) {
        Toast t = Toast.makeText(Ajout.this, "Problème d'appel HTTP: " + e.getMessage(),
        Toast.LENGTH_LONG);
        t.show();
    }
})
{
    @Override
    public Map<String, String> getParams() throws AuthFailureError {
        HashMap<String, String> headers = new HashMap<String, String>();
        headers.put("titre", edTitre.getText().toString());
        headers.put("nbPage", edNbPage.getText().toString());
        return headers;
    }
};
queue.add(stringRequest);
}
}

```

Suppression.java

```

package com.bib;
public class Suppression extends Activity {
    private Spinner spLivre;
    private Button btnSupprimer;
    private Button btnRetour;
    private ArrayAdapter<Livre> adpLivre;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.suppression);
        init();
    }

    private void init() {
        spLivre = (Spinner) findViewById(R.id.spLivre1);
        btnSupprimer = (Button) findViewById(R.id.btnSupprimer);
        btnRetour = (Button) findViewById(R.id.btnRetour);
        adpLivre = new ArrayAdapter<Livre>(this, android.R.layout.simple_list_item_1);
        adpLivre.setDropDownViewResource(android.R.layout.simple_list_item_1);
        spLivre.setAdapter(adpLivre);
        remplir();
        ajouterEcouteur();
    }

    private void remplir() {
        RequestQueue queue = Volley.newRequestQueue(this);
        String url = Config.URL + "liste/ListeLivres.php";
        JsonObjectRequest jsObjRequest = new JsonObjectRequest(Request.Method.POST, url, null,
            new Response.Listener<JSONObject>() {
                @Override

```

```

public void onResponse(JSONObject response) {
    try {

        JSONArray a = response.getJSONArray("liste");
        adpLivre.clear();
        for (int i = 0; i < a.length(); i++) {
            JSONObject o = a.getJSONObject(i);
            int id = o.getInt("id");
            String titre = o.getString("titre");
            int nbPage = o.getInt("nbPage");
            Livre l = new Livre(id, titre, nbPage);
            adpLivre.add(l);
        }
    } catch (JSONException e) {
        Toast t = Toast.makeText(Suppression.this, "Problème d'analyse JSON: " + e.getMessage(),
            Toast.LENGTH_LONG);
        t.show();
    }
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast t = Toast.makeText(Suppression.this, "Problème d'appel HTTP: " + error.getMessage(),
            Toast.LENGTH_LONG);
        t.show();
    }
});
queue.add(jsObjRequest);
}

protected void supprimerLivre() {
    if (spLivre.getSelectedItemPosition() >= 0) {
        final Livre l = (Livre) spLivre.getSelectedItem();
        RequestQueue queue = Volley.newRequestQueue(this);
        String url = Config.URL + "suppression/Suppression.php";
        StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new
        Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject r = new JSONObject(response);
                    String etat = r.getString("ETAT");
                    if (etat.equals("SUCCES")) {
                        remplir();
                    } else {
                        Toast t = Toast.makeText(Suppression.this, "Echec de la suppression", Toast.LENGTH_LONG);
                        t.show();
                    }
                } catch (JSONException error) {
                    Toast t = Toast.makeText(Suppression.this, "Problème d'analyse JSON: " + error.getMessage(),
                        Toast.LENGTH_LONG);
                    t.show();
                }
            }
        });
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError e) {
            Toast t = Toast.makeText(Suppression.this, "Problème d'appel HTTP: " + e.getMessage(),
                Toast.LENGTH_LONG);
            t.show();
        }
    })
}

```

```

    {
        @Override
        public Map<String, String> getParams() throws AuthFailureError {
            HashMap<String, String> headers = new HashMap<String, String>();
            headers.put("id", l.getId() + "");
            return headers;
        }
    };
    queue.add(stringRequest);
}
}

protected void retourner() {
    finish();
}
}

```

Modification.java

```

package com.bib;
public class Modification extends Activity {
    private Spinner spLivre;
    private EditText edTitre;
    private EditText edNbPage;
    private Button btnModifier;
    private Button btnRetour;
    private ArrayAdapter<Livre> adpLivre;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.modification);
        init();
    }

    private void init() {
        spLivre = (Spinner) findViewById(R.id.spLivre2);
        edTitre = (EditText) findViewById(R.id.edTitre01);
        edNbPage = (EditText) findViewById(R.id.edNbPage01);
        btnModifier = (Button) findViewById(R.id.btnModifier);
        btnRetour = (Button) findViewById(R.id.btnRetour);
        adpLivre = new ArrayAdapter<Livre>(this, android.R.layout.simple_list_item_1);
        adpLivre.setDropDownViewResource(android.R.layout.simple_list_item_1);
        spLivre.setAdapter(adpLivre);
        remplir();
        ajouterEcouteur();
    }

    private void remplir() {
        RequestQueue queue = Volley.newRequestQueue(this);
        String url = Config.URL + "liste/ListeLivres.php";
        JsonObjectRequest jsObjRequest = new JsonObjectRequest(Request.Method.POST, url, null,
                new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {
                        try {
                            JSONArray a = response.getJSONArray("liste");
                            adpLivre.clear();
                            for (int i = 0; i < a.length(); i++) {
                                JSONObject o = a.getJSONObject(i);
                                int id = o.getInt("id");
                                String titre = o.getString("titre");
                                int nbPage = o.getInt("nbPage");
                                Livre l = new Livre(id, titre, nbPage);
                                adpLivre.add(l);
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                });
        queue.add(jsObjRequest);
    }
}

```

```

    }

} catch (JSONException e) {
    Toast t = Toast.makeText(Modification.this, "Problème d'analyse JSON: " + e.getMessage(),
        Toast.LENGTH_LONG);
    t.show();
}
},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Toast t = Toast.makeText(Modification.this, "Problème d'appel HTTP: " + error.getMessage(),
        Toast.LENGTH_LONG);
    t.show();
}
});
queue.add(jsObjRequest);
}

private void ajouterEcouteur() {
btnModifier.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
    modifierLivre();
}
});
btnRetour.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
    retourner();
}
});
spLivre.setOnItemSelectedListener(new OnItemSelectedListener() {
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
    actualiser();
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
}
});
}

protected void actualiser() {
Livre l = (Livre) spLivre.getSelectedItem();
if (l != null) {
    edTitre.setText(l.getTitre());
    edNbPage.setText(l.getNbpage() + "");
}
}

protected void retourner() {
finish();
}

protected void modifierLivre() {
if (spLivre.getSelectedItemPosition() >= 0) {
    final Livre l = (Livre) spLivre.getSelectedItem();
    RequestQueue queue = Volley.newRequestQueue(this);
    String url = Config.URL + "modification/Modification.php";
}
}

```

```

StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject r = new JSONObject(response);
            String etat = r.getString("ETAT");
            if (etat.equals("SUCCES")) {
                remplir();
            } else {
                Toast t = Toast.makeText(Modification.this, "Echec de la modification", Toast.LENGTH_LONG);
                t.show();
            }
        } catch (JSONException error) {
            Toast t = Toast.makeText(Modification.this, "Problème d'analyse JSON:" + error.getMessage(),
                Toast.LENGTH_LONG);
            t.show();
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError e) {
        Toast t = Toast.makeText(Modification.this, "Problème d'appel HTTP:" + e.getMessage(),
            Toast.LENGTH_LONG);
        t.show();
    }
}) {
    @Override
    public Map<String, String> getParams() throws AuthFailureError {
        HashMap<String, String> headers = new HashMap<String, String>();
        headers.put("id", l.getId() + "");
        headers.put("titre", edTitre.getText().toString());
        headers.put("nbPage", edNbPage.getText().toString());
        return headers;
    }
};
queue.add(stringRequest);
}
}
}

```

Consultation.java

```

package com.bib;
public class Consultation extends Activity {
    private ListView lstLivre;
    private Button btnRetour;
    private ArrayAdapter<Livre> adpLivre;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.consultation);
        init();
    }

    private void init() {
        lstLivre = (ListView) findViewById(R.id.lstLivre);
        btnRetour = (Button) findViewById(R.id.btnRetour);
        adpLivre = new ArrayAdapter<Livre>(this, android.R.layout.simple_list_item_1);
        adpLivre.setDropDownViewResource(android.R.layout.simple_list_item_1);
        lstLivre.setAdapter(adpLivre);
        ajouterEcouteur();
        remplir();
    }
}

```

```

private void ajouterEcouteur() {
    btnRetour.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            retourner();
        }
    });
}

protected void retourner() {
    finish();
}

private void remplir() {
    RequestQueue queue = Volley.newRequestQueue(this);
    String url = Config.URL + "liste/ListeLivres.php";
    StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject r = new JSONObject(response);
            JSONArray a = r.getJSONArray("liste");
            adpLivre.clear();
            for (int i = 0; i < a.length(); i++) {
                JSONObject o = a.getJSONObject(i);
                int id = o.getInt("id");
                String titre = o.getString("titre");
                int nbPage = o.getInt("nbPage");
                Livre l = new Livre(id, titre, nbPage);
                adpLivre.add(l);
            }
        } catch (JSONException error) {
            Toast t = Toast.makeText(Consultation.this, "Problème d'analyse JSON:" + error.getMessage(),
                Toast.LENGTH_LONG);
            t.show();
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError e) {
        Toast t = Toast.makeText(Consultation.this, "Problème d'appel HTTP:" + e.getMessage(),
            Toast.LENGTH_LONG);
        t.show();
    }
});

queue.add(stringRequest);
}
}

```