

**I. MỤC TIÊU**

- Tạo một WebService đơn giản với PHP và MySQL
- Sử dụng thư viện Volley để kết nối tới WebService

**II. TÓM TẮT LÝ THUYẾT****1. Web service**

Web service là một tập các phương thức được gọi thực hiện từ xa thông qua một địa chỉ http url. Kết quả trả về của web service thường dưới dạng json hoặc xml. Web service thường được sử dụng để tạo các ứng dụng phân tán.

Web service cho phép ta giao tiếp qua mạng, không phải P2P. Nó cung cấp API cho phép ta sử dụng nó để tương tác với service hoặc ta có thể tự tạo web service bằng PHP và MYSQL như trong bài viết này.

Ví dụ ta dùng một ứng dụng di động, ứng dụng desktop, web. Thì các ứng dụng đó gửi thông tin lên webservice rồi webservice xử lý dữ liệu ta gửi từ app đến sever và sever xử lý xong nó sẽ trả về các dữ liệu cho ta bằng json hay xml, để xử lý dữ liệu ở app ta sẽ phải parse json hay parse xml.

**Đặc điểm của web service:**

- Chi phí thấp, dễ bảo trì
- Có thể được truy cập từ bất kỳ ứng dụng nào
- Không phụ thuộc vào ngôn ngữ lập trình: PHP, JAVA, .NET,...
- Hỗ trợ thao tác giữa các thành phần không đồng nhất

**2. Volley**

Volley là một Networking Library để quản lý các request Network và lưu trữ những response không cần các Developer viết nhiều code như trước .

Các tools bên trong Volley cho phép cùng một lúc thực hiện nhiều request trên các thread khác nhau với các mức độ ưu tiên (priority ) khác nhau. Tất cả request được thực hiện và lưu trữ trong bộ nhớ cache giúp cho việc reload lại dữ liệu nhanh hơn. Toàn bộ response được lưu trong memory vậy nên nó không phải là 1 ý tưởng hay cho việc tải về những dữ liệu lớn như music hay movies. Nhưng nó lại là tuyệt vời với những loại dữ liệu như JSON, Image, String....

Volley có các điểm nổi bật dưới đây:

- Tự động lập lịch (scheduling) cho các request
- Caching Response
- Hỗ trợ set độ ưu tiên cho các request (priority)
- Hỗ trợ nhiều kết quả trả về (String, JSONObject, JSONArray, Bitmap...)
- Có thể hủy Request.

Các class sử dụng trong Volley:

- RequestQueue: Là hàng đợi giữ các Request
- Request: là lớp cơ sở của các Request trong Volley, chứa thông tin về request HTTP
- StringRequest: Kết thừa từ Request, là class đại diện cho request trả về String
- JSONObjectRequest: Là HTTP request có kết quả trả về là JSONObject
- JSONArrayRequest: Là HTTP request có kết quả trả về là JSONArray
- ImageRequest: Là HTTP request có kết quả trả về là Bitmap.

**3. Sử dụng thư viện Volley để kết nối tới WebService**

Khai báo dòng sau trong mục dependencies trong file build.gradle của module app:

```
implementation 'com.android.volley:volley:1.1.1'
```

Cấp quyền sử dụng Internet trong AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Tạo một RequestQueue:

```
RequestQueue requestQueue =  
Volley.newRequestQueue(MainActivity.this);
```

Khi đã có RequestQueue chúng ta chỉ cần các loại Request và add các request đó vào RequestQueue:

```
requestQueue.add(request);
```

StringRequest Volley:

**GET StringRequest Volley**

```
String url = "https://google.com.vn";  
StringRequest stringRequest = new StringRequest(  
    Request.Method.GET,  
    url,  
    new Response.Listener<String>() {  
        @Override  
        public void onResponse(String response) {  
            Log.e(TAG, "StringRequest onResponse: " +  
response);  
        }  
    },  
    new Response.ErrorListener() {  
        @Override  
        public void onErrorResponse(VolleyError error) {  
            Log.e(TAG, "StringRequest onErrorResponse: " +  
error.getMessage());  
        }  
    })
```

**POST StringRequest Volley**

```
String url = "https://google.com.vn";  
StringRequest stringRequest = new StringRequest(  
    Request.Method.POST,  
    url,  
    new Response.Listener<String>() {  
        @Override  
        public void onResponse(String response) {  
            Log.e(TAG, "onResponse: " + response);  
        }  
    },  
    new Response.ErrorListener() {  
        @Override  
        public void onErrorResponse(VolleyError error) {  
            Log.e(TAG, "onErrorResponse: " +  
error.getMessage());  
        }  
    }) {  
    @Override  
    protected Map<String, String> getParams() throws
```

```
AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("para1", "value1");
    params.put("para1", "value2");
    return params;
}
};
```

Cách tạo request đối với JSONObjectRequest, JSONArrayObject, ImageRequest đều tương tự như với StringRequest. Constructor sẽ có những thành phần dưới đây:

- Method POST, GET.
- Constructor truyền vào url của request
- Hai Listener là Response.Listener và Response.ErrorListener

**Lưu ý:** Khi app giao tiếp với server bằng cách sử dụng cleartext (ví dụ như HTTP) sẽ làm tăng nguy cơ bị rò rỉ và giả mạo nội dung. Các bên thứ ba có thể thêm bớt dữ liệu trái phép hoặc trích xuất thông tin của người dùng. Đó là lý do tại sao chúng ta được khuyến khích sử dụng các phương thức bảo mật thông tin (ví dụ như HTTPS thay thế cho HTTP). Tuy nhiên, trong trường hợp bắt buộc phải sử dụng cleartext thì chúng ta có thể gán thuộc tính useCleartextTraffic giá trị là "true" trong AndroidManifest.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="...">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        ...
        android:usesCleartextTraffic="true">
        <activity ...>
        </activity>
    </application>
</manifest>
```

### III. NỘI DUNG THỰC HÀNH

#### 1. Quản lý sinh viên

Xây dựng hệ thống Quản lý sinh viên đơn giản, bao gồm những phần:

- Database: cơ sở dữ liệu dbsinhvien gồm 1 bảng: sinhvien\_tbl với 2 trường là *masv* (int, pk) và *tensv* (text)
- WebService: được xây dựng bằng PHP với các service sau:
  - + api.php?action=getall: trả về chuỗi JSON chứa danh sách tất cả sinh viên. Không cần tham số bổ sung.
  - + api.php?action=insert: thêm 1 sinh viên mới, trả về chuỗi JSON thông báo kết quả true/false. Tham số bổ sung theo phương thức GET: *masv* và *tensv*.
  - + api.php?action=delete: xóa sinh viên theo mã, trả về chuỗi JSON thông báo kết quả true/false. Tham số bổ sung theo phương thức GET: *masv*.
  - + api.php?action=update: cập nhật thông tin sinh viên (cập nhật tên, không cập nhật mã), trả về chuỗi JSON thông báo kết quả true/false. Tham số bổ sung theo phương thức GET: *masv* và *tensv*.
- Ứng dụng android: gồm 1 activity dùng để hiển thị danh sách, thêm, xóa, sửa thông tin sinh viên như hình dưới:

## LAB 8: WebService

- + Khi nhấn vào 1 sinh viên thì đưa các thông tin về sinh viên đó lên các view để cập nhật (không cho sửa mã). Sau khi cập nhật xong thông tin, nhấn nút Lưu thì sẽ gọi lên service để cập nhật trên server, sau đó cập nhật lại listview.
- + Khi nhấn giữ vào 1 sinh viên thì gọi lên service để xóa thông tin sinh viên đó trên server, sau đó cập nhật lại listview.

**Lab08\_WebService**

**Web Service**

Mã SV:

Tên SV:

LƯU

Mã: 1  
Tên: Nguyen Ti

Mã: 2  
Tên: Tran Suu

Mã: 3  
Tên: Le Dan

### Database (giảng viên cung cấp)

Sinh viên import file dbsinhvien.sql vào phpmyadmin để có cơ sở dữ liệu dbsinhvien gồm 1 bảng (sinhvien\_tbl) với 2 trường là masv (int, pk) và tensv (text).

### WebService (giảng viên cung cấp)

#### dbconnection.php

```
1 <?php
2 class DBConnection{
3     private $host = "localhost";
4     private $db_name = "dbsinhvien";
5     private $username = "root";
6     private $password = "";
7     private $conn;
8
9     /**
10      * Khởi tạo - Mở kết nối đến database
11      *
12      * @param none
13      * @return database
14      */
15     public function __construct(){
16         $this->conn = new mysqli($this->host, $this->username, $this->password,
17                                 $this->db_name);
18     }
19
20     /**
21      * Hủy - Đóng kết nối tới database
22      *
23      * @param none
24      * @return none
25      */
```

```

25     function __destruct()
26     {
27         $this->conn->close();
28     }
29
30     /**
31     * getConnection
32     *
33     * @param none
34     * @return connection
35     */
36     public function getConnection() {
37         return $this->conn;
38     }

```

**sinhviendal.php**

```

1  <?php
2  // the dbconnection file
3  require_once 'dbconnection.php';
4
5  class SinhvienDAL{
6      /**
7       * Lấy danh sách sinh viên
8       *
9       * @param none
10      * @return array[] Danh sách sinh viên
11      */
12      public function getAll()
13      {
14          $dbConnection = new DBConnection();
15          $conn = $dbConnection->getConnection();
16          $query = 'SELECT masv, tensv FROM sinhvien_tbl';
17          $list = array();
18          $result = $conn->query($query);
19          while ($row = $result->fetch_assoc())
20          {
21              $list[] = $row;
22          }
23          return $list;
24      }
25
26      /**
27      * Thêm 1 sinh viên vào CSDL
28      *
29      * @param integer masv   Mã sinh viên
30      * @param string tensv   Tên sinh viên
31      * @return true/false   Kết quả thực hiện câu sql
32      */
33      function insert($masv, $tensv)
34      {
35          $dbConnection = new DBConnection();
36          $conn = $dbConnection->getConnection();
37          $query = 'INSERT INTO sinhvien_tbl (masv, tensv) VALUES (?, ?)';
38          $stmt = $conn->prepare($query);
39          $stmt->bind_param("is", $masv, $tensv);
40          return $stmt->execute();
41      }
42
43      /**
44      * Thêm nhiều sinh viên vào CSDL
45      *
46      * @param string data     Chuỗi JSON chứa thông tin của các sinh viên cần thêm
47      * @return int            Số dòng thêm thành công
48      */

```

```

49     function insertsome($data)
50     {
51         $result = 0;
52         $dbConnection = new DBConnection();
53         $conn = $dbConnection->getConnection();
54         $query = 'INSERT INTO sinhvien_tbl (masv, tensv) VALUES (?, ?)';
55         $arrSinhvien = json_decode($data, TRUE);
56
57         foreach ($arrSinhvien as $sinhvien) {
58             var_dump($sinhvien);
59             $stmt = $conn->prepare($query);
60             $stmt->bind_param("is", $sinhvien["masv"], $sinhvien["tensv"]);
61             if ($stmt->execute()) $result++;
62         }
63         return $result;
64     }
65
66     /**
67     * Xóa 1 sinh viên
68     *
69     * @param integer masv    Mã sinh viên
70     * @return true/false    Kết quả thực hiện câu sql
71     */
72     function delete($masv)
73     {
74         $dbConnection = new DBConnection();
75         $conn = $dbConnection->getConnection();
76         $query = 'DELETE FROM sinhvien_tbl WHERE masv = ?';
77         $stmt = $conn->prepare($query);
78         $stmt->bind_param("i", $masv);
79         return $stmt->execute();
80     }
81
82     /**
83     * Cập nhật 1 sinh viên
84     *
85     * @param integer masv    Mã sinh viên
86     * @param string tensv    Tên sinh viên
87     * @return true/false    Kết quả thực hiện câu sql
88     */
89     function update($masv, $tensv)
90     {
91         $dbConnection = new DBConnection();
92         $conn = $dbConnection->getConnection();
93         $query = 'UPDATE sinhvien_tbl SET tensv = ? WHERE masv = ?';
94         $stmt = $conn->prepare($query);
95         $stmt->bind_param("si", $tensv, $masv);
96         return $stmt->execute();
97     }
98 }

```

**api.php**

```

1 <?php
2 // Turn off all error reporting
3 error_reporting(0);
4
5 // the sinhvienDAL file
6 require_once 'sinhvienDAL.php';
7
8 // message to return
9 $message = array();
10
11 $dal = new SinhvienDAL();

```



```

12
13 switch($_GET["action"])
14 {
15     case 'getall':
16         $message = $dal->getAll();
17         break;
18
19     case 'insert':
20         $masv = $_GET["masv"];
21         $tensv = $_GET["tensv"];
22         $result = $dal->insert($masv, $tensv);
23         $message = ["message" => json_encode($result)];
24         break;
25
26     case 'insertsome':
27         $data = $_GET["data"];
28         $result = $dal->insertsome($data);
29         $message = ["message" => json_encode($result)];
30         break;
31
32     case 'delete':
33         $masv = $_GET["masv"];
34         $result = $dal->delete($masv);
35         $message = ["message" => json_encode($result)];
36         break;
37
38     case 'update':
39         $masv = $_GET["masv"];
40         $tensv = $_GET["tensv"];
41         $result = $dal->update($masv, $tensv);
42         $message = ["message" => json_encode($result)];
43         break;
44
45     default:
46         $message = ["message" => "Unknown method " . $_GET["action"]];
47         break;
48 }
49
50 //The JSON message
51 header('Content-type: application/json; charset=utf-8');
52
53 //Clean (erase) the output buffer
54 ob_clean();
55
56 echo json_encode($message);

```

**Project Android****Gradle Scripts/build.gradle (Module:app)**

Thêm dòng sau vào cuối mục dependencies:

```
implementation 'com.android.volley:volley:1.1.1'
```

\* Chú ý: phiên bản 1.1.1 có thể thay đổi tùy vào thời điểm viết mã nguồn

**AndroidManifest.xml**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="vn.edu.stu.lab08_webservice">
4
5     <uses-permission android:name="android.permission.INTERNET" />
6
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"

```

## LAB 8: Webservice

```
9         android:label="@string/app_name"
10        android:roundIcon="@mipmap/ic_launcher_round"
11        android:supportsRtl="true"
12        android:theme="@style/AppTheme"
13        android:usesCleartextTraffic="true">
14        <activity android:name=".MainActivity">
15            <intent-filter>
16                <action android:name="android.intent.action.MAIN" />
17
18                <category android:name="android.intent.category.LAUNCHER" />
19            </intent-filter>
20        </activity>
21    </application>
22</manifest>
```

### model/Sinhvien.java

```
1  package vn.edu.stu.model;
2
3  public class Sinhvien {
4      private int maSV;
5      private String tenSV;
6
7      public Sinhvien() {
8      }
9
10     public Sinhvien(int maSV, String tenSV) {
11         this.maSV = maSV;
12         this.tenSV = tenSV;
13     }
14
15     public int getMaSV() {
16         return maSV;
17     }
18
19     public void setMaSV(int maSV) {
20         this.maSV = maSV;
21     }
22
23     public String getTenSV() {
24         return tenSV;
25     }
26
27     public void setTenSV(String tenSV) {
28         this.tenSV = tenSV;
29     }
30
31     @Override
32     public String toString() {
33         return "Mã: " + maSV + "\n"
34             + "Tên: " + tenSV;
35     }
36 }
```



**activity\_main.xml (giảng viên cung cấp)**

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:id="@+id/textView"
12         android:layout_width="0dp"
13         android:layout_height="wrap_content"
14         android:layout_marginStart="8dp"
15         android:layout_marginEnd="8dp"
16         android:background="#ffff00"
17         android:gravity="center"
18         android:text="Web Service"
19         android:textSize="24sp"
20         android:textStyle="bold"
21         app:layout_constraintEnd_toEndOf="parent"
22         app:layout_constraintStart_toStartOf="parent"
23         app:layout_constraintTop_toTopOf="parent" />
24
25     <TextView
26         android:id="@+id/textView2"
27         android:layout_width="55dp"
28         android:layout_height="0dp"
29         android:gravity="center_vertical"
30         android:text="Mã SV: "
31         app:layout_constraintBottom_toBottomOf="@+id/txtMaSv"
32         app:layout_constraintStart_toStartOf="@+id/textView"
33         app:layout_constraintTop_toBottomOf="@+id/textView" />
34
35     <EditText
36         android:id="@+id/txtMaSv"
37         android:layout_width="0dp"
38         android:layout_height="wrap_content"
39         android:layout_marginEnd="0dp"
40         android:inputType="number"
41         app:layout_constraintEnd_toEndOf="@+id/textView"
42         app:layout_constraintStart_toEndOf="@+id/textView2"
43         app:layout_constraintTop_toTopOf="@+id/textView2" />
44
45     <TextView
46         android:id="@+id/textView4"
47         android:layout_width="55dp"
48         android:layout_height="0dp"
49         android:gravity="center_vertical"
50         android:text="Tên SV: "
51         app:layout_constraintBottom_toBottomOf="@id/txtTenSv"
52         app:layout_constraintStart_toStartOf="@+id/textView"
53         app:layout_constraintTop_toBottomOf="@id/textView2"
54         app:layout_constraintVertical_bias="1.0" />
55
56     <EditText
57         android:id="@+id/txtTenSv"
```

```
58         android:layout_width="0dp"
59         android:layout_height="wrap_content"
60         android:inputType="textPersonName"
61         app:layout_constraintEnd_toEndOf="@+id/textView"
62         app:layout_constraintStart_toEndOf="@+id/textView2"
63         app:layout_constraintTop_toTopOf="@id/textView4" />
64
65     <Button
66         android:id="@+id/btnLuu"
67         android:layout_width="wrap_content"
68         android:layout_height="wrap_content"
69         android:layout_marginStart="8dp"
70         android:layout_marginEnd="8dp"
71         android:text="Luu"
72         app:layout_constraintEnd_toEndOf="parent"
73         app:layout_constraintStart_toStartOf="parent"
74         app:layout_constraintTop_toBottomOf="@+id/txtTenSv" />
75
76     <ListView
77         android:id="@+id/lvSv"
78         android:layout_width="0dp"
79         android:layout_height="0dp"
80         android:layout_marginStart="8dp"
81         android:layout_marginTop="8dp"
82         android:layout_marginEnd="8dp"
83         android:layout_marginBottom="8dp"
84         app:layout_constraintBottom_toBottomOf="parent"
85         app:layout_constraintEnd_toEndOf="parent"
86         app:layout_constraintStart_toStartOf="parent"
87         app:layout_constraintTop_toBottomOf="@+id/btnLuu" />
88 </android.support.constraint.ConstraintLayout>
```

**MainActivity.java**

```
1  package vn.edu.stu.lab08_webservice;
2
3  import android.net.Uri;
4  import android.os.Bundle;
5  import android.support.v7.app.AppCompatActivity;
6  import android.view.View;
7  import android.widget.AdapterView;
8  import android.widget.ArrayAdapter;
9  import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ListView;
12 import android.widget.Toast;
13
14 import com.android.volley.DefaultRetryPolicy;
15 import com.android.volley.Request;
16 import com.android.volley.RequestQueue;
17 import com.android.volley.Response;
18 import com.android.volley.VolleyError;
19 import com.android.volley.toolbox.StringRequest;
20 import com.android.volley.toolbox.Volley;
21
22 import org.json.JSONArray;
23 import org.json.JSONObject;
24
25 import java.util.ArrayList;
```

```

26
27 import vn.edu.stu.model.Sinhvien;
28
29 public class MainActivity extends AppCompatActivity {
30     // Server chứa webservice. Các bạn đổi thành sever của mình
31     final String SERVER = "http://192.168.1.8/ws/api.php";
32
33     EditText txtMaSv, txtTenSv;
34     Button btnLuu;
35     ArrayList<Sinhvien> dsSv;
36     ArrayAdapter<Sinhvien> adapter;
37     ListView lvSv;
38     Sinhvien chon = null;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_main);
44         addControls();
45         hienthiDanhSach();
46         addEvents();
47     }
48
49     private void addControls() {
50         txtMaSv = findViewById(R.id.txtMaSv);
51         txtTenSv = findViewById(R.id.txtTenSv);
52         btnLuu = findViewById(R.id.btnLuu);
53         lvSv = findViewById(R.id.lvSv);
54         dsSv = new ArrayList<>();
55         adapter = new ArrayAdapter<>(
56             MainActivity.this,
57             android.R.layout.simple_list_item_1,
58             dsSv
59         );
60         lvSv.setAdapter(adapter);
61     }
62
63     private void addEvents() {
64         btnLuu.setOnClickListener(new View.OnClickListener() {
65             @Override
66             public void onClick(View v) {
67                 if (chon == null) {
68                     // Thêm mới
69                     int maSv = Integer.parseInt(txtMaSv.getText().toString());
70                     String tenSv = txtTenSv.getText().toString();
71                     Sinhvien sv = new Sinhvien(maSv, tenSv);
72                     xuliThemSv(sv);
73                     txtMaSv.setText("");
74                     txtTenSv.setText("");
75                     txtMaSv.requestFocus();
76                 } else {
77                     // Cập nhật
78                     String tenSv = txtTenSv.getText().toString();
79                     chon.setTenSV(tenSv);
80                     xuliCapnhatSv(chon);
81                     chon = null;
82                     txtMaSv.setText("");
83                     txtTenSv.setText("");

```

```

84         txtMaSv.setEnabled(true);
85         txtMaSv.requestFocus();
86     }
87 }
88 });
89 lvSv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
90     @Override
91     public void onItemClick(AdapterView<?> parent, View view,
92                             int position, long id) {
93         if (position >= 0 && position < dsSv.size()) {
94             chon = dsSv.get(position);
95             txtMaSv.setText(chon.getMaSV() + "");
96             txtTenSv.setText(chon.getTenSV());
97             txtMaSv.setEnabled(false);
98         }
99     }
100 });
101 lvSv.setOnItemLongClickListener(
102     new AdapterView.OnItemLongClickListener() {
103         @Override
104         public boolean onItemLongClick(AdapterView<?> parent, View view,
105                                       int position, long id) {
106             if (position >= 0 && position < dsSv.size()) {
107                 Sinhvien sv = dsSv.get(position);
108                 xuliXoaSv(sv);
109             }
110             return true;
111         }
112     });
113 }
114
115 private void hienthiDanhSach() {
116     // Hàng đợi các request lên server
117     RequestQueue requestQueue = Volley.newRequestQueue(
118         MainActivity.this
119     );
120
121     // Lắng nghe kết quả trả về
122     Response.Listener<String> responseListener =
123         new Response.Listener<String>() {
124             @Override
125             public void onResponse(String response) {
126                 try {
127                     dsSv.clear();
128                     // Server trả về một chuỗi có dạng mảng JSON,
129                     // nên ta ép nó thành JSONArray rồi lặp
130                     // trên Array để lấy ra từng JSONObject
131                     JSONArray jsonArray = new JSONArray(response);
132                     int len = jsonArray.length();
133                     for (int i = 0; i < len; i++) {
134                         JSONObject jsonObject = jsonArray.getJSONObject(i);
135                         int ma = jsonObject.getInt("masv");
136                         String ten = jsonObject.getString("tensv");
137                         dsSv.add(new Sinhvien(ma, ten));
138                     }
139                     adapter.notifyDataSetChanged();
140                 } catch (Exception ex) {
141

```

```

142     }
143     };
144
145     // Lắng nghe lỗi trả về (thường là lỗi kết nối)
146     Response.ErrorListener errorListener = new Response.ErrorListener() {
147         @Override
148         public void onErrorResponse(VolleyError error) {
149             Toast.makeText(
150                 MainActivity.this,
151                 error.getMessage(),
152                 Toast.LENGTH_LONG
153             ).show();
154         }
155     };
156
157     // Tao url đến service
158     Uri.Builder builder = Uri.parse(SERVER).buildUpon();
159
160     // Chèn thêm tham số cho url, dùng trong phương thức $_GET
161     builder.appendQueryParameter("action", "getall");
162     String url = builder.build().toString();
163     StringRequest request = new StringRequest(
164         Request.Method.GET, // nếu dùng $_POST thì đổi thành POST
165         url,
166         responseListener,
167         errorListener
168     );
169
170     // Volley có xu hướng thực hiện nhiều cuộc gọi đến máy chủ chậm
171     // vì nó không nhận được phản hồi từ yêu cầu đầu tiên,
172     // nên cần cấu hình thông tin thử lại (Retry)
173     // DEFAULT_TIMEOUT_MS: Thời gian chờ tối đa trong mỗi lần thử lại. Mặc
174     // định 2500ms.
175     // DEFAULT_MAX_RETRIES: Số lần thử lại tối đa. Mặc định 1.
176     // DEFAULT_BACKOFF_MULT: Hệ số được xác định thời gian theo hàm mũ
177     // được gán cho socket trong mỗi lần thử lại. Mặc định 1.0f
178     request.setRetryPolicy(
179         new DefaultRetryPolicy(
180             DefaultRetryPolicy.DEFAULT_TIMEOUT_MS,
181             DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
182             DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
183         )
184     );
185     requestQueue.add(request);
186 }
187
188 @
189 private void xuliThemSv(Sinhvien sv) {
190     RequestQueue requestQueue = Volley.newRequestQueue(
191         MainActivity.this
192     );
193     Response.Listener<String> responseListener =
194         new Response.Listener<String>() {
195             @Override
196             public void onResponse(String response) {
197                 try {
198                     // Server trả về một chuỗi response có dạng đối tượng
199                     // JSON, nên ta ép nó thành JSONObject
200                     JSONObject jsonObject = new JSONObject(response);

```

```

200         boolean result = jsonObject.getBoolean("message");
201         if (result) {
202             Toast.makeText(
203                 MainActivity.this,
204                 "Thêm thành công",
205                 Toast.LENGTH_SHORT
206             ).show();
207             hienthiDanhsach();
208         } else {
209             Toast.makeText(
210                 MainActivity.this,
211                 "Thêm thất bại",
212                 Toast.LENGTH_SHORT
213             ).show();
214         }
215     } catch (Exception ex) {
216     }
217 }
218 };
219 Response.ErrorListener errorListener = new Response.ErrorListener() {
220     @Override
221     public void onErrorResponse(VolleyError error) {
222         Toast.makeText(
223             MainActivity.this,
224             error.getMessage(),
225             Toast.LENGTH_LONG
226         ).show();
227     }
228 };
229 Uri.Builder builder = Uri.parse(SERVER).buildUpon();
230 builder.appendQueryParameter("action", "insert");
231 builder.appendQueryParameter("masv", sv.getMaSV() + "");
232 builder.appendQueryParameter("tensv", sv.getTenSV());
233 String url = builder.build().toString();
234 StringRequest request = new StringRequest(
235     Request.Method.GET,
236     url,
237     responseListener,
238     errorListener
239 );
240 request.setRetryPolicy(
241     new DefaultRetryPolicy(
242         DefaultRetryPolicy.DEFAULT_TIMEOUT_MS,
243         DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
244         DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
245     )
246 );
247 requestQueue.add(request);
248 }
249
250 private void xuliXoaSv(Sinhvien sv) {
251     // Sinh viên tự viết
252 }
253
254 private void xuliCapnhatSv(Sinhvien sv) {
255     // Sinh viên tự viết
256 }
257 }

```



**Yêu cầu sinh viên:**

- Tiến hành cài đặt lại và chạy thử
- Viết code cho hàm xử lý xóa sinh viên và hàm xử lý cập nhật sinh viên dựa trên hàm xử lý thêm sinh viên

**2. Quản lý sinh viên – Phương thức POST**

Xây dựng chương trình quản lý sinh viên như bài 1 nhưng sử dụng phương thức POST của server:

**Webservice (giảng viên cung cấp)**

Sao chép file api.php thành file api\_post.php, chỉnh sửa tất cả các phương thức GET thành phương thức POST. Các tập tin còn lại giữ nguyên.

**Project Android**

Sao chép thư mục project Lab08\_WebService thành Lab08\_WebService\_POST. Tiến hành chỉnh sửa mã nguồn cho tập tin MainActivity.java như sau:

```
1 package vn.edu.stu.lab08_webservice;
2
3 import android.net.Uri;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.view.View;
7 import android.widget.AdapterView;
8 import android.widget.AdapterView.OnItemClickListener;
9 import android.widget.ArrayAdapter;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.ListView;
13 import android.widget.Toast;
14 import com.android.volley.AuthFailureError;
15 import com.android.volley.DefaultRetryPolicy;
16 import com.android.volley.Request;
17 import com.android.volley.RequestQueue;
18 import com.android.volley.Response;
19 import com.android.volley.VolleyError;
20 import com.android.volley.toolbox.StringRequest;
21 import com.android.volley.toolbox.Volley;
22
23 import org.json.JSONArray;
24 import org.json.JSONObject;
25
26 import java.util.ArrayList;
27 import java.util.HashMap;
28 import java.util.Map;
29
30 import vn.edu.stu.model.Sinhvien;
31
32 public class MainActivity extends AppCompatActivity {
33     // Server chứa webservice. Các bạn đổi thành sever của mình
34     final String SERVER = "http://192.168.1.8/ws/api_post.php";
35
36     EditText txtMaSv, txtTenSv;
37     Button btnLuu;
38     ArrayList<Sinhvien> dsSv;
39     ArrayAdapter<Sinhvien> adapter;
40     ListView lvSv;
41     Sinhvien chon = null;
```



```

42
43     @Override
44     protected void onCreate(Bundle savedInstanceState) {
45         super.onCreate(savedInstanceState);
46         setContentView(R.layout.activity_main);
47         addControls();
48         hienthiDanhsach();
49         addEvents();
50     }
51
52     private void addControls() {
53         txtMaSv = findViewById(R.id.txtMaSv);
54         txtTenSv = findViewById(R.id.txtTenSv);
55         btnLuu = findViewById(R.id.btnLuu);
56         lvSv = findViewById(R.id.lvSv);
57         dsSv = new ArrayList<>();
58         adapter = new ArrayAdapter<>(
59             MainActivity.this,
60             android.R.layout.simple_list_item_1,
61             dsSv
62         );
63         lvSv.setAdapter(adapter);
64     }
65
66     private void addEvents() {
67         btnLuu.setOnClickListener(new View.OnClickListener() {
68             @Override
69             public void onClick(View v) {
70                 if (chon == null) {
71                     // Thêm mới
72                     int maSv = Integer.parseInt(txtMaSv.getText().toString());
73                     String tenSv = txtTenSv.getText().toString();
74                     Sinhvien sv = new Sinhvien(maSv, tenSv);
75                     xuliThemSv(sv);
76                     txtMaSv.setText("");
77                     txtTenSv.setText("");
78                     txtMaSv.requestFocus();
79                 } else {
80                     // Cập nhật
81                     String tenSv = txtTenSv.getText().toString();
82                     chon.setTenSV(tenSv);
83                     xuliCapnhatSv(chon);
84                     chon = null;
85                     txtMaSv.setText("");
86                     txtTenSv.setText("");
87                     txtMaSv.setEnabled(true);
88                     txtMaSv.requestFocus();
89                 }
90             }
91         });
92         lvSv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
93             @Override
94             public void onItemClick(AdapterView<?> parent, View view,
95                                     int position, long id) {
96                 if (position >= 0 && position < dsSv.size()) {
97                     chon = dsSv.get(position);
98                     txtMaSv.setText(chon.getMaSV() + "");
99                     txtTenSv.setText(chon.getTenSV());

```

```

100         txtMaSv.setEnabled(false);
101     }
102 }
103 });
104 lvSv.setOnItemClickListener(
105     new AdapterView.OnItemClickListener() {
106         @Override
107         public boolean onItemClick(AdapterView<?> parent, View view,
108             int position, long id) {
109             if (position >= 0 && position < dsSv.size()) {
110                 Sinhvien sv = dsSv.get(position);
111                 xuliXoaSv(sv);
112             }
113             return true;
114         }
115     });
116 }
117
118 private void hienthiDanhSach() {
119     // Hàng đợi các request lên server
120     RequestQueue requestQueue = Volley.newRequestQueue(
121         MainActivity.this
122     );
123
124     // Lắng nghe kết quả trả về
125     Response.Listener<String> responseListener =
126         new Response.Listener<String>() {
127             @Override
128             public void onResponse(String response) {
129                 try {
130                     dsSv.clear();
131                     // Server trả về một chuỗi có dạng mảng JSON,
132                     // nên ta ép nó thành JSONArray rồi lặp
133                     // trên Array để lấy ra từng JSONObject
134                     JSONArray jsonArray = new JSONArray(response);
135                     int len = jsonArray.length();
136                     for (int i = 0; i < len; i++) {
137                         JSONObject jsonObject = jsonArray.getJSONObject(i);
138                         int ma = jsonObject.getInt("masv");
139                         String ten = jsonObject.getString("tensv");
140                         dsSv.add(new Sinhvien(ma, ten));
141                     }
142                     adapter.notifyDataSetChanged();
143                 } catch (Exception ex) {
144                 }
145             }
146         };
147
148     // Lắng nghe lỗi trả về (thường là lỗi kết nối)
149     Response.ErrorListener errorListener = new Response.ErrorListener() {
150         @Override
151         public void onErrorResponse(VolleyError error) {
152             Toast.makeText(
153                 MainActivity.this,
154                 error.getMessage(),
155                 Toast.LENGTH_LONG
156             ).show();
157         }
158     };

```

```
158     };
159
160     // Tao url đến service
161     Uri.Builder builder = Uri.parse(SERVER).buildUpon();
162
163     // Chèn thêm tham số cho url, dùng trong phương thức $_GET
164     builder.appendQueryParameter("action", "getall");
165     String url = builder.build().toString();
166     StringRequest request = new StringRequest(
167         Request.Method.POST, // nếu dùng $_POST thì đổi thành POST
168         url,
169         responseListener,
170         errorListener
171     ){
172         @Override
173         protected Map<String, String> getParams() throws AuthFailureError {
174             Map<String, String> params = new HashMap<>();
175             params.put("action", "getall");
176             return params;
177         }
178     };
179
180     // Volley có xu hướng thực hiện nhiều cuộc gọi đến máy chủ chậm
181     // vì nó không nhận được phản hồi từ yêu cầu đầu tiên,
182     // nên cần cấu hình thông tin thử lại (Retry)
183     // DEFAULT_TIMEOUT_MS: Thời gian chờ tối đa trong mỗi lần thử lại. Mặc
184     // định 2500ms.
185     // DEFAULT_MAX_RETRIES: Số lần thử lại tối đa. Mặc định 1.
186     // DEFAULT_BACKOFF_MULT: Hệ số được xác định thời gian theo hàm mũ
187     // được gán cho socket trong mỗi lần thử lại. Mặc định 1.0f
188     request.setRetryPolicy(
189         new DefaultRetryPolicy(
190             DefaultRetryPolicy.DEFAULT_TIMEOUT_MS,
191             DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
192             DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
193         )
194     );
195     requestQueue.add(request);
196 }
197
198 @ private void xuliThemSv(final Sinhvien sv) {
199     RequestQueue requestQueue = Volley.newRequestQueue(
200         MainActivity.this
201     );
202     Response.Listener<String> responseListener =
203         new Response.Listener<String>() {
204             @Override
205             public void onResponse(String response) {
206                 try {
207                     // Server trả về một chuỗi response có dạng đối tượng
208                     // JSON, nên ta ép nó thành JSONObject
209                     JSONObject jsonObject = new JSONObject(response);
210                     boolean result = jsonObject.getBoolean("message");
211                     if (result) {
212                         Toast.makeText(
213                             MainActivity.this,
214                             "Thêm thành công",
215                             Toast.LENGTH_SHORT
```

```

216         }.show();
217         hienthiDanhsach();
218     } else {
219         Toast.makeText(
220             MainActivity.this,
221             "Thêm thất bại",
222             Toast.LENGTH_SHORT
223         ).show();
224     }
225 } catch (Exception ex) {
226 }
227 }
228 };
229 Response.ErrorListener errorListener = new Response.ErrorListener() {
230     @Override
231     public void onErrorResponse(VolleyError error) {
232         Toast.makeText(
233             MainActivity.this,
234             error.getMessage(),
235             Toast.LENGTH_LONG
236         ).show();
237     }
238 };
239 Uri.Builder builder = Uri.parse(SERVER).buildUpon();
240 builder.appendQueryParameter("action", "insert");
241 builder.appendQueryParameter("masv", sv.getMaSV() + "");
242 builder.appendQueryParameter("tensv", sv.getTenSV());
243 String url = builder.build().toString();
244 StringRequest request = new StringRequest(
245     Request.Method.POST,
246     url,
247     responseListener,
248     errorListener
249 );
250 @Override
251 protected Map<String, String> getParams() throws AuthFailureError {
252     Map<String, String> params = new HashMap<>();
253     params.put("action", "insert");
254     params.put("masv", sv.getMaSV() + "");
255     params.put("tensv", sv.getTenSV());
256     return params;
257 }
258 };
259 request.setRetryPolicy(
260     new DefaultRetryPolicy(
261         DefaultRetryPolicy.DEFAULT_TIMEOUT_MS,
262         DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
263         DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
264     )
265 );
266 requestQueue.add(request);
267 }
268
269 private void xuliXoaSv(Sinhvien sv) {
270     // Sinh viên tự viết
271 }
272
273 private void xuliCapnhatSv(Sinhvien sv) {

```

```
274         // Sinh viên tự viết
275     }
276 }
```

**Yêu cầu sinh viên:**

- Tiến hành cài đặt lại và chạy thử
- Tiếp tục chỉnh code cho hàm xử lý xóa sinh viên và hàm xử lý cập nhật sinh viên dựa trên hàm xử lý thêm sinh viên

**IV. BÀI TẬP LÀM THÊM****1. Đăng kí – Đăng nhập**

Xây dựng hệ thống quản lý người dùng gồm 2 chức năng:

- Đăng kí người dùng mới
- Đăng nhập

Thông tin người dùng gồm username, password, email

**Yêu cầu:**

- Sử dụng cơ sở dữ liệu mysql
- Xây dựng WebService bằng PHP cho các chứng năng của hệ thống

**2. Tỉ giá ngoại tệ**

Xây dựng hệ thống truy vấn thông tin tỉ giá ngoại tệ gồm 2 bảng:

- Thông tin ngoại tệ: gồm mã ngoại tệ, giá mua vào, giá bán ra (sinh viên tự làm CSDL)
- Người dùng: gồm username, password, email (có thể kết hợp với bài làm thêm 1)

Chỉ những người dùng có đăng kí mới được tra cứu thông tin ngoại tệ.