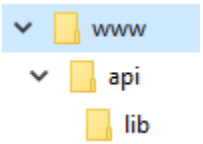
Kiểm tra lần 03 – LT Di động – Thời gian: 120 phút

Thực hiện các yêu cầu sau:



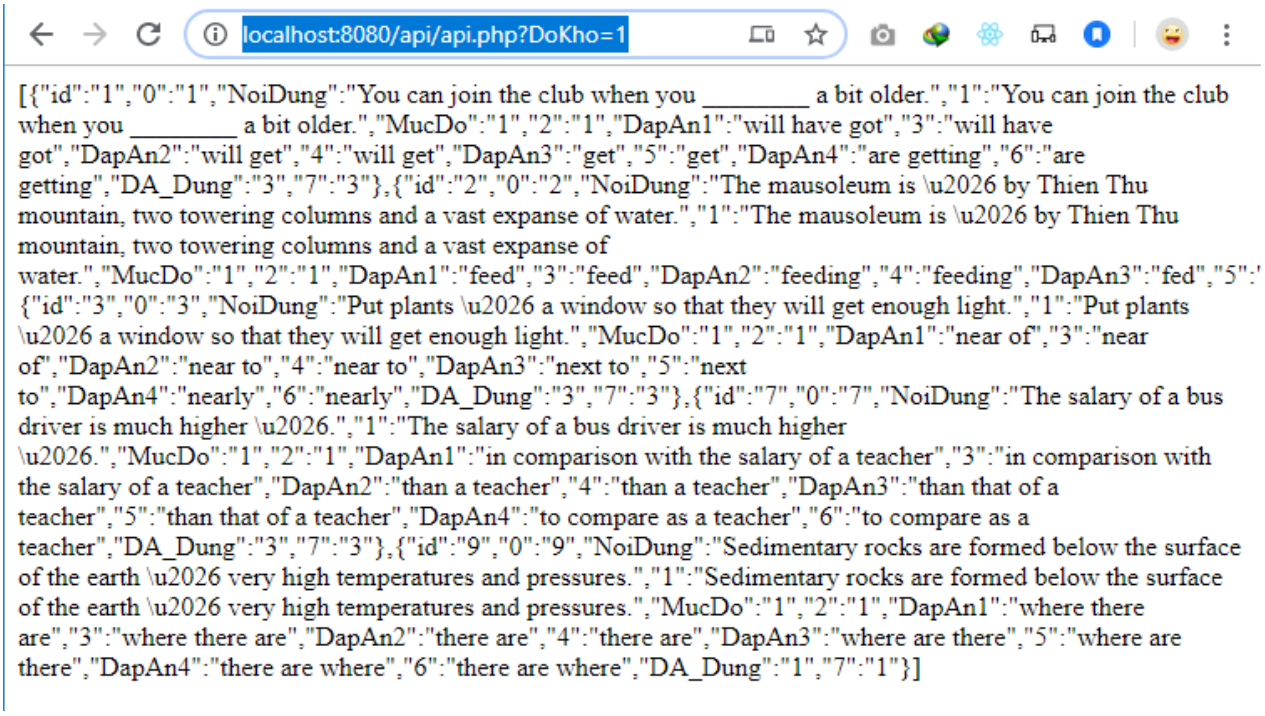
1. Mở wamp/xamp, tạo CSDL MySQL db_question, thêm dữ liệu từ tập tin question.sql
2. Copy thư mục api được cung cấp vào thư mục www (đối với wamp) hoặc htdocs (đối với xampp)
3. Với tập tin api.php, viết xử lý trả về chuỗi jSon chứa danh sách câu hỏi lấy theo độ khó

Gợi ý:

- Kiểm tra tên CSDL, tài khoản đăng nhập CSDL và mật khẩu trong db.php
- Viết xử lý gợi ý như sau:

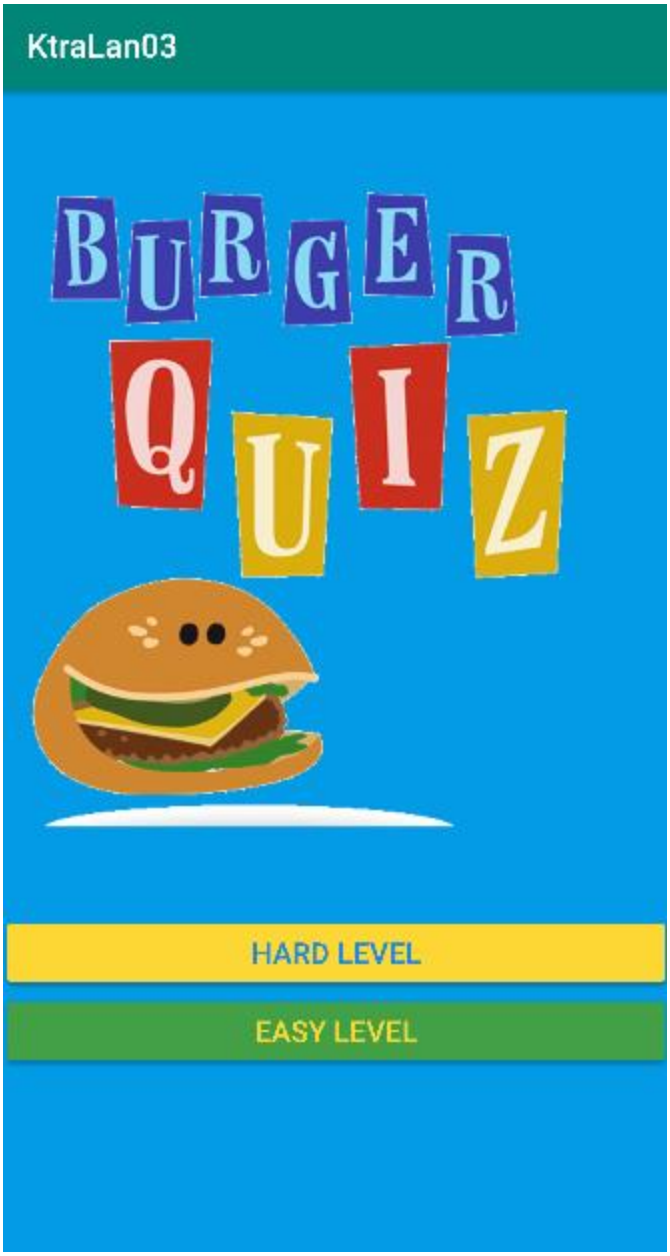
```
require_once('lib/db.php');//tập tin kết nối dữ liệu
$r=DP::run_query('SELECT `id`, `NoiDung`, `MucDo`, `DapAn1`, `DapAn2`, `DapAn3`, `DapAn4`, `DA_Dung` FROM
 `question` WHERE `MucDo`=?',[$_GET['DoKho']],2); //Truy vấn lấy về danh sách câu hỏi
echo json_encode($r); //Trả về danh sách câu hỏi dưới dạng mảng jSon
```

- Kiểm tra kết quả trên trình duyệt:



Lưu ý: Đối với việc gọi API trong android cần sử dụng địa chỉ ip được cấp cho máy, ví dụ: http://192.168.0.105:8080/api/api.php?DoKho=1

4. Tạo giao diện chính sử dụng Empty Activity như sau



Có thể sử dụng hình gif như sau:

Vào trong gradle file của module, thêm vào:

```
repositories {
    mavenCentral()
    google()
}

dependencies {
    implementation 'com.github.bumptech.glide:glide:4.10.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.10.0'
}
```

Trong mainactivity thêm lệnh:

```
Glide.with(this).load(R.drawable.logo).into((ImageView)findViewById(R.id.imageView));
```

- Khi nhấn vào HARD LEVEL sẽ tiến hành lấy danh sách câu hỏi khó về lưu vào ArrayList, chuyển sang giao diện trả lời câu hỏi, hiển thị câu hỏi đầu tiên.
- Khi nhấn vào EASY LEVEL sẽ tiến hành lấy danh sách câu hỏi khó về lưu vào ArrayList, chuyển sang giao diện trả lời câu hỏi, hiển thị câu hỏi đầu tiên.
- Thiết kế AsyncTask class cho việc gọi api lấy danh sách câu hỏi (Có thể sử dụng AsyncTask Loader)
- Thiết kế Class APICauHoi để lấy danh sách câu hỏi từ api

Tạo APIQuestion để lấy dữ liệu từ API

```
public class APIQuestion {
    static String getQuestions(String Level){
        HttpURLConnection urlConnection = null;
        BufferedReader reader = null;
        String result = null;
        try {
            URL requestURL = new URL("http://192.168.0.105:8080/api/api.php?DoKho="+Level);
            urlConnection = (HttpURLConnection) requestURL.openConnection();
            urlConnection.setRequestMethod("GET");
            urlConnection.connect();
            InputStream inputStream = urlConnection.getInputStream();
            reader = new BufferedReader(new InputStreamReader(inputStream));
            StringBuilder builder = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
                builder.append(line);
            }
            if (builder.length() == 0) {
                return null;
            }
            result = builder.toString();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (urlConnection != null) {
                urlConnection.disconnect();
            }
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        return result;
    }
}
```

Tạo AsynTask để lấy dữ liệu API

- Viết phương thức khởi tạo truyền vào Context (là MainActivity hiện tại để có thể lấy dữ liệu gửi sang activity tiếp theo, hoặc có thể xuất thông báo)

```
public class APIGetting extends AsyncTask<String, String, String> {
    private Context m_con;
    public  APIGetting(Context con)
    {
        m_con =con;
    }
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        //Toast.makeText(m_con, s, Toast.LENGTH_LONG).show();
```

```
        Intent intent = new Intent(m_con, Question.class);
        intent.putExtra("message", s);
        Activity activity = (Activity) m_con;
        activity.startActivity(intent);
    }

    @Override
    protected String doInBackground(String... level) {
        return APIQuestion.getQuestions(level[0]);
    }
}
```

Xử lý nút

```
public void btnLevel_click(View v) throws ExecutionException, InterruptedException {
    Button btn = (Button)v;
    String lst=new APIGetting(this).execute(v.getId()==R.id.btnHard?"1":"2").get();
}
```

- Thiết kế Class CauHoi để chứa các đối tượng câu hỏi và lưu vào danh sách.

```
public class QuizObject {
    public  String NoiDung;
    public  String DapAn1;
    public  String DapAn2;
    public  String DapAn3;
    public  String DapAn4;
    public  String Dung;
    public  String Chon;
    public QuizObject(){
    }
}
```

- Lưu ý: Viết 1 phương thức sử lý sự kiện cho cả 2 nút nhấn, sử dụng AsynTask để lấy tất cả dữ liệu về rồi mới hiển thị ở giao diện trả lời câu hỏi.

5. Giao diện trả lời câu hỏi

- Hiển thị danh sách câu hỏi theo thứ tự lấy xuống
    - Xử lý hiển thị các câu hỏi tương ứng khi nhấn PRE: câu hỏi trước, NEXT: câu hỏi tiếp theo
    - Ghi nhận kết quả trả lời của người dùng mỗi khi nhấn nút (có thể sử dụng cách gì tùy ý)
    - Sử dụng AsyncTask để đếm tổng thời gian trả lời trong vòng 1 phút. (Trong lúc chạy thử thì có thể để thấp hơn để tiết kiệm thời gian)
    - Xuất kết quả là số câu đúng ra màn hình khi hết thời gian
    - Xử lý xáo câu hỏi và đáp án trước khi hiển thị lên.

Gợi ý:

Kết nối API:

```
HttpURLConnection urlConnection = null;
URL requestURL = new URL("http://192.168.0.105:8080/api/api.php?DoKho=1");
urlConnection = (HttpURLConnection) requestURL.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
```

Lấy kết quả trả về:

```
InputStream inputStream = urlConnection.getInputStream();
BufferedReader reader = null;
StringBuilder builder = new StringBuilder();
String line;
while ((line = reader.readLine()) != null) {
    builder.append(line);
}
```

Chuyển sang jSon Array, jSon Object, lấy giá trị thuộc tính:

```
JSONArray jr = new JSONArray(<Chuỗi mảng đối tượng json trả về>);
JSONObject jb = (JSONObject)jr.getJSONObject(<vị trí đối tượng json trong mảng tính từ 0>);
String <tên chuỗi> = jb.getString(<Tên thuộc tính>);
```

```
public class Question extends AppCompatActivity {
    ArrayList<QuizObject> lst_cauhoi;
    int point=0;
    int cur=0;
    TextView m_txt_num;
    TextView m_txt_content;
    TextView m_rad_DA1;
    TextView m_rad_DA2;
    TextView m_rad_DA3;
    TextView m_rad_DA4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_question);
        m_txt_num = (TextView) findViewById(R.id.txtNum);
        m_txt_content = (TextView) findViewById(R.id.txtContent);
        m_rad_DA1 = (RadioButton) findViewById(R.id.radDA1);
        m_rad_DA2 = (RadioButton) findViewById(R.id.radDA2);
        m_rad_DA3 = (RadioButton) findViewById(R.id.radDA3);
        m_rad_DA4 = (RadioButton) findViewById(R.id.radDA4);
        Intent intent = getIntent();
        String jSonString = intent.getStringExtra("message");

        if( get_lst_cauhoi(jSonString)==true)
        {
            m_txt_num.setText("Câu hỏi 1:");
            m_txt_content.setText(lst_cauhoi.get(0).NoiDung);
            m_rad_DA1.setText("A. "+lst_cauhoi.get(0).DapAn1);
            m_rad_DA2.setText("B. "+lst_cauhoi.get(0).DapAn2);
            m_rad_DA3.setText("C. "+lst_cauhoi.get(0).DapAn3);
            m_rad_DA4.setText("D. "+lst_cauhoi.get(0).DapAn4);
        }
        else
        {
            m_txt_content.setText("API can not run.");
            m_txt_num.setVisibility(View.INVISIBLE);
            m_rad_DA1.setVisibility(View.INVISIBLE);
            m_rad_DA2.setVisibility(View.INVISIBLE);
            m_rad_DA3.setVisibility(View.INVISIBLE);
            m_rad_DA4.setVisibility(View.INVISIBLE);
        }
    }
    private Boolean get_lst_cauhoi(String jSonString){
        try {
            lst_cauhoi=new ArrayList();
            JSONArray jr = new JSONArray(jSonString);
            int num = jr.length();
            for(int i=0;i<num;i++)
            {
                JSONObject jb = (JSONObject)jr.getJSONObject(i);
                QuizObject quiz=new QuizObject();
                quiz.NoiDung = jb.getString("NoiDung");
```

```
                quiz.DapAn1 = jb.getString("DapAn1");
                quiz.DapAn2 = jb.getString("DapAn2");
                quiz.DapAn3 = jb.getString("DapAn3");
                quiz.DapAn4 = jb.getString("DapAn4");
                quiz.Dung = jb.getString("DA_Dung");
                quiz.Chon="0";
                lst_cauhoi.add(quiz);
            }
            return true;
        } catch (JSONException e) {
            e.printStackTrace();
            return false;
        }
    }
}
```