

ارتباط با وب سرویس WooCommerce REST API در اندروید

مدرس : خانم خدابنده

www.AndroidDevelop.org

برای آموزش های بیشتر عضو کانال تلگرام ما بشین [@AndroidDevelop](https://t.me/AndroidDevelop)



در این آموزش قصد داریم برای یک سایت وردپرسی که با افزونه ووکامرس دارای فروشگاه می باشد اپلیکیشن اندروید آماده کنیم .

مواردی که به کار گرفته شده در این نمونه:

- ✓ استفاده از `woocommerce-rest-api`
- ✓ استفاده از کتابخانه `volley` برای ارتباط با وب سرویس در اندروید
- ✓ استفاده از `RecyclerView` و `CardView` برای نمایش لیست محصولات

به این منظور ابتدا می بایست در بخش مدیریت وردپرس تنظیمات زیر را انجام دهیم:

1) وارد بخش پیکربندی ووکامرس می شویم.



2) تب API را انتخاب می کنیم.



در قسمت پیکر بندی تیک مربوط به فعال سازی REST API را می زنیم که امکان استفاده از وب سرویس مربوطه فراهم شود.

3)



در تب کلیدها/برنامه ها بروی افزودن کلید کلیک میکنیم.

سپس در پنجره ای که نمایش داده می شود در بخش توضیحات و کاربر و دسترسی مشابه زیر اطلاعات مورد نظر را ثبت کرده و بر روی دکمه ساخت کلید API کلیک میکنیم.(دسترسی از نوع خواندن/نوشتن میبایست انتخاب شود).

همگانی	محصولات	مالیات	تسویه حساب	حمل و نقل	حساب های کاربری	ایمیل ها	API
بیکربندی کلیدها/برنامه ها وب هوکها							
جزئیات کلید							
توضیحات							
<input type="text"/>							
کاربر							
<input type="text" value="admin (#1 – aa.bb@yahoo.com)"/>							
دسترسی ها							
<input type="text" value="خواندن/نوشتن"/>							
ساخت کلید API							

پس از کلیک بر روی ساخت کلید مانند شکل زیر کلید مصرف کننده و رمز مصرف کننده ساخته می شوند .
با استفاده از دکمه کپی هر دو را ذخیره کنید که در مراحل بعدی می بایست از آنها استفاده شود.

کلید API بخوبی ساخته شد. مطمئن شوید که هم اکنون یک کپی از کلیدهای API جدید را دارید. شما نمی توانید دوباره آنها را ببینید!

کپی	ck_53638fbf9d10985a0dd15be17b28f5f824366b8f	کلید مصرف کننده
کپی	cs_a6de157e6f5a2e5699814488e6a936b2942e7c57	رمز مصرف کننده
		بارکد QR

(4)

از آدرس <https://github.com/kloon/WooCommerce-REST-API-Client-Library>

پروژه نمونه را دانلود کرده و در هاست پروژه وردپرسی خود یک پوشه به نام app ایجاد کرده و در پوشه دیگری به نام test فایل‌های دانلود شده از این آدرس را قرار می‌دهیم. (نام‌ها اختیاری است و مواردی که ذکر شد در این نمونه می‌باشد).

در پوشه test فایل‌ی به نام a.php قرار داده ایم که کدهای قرار گرفته در آن مانند زیر است:

```
<?php
//header('Content-Type: application/json; charset=utf-8');
require_once 'lib/woocommerce-api.php';

$options = array(
    'debug' => false,
    'return_as_array' => false,
    'validate_url' => false,
    'timeout' => 30,
    'ssl_verify' => false,
);

try {
    $client = new WC_API_Client('http://www.sohannet.ir',
    'ck_247bb180d6409c43fb0e4d60b9d3bdb463e9a015',
    'cs_67f6121283594a485c951257bab240372a7103bf', $options);
```

آدرس سایت و کلیدهایی که در بخش بالا پیدا کردیم را در این قسمت قرار می‌دهیم.

برای نمایش محصولات در قالب
json

این دوخط اضافه شده است.

```
// products
// print_r($client->products->get());
$json = json_encode($client->products->get(), JSON_UNESCAPED_UNICODE);
echo $json;

//print_r( $client->products->get( $product_id ) );
//print_r( $client->products->get( $variation_id ) );
//print_r( $client->products->get_by_sku( 'a-product-sku' ) );
//print_r( $client->products->create( array( 'title' => 'Test Product',
'type' => 'simple', 'regular_price' => '9.99', 'description' => 'test' ) ) );
//print_r( $client->products->update( $product_id, array( 'title' => 'Yet
another test product' ) ) );
//print_r( $client->products->delete( $product_id, true ) );
//print_r( $client->products->get_count() );
//print_r( $client->products->get_count( array( 'type' => 'simple' ) ) );
//print_r( $client->products->get_categories() );
//print_r( $client->products->get_categories( $category_id ) );
} catch (WC_API_Client_Exception $e) {

    echo $e->getMessage() . PHP_EOL;
    echo $e->getCode() . PHP_EOL;

    if ($e instanceof WC_API_Client_HTTP_Exception) {

        print_r($e->get_request());
        print_r($e->get_response());
    }
}
```

خروجی کار ما در این قسمت یک فایل json میباشد که از طریق `url`:

`http://sohannet.ir/app/test/a.php` قابل دسترسی است و در اپلیکیشن اندروید می بایست اطلاعات

این فایل را گرفته و تجزیه (parse) و نمایش دهیم.

تصویر زیر بخشی از این فایل را نشان می دهد:

```

{"products":[{"title":"سوهان
گزی","id":23,"created_at":"2015-12-16T12:48:50Z","updated_at":"2015-12-16T12:48:50Z","type":"simple","status":"publish","downloadable":false,"virtual":false,"permalink":"http://www.sohannet.ir/product/%d8%b3%d9%88%d9%87%d8%a7%d9%86-%da%af%d8%b2%db%8c
V","sku":"5567","price":"28000","regular_price":"28000","sale_price":null,"price_html":"28.000 تومان</span>","taxable":false,"tax_status":"taxable","tax_class":"","managing_stock":false,"stock_qu
{"length":"","width":"","height":"","unit":"cm"},"shipping_required":true,"shipping_taxable":true,"shipping_class":"","shipping_class_id":null,"description":
سوهان
</p>\n
سوهان
short_description":"","reviews_allowed":true,"average_rating":"0.00","rating_count":0,"related_ids":[19],"upsell_ids":
[],"cross_sell_ids":[],"parent_id":0,"categories":["سوهان مرتضوی","سوهان ساعدی","سوهان ساعدی"],"tags":[],"images":[{"id":24,"created_at":"2015-12-16T12:48:38Z","updated_at":"2015-12-16T12:48:38Z","src":"http://www.sohannet.ir/wp-content/uploads/2015/12/vq1.jpg","title":"q1","alt":"","position":0},"featured_src":"http://www.sohannet.ir/wp-content/uploads/2015/12/vq1.jpg","attributes":
[],"downloads":[],"download_limit":0,"download_expiry":0,"download_type":"","purchase_note":"","total_sales":0,"variations":[],"parent":{},"title":"سوهان
سوهان","id":19,"created_at":"2015-12-16T09:18:01Z","updated_at":"2015-12-16T09:18:01Z","type":"simple","status":"publish","downloadable":false,"virtual":false,"permalink":"http://www.sohannet.ir/product/%d8%b3%d9%88%d9%87%d8%a7%d9%86-%d8%b9%d8%b3%d9%84%db%8c
عسل","sku":"5566","price":"36000","regular_price":"36000","sale_price":null,"price_html":"36.000 تومان</span>","taxable":false,"tax_status":"taxable","tax_class":"","managing_stock":false,"stock_qu
{"length":"","width":"","height":"","unit":"cm"},"shipping_required":true,"shipping_taxable":true,"shipping_class":"","shipping_class_id":null,"description":

```

چنانچه دقت کنید محصولات فروشگاه در آرایه ای به نام **products** قرار دارند و عنوان محصول در المان **title** و توضیحات محصول در المان **description** و قیمت محصول در **price** و همچنین عکس محصول در المان **src** در آرایه ای به نام **images** قابل دسترسی است.

Json چیست؟

JSON شیوه‌ای است که با کمک آن اشیا و اطلاعات برنامه‌نویسی به متن قابل پردازش توسط رایانه و قابل فهم برای انسان تبدیل می‌شوند.

نکته مهم در این شیوه قابلیت فهم آسان توسط انسان است که در دیگر شیوه‌ها مانند XML کمتر به آن توجه شده است .

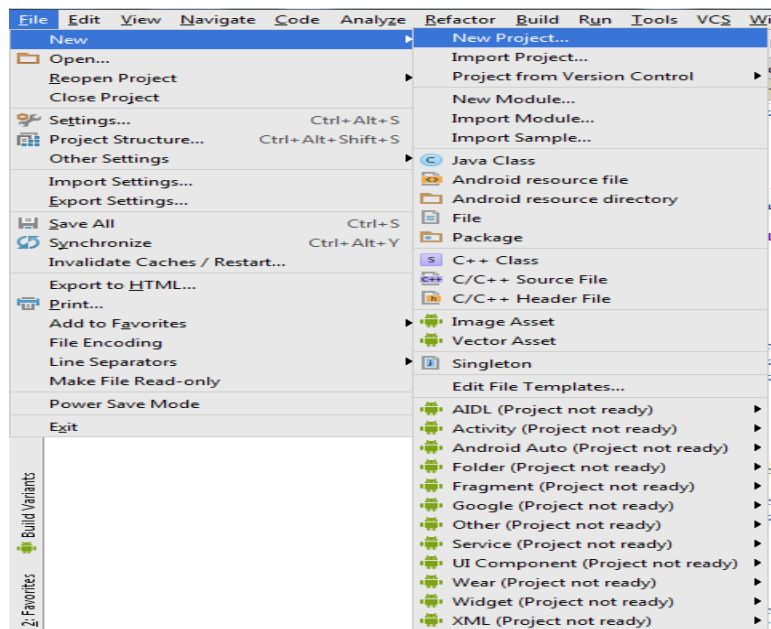
تصویر زیر نمونه ای از Json می باشد که در آن **employees** نام آرایه می باشد و این آرایه دارای المان های **firstname** و **lastname** می باشد که هر کدام از عناصر آرایه درون **{ }** قرار می گیرند و با , از هم جدا می شوند. **[]** نماد آرایه می باشد.

```
{ "employees": [
  { "firstName": "John", "lastName": "Doe" },
  { "firstName": "Anna", "lastName": "Smith" },
  { "firstName": "Peter", "lastName": "Jones" }
]}
```

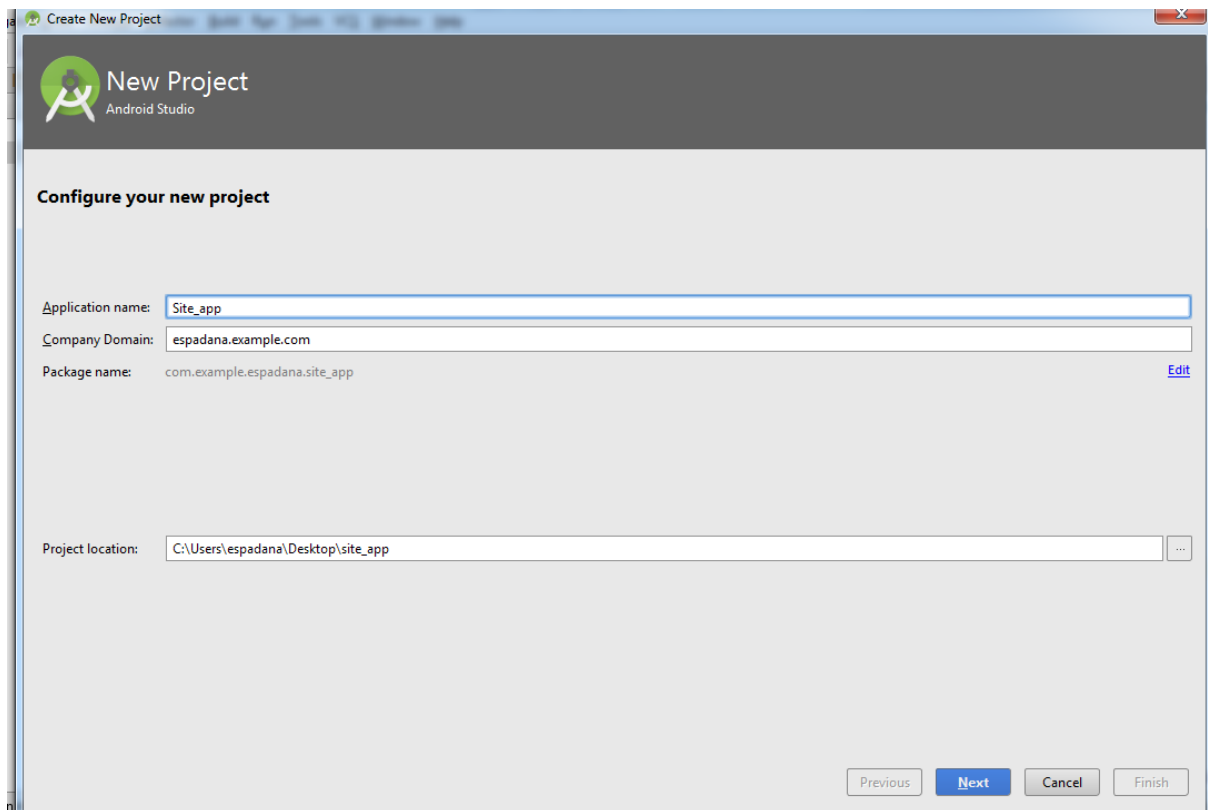
اندروید استادیو

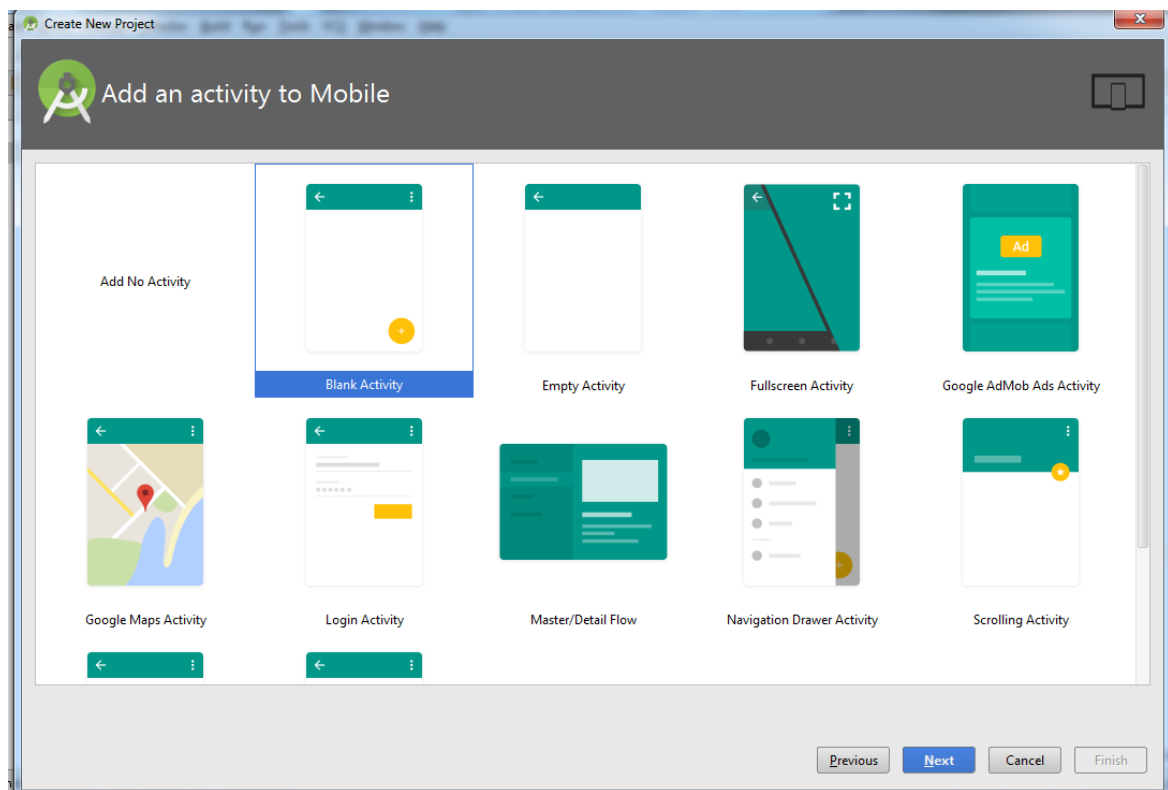
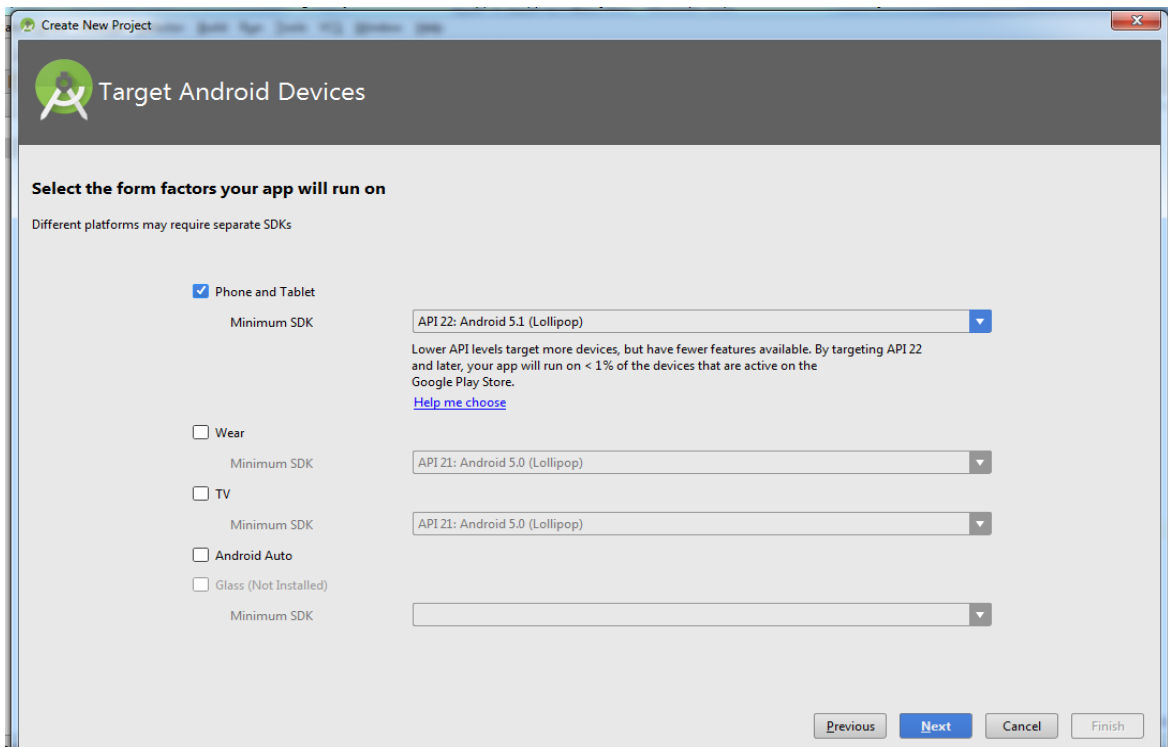
پروژه جدید ایجاد میکنیم:

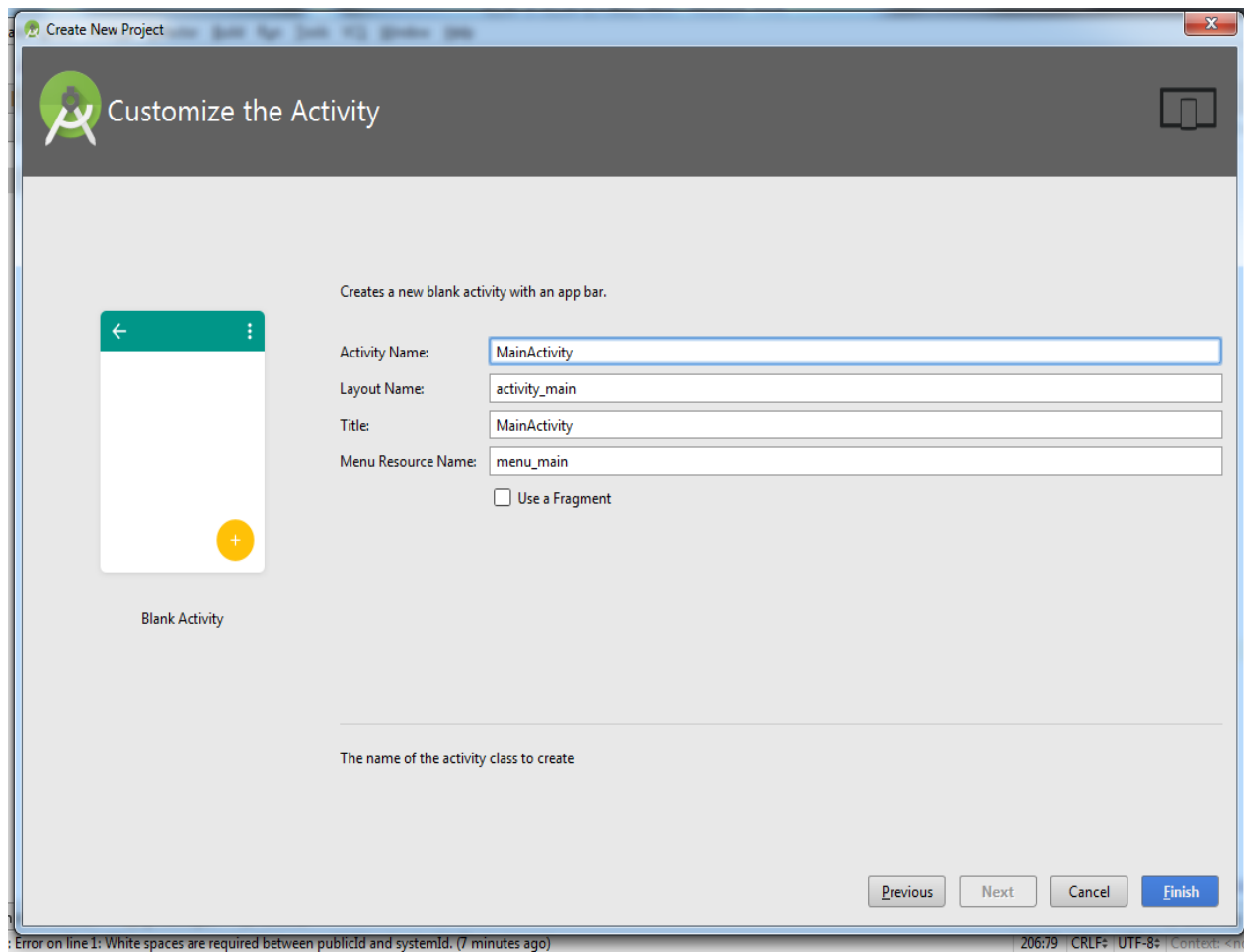
File → New → New Project

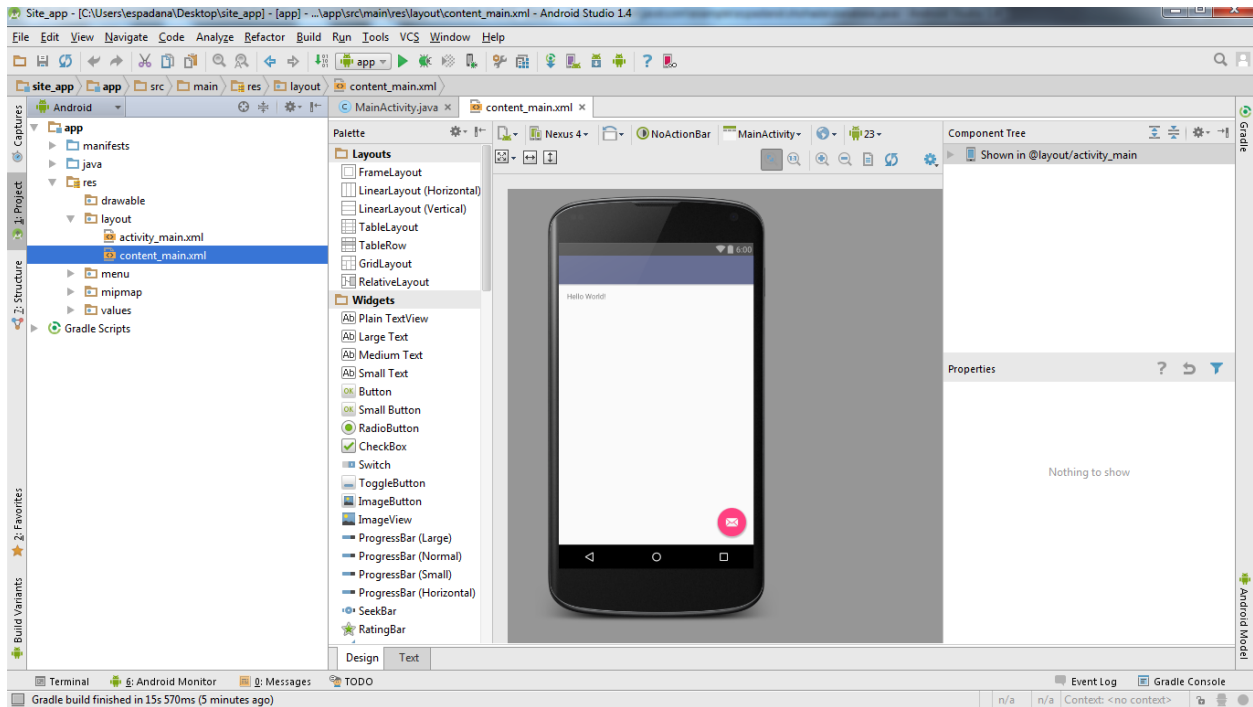


نام و مسیر دخیره پروژه اپلیکیشن را وارد میکنیم:



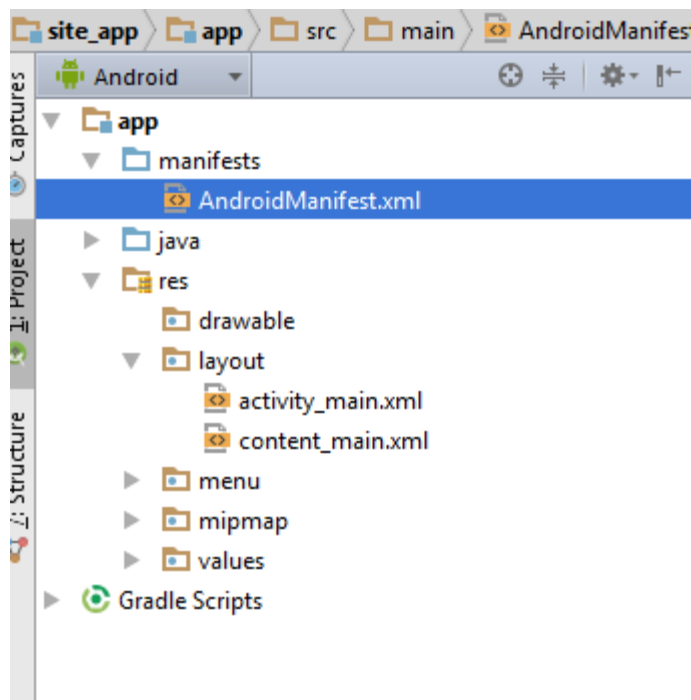






ابتدا در فایل manifest دسترسی اینترنت را اضافه می کنیم:

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

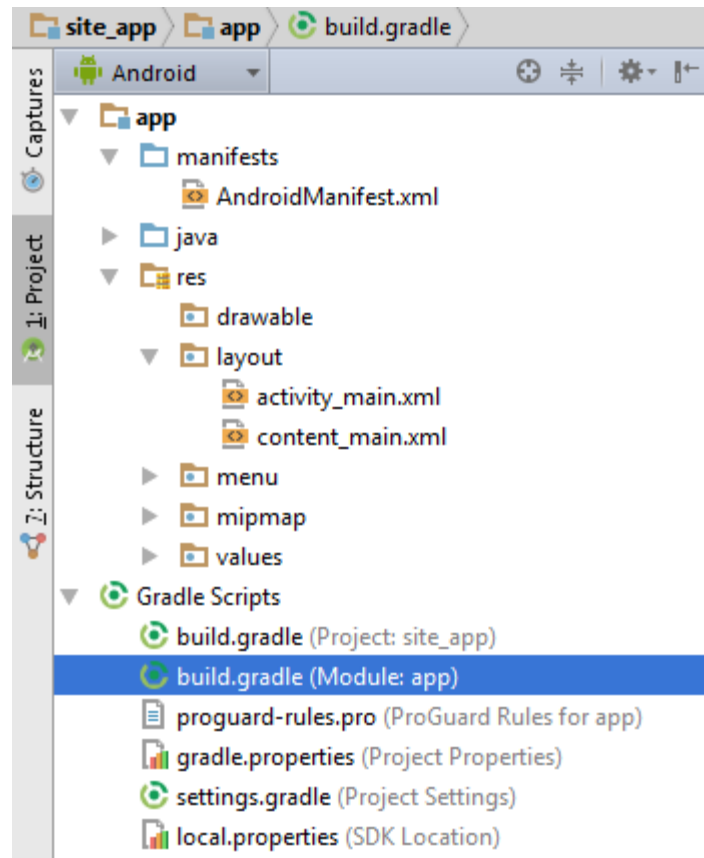


```
MainActivity.java × content_main.xml × AndroidManifest.xml ×
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.espadana.site_app" >
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Site_app"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="Site_app"
            android:theme="@style/AppTheme.NoActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

در فایل build.gradle جهت استفاده از کتابخانه volley خط زیر را اضافه میکنیم:

```
compile 'com.mcxiaoke.volley:library:1.0.18'
```



```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.1.1'  
    compile 'com.android.support:design:23.1.1'  
    compile 'com.mcxiaoke.volley:library:1.0.18'  
}
```

کتابخانه volley چیست؟

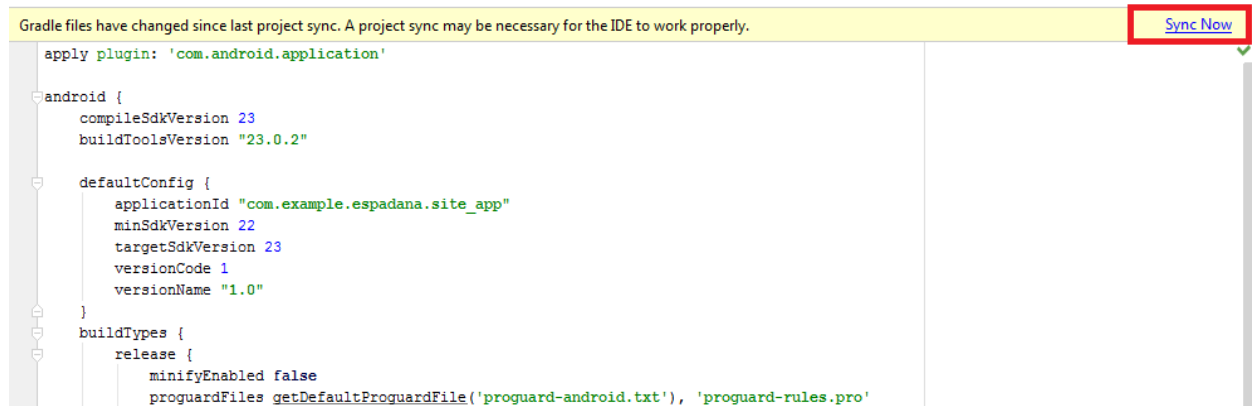
همچنین برای استفاده از **recyclerview** و **cardview** دو خط زیر را در همین قسمت اضافه میکنیم:

```
compile 'com.android.support:recyclerview-v7:+'
compile 'com.android.support:cardview-v7:+'
```

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'

    compile 'com.mcxiaoke.volley:library:1.0.18'
    compile 'com.android.support:recyclerview-v7:+'
    compile 'com.android.support:cardview-v7:+'
}
```

سپس روی **Sync Now** در قسمت بالا کلیک میکنیم:



RecyclerView چیست؟

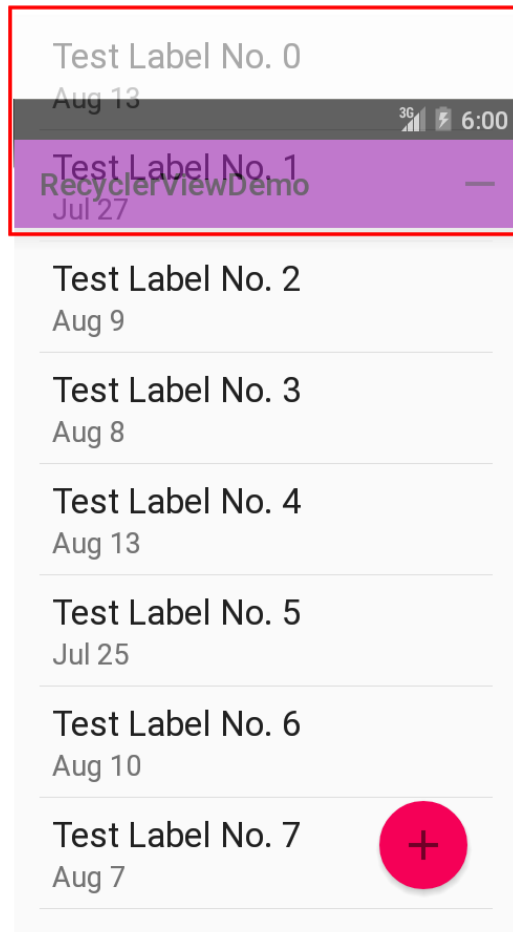
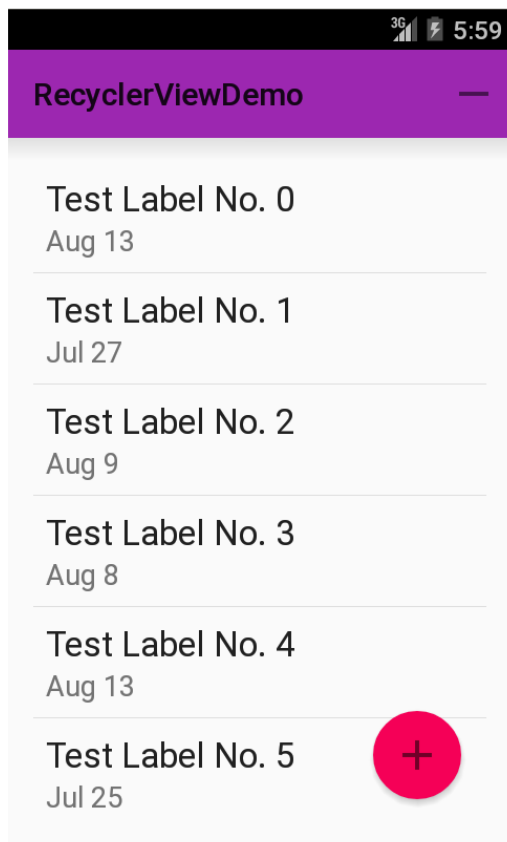
بنابر تعریف Documentation رسمی اندروید RecyclerView نسخه بهینه تر و انعطاف پذیرتری از ListView می باشد. توصیه شده وقتی که میخوایم حجم بالایی از اطلاعات رو در لیست نمایش بدیم و وقتی که قراره محتویات لیست دائما تغییر کنه (مثلا با دریافت اطلاعات از اینترنت یا مثلا حذف و اضافه آیتم ها توسط کاربر) به جای listview از RecyclerView استفاده بشه. اما اگر مثلا قراره در برنامه یه لیست ایستا شامل مثلا 20 آیتم داشته باشید هیچ دلیلی نداره که از RecyclerView استفاده کنید و این ویجت برای ایجاد لیست های پیشرفته طراحی شده (احتمالا در آینده به طور کلی جای ListView رو بگیره).

برخی از ویژگی های RecyclerView:

برخلاف ListView رعایت الگوی طراحی ViewHolder در RecyclerView الزامیست.

برخلاف ListView که تنها حالت قرارگیری آیتم های لیست به صورت عمودی بود در RecyclerView می توان به چندین حالت آیتم های لیست در کنارهم قرار بگیرند (البته در ListView هم تا حدودی این کار قابل انجامه اما کار بیشتری میبره)

در RecyclerView می توان به راحتی برای حذف و اضافه آیتم ها از لیست انیمیشن تعریف کرد.



RecyclerView با سه عنصر اصلی زیر کار میکند:

ViewHolder: ویو هولدر یک وظیفه ساده دارد نگهداری یک آیتم از لیست. اگر یک **ViewHolder** خالی باشد یعنی باید یک **View** جدید (آیتم جدید) ایجاد بشود و داخل ویو هولدر قرار بگیرد اگر هم از قبل یک **View** نگهداری کنه آنوقت از همان قبلی استفاده میشود.

Adapter: ادپتر دو وظیفه بسیار مهم دارد: ساخت **View** یا همان آیتم لیست و قرار دادن آن در یک **ViewHolder** و چسباندن اطلاعات به هر آیتم (**Data Binding**)

LayoutManager : نحوه نمایش آیتم های لیست (عمودی، افقی، گرید و ...) به عهده LayoutManager

است.

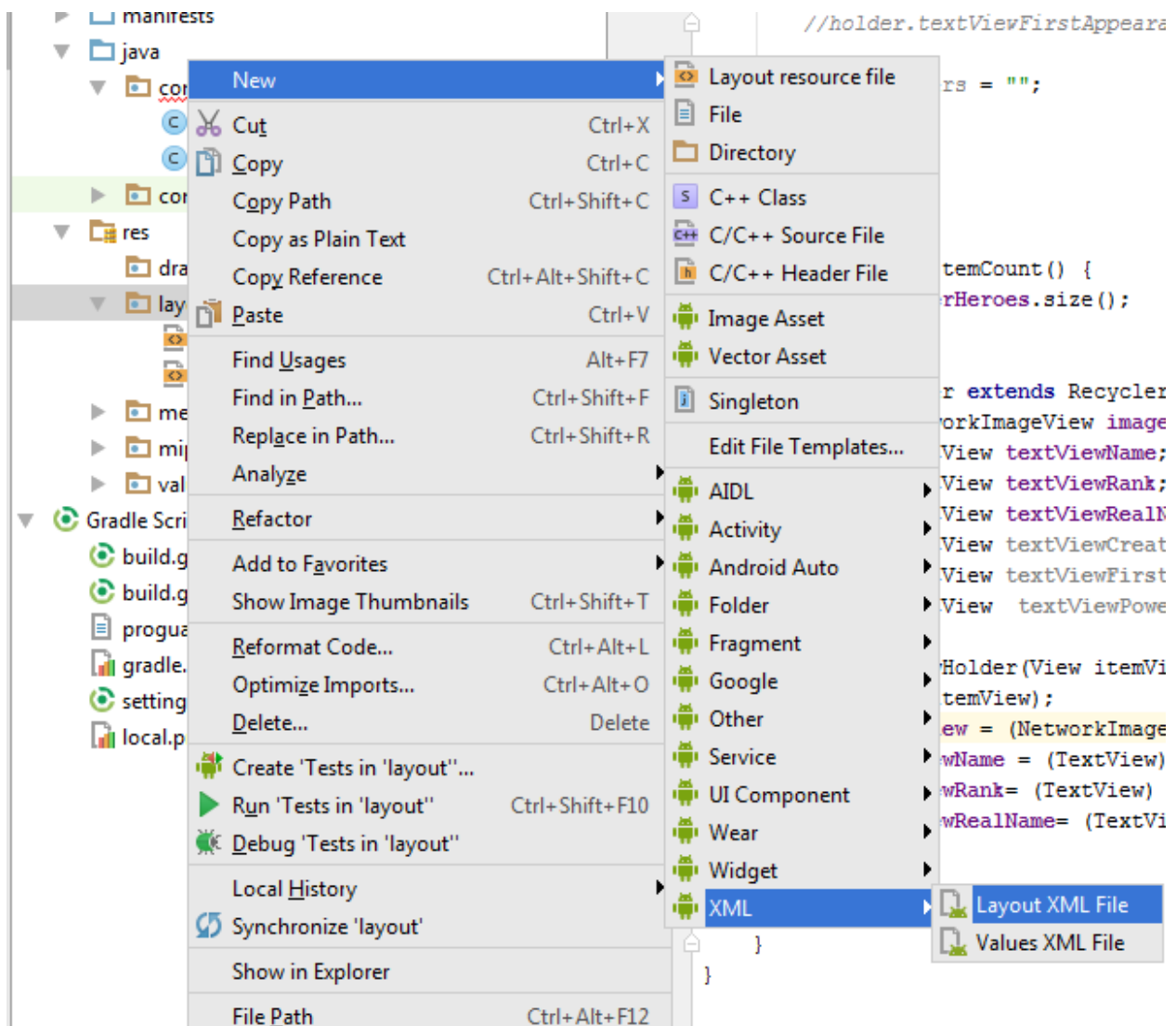
در قسمت layout <-- activity_main یک RecyclerView اضافه می کنیم:

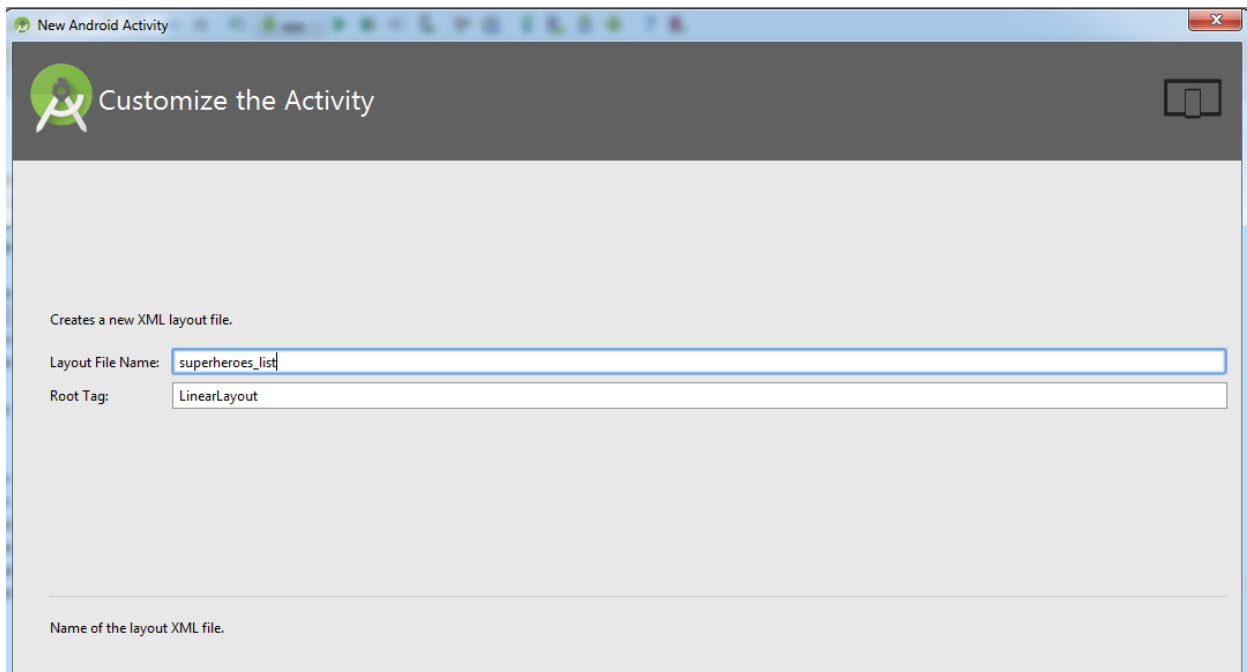
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:paddingLeft="40dp">
    </android.support.v7.widget.RecyclerView>

</LinearLayout>
```

همچنین فایلی به نام superheroes_list در بخش layout ایجاد می کنیم و از طریق کدهای زیر ویجت هایی که می خواهیم در Recycle View نمایش دهیم را مشخص می کنیم:





```

<?xml version="1.0" encoding="utf-8" ?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:padding="@dimen/activity_horizontal_margin"
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >

            <com.android.volley.toolbox.NetworkImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/imageViewHero" />

            <TableLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

                <TableRow>
                    <TextView
                        android:id="@+id/textViewName"
                        android:textStyle="bold"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                    />
                    <TextView
                        android:text="محصول نام"

```

```

        android:paddingRight="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

</TableRow>

<TableRow>
    <TextView
        android:id="@+id/textViewRank"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
    <TextView
        android:text=": قیمت"
        android:paddingRight="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

</TableRow>

<TableRow>
    <TextView
        android:id="@+id/textViewRealName"
        android:textStyle="bold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
    <TextView
        android:text="توضیحات"
        android:paddingRight="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

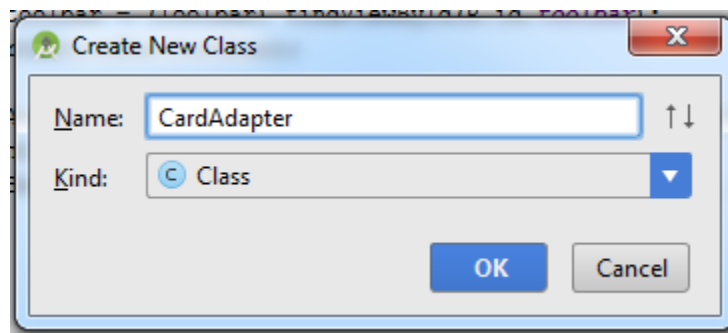
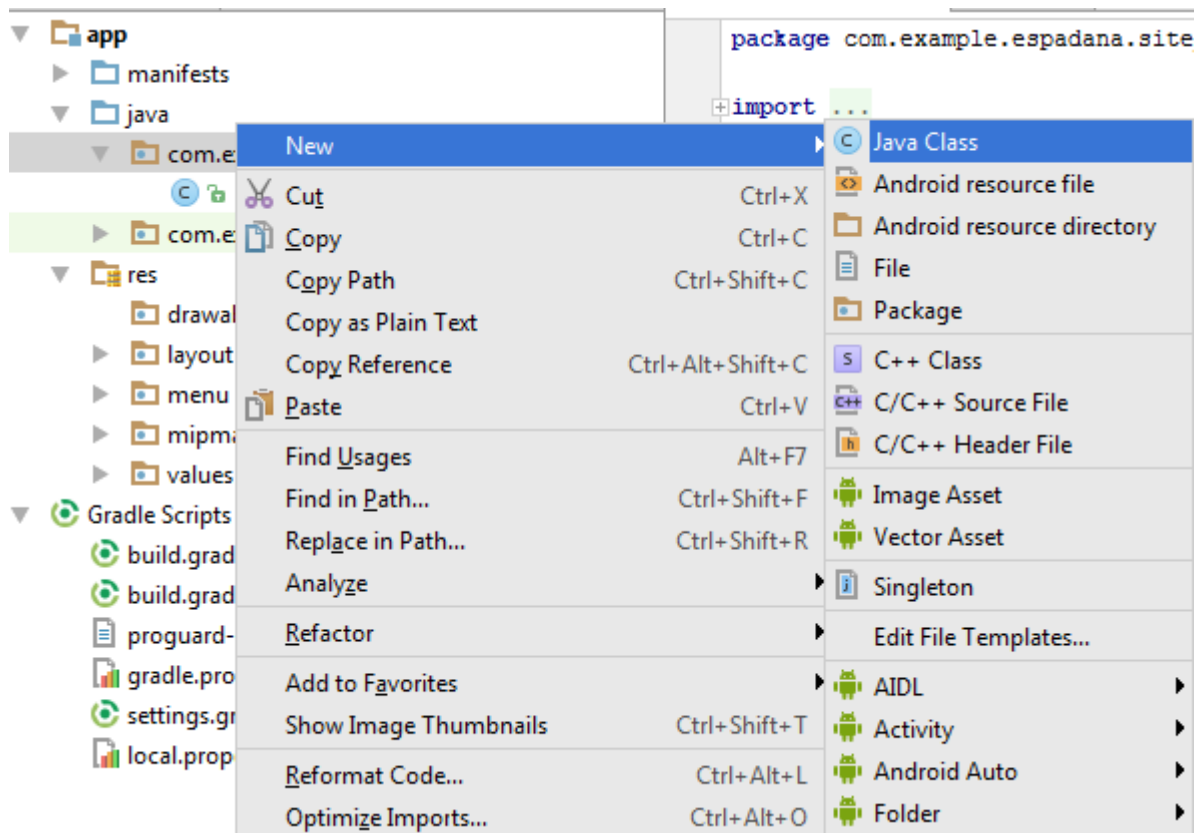
</TableRow>

</TableLayout>
</LinearLayout>
</android.support.v7.widget.CardView>

</RelativeLayout>

```

در بخش جاوا کلاسی به نام CardAdapter ایجاد کرده و کدهای زیر را در آن اضافه میکنیم:



```

import android.content.Context;
import android.media.Image;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;

import org.w3c.dom.Text;

```

```

import java.util.ArrayList;
import java.util.List;

public class CardAdapter extends RecyclerView.Adapter<CardAdapter.ViewHolder> {

    private ImageLoader imageLoader;
    private Context context;

    //List of superHeroes
    List<SuperHeroes> superHeroes;

    public CardAdapter(List<SuperHeroes> superHeroes, Context context){
        super();
        //Getting all the superheroes
        this.superHeroes = superHeroes;
        this.context = context;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.superheroes_list, parent, false);
        ViewHolder viewHolder = new ViewHolder(v);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {

        SuperHeroes superHero = superHeroes.get(position);

        imageLoader = CustomVolleyRequest.getInstance(context).getImageLoader();
        imageLoader.get(superHero.getImageUrl(),
            ImageLoader.getImageListener(holder.imageView, R.mipmap.ic_launcher,
            android.R.drawable.ic_dialog_alert));

        holder.imageView.setImageUrl(superHero.getImageUrl(), imageLoader);
        holder.textViewName.setText(superHero.getName());
        holder.textViewRank.setText(String.valueOf(superHero.getRank()));
        holder.textViewRealName.setText(superHero.getRealName());
        // holder.textViewCreatedBy.setText(superHero.getCreatedBy());
        //holder.textViewFirstAppearance.setText(superHero.getFirstAppearance());

        String powers = "";

    }

    @Override
    public int getItemCount() {
        return superHeroes.size();
    }

    class ViewHolder extends RecyclerView.ViewHolder{
        public NetworkImageView imageView;
        public TextView textViewName;
        public TextView textViewRank;
        public TextView textViewRealName;
        public TextView textViewCreatedBy;
        public TextView textViewFirstAppearance;
        public TextView textViewPowers;
    }
}

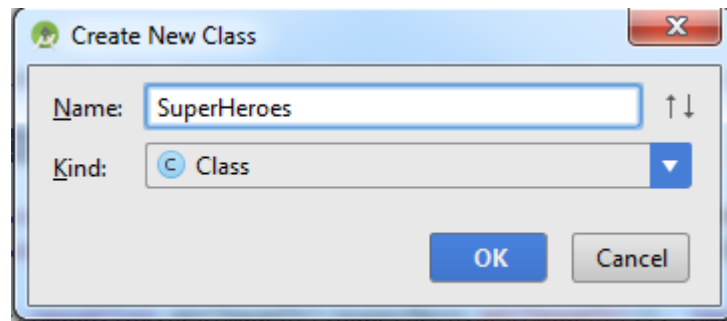
```

```

public ViewHolder(View itemView) {
    super(itemView);
    imageView = (NetworkImageView) itemView.findViewById(R.id.imageViewHero);
    textViewName = (TextView) itemView.findViewById(R.id.textViewName);
    textViewRank = (TextView) itemView.findViewById(R.id.textViewRank);
    textViewRealName = (TextView) itemView.findViewById(R.id.textViewRealName);
}
}
}

```

همچنین کلاس دیگری به نام SuperHeroes ایجاد میکنیم و کدهای زیر را در آن قرار می دهیم:



```

import java.util.ArrayList;

public class SuperHeroes {

    //Data Variables
    private String imageUrl;
    private String name;
    private String rank;
    private String realName;
    private String createdBy;
    private String firstAppearance;
    private ArrayList<String> powers;

    //Getters and Setters
    public String getImageUrl() {
        return imageUrl;
    }

    public void setImageUrl(String imageUrl) {
        this.imageUrl = imageUrl;
    }

    public String getName() {
        return name;
    }
}

```

```

public void setName(String name) {
    this.name = name;
}

public String getRank() {
    return rank;
}

public void setRank(String rank) {
    this.rank = rank;
}

public String getRealName() {
    return realName;
}

public void setRealName(String realName) {
    this.realName = realName;
}

public String getCreatedBy() {
    return createdBy;
}

public void setCreatedBy(String createdBy) {
    this.createdBy = createdBy;
}

public String getFirstAppearance() {
    return firstAppearance;
}

public void setFirstAppearance(String firstAppearance) {
    this.firstAppearance = firstAppearance;
}

public ArrayList<String> getPowers() {
    return powers;
}

public void setPowers(ArrayList<String> powers) {
    this.powers = powers;
}
}

```

کلاس دیگری به نام CustomVolleyRequest ایجاد کرده و کدهای زیر را ر آن قرار می دهیم:

```

import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;

import com.android.volley.Cache;
import com.android.volley.Network;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.BasicNetwork;
import com.android.volley.toolbox.DiskBasedCache;

```



```

import com.android.volley.toolbox.HurlStack;
import com.android.volley.toolbox.ImageLoader;

public class CustomVolleyRequest {

    private static CustomVolleyRequest customVolleyRequest;
    private static Context context;
    private RequestQueue requestQueue;
    private ImageLoader imageLoader;

    private CustomVolleyRequest(Context context) {
        this.context = context;
        this.requestQueue = getRequestQueue();

        imageLoader = new ImageLoader(requestQueue,
            new ImageLoader.ImageCache() {
                private final LruCache<String, Bitmap>
                    cache = new LruCache<String, Bitmap>(20);

                @Override
                public Bitmap getBitmap(String url) {
                    return cache.get(url);
                }

                @Override
                public void putBitmap(String url, Bitmap bitmap) {
                    cache.put(url, bitmap);
                }
            });
    }

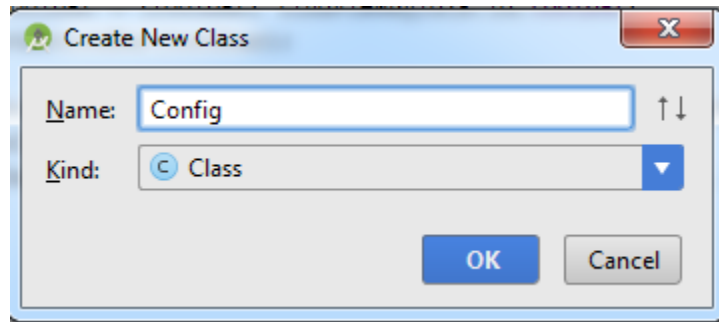
    public static synchronized CustomVolleyRequest getInstance(Context context) {
        if (customVolleyRequest == null) {
            customVolleyRequest = new CustomVolleyRequest(context);
        }
        return customVolleyRequest;
    }

    public RequestQueue getRequestQueue() {
        if (requestQueue == null) {
            Cache cache = new DiskBasedCache(context.getCacheDir(), 10 * 1024 * 1024);
            Network network = new BasicNetwork(new HurlStack());
            requestQueue = new RequestQueue(cache, network);
            requestQueue.start();
        }
        return requestQueue;
    }

    public ImageLoader getImageLoader() {
        return imageLoader;
    }
}

```

کلاس دیگری به نام Config داریم که در آن آدرس فایل json و عناصر json که نیاز به تجزیه داریم را مشخص می کنیم:



```
public class Config {  
    //URL of my API  
    public static final String DATA_URL = "http://sohannet.ir/app/test/a.php";  
  
    //Tags for my JSON  
    public static final String TAG_IMAGE = "images";  
    public static final String TAG_IMAGE_URL = "src";  
    public static final String TAG_NAME = "title";  
    public static final String TAG_RANK = "price";  
    public static final String TAG_REAL_NAME = "description";  
  
}
```

کدهای مربوط به کلاس MainActivity نیز مانند زیر می باشد :

```
import android.app.ProgressDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonArrayRequest;  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.Volley;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;
```

```

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    //Creating a List of superheroes
    private List<SuperHeroes> listSuperHeroes;

    //Creating Views
    private RecyclerView recyclerView;
    private RecyclerView.LayoutManager layoutManager;
    private RecyclerView.Adapter adapter;

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

        //Initializing Views
        recyclerView = (RecyclerView) findViewById(R.id.recyclerView);

        //Initializing our superheroes list
        listSuperHeroes = new ArrayList<>();

        //Calling method to get data
        getData();
    }

    //This method will get data from the web api
    private void getData(){
        //Showing a progress dialog
        final ProgressDialog loading = ProgressDialog.show(this, "اطلاعات دریافت در حال", "لطفاً
        کنید صبر", false, false);

        //Creating a json array request
        JsonObjectRequest jsonArrayRequest = new JsonObjectRequest(Config.DATA_URL,
            new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    //Dismissing progress dialog
                    loading.dismiss();
                    try {
                        JSONArray obj_products =
response.getJSONArray("products");

                        parseData(response.getJSONArray("products"));
                    }
                    catch (Exception e)
                    {

                    }

                    //calling method to parse json array
                }
            },
            new Response.ErrorListener() {

```

```

        @Override
        public void onErrorResponse(VolleyError error) {

        }
    });

    //Creating request queue
    RequestQueue requestQueue = Volley.newRequestQueue(this);

    //Adding request to the queue
    requestQueue.add(jsonArrayRequest);
}

//This method will parse json data
private void parseData(JSONArray array) {
    for(int i=0; i<array.length(); i++) {

        SuperHeroes superHero = new SuperHeroes();
        JSONObject json = null;
        try {

            json = array.getJSONObject(i);

            superHero.setName(json.getString(Config.TAG_NAME));
            superHero.setRank(json.getString(Config.TAG_RANK));

superHero.setRealName(json.getString(Config.TAG_REAL_NAME).toString().replace("<p>", "
").replace("</p>", " "));

            JSONArray products = json.getJSONArray(Config.TAG_IMAGE);
            String image_url="";
            for(int k=0;k<products.length();k++) {
                JSONObject new_json = products.getJSONObject(k);
                image_url = new_json.getString(Config.TAG_IMAGE_URL);
            }
            superHero.setImageUrl(image_url);

            ArrayList<String> addList = new ArrayList<String>();
            JSONArray addArray = json.getJSONArray(Config.TAG_IMAGE);
            int size = addArray.length();

        } catch (JSONException e) {
            e.printStackTrace();
        }
        listSuperHeroes.add(superHero);
    }

    //Finally initializing our adapter
    adapter = new CardAdapter(listSuperHeroes, this);
    recyclerView.setLayoutManager(new LinearLayoutManager(MainActivity.this));
    recyclerView.setHasFixedSize(true);
    //Adding adapter to recyclerview
    recyclerView.setAdapter(adapter);
}
}

```

قسمت مهم مربوط به گرفتن و تجزیه اطلاعات در این فایل می باشد که قسمتی از کدها توضیح داده می شوند:

در ابتدای ایجاد اکتیوییتی تابع `getData` فراخوانی می شود که در آن ابتدا برای نمایش مقدار پیشرفت، کد

```
final ProgressDialog loading = ProgressDialog.show
```

قرار داده شده و بعد از آن

```
JsonObjectRequest jsonArrayRequest = new JsonObjectRequest(Config.DATA_URL,  
    new Response.Listener<JSONObject>() {
```

استفاده شده که همانطور که میبینید از طریق پارامتر اول (`Config.DATA_URL`) آدرس `url` که در بخش قبل آماده کردیم را میفرستیم .

`JsonObjectRequest` دارای دو تابع می باشد :

`onResponse`: کدهایی که مربوط به درست برقرار شدن ارتباط نیاز است در این قسمت قرار می گیرند.

`onErrorResponse`: چنانچه در ارتباط خطایی رخ دهد به منظور کنترل آنها در این قسمت کدهای مربوطه مثل لاگ ها قرار میگیرند.

که ما در بخش `onResponse` کدهای زیر را قرار داده ایم :

```
loading.dismiss();  
try {  
  
    parseData(response.getJSONArray("products"));  
}  
catch (Exception e)  
{  
  
}
```

که در آن `response` را به صورت آرایه به تابع `parseData` ارسال می کنیم (نام آرایه در فایل `json`، `products` می باشد).

در تابع `parseData` نیز عناصر آرایه گرفته شده را تجزیه می کنیم.

```
private void parseData(JSONArray array){  
    for(int i =0; i<array.length(); i++) {  
  
        SuperHeroes superHero = new SuperHeroes();  
        JSONObject json = null;  
        try {  
  
            json = array.getJSONObject(i);  
  
            superHero.setName(json.getString(Config.TAG_NAME));  
            superHero.setRank(json.getString(Config.TAG_RANK));  
            superHero.setRealName(json.getString(Config.TAG_REAL_NAME));
```

```

JSONArray products = json.getJSONArray(Config.TAG_IMAGE);
String image_url="";
for(int k=0;k<products.length();k++) {
    JSONObject new_json = products.getJSONObject(k);
    image_url = new_json.getString(Config.TAG_IMAGE_URL);
}
superHero.setImageUrl(image_url);

ArrayList<String> addList = new ArrayList<String>();
JSONArray addArray = json.getJSONArray(Config.TAG_IMAGE);
int size = addArray.length();

} catch (JSONException e) {
    e.printStackTrace();
}
listSuperHeroes.add(superHero);
}

//Finally initializing our adapter
adapter = new CardAdapter(listSuperHeroes, this);

//Adding adapter to recyclerview
recyclerView.setAdapter(adapter);

recyclerView.setLayoutManager(new LinearLayoutManager(MainActivity.this));
recyclerView.setHasFixedSize(true);
}

```

دقت کنید آدرس تصویر محصول چون در آرایه دیگری قرار دارد از یک حلقه for برای بدست آوردن آن در المان SRC استفاده شده است.

و در نهایت اطلاعات تجزیه شده را در قالب Adapter به RecyclerView ، Set می‌کنیم .

RequestQueue: برای دسته بندی و صف بندی اطلاعات استفاده می شود.

```

RequestQueue requestQueue = Volley.newRequestQueue(this);

//Adding request to the queue
requestQueue.add(jsonArrayRequest);

```


تصویر زیر خروجی مربوطه را در Genymotion نشان می دهد.(بقیه محصولات از طریق اسکروال قابل مشاهده است.)

در حال دریافت اطلاعات



لطفا صبر کنید...

Genymotion for personal use - Custom Tablet - 5.1.0 - API 22 - 2560x1600 (2560x1600, 320dpi) - 192.168.83.10



نام محصول سوهان گزی
قیمت
28000
توضیحات ترکیبی از گز و سوهان ترکیبی از گز و سوهان ترکیبی از گز و سوهان ترکیبی از گز و سوهان



نام محصول سوهان عسلی
قیمت
36000
توضیحات این سوهان عسلی است. این سوهان عسلی است. این سوهان عسلی است. این سوهان عسلی است.

[/https://docs.woothemes.com/document/woocommerce-rest-api](https://docs.woothemes.com/document/woocommerce-rest-api)

<https://github.com/kloon/WooCommerce-REST-API-Client-Library>

<https://www.simplifiedcoding.net/android-custom-listview-with-images-using-recyclerview-and-volley>

<http://p30droid.com/Thread-%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%DA%A9%D8%A7%D9%85%D9%84-RecyclerView?pid=18969>

<http://android-learn.ir>

عضویت در کانال :

https://telegram.me/joinchat/B1f7ETvpFe7AsduE7_6UPw

گروه پرسش و پاسخ :

https://telegram.me/joinchat/B1f7ETv_ZoJDk2dr0ES0rQ

گروه پرسش و پاسخ 2 :

<https://telegram.me/joinchat/B1f7ETxHZNwJxmxr-QG8bQ>

نکته : اگر عضو گروه اول هستید نیازی به عضویت در گروه دوم نیست! و هر سوال فقط باید در یک گروه پرسیده بشه و اگر در هر دو گروه بپرسین حذف میشه! ممنون از توجهتون. قبل از پیام دادن در گروه ها با سرچ کردن هشتگ #قوانین قوانین گروه رو بخونین .

این آموزش ها اختصاصی هستند و ساعت ها برای هر آموزش زحمت کشیده شده پس با کپی کردن بدون ذکر منبع تمامی زحمات مدرسین رو ضایع نکنین .

به زودی تمامی مطالب در سایت بارگزاری میشه 😊

www.AndroidDevelop.org