شرح ال custom ArrayAdapter بالتفصيل الممل

haythem.kmal **في ١ مارس**

السلام عليكم مبرمجي المستقبل وأصدقائي الأعزاء
أذكركم أنى قد قمت بشرح ال ArrayAdapter في مقال سابق

> **شرح ال ArrayAdapter كمقدمة لل custom ArrayAdapter**
>
> For English readers here is the English copy **https://macdiscussions.udacity.com/t/explanation-for-arrayadapter-an-introduction-for-custom-arrayadapter/119812?u=haythem.kmal** السلام عليكم اولا. ArrayAdapter ولكني وجدت ضرورة فهم ال custom ArrayAdapter أصداقئي الأعزاء كان من المفترض أن اقوم بشرح ال وكالعادة فان المقال طويل لزيادة الايضاح فقط ولا يعني باي حال من الاحوال صعوبة الموضوع...حتى اذا لم تفهم بعض النقاط حاول قراءة المقال حتى النهاية فلربما اتضح هذا الشيء فيما بعد خصوصا واني بدأت المقال بمقد...

وقد كان الغرض من هذا المقال أن يكون مقدمة لل custom ArrayAdapter فمن فضلك اذا لم يكن مفهوم ال arrayAdapter مفهوم لديك بشكل كامل قم بقراءة المقال السابق بتأن وبهدوء حيث أنى أعتمد عليه في هذا الشرح هنا .
(تحذير)
هذا الموضوع كعادتي طويل ومفصل تفصيلا مملا ...حاول قراءته بتأن ولا تستعجل الفهم فبعض الاجزاء اللاحقة تشرح السابقة والعكس .
(نظرة عامة)

لدينا ListView و ArrayAdapter وArrayList . ال ArrayList الذي لدينا ليس مجرد array عادي فال element الواحد يحتوي على عدة بيانات (حيث سنقوم بانشاء نوع بيانات معين) ولذلك لا نستطيع استخدام ال ArrayAdapter العادي بدون التعديل على بعض أوامره التي تتيح لنا عرض هذه البيانات.ولذلك نقوم بتوفير ملف xml به ال views التي تناسب طبيعة هذه البيانات ونقوم بتمريره لل ArrayAdapter وهو سيقوم باستخدام هذا ال xml وعرض البيانات عليه ثم سيقوم بتسليم ال View وبه البيانات لل ListView ليقوم بعرضها لنا.الأمر المسؤول عن وضع البيانات في الView المنشأ من ملف ال xml هو أمر يسمى getView .اذن فهمنا لهذا الأمر سيسهل لنا كيفية عمل custom ArrayAdapter.

(مقدمة و معلومات تمهيدية يجب أن تكون مألوفة لديك)

- في خلال الشرح سأقوم بالتعامل مع المكونات الثلاث الاتية ال ListView كمثال لل ArrayView و ال ArrayAdapter كمثال لل Adapter و ال ArrayList كمثال لل Data.
- الغرض الأساسي من ال custom ArrayAdapter ان تقوم بعرض البيانات في صورة View (اي View مهما يكن) غير ال TextView فقد تحتاج الى عرض البيانات (data) التي لديك في هيئة اثنان TextView و بجانبهماImageView وبجانبهما Button .فكما تعلم ال ArrayAdapter مهيأ لعرض البيانات التي لديك في صورة TextView وحيد فاذا كانت هذه رغبتك فأهلا وسهلا وان لم تكن فعليك بانشاء ال custom ArrayAdapter> المدخل لفعل هذا الشيء هو التعديل على أمر يقوم ال ListView بطلب تنفيذه من ال ArrayAdapter هذا الأمر يدعى getView ومن اسمه يتضح ذلك فعلا فنحن في النهاية نريد البيان معروض في View معين وقدرتنا على التلاعب في هذا الأمر تتيح لنا تغيير شكل ال view المرسل لنا وبداخله البيانات.
- عندما تقوم بعمل scroll لل ListView لمشاهدة باقي البيانات هذا مايحدث في الخلفية ...يقوم ال ListView بالطلب من ال ArrayAdapter أن يقوم بارسال البيان التالي (في صورة View)وذلك بطلب تنفيذ الأمر getView .
هنا يجب عليك أن تفهم هذا الأمر جيدا فهو المدخل لل custom ArrayAdapter
حسنا لندخل الى الأكواد ونربطها نظريا بما يحدث
هل تتذكر أننا قلنا أنه يجب تمرير ملف xml به شكل ال views التي نريد أن نعرض عليها البيانات وهذا الملف هوا مايتم تمريره الى ال custom ArrayAdapter الذي سنقوم بانشاءه؟؟ حسنا لنقم بانشاء هذا الملف كالاتي وفيما بعد سأريك ماذا يحدث لهذا الملف بالتحديد
اذن سأقوم بانشاء ملف xml بالطريقة المعتادة و ِاختار اسم له وليكن my_layout وسيحتوي على الاتي

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

```
<LinearLayout

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

<TextView
    android:id="@+id/firstTV"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    />
<TextView
    android:id="@+id/secondTV"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    />
</LinearLayout>
```

اذن من هنا يتضح أننا قمنا بانشاء ملف xml به LinearLayout واثنان TextViews وكل منهما له id مميز له وسيتم استخدام هذان ال TextView لعرض ما أريد عرضه
ولكي نستطيع عمل ذلك لابد من انشاء ArrayAdapter متخصص وهذا مايطلق عليه custom ArrayAdapter و لعل كلمة custom توضح لك فائدة هذا الفعل فكلنا يعلم كما ذكرت سابقا أن ال ArrayAdapter العادي يقوم بعرض البيانات على TextView واحد فقط وهذا أصلا ماسقيبله منك وفي حالة تمرير ملف xml به أكثر من TextView فلن يستطيع التعامل معك بياناتك .على اية حال قبل الخوض في ال custom ArrayAdapter دعنا ننشىء ال data ألتي نريدها .
عزيزي المبرمج أنا أعلم أنه في خلال الدرس قمنا بانشاء ArrayList من نوع word ولكني لا أريد أن تتشتت مني.أريدك من خلال هذا الدرس أن تعلم كيف تتلاعب بال ArrayAdapter فحتى لو كان لديك ArrayList وبه items من نوع String بامكانك عمل custom ArrayAdapter لعرض هذه البيانات البسيطة ففي مثالنا هذا سنقوم بعمل arrayList يحتوي على أسماء لاعبين كالاتي

```
ArrayList<String> myArrayList = new ArrayList<String>();
```

لنقم باضافة بعض الأسماء القليلة)

```
myArrayList.add("Mohamed Salah");
myArrayList.add("Abou Trika");
myArrayList.add("C.Ronaldo");
myArrayList.add("Messi");
```

(ملحوظة هذا ال ArrayList بطريقته البدائية هذه يمكن معها استخدام ArrayAdapter غير متخصص ولكن ماذا لو أردت مثلا كنوع من التجربة أن أكرر الأسماء وبذلك سأحتاج الى اثنان من TextView وهذا ماقمنا بانشاءه في ملف ال xmlالذي نريد من ال arrayAdapter أن يستخدمه لعرض مانريد عليه ثم ارجاع هذه ال views ومابها من بيانات اذن أنا استخدمت arrayList بسيط ولكن سأقوم بالعرض بطريقة مختلفة وذلك حتى أبعد تركيزك عن كيفية انشاء ArrayList به بيانات معقدة وبالتالي سيحتاج الى custom ArrayAdapter لعرض هذه البيانات المعقدة)
حسنا لنذهب الى كيفية انشاء ال custom ArrayAdapter
ستقوم بطبيعة الحال بانشاء ملف جافا لانشاء class جديد يقوم بعمل extends لل ArrayAdapter class ولكننا سنقوم بتغيير طريقة عمل بعض ال methods التي تتعامل مع كيفية ارجاع ال views وبها البيانات
اذن نحن نريد أن نرث جميع ال methods الخاصة بهذا ال ArrayAdapter class الا أننا في نفس الوقت نريد تغيير أمر يسمى getView هذا الأمر هوا مايقوم بترتيب البيانات في ال Views وفي حالتنا هذه في ال TextViews .
حسنا سنقوم بانشاء class سأسميه myCustomArrayAdapter وسيحتوي على الأكواد الاتية
سأقوم بشرح كل سطر بها باذن الله فلا تقلق
لاحظ أنني لا أتبع الكود المذكور في الشرح وذلك لاعتقادي أن هذا الكود أسهل للفهم وثانيا لتوسعة مداركك لتعلم أنك في كل الأحوال غير ملزم بكود معين فكل الطرق تؤدي الى روما

```
public class myCustomArrayAdapter extends ArrayAdapter {
```

```
    public ArrayList<String> myData ;
    public myCustomArrayAdapter(@NonNull Context context,  @NonNull ArrayList<Strir
        super(context, 0, objects);
        myData = objects ;
    }


    @Override
    public View getView(int position,  View convertView,  ViewGroup parent) {

        if (convertView == null){
            LayoutInflater myXmlInflater = LayoutInflater.from(getContext());
            convertView = myXmlInflater.inflate(R.layout.my_layout,parent,false);
        }

        TextView myFirstTV = (TextView) convertView.findViewById(R.id.firstTV);
        TextView mySecondTV = (TextView) convertView.findViewById(R.id.secondTV);

        myFirstTV.setText(myData.get(position));

        mySecondTV.setText(myData.get(position));

        return convertView;
    }
    }
```

بدأنا بهذا الأمر

```
public class myCustomArrayAdapter extends ArrayAdapter
```

هذا السطر كما ذكرنا يقوم بانشاء class خاص بنا أسميناه myCustomArrayAdapter وهو يرث جميع ال methods الموجودة في ال class الذي يسمى ArrayAdapter وأكرر أننا قمنا بفعل هذا الشيء لأننا نرى أن ال ,method المسماة getView قاصرة بعض الشيء ولا تقم بفعل مانريده .
لننتقل الى السكر التالي

```
public ArrayList<String> myData ;
```

هنا قمت بانشاء متغير من نوع ArrayList و أسميت المتغير هذا myData وذلك لأني سأحتاج أن أصل الى ArrayList التي سأقوم بتمريرها فيما بعد وهذا طبيعي فأنا أريد الحصول على البيانات التي أريد عرضها بطريقتي .
لننتقل الى السطر التالي

```
public myCustomArrayAdapter(Context context,  ArrayList<String> objects) {
    super(context, 0, objects);
    myData = objects ;
}
```

حسنا هنا نقوم بانشاء ال constructor وهذا طبيعي فنحن نريد تمرير بعض ال arguments فيما بعد حتى نستطيع التعامل معها .نحن الان نريد فقط ال context الذي يتيح لنا الوصول الى موارد ال activity الذي نحن بداخله (لا تشغل بالك حاليا ف فهم ال context) وثانيا نريد تمرير ال arrayList وهذا فقط مانريده ... لاحظ أنه وبالمقارنة مع ال ArrayAdapter العادي لا نقوم باستقبال ملف xml والذي كان سابقا يحتوي على TextView وحيد بل بالعكس نحن نرفض اي ملف xml أصلا وهذا يتضح من خلال السطر

```
super(context, 0, objects);
```

فكلمة super تقوم باستدعاء ال constructor الأصلي لل arrayAdapter . هل تلاحظ ماقمناه بتمريره بدلا من ملف ال xml ??للقد قمنا بتمرير 0 فنحن نري أن نتحكم في عرض البيانات و قمنا بانشاء ملف ال xml خاصتنا لهذا الغرض فكيف نستقبل ملف xml من الخارج.
حسنا في النهاية قمت باستقبال ال ArrayList في المتغير الذي قمت بانشاءه سابقا باسم myData وهذا يتضح من السطر التالي

```
myData = objects ;
```

حسنا لننتقل الى الأمر الذي قمنا بفعل كل هذا من أجله

```
public View getView(int position,  View convertView,  ViewGroup parent)
```

أولا عليك ملاحظة نوع ال object الذي يتم ارجاعه لهذا الأمر .هل تلاحظ ؟؟ انه من النوع View اذن ف النهاية هذا الأمر كما شرحنا سيقوم بارجاع View قد يكون هذا ال View عبارة عن TextView وحيد كما في حالة ال ArrayAdapter العادي أو قد يكون عبارة عن LinearLayout وبه اثنان من ال TextView وسترى في حالتنا هذه كيف نقوم بارجاع هذا ال View الناشيء أصلا من ملف ال xml الذي قمنا بانشاءه سابقا.هل تتذكر؟؟
حسنا لنشرح ال parameters التي تستقبلها هذه ال method

1. ال psition هذا عبارة عن ال Index الذي من خلاله يمكن عرض البيان من ال ArrayList
وسترى فيما بعد كيف أستخدم هذا الpositionفي استخراج ال data من ال arrayList

2. ال convertView هذا هوا أحد ال viewsالتي قمنا بانشائها سابقا . من فضلك ركز في هذه النقطة ؟؟
الان لديك ListView عند تحميل تطبيقك وفي بداية ملىء ال ListView فان هذا مايحدث ... يقوم ال ListView أولا بطلب تنفيذ الأمر getView من ال arrayAdapter المرتبط به ولأنه هوا الذي يطلب تنفيذ هذا الأمر فانه يقوم بتمرير ال arguments الى هذا الأمر . فهوا يقوم بتمرير قيمة ال position وأيضا ال convertView وأخيرا يقوم بتمرير قيمة ال parent> لنركز في قيمة ال convertView حسنا قلنا أنه في بداية ملىء ال ListView فانه سيقوم بتمرير قيمة ال convertView هذه ال convertView عبارة أصلا عن Views قد تم انشاءها من قبل ولم نعد في حاجة اليها وتعريف عدم حاجتنا لل View يعني اننا قمنا بعمل scroll أي سحب الشاشة لأعلى مثلا وبالتالي سيتكون لدينا بعض ال views التي لا نستطيع أن نراها الا أن ال ListView يحتفظ بها ولكن مهلا مازلنا في بداية ملىء ال ListView اذن في هذه الحالة سيقوم ال listView بتمرير القيمة null لكي يوضح لل arrayAdapter أنني مازلت في مرحلة التحميل والمستخدم لم يقم بعمل scrollبعد ولكن بعد اكتمال ملء ال ListView بعدد من ال Views التي تستوعب مساحة شاشتك فانه عند عمل scroll فانك بطبيعة الحال ستقوم باخفاء بعض هذه ال Views هذه ال Views لا يتم تدميرها بل يتم الاحتفاظ بها فيما يسمى pile scrap وحينما تقوم بعمل scroll فان ال ListView يقوم بتمرير قيمة ال converView ك View قديم بدلا من القيمة null وبذلك يجبر ال AdapterView على اعادة استخدام هذا ال view ألقديم (سترى الان الأسطر البرمجية التي تقوم بفعل هذا الشيء الرائع)

3. ال parent هذا هوا ال ListView خاصتنا ويتم تمريره لأننا نريد تحميل ال Views داخل row لهذا ال ListView.
حسنا لننتقل الى السطر التالي

```
if (convertView == null){
 LayoutInflater myXmlInflater = LayoutInflater.from(getContext());
 convertView = myXmlInflater.inflate(R.layout.my_layout,parent,true);
}
```

هل تتذكر ماقلناه أنه في بداية ملىء ال ListView يقوم بتمرير ال converView بالقيمة null حسنا هنا نقوم بالتأكد من هذه القيمة فان كانت null فان ال ListView سيحتاج حتما الى انشاء View جديد فلا نستطيع استخدام convertView قديم فالقيمة هنا null بمعنى أن ال ListView يقول أنا لا أملك حاليا أي views قديمة من فضلك قم بانشاء view جديد و املأه بالبيانات و ارجعه الي .
هنا السؤال كيف نقوم بانشاء view ؟؟ هل تتذكر أنه لدينا ملف xml اذن كل مانحتاجه هو أمر يقوم بتحويل ملف ال xml هذا ال Veiws ألموجود بداخله. هذا الأمر يسمى inflate ويتم استخدامه من خلال LAyoutInflater object
السطر التالي يقوم بانشاء هذا ال object من نوع LayoutInflater باسم myXmlinflater

```
LayoutInflater myXmlInflater = LayoutInflater.from(getContext());
```

بعدها أقوم باستخدام هذا الأمر لتحويل ملف ال xml الذي قمت بانشاءه سابقا الى ال Views التي بداخله

```
convertView = myXmlInflater.inflate(R.layout.my_layout,parent,true);
```

لكي تستوعب الأمر inflate تخيل أن ملف ال xml عبارة عن بالونة غير منتفخة وانت ترسم ال TextViews عليها او اي شيء تريد .هذه البالونة في حالتها هذه غير صالحة للاستخدام كما أن مارسمته لا يظهر بشكله الحقيقي مايفعله الأمر Inflate أن نفخ هذا البالون بحيث يظهر لك مارسمته بألوانه الرائعة وبذلك تكون هذه البالونة صالحة للعب .اذن ملف ال xml يجب أن يتم تحويله الى View object عن طريق الأمر inflate وبعدها نحفظه في ال converView وهذا ماسوف نقدمه لل ListView بعد ذلك حتى يقوم بعرضه.
حسنا لننتقل الى الأسطر الاخيرة

```
TextView myFirstTV = (TextView) convertView.findViewById(R.id.firstTV);
TextView mySecondTV = (TextView) convertView.findViewById(R.id.secondTV);
```

هنا أنا أعلم الان أن ال convertView سواء أكان قديما أو قمت بانشاءه فانه حتما سيحتوي على الاثنان TextViews كل بما لديه من id مميز .
هل تستوعب الان ما طبيعة ال converView ؟؟ أعتقد أن الأمر اتضح بعض الشيء فال convertView عبارة عن ال LinearLayout وبه الاثنان TextViews بالضبط كما حددنا ذلك في ملف ال xml الا ان ال converView عبارة عن View صالح للاستخدام على عكس ملف ال xml .
حسنا مادمنا استكعنا الوصول الى ال TextViews لنعدل من النص التي بداخلها كالاتي

```
myFirstTV.setText(myData.get(position));

mySecondTV.setText(myData.get(position));
```

كلنا يعلم الأمر setText. هل تتذكر المتغير myData الذي استخدمته سابقا في حفظ ArrayList التي ستمرر لي فيما بعد ...أنا أعلم بما أن myData عبارة عن ArrayList فحتما ستحتوي على الأمر get وأقوم بتمرير قيمة ال position لكي أتحصل على القيمة الموجودة في هذا ال index/position

ف النهاية أقوم بارجاع ال converView الذي يمثل لى الان LinearLayout و TextViews قيمة نصوصهما عدلت على حسب ما قمت بتحديده سابقا
أقوم بارجاع قيمة ال converView بالأمر التالي

```
return convertView;
```

أخيرا أقوم باستخدام هذا ال custom ArrayAdapter الذي أسميته myCustomArrayAdapter في انشاء ArrayAdapter يقوم بعرض البيانات بالطريقة التي أحبها .لعلك تلاحظ أنني لم أمرر أي ملف xmlحيث أني أتحكم بشكل العرض من خلال getView

```
myCustomArrayAdapter myAdapter = new myCustomArrayAdapter(this,myArrayList);


ListView myListView = findViewById(R.id.myListView);
myListView.setAdapter(myAdapter);
```

لقد هرمت من اجل هذه اللحظة 😂 أخيرا انتهينا من الشرح
كما قلت سابقاأنا أعلم أن مقالاتي طويلة الا أنى أظن أن قراءتها بتأن ستجعلك أكثر فهما بما يحدث وبالتالي ستكون أكثر قدرة على تعديل الكود والتلاعب بالكود كيفما شئت.
بالتوفيق للجميع

السلام عليكم ورحمة الله , ممكن توضيح ماهو word adapter, word في درس 12 اندرويد

الدرس 12 الحلقه 28

ممكن خطوات عمل custom adapter

Lesson 12, Custom ArrayAdapter

---

soliman.sabry83 **في ١ مارس**

ونحن كمان هرمنا ... هرمنا ... من أجل هذه اللحظة التاريخية ... حمد الله بالسلامة ياكبير 😂