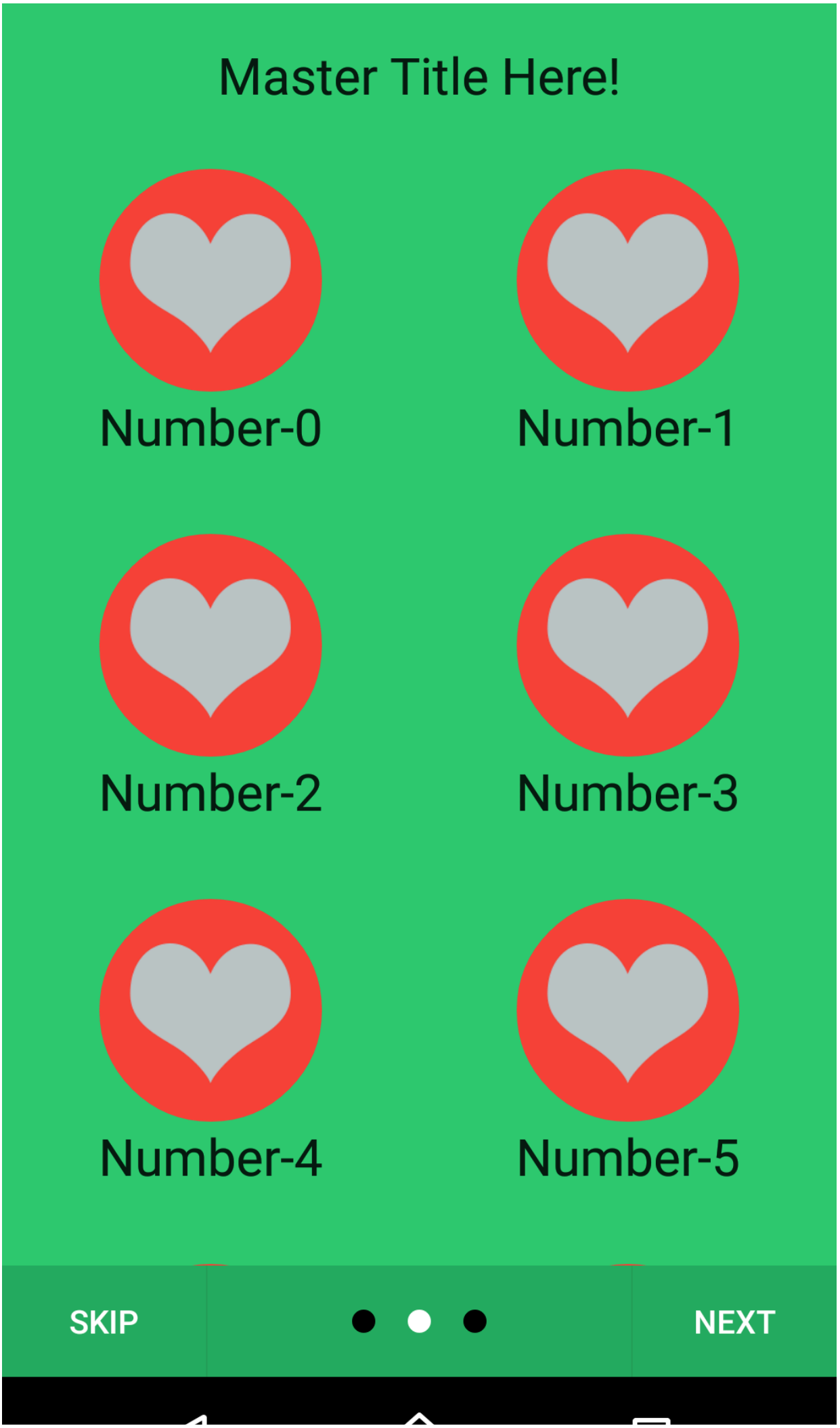
# How to set the height of an item row in GridLayoutManager

Asked 8 years, 10 months ago Modified 1 year, 6 months ago Viewed 80k times

Part of Mobile Development Collective


Master Title Here!
Number-0
Number-1
Number-2
Number-3
Number-4
Number-5
SKIP
NEXT

My Recycler Item which inflate in onCreateViewHolder

```
<?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:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <ImageView
        android:id="@+id/gridListImageView"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:src="@drawable/a" />

    <TextView
        android:id="@+id/gridListView_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge" />
</LinearLayout>
```

I want to display something like this Which has one row of half the height of recycler View? And add padding to the rest of the space? Can i do this by GridLayoutManager?

And this is my GridLayoutManager

```
        GridLayoutManager glm = new GridLayoutManager(getActivity(), 2);
        recyclerView.setLayoutManager(glm);
```

MD android android-recyclerview gridlayoutmanager

Share

Improve this question

Follow

edited Feb 5, 2016 at 11:32

Rajesh Satvara

3,954 ● 2 ● 32 ● 50

asked Feb 5, 2016 at 10:18

Kunal

973 ● 1 ● 10 ● 15

If you set a specific height for your `RecyclerView`, you could just set half of that as height for the `LinearLayout`. If not, for example if you want one item to be half of the screen height, you have to do it programmatically, get the screen height, do some calculations and set the LayoutParams. – yennsarah Feb 5, 2016 at 10:24

you have to calculate the screen height- minus nav bar height and then divide it by 2 to get height of a row – JAAD Feb 5, 2016 at 10:28

why dont you try with LinearLayout, will be easy with using weightSum – Amit Vaghela Feb 5, 2016 at 11:39

@AmitVaghela I have to add more items so i have to use recycler view! – Kunal Feb 5, 2016 at 14:04

@Amy, I'm getting view.getLayoutParams().height as -1 and view.getMeasuredHeight() as 0. What to do? – Kunal Feb 5, 2016 at 14:07

## 9 Answers

Sorted by: Highest score (default)

70

When inflating layout for your views in adapter, you can set their height programmatically. In order to evaluate proper height to use you can rely on parent ViewGroup (that is the RecyclerView itself). Here it is a sample:

```
@Override
public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
    View itemView = mLayoutInflater.inflate(R.layout.view_item, parent, false);
    // work here if you need to control height of your items
    // keep in mind that parent is RecyclerView in this case
    int height = parent.getMeasuredHeight() / 4;
    itemView.setMinimumHeight(height);
    return new ItemViewHolder(itemView);
}
```

Hope this could help.

Share Improve this answer Follow

answered Feb 6, 2016 at 7:26

andrea.petreri
4,137 ● 2 ● 24 ● 22

Hello @thetonrifles, I tried your sample code already but looks like not help me in `(this case)` [stackoverflow.com/questions/41338627/... that I don't know why. Please help me take a look. Thank you, – Huy Tower Dec 27, 2016 at 3:32

3 It seems that as of version recyclerview-v7:25.1.1 doesn't support or misbehave now. By invoking `parent.getMeasuredHeight()` the return value is always zero. – drindt Feb 10, 2017 at 13:04

I just tested to ensure this and switched back as your project uses version 23.1.1 and now it works. – drindt Feb 10, 2017 at 13:20

@drindt Thanks for raising this. I will take a look at this and provide an update to the answer! – andrea.petreri Feb 10, 2017 at 13:25

Did you figure out a way to deal with this with the new recyclerview api? I'm getting 0 for `parent.getMeasuredHeight()` still – b.lyte Jun 9, 2017 at 4:38

31

As of support library 25.1.0, ABOVE ANSWER DOESN'T WORK. I suggest below modifications:

```
    public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.list_item, parent,
false);
        GridLayoutManager.LayoutParams lp = (GridLayoutManager.LayoutParams)
v.getLayoutParams();
        lp.height = parent.getMeasuredHeight() / 4;
        v.setLayoutParams(lp);
        return new ViewHolder(v);
    }
```

Share

Improve this answer

Follow

edited Jan 3, 2019 at 14:11

answered Apr 3, 2017 at 21:59

abedfar

**1,999** ● 24 ● 21

why divided by 4 ? – Gopal Singh Sirvi Jul 3, 2020 at 13:48

@GopalSinghSirvi if the guy wants 4 row, divide by 6, if required 6 rows. – Axay Prajapati Oct 19, 2020 at 12:45

What if I need a different height for each row? I have posted a question but haven't been answered link – JP711 Feb 26, 2022 at 17:29

22

You don't need to set the height of an item. The problem here is that image tries to fill all the space. Just add `android:adjustViewBounds="true"` to your ImageView and it will not add blank spaces

Share Improve this answer Follow

answered Feb 25, 2018 at 22:41

Valentina Konyukhova

**5,234** ● 2 ● 29 ● 37

Thank you, If someone needed to use `FrameLayout` for forground ripple effect on an `ImageView` in a `RecyclerView`, this is the way to go. – M. Reza Nasirloo Aug 23, 2019 at 9:40

Inside `onCreateViewHolder`, simply

12

```
v.getLayoutParams().width = parent.getMeasuredWidth() / 2;
```

or

```
v.getLayoutParams().height = parent.getMeasuredWidth() / 2;
```

Share

Improve this answer

Follow

edited Jun 1, 2023 at 11:21

Fattie

12.5k ● 75 ● 447 ● 752

answered Apr 28, 2018 at 9:35

Pranav Goswami

234 ● 2 ● 8

pls add some more information to your answer like why it's correct or what differs to other answers – tung Apr 28, 2018 at 10:21

3 none of the above works for me, but this one does. tell what other information you need? do you want me to post full function here? – Pranav Goswami Apr 30, 2018 at 20:20

@PranavGoswami i just formated your answer not edited kindly check again – AskNilesh Oct 31, 2018 at 5:33

6

**kotlin version**

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
    val binding = ItemBinding.inflate(
        LayoutInflater.from(parent.context),
        parent,
        false
    )
    binding.root.post {
        binding.root.layoutParams.height = parent.width/3
        binding.root.requestLayout()
    }
    return ViewHolder(binding)
}
```

here 3 is the span count of your `GridLayoutManager` . You can replace `binding.root` with your `itemView` , if you are not using Databinding

Share Improve this answer Follow

answered Mar 13, 2020 at 6:46

a0x2

2,111 ● 1 ● 19 ● 26

3

sometime, getting size of inflate view in adapter return 0 or negative. another approach is get required size from out side the adapter, manipulate it and set it into

view. in my case, another problem was size set effectless. so i set the size using layout parameter

here is setting my adapter in activity:

```
Display display = MainActivity.this.getWindowManager().getDefaultDisplay();
                    Point size = new Point();
                    display.getSize(size);
                    int  y = size.y;
                    y=(int)y/2;
 GridLayoutManager linearLayoutManager = new
GridLayoutManager(MainActivity.this,2);
                    recyclerView.setLayoutManager(linearLayoutManager);
                    NewOrderAdapter newOrderAdapter=new
NewOrderAdapter(MainActivity.this,arrayListname,arrayListimage,y);
                    recyclerView.setAdapter(newOrderAdapter);
```

and i set view size like this:

```
@Override
    public myviewholder onCreateViewHolder(ViewGroup viewGroup, int i) {
        View view = inflator.inflate(R.layout.new_order_row, viewGroup, false);
        GridLayoutManager.LayoutParams params =
(GridLayoutManager.LayoutParams) view.getLayoutParams();
        params.height = ysize;
        view.setLayoutParams(params);
        myviewholder holder = new myviewholder(view);
        return holder;
    }
```

and dont forget to set a height to layout in your layout for initializatiin

Share Improve this answer Follow

answered Jan 15, 2017 at 15:32

iman kazemayni

1,343 ● 1 ● 20 ● 22

---

3

On a recent API Level currently 25 the height from the recycler in onCreateViewHolder is always empty. This snippet is to set the hight after the recycler view's onMeasure is invoked and set the correct height to the inflated list view.

```
@Override
public DataBindingViewHolder onCreateViewHolder(final ViewGroup parent, int
viewType) {
    // You may inflate your view here.
    parent.post(new Runnable() {

        @Override
        public void run() {
            int height = parent.getMeasuredHeight() / rows;
            View view = holder.getBinding().getRoot();
            view.getLayoutParams().height = height;
```

```
        }
    });
    return holder;
}
```

Share Improve this answer Follow

answered Feb 13, 2017 at 11:42

drindt

899 ● 11 ● 15

nice solution, but 2 questions: 1) how can you be sure onMeasure was called when the runnable runs? 2) what is DataBindingViewHolder? – Itay Bianco Jul 4, 2017 at 8:53

1

This is my code for adjusting height of recycle view element based on actual aspect-ration required.

```
@Override
public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View itemView = LayoutInflater.from(parent.getContext())
.inflate(R.layout.adapter_offers, parent, false);

    int width =  parent.getMeasuredWidth();
    float height = (float) width / Config.ASPECT_RATIO;//(Width/Height)
    RecyclerView.LayoutParams params = (RecyclerView.LayoutParams)
itemView.getLayoutParams();
    params.height = Math.round(height);
    itemView.setLayoutParams(params);
    return new MyViewHolder(itemView);
}
```

Share Improve this answer Follow

answered Jun 10, 2017 at 6:11



Vinil Chandran

1,561 ● 1 ● 19 ● 33

1

I had a similar requirement which needed a dynamic height for each row in the grid with the ability to add and remove items from the RecyclerView as well. The problem with setting the height layout params in `onCreateViewHolder` as few of the answers suggests here, is that when an item is added/removed from the RecyclerView, the view is recycled and the `onCreateViewHolder` is not called again but rather a view within the pool is re-used (if available) and the layout manager would do a pass to calculate the height/width for the item and we would lose the original intended height/width set in the `onCreateViewHolder`.

This approach below might help who is facing a similar issue.

The key step here is to extend `GridLayoutManager` and override the following -

1. `generateDefaultLayoutParams`
2. `generateLayoutParams`
3. `checkLayoutParams`

The layout manager would looks something like this -

```
SpanGridLayoutManager : GridLayoutManager {

constructor(context: Context, attrs: AttributeSet?, defStyleAttr: Int,
defStyleRes: Int) :
        super(context, attrs, defStyleAttr, defStyleRes)

constructor(context: Context, spanCount: Int) : super(context, spanCount)
constructor(context: Context, spanCount: Int, orientation: Int, reverseLayout:
Boolean) :
        super(context, spanCount, orientation, reverseLayout)

override fun generateDefaultLayoutParams(): RecyclerView.LayoutParams {
    return spanLayoutSize(super.generateDefaultLayoutParams())
}

override fun generateLayoutParams(c: Context, attrs: AttributeSet):
RecyclerView.LayoutParams {
    return spanLayoutSize(super.generateLayoutParams(c, attrs))
}

override fun generateLayoutParams(lp: ViewGroup.LayoutParams):
RecyclerView.LayoutParams {
    return spanLayoutSize(super.generateLayoutParams(lp))
}

override fun checkLayoutParams(lp: RecyclerView.LayoutParams): Boolean {
    val layoutParams = generateDefaultLayoutParams()
    return super.checkLayoutParams(lp) &&
            layoutParams.width == lp.width &&
            layoutParams.height == lp.height
}

private fun spanLayoutSize(layoutParams: RecyclerView.LayoutParams):
RecyclerView.LayoutParams {
    layoutParams.height = if (some_condition) x else y
    return layoutParams
}
```

Here `x` and `y` can be the height in pixels that you would need to supply.

Share

Improve this answer

Follow

edited Mar 24, 2021 at 5:35

Dharman ♦

33.2k ● 27 ● 99 ● 146

answered Mar 24, 2021 at 5:29

Tejas

447 ● 5 ● 12