
CIS 470

Mobile App Development

Lecture 8

Wenbing Zhao

Department of Electrical Engineering and Computer Science

Cleveland State University

wenbing@ieee.org

User Interface

- Basic views: Commonly used views, such as the TextView, EditText, and Button views
- Picker views: Views that enable users to select from a list, such as the TimePicker and DatePicker views
- List views: Views that display a long list of items, such as the ListView and the SpinnerView views

Basic views

- TextView — to display text to the user
- EditText —A subclass of the TextView view, which allows users to edit its text content
- Button—Represents a push-button widget
- ImageButton—Similar to the Button view, except that it also displays an image
- CheckBox—A special type of button that has two states: checked or unchecked
- RadioGroup and RadioButton—The RadioButton has two states: either checked or unchecked. A RadioGroup is used to group one or more RadioButton views, thereby allowing only one RadioButton to be checked within the RadioGroup
- ToggleButton—Displays checked/unchecked states using a light indicator

Basic views

- One thing that has been consistent throughout this example is that each view has the id attribute set to a particular value, such as in the case of the Button
- The id attribute is an identifier for a view, which allows it to be retrieved using the `View.findViewById()` or `Activity.findViewById()` methods

```
<Button android:id="@+id/btnSave"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="@string/save" />
```

Using basic views

- Using Android Studio, create an Android project and name it BasicViews
- Replace the activity_main.xml with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button android:id="@+id/btnSave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="save" />
    <Button android:id="@+id/btnOpen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open" />
    <ImageButton android:id="@+id/btnImg1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher" />
```

```
<EditText android:id="@+id/txtName"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<CheckBox android:id="@+id/chkAutosave"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Autosave" />
<CheckBox android:id="@+id/star"
    style="?android:attr/starStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<RadioGroup android:id="@+id/rdbGp1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton android:id="@+id/rdb1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 1" />
    <RadioButton android:id="@+id/rdb2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Option 2" />
</RadioGroup>
<ToggleButton android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Using basic views

- Modify MainActivity.java (add bolded lines):

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {

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

        //---Button view---
        Button btnOpen = (Button) findViewById(R.id.btnOpen);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                DisplayToast("You have clicked the Open button");
            }
        });
    }
}
```

Using basic views

- Modify MainActivity.java (add bolded lines):

//---Button view---

```
Button btnSave = (Button) findViewById(R.id.btnSave);  
btnSave.setOnClickListener(new View.OnClickListener()  
{  
    public void onClick(View v) {  
        DisplayToast("You have clicked the Save button");  
    }  
});
```

//---CheckBox---

```
CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);  
checkBox.setOnClickListener(new View.OnClickListener()  
{  
    public void onClick(View v) {  
        if (((CheckBox)v).isChecked())  
            DisplayToast("CheckBox is checked");  
        else  
            DisplayToast("CheckBox is unchecked");  
    }  
});
```

Alternative way of adding an event handler

```
<Button android:id="@+id/btnSave"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/save"  
    android:onClick="btnSaved_clicked"/>
```


Using basic views

- Modify MainActivity.java (add bolded lines):

//---RadioButton---

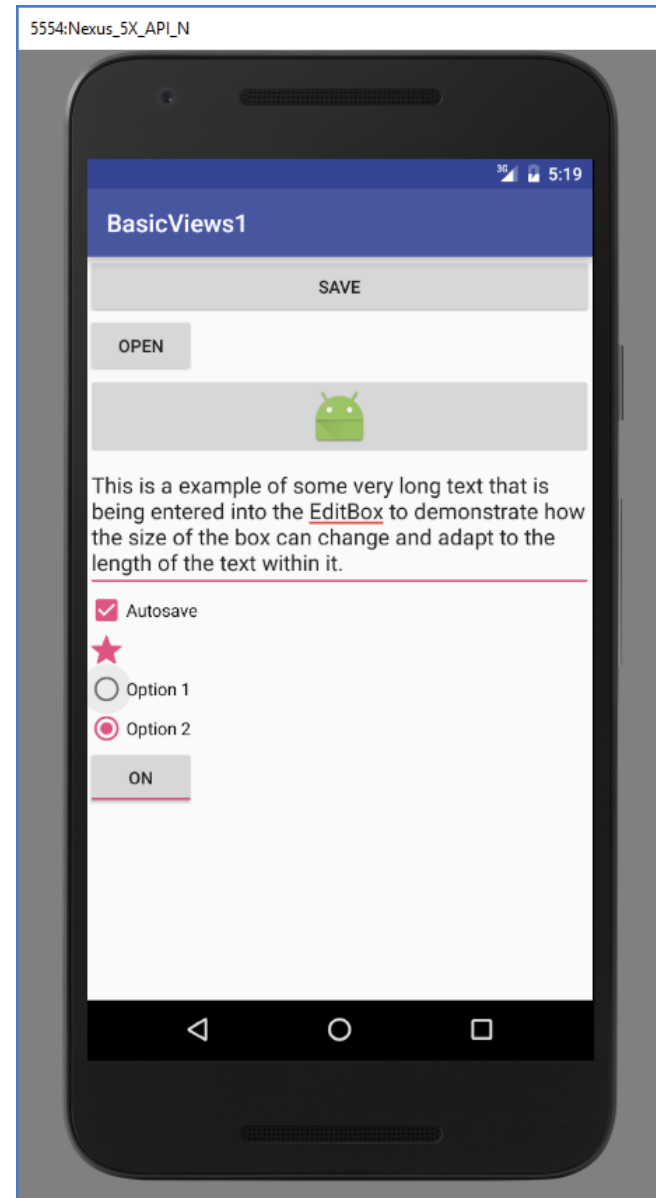
```
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);  
radioGroup.setOnCheckedChangeListener(  
    new RadioGroup.OnCheckedChangeListener() {  
        public void onCheckedChanged(RadioGroup group, int checkedId) {  
            RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);  
            if (rb1.isChecked()) {  
                DisplayToast("Option 1 checked!");  
            } else {  
                DisplayToast("Option 2 checked!");  
            }  
        }  
    });
```

//---ToggleButton---

```
ToggleButton toggleButton = (ToggleButton) findViewById(R.id.toggle1);  
toggleButton.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        if (((ToggleButton)v).isChecked())  
            DisplayToast("Toggle button is On");  
        else  
            DisplayToast("Toggle button is Off");  
        }  
    });  
}
```

```
private void DisplayToast(String msg)  
{  
    Toast.makeText(getApplicationContext(), msg,  
        Toast.LENGTH_SHORT).show();  
}
```

Using basic views



ProgressBar view

- The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background
- Using Android Studio, create an Android project and name it ProgressBarTest
- Change activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <ProgressBar android:id="@+id/progressbar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@android:style/Widget.ProgressBar.Horizontal" />
</LinearLayout>
```

ProgressBar view

■ Change MainActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {
    static int progress;
    ProgressBar progressBar;
    int progressStatus = 0;
    Handler handler = new Handler();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        progress = 0;
        progressBar = (ProgressBar) findViewById(R.id.progressbar);
        progressBar.setMax(200); // default is 100
    }
}
```

```

//---do some work in background thread---
new Thread(new Runnable() {
    public void run() {
        //---do some work here---
        while (progressStatus < 100)
        {
            progressStatus = doSomeWork();
            //---Update the progress bar---
            handler.post(new Runnable()
            {
                public void run() {
                    progressBar.setProgress(progressStatus);
                }
            });
        }
        //---hides the progress bar---
        handler.post(new Runnable() {
            public void run() {
                //---0 - VISIBLE; 4 - INVISIBLE; 8 - GONE---
                progressBar.setVisibility(View.GONE);
            }
        });
    }
    //---do some long running work here---
    private int doSomeWork() {
        try {
            //---simulate doing some work---
            Thread.sleep(500);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        return ++progress;
    }
}).start();
}
}

```

ProgressBar Styles

- `Widget.ProgressBar.Horizontal`
- `Widget.ProgressBar.Small`
- `Widget.ProgressBar.Large`
- `Widget.ProgressBar.Inverse`
- `Widget.ProgressBar.Small.Inverse`
- `Widget.ProgressBar.Large.Inverse`



Picker Views

- TimePicker and DatePicker views: select a data and time
- The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode
- Using Android Studio, create an Android project and name it TimePickerTest
- Modify activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TimePicker android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button android:id="@+id/btnSet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am all set!"
        android:onClick="onClick" />
</LinearLayout>
```

TimePicker Views

■ Modify MainActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
    TimePicker timePicker;
```

@Override

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

```
    timePicker = (TimePicker) findViewById(R.id.timePicker);
    timePicker.setIs24HourView(true);
```

```
}
```

```
public void onClick(View view) {
    Toast.makeText(getApplicationContext(),
        "Time selected:" +
        timePicker.getHour() +
        ":" + timePicker.getMinute(),
        Toast.LENGTH_SHORT).show();
    }
}
```


DatePicker Views

- Using Android Studio, create an Android project and name it DatePickerTest
- Modify activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button android:id="@+id/btnSet"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am all set!"
        android:onClick="onClick" />
    <DatePicker android:id="@+id/datePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TimePicker android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

DatePicker Views

■ Modify MainActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.TimePickerDialog;
import android.icu.text.SimpleDateFormat;
import android.view.View;
import android.widget.DatePicker;
import android.widget.TimePicker;
import android.widget.Toast;
import java.util.Date;

public class MainActivity extends AppCompatActivity {
    TimePicker timePicker; DatePicker datePicker; int hour, minute;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);
        datePicker = (DatePicker) findViewById(R.id.datePicker);
    }
}
```

DatePicker Views

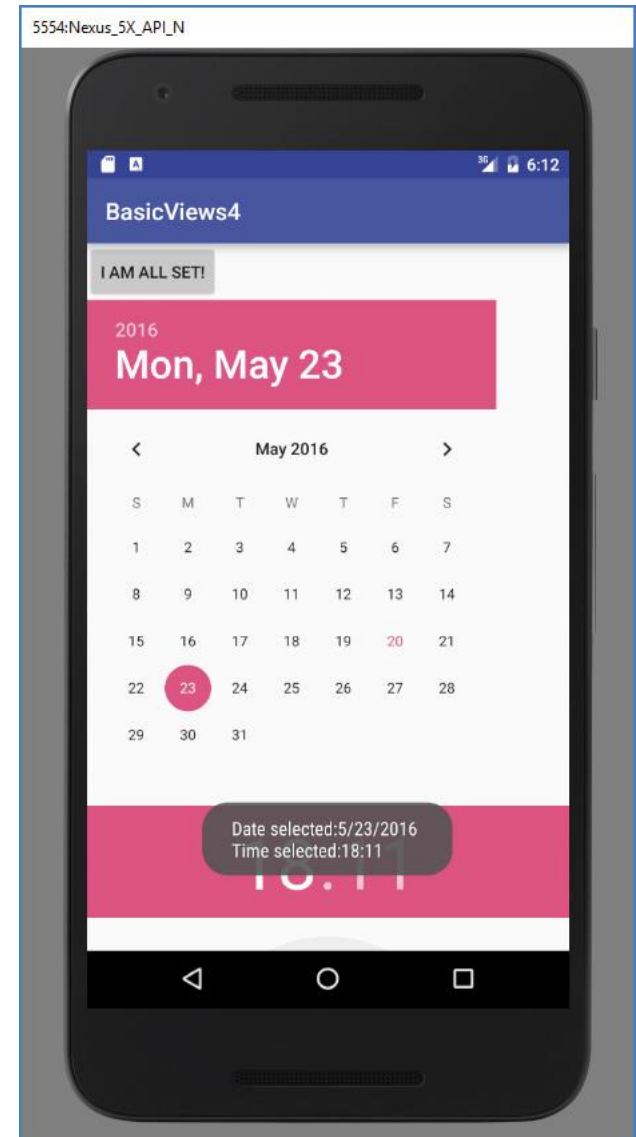
- Modify MainActivity.java

```
private TimePickerDialog.OnTimeSetListener mTimeSetListener =
    new TimePickerDialog.OnTimeSetListener() {
        public void onTimeSet(
            TimePicker view, int hourOfDay, int minuteOfHour) {
            hour = hourOfDay; minute = minuteOfHour;
            SimpleDateFormat timeFormat = new SimpleDateFormat("hh:mm aa");
            Date date = new Date(); String strDate = timeFormat.format(date);
            Toast.makeText(getApplicationContext(),
                "You have selected " + strDate,
                Toast.LENGTH_SHORT).show();
        }
    };

public void onClick(View view) {
    Toast.makeText(getApplicationContext(),
        "Date selected:" + (datePicker.getMonth() + 1) +
        "/" + datePicker.getDayOfMonth() +
        "/" + datePicker.getYear() + "\n" +
        "Time selected:" + timePicker.getHour() +
        ":" + timePicker.getMinute(),
        Toast.LENGTH_SHORT).show();
}
```

DatePicker Views

- **Exercise** (required): If you look closely, the TimePicker is not entirely displayed. Fix the problem using a layout we learned in lecture 7 so that nothing is cut off from the screen.



List View

- List views are views that enable you to display a long list of items. In Android, there are two types of list views: ListView and SpinnerView.
- Using Android Studio, create an Android project and name it ListViewTest
- Modify activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button
        android:id="@+id/btn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Show selected items"
        android:onClick="onClick"/>
    <ListView
        android:id="@+id/android:list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

List View

- Modify strings.xml under res/values

```
<resources>
  <string name="app_name">ListViewTest</string>
  <string-array name="presidents_array">
    <item>Dwight D. Eisenhower</item>
    <item>John F. Kennedy</item>
    <item>Lyndon B. Johnson</item>
    <item>Richard Nixon</item>
    <item>Gerald Ford</item>
    <item>Jimmy Carter</item>
    <item>Ronald Reagan</item>
    <item>George H. W. Bush</item>
    <item>Bill Clinton</item>
    <item>George W. Bush</item>
    <item>Barack Obama</item>
    <item>Donald Trump</item>
  </string-array>
</resources>
```

List View

■ Modify MainActivity.java

```
import android.os.Bundle;
import android.app.ListActivity;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends ListActivity {
    String[] presidents;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ListView lstView = getListView();
        lstView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
        lstView.setTextFilterEnabled(true);
        presidents = getResources().getStringArray(R.array.presidents_array);
        setListAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_checked, presidents));
    }
}
```

List View

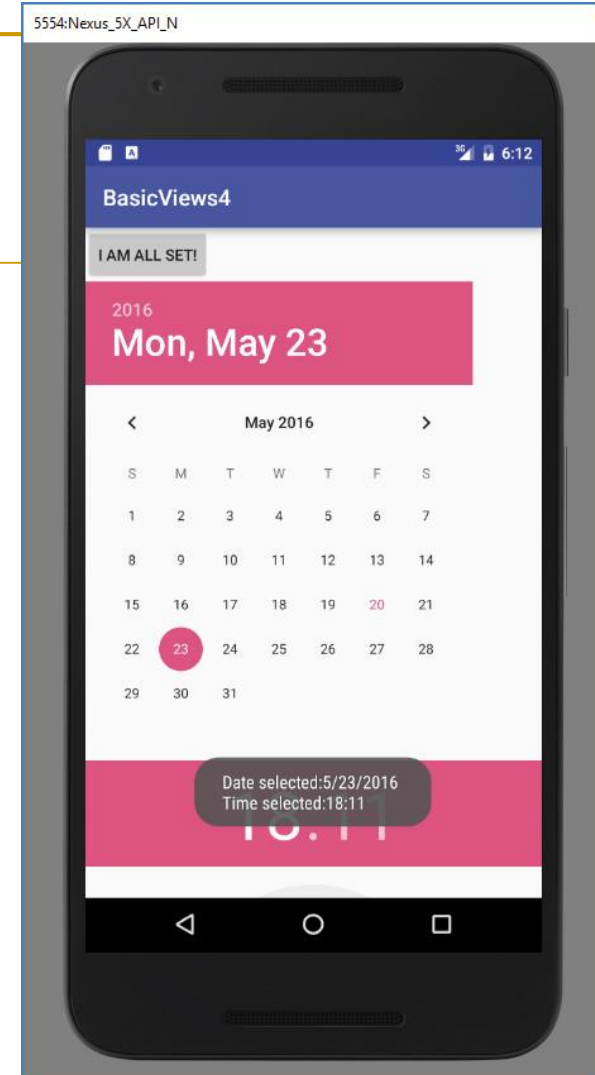
■ Modify MainActivity.java

```

public void onListItemClick(
    ListView parent, View v, int position, long id)
{
    Toast.makeText(this,
        "You have selected " + presidents[position],
        Toast.LENGTH_SHORT).show();
}

public void onClick(View view) {
    ListView lstView = getListView();
    String itemsSelected = "Selected items: \n";
    for (int i=0; i<lstView.getCount(); i++) {
        if (lstView.isItemChecked(i)) {
            itemsSelected += lstView.getItemAtPosition(i) + "\n";
        }
    }
    Toast.makeText(this, itemsSelected, Toast.LENGTH_LONG).show();
}

```



SpinnerView

- The SpinnerView displays one item at a time from a list and enables users to choose from them
- Create an Android project and name it SpinnerViewTest
- Modify activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawSelectorOnTop="true" />
</LinearLayout>
```

SpinnerView

- Modify strings.xml

```
<resources>
  <string name="app_name">SpinnerViewTest</string>
  <string-array name="presidents_array">
    <item>Dwight D. Eisenhower</item>
    <item>John F. Kennedy</item>
    <item>Lyndon B. Johnson</item>
    <item>Richard Nixon</item>
    <item>Gerald Ford</item>
    <item>Jimmy Carter</item>
    <item>Ronald Reagan</item>
    <item>George H. W. Bush</item>
    <item>Bill Clinton</item>
    <item>George W. Bush</item>
    <item>Barack Obama</item>
    <item>Donald Trump</item>
  </string-array>
</resources>
```

SpinnerView

■ Modify MainActivity.java

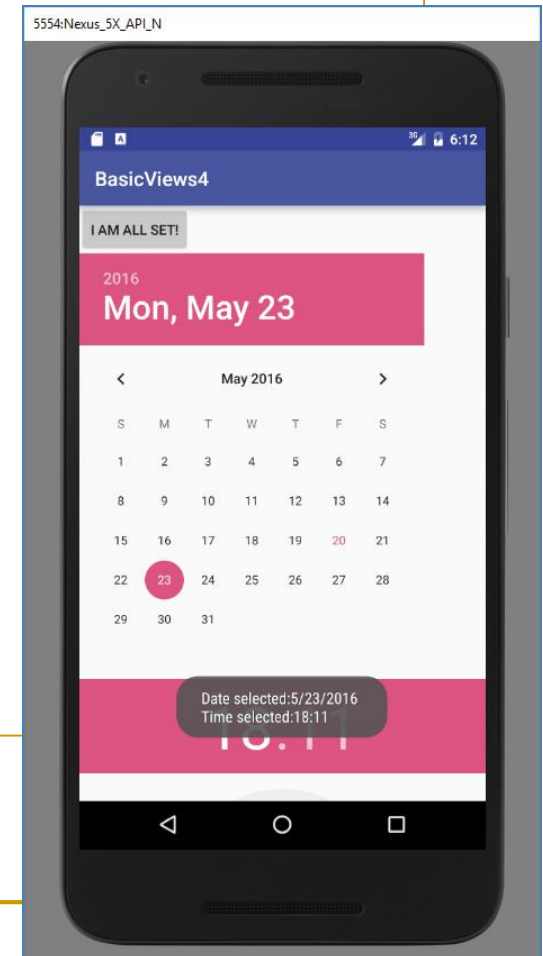
```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    String[] presidents;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        presidents = getResources().getStringArray(R.array.presidents_array);
        Spinner s1 = (Spinner) findViewById(R.id.spinner1);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_single_choice, presidents);
        s1.setAdapter(adapter);
    }
}
```

SpinnerView

■ Modify MainActivity.java

```
s1.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0,
                            View arg1, int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        Toast.makeText(getApplicationContext(),
            "You have selected item : " + presidents[index],
            Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) { }
});
}
```



Challenge Task

- Build a calendar app (without persistency and reminder)