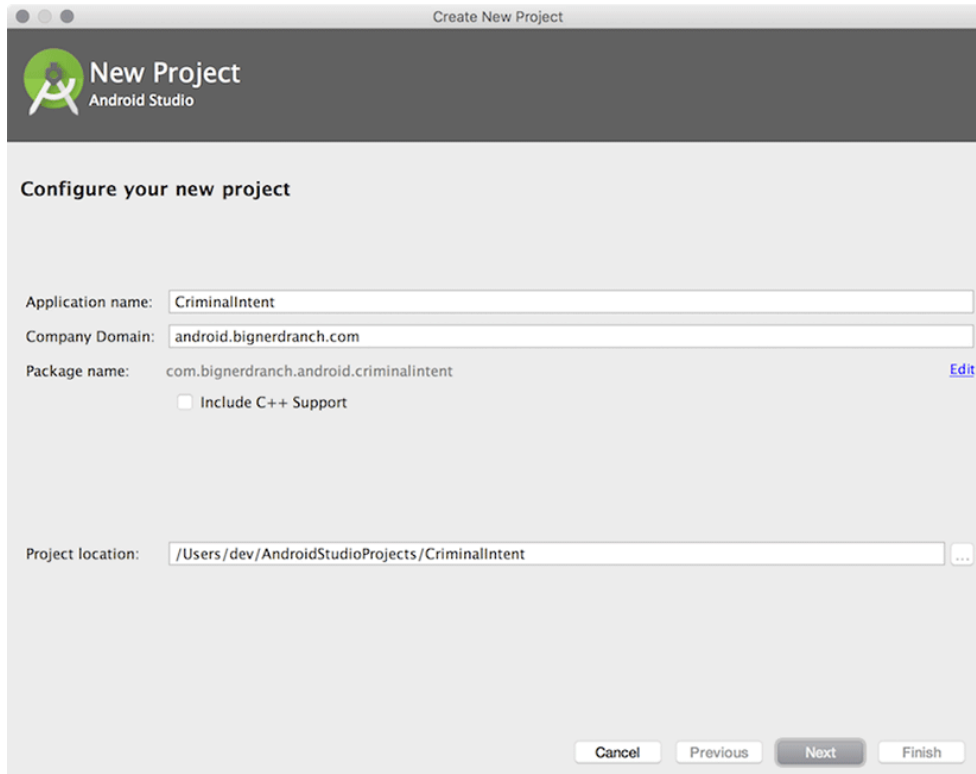


UI Fragments and the Fragment Manager-09

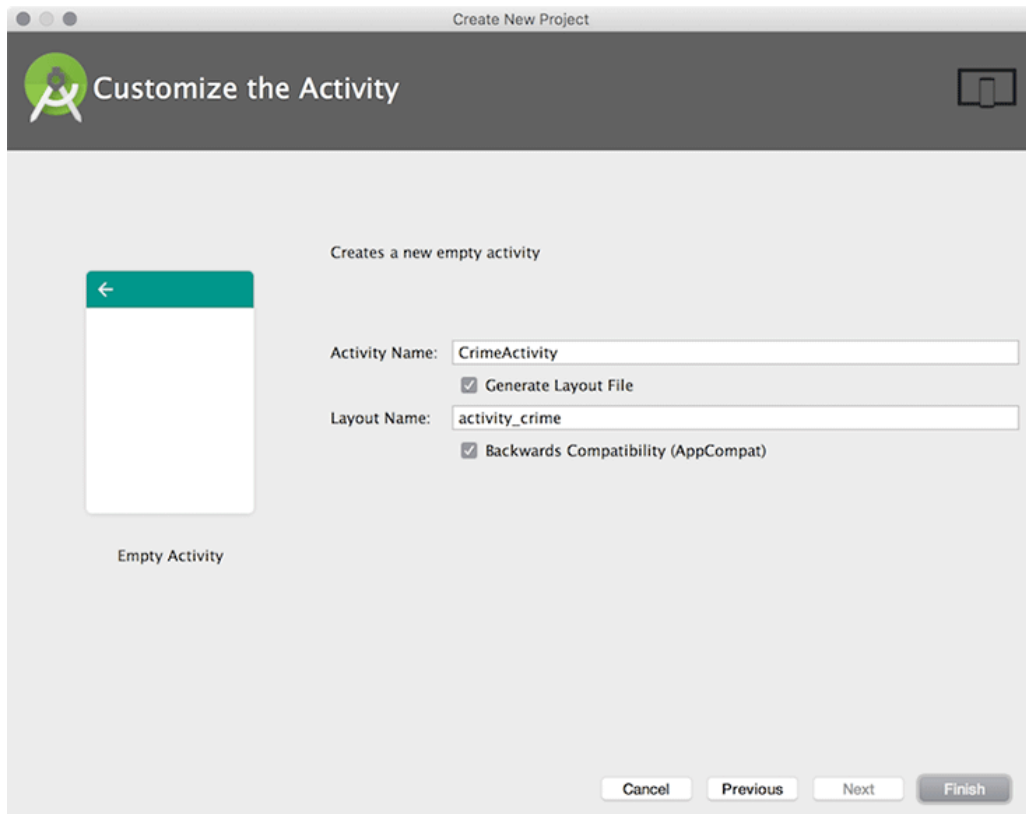
In this assignment, you will start building an app named CriminalIntent. CriminalIntent records the details of “office crimes” - things like leaving dirty dishes in the breakroom sink or walking away from an empty shared printer after documents have printed.

1. Create a new Android application (File → New Project...). Name the application CriminalIntent and make sure the company domain is android.bignerdranch.com. Also be sure to create a new repository on GitHub (add your instructor as a collaborator) and perform an initial commit:



2. Click Next and specify a minimum SDK of API 19: Android 4.4. Also ensure that **only** the Phone and Tablet application type is checked.
3. Click Next again to select the type of activity to add. Choose Empty Activity and continue along in the wizard.

4. In the final step of the New Project wizard, name the activity CrimeActivity and click Finish:

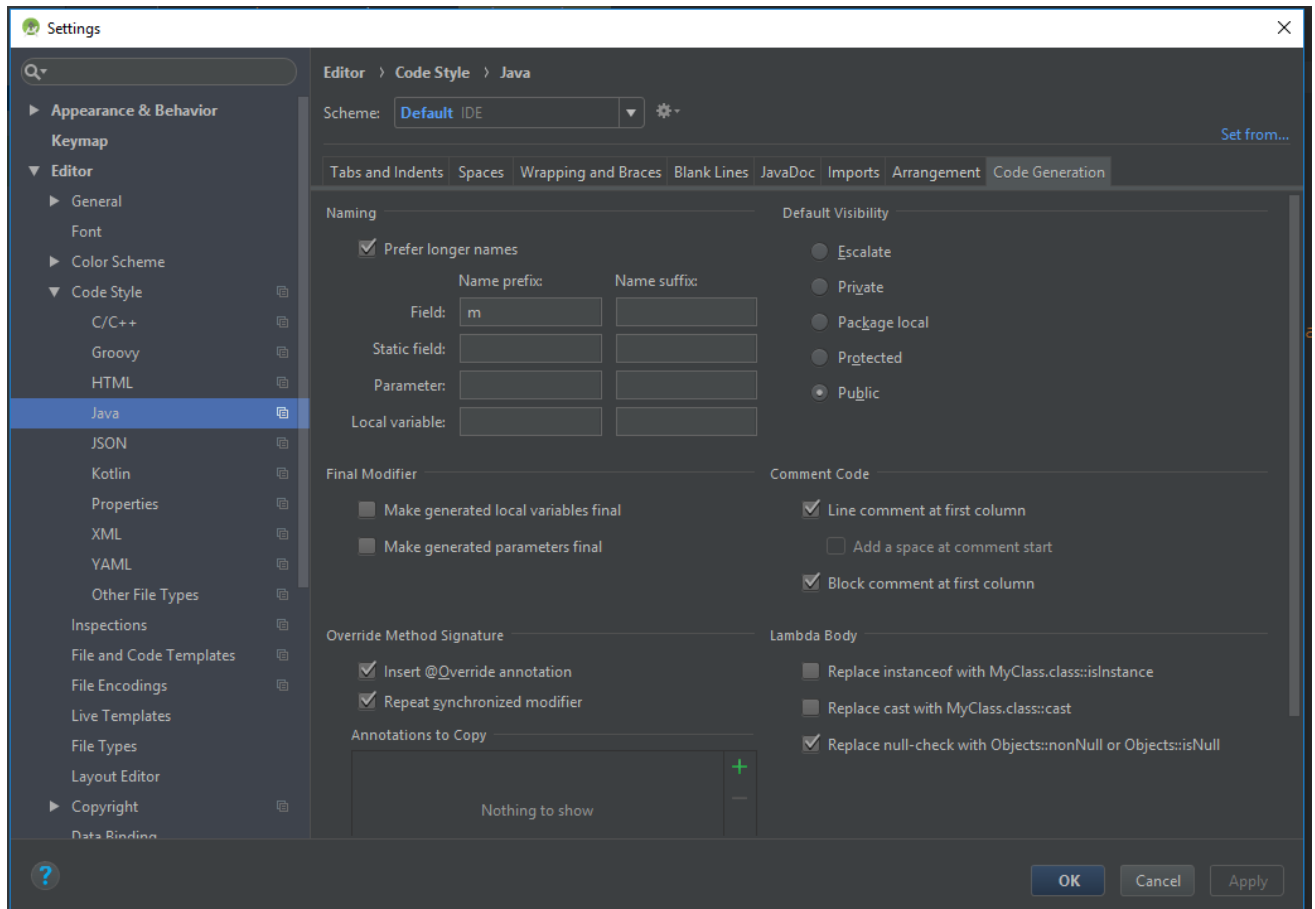


5. Let's create the Crime class. In the project tool window, right-click the com.bignerdranch.android.criminalintent package and select New → Java Class. Name the class Crime and click OK.
6. In Crime.java, add fields to represent the crime's ID (which will be read-only), title, date, and status and a constructor that initializes the ID and date fields:

```
public class Crime {  
    private UUID mId;  
    private String mTitle;  
    private Date mDate;  
    private boolean mSolved;  
  
    public Crime() {  
        mId = UUID.randomUUID();  
        mDate = new Date();  
    }  
}
```

7. Click the items that are highlighted red and press Alt-Enter to have their import statements automatically typed at the top of your class (be sure to use java.util.Date for the date).

- Before moving on, ensure you have told Android Studio we will use a “m” prefix on our member/class variables. Click File → Settings, then navigate to Editor → Code Style → Java and ensure there is a m in the Field name prefix box:



- Generate a getter for the read-only mId and both a getter and setter for mTitle, mDate, and mSolved. Right-click after the constructor and select Generate... → Getter and select the mId variable and click ok.
- Again, right-click after the constructor and select Generate... → Getter and Setter, highlight mTitle, mDate, and mSolved, then click ok.
- Open CrimeActivity's layout at res/layout/activity_crime.xml and replace the default layout with this FrameLayout XML:

<FrameLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/fragment_container"
android:layout_width="match_parent"
android:layout_height="match_parent" />
```

12. Define the strings that the user will see in res/values/strings.xml

```
<string name="crime_title_hint">Enter a title for the crime.</string>
<string name="crime_title_label">Title</string>
<string name="crime_details_label">Details</string>
<string name="crime_solved_label">Solved</string>
```

13. Define the UI. The layout for CrimeFragment will consist of a vertical LinearLayout that contains two TextViews, an EditText, a Button, and a Checkbox. To create a layout file, right-click the res/layout folder in the project tool window and select New → Layout resource file. Name this file fragment_crime.xml and enter LinearLayout as the root element. Click OK and Android Studio will generate the file for you.

14. Update the layout to include the UI components:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp"
    android:orientation="vertical">
```

```
<TextView
    style="?android:listSeparatorTextViewStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/crime_title_label"/>
```

```
<EditText
    android:id="@+id/crime_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/crime_title_hint"/>
```

```
<TextView
    style="?android:listSeparatorTextViewStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/crime_details_label"/>
```

```
<Button
    android:id="@+id/crime_date"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

```
<CheckBox
    android:id="@+id/crime_solved"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/crime_solved_label"/>
```

```
</LinearLayout>
```

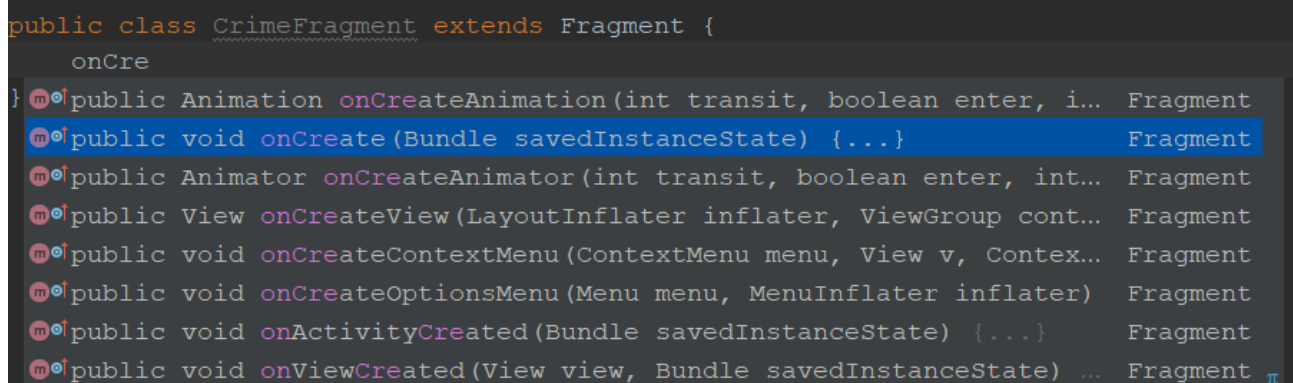
15. Create the Java class for the fragment, right-click the com.bignerdranch.android.criminalintent package and select New → Java Class. Name the class CrimeFragment and click OK to generate the class.

16. Turn this class into a fragment. Update CrimeFragment to subclass the Fragment class:

```
public class CrimeFragment extends Fragment {

}
```

17. As you subclass the Fragment class, you will notice that Android Studio finds two classes with the Fragment name. You will see Fragment (android.app) and Fragment (androidx.fragment.app). The android.app Fragment is the version of fragments built into the Android OS. You will use the support library version, so be sure to select the androidx.fragment.app version of the Fragment class when you see the dialog (or when you press Alt-Enter to resolve the reference error).
18. In CrimeFragment.java, add a member variable for the Crime instance and an implementation of Fragment.onCreate(Bundle). Type the first few characters of the method (e.g. "onCre") where you want to place the method. Android Studio will provide a list of suggestions, select the onCreate() method and press enter.



```
public class CrimeFragment extends Fragment {
    onCreate
}
m public Animation onCreateAnimation(int transit, boolean enter, i... Fragment
m public void onCreate(Bundle savedInstanceState) {...} Fragment
m public Animator onCreateAnimator(int transit, boolean enter, int... Fragment
m public View onCreateView(LayoutInflater inflater, ViewGroup cont... Fragment
m public void onCreateContextMenu(ContextMenu menu, View v, Contex... Fragment
m public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) Fragment
m public void onActivityCreated(Bundle savedInstanceState) {...} Fragment
m public void onViewCreated(View view, Bundle savedInstanceState) ... Fragment
```

19. Create a member variable to hold a Crime and initialize it to a new Crime in your onCreate method:

```
public class CrimeFragment extends Fragment {  
    private Crime mCrime;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        mCrime = new Crime();  
    }  
}
```

20. In CrimeFragment.java, add an implementation of onCreateView(...) that inflates fragment_crime.xml. You can use the same trick from above to fill out the method definition:

```
public class CrimeFragment extends Fragment {  
    private Crime mCrime;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        mCrime = new Crime();  
    }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View v = inflater.inflate(R.layout.fragment_crime, container, false);  
        return v;  
    }  
}
```

21. Wire up your EditText, first add a member variable for it:

```
private EditText mTitleField;
```

22. Next, get a reference to your EditText and add a TextWatcher (put the code in the onCreateView method, before you return the view):

```
mTitleField = (EditText) v.findViewById(R.id.crime_title);
mTitleField.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(
        CharSequence s, int start, int count, int after) {
        // This space intentionally left blank
    }

    @Override
    public void onTextChanged(
        CharSequence s, int start, int before, int count) {
        mCrime.setTitle(s.toString());
    }

    @Override
    public void afterTextChanged(Editable s) {
        // This space intentionally left blank
    }
});
```

23. Wire up your Button, first add a member variable for it:

```
private Button mDateButton;
```

24. Next get a reference to your button and set the text to be the date of the crime (put the code in the onCreateView method, before you return the view):

```
mDateButton = (Button) v.findViewById(R.id.crime_date);
mDateButton.setText(mCrime.getDate().toString());
mDateButton.setEnabled(false);
```

25. Wire up your CheckBox, first add a member variable for it:

```
private CheckBox mSolvedCheckBox;
```

26. Next get a reference to your CheckBox and add a OnCheckedChangeListener (put the code in the onCreateView method, before you return the view):

```
mSolvedCheckBox = (CheckBox)v.findViewById(R.id.crime_solved);  
mSolvedCheckBox.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        mCrime.setSolved(isChecked);  
    }  
});
```

27. You will receive an error as Android Studio will not know what OnCheckedChangeListener is. Also, when you press Alt+Enter, it will give you two options, ensure you select android.widget.CompoundButton.

28. Add your fragment to your CriminalActivity by opening CriminalActivity.java, getting a reference to the FragmentManager in the onCreate() method, create an instance of your fragment and add it to the transaction:

```
public class CrimeActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_crime);  
  
        FragmentManager fm = getSupportFragmentManager();  
        Fragment fragment = fm.findFragmentById(R.id.fragment_container);  
  
        if (fragment == null) {  
            fragment = new CrimeFragment();  
            fm.beginTransaction()  
                .add(R.id.fragment_container, fragment)  
                .commit();  
        }  
    }  
}
```


Final App Example

7:00

CriminalIntent

TITLE

Enter a title for the crime.

DETAILS

THU NOV 17 09:45:07 EST 2016

☐ Solved