

Директория res

Внутри Android проекта вы можете найти директорию **res**. Она находится в `<project-name>\app\src\<main или debug>\` или `<project-name>\app\src\<ваш build variant или build type или product flavor>\`

Что такое build variant, build type, product flavor вы узнаете чуть позднее. Пока что не заморачивайтесь по этому поводу.

Главное помнить, что все ресурсы и исходные коды с которыми вы, в основном, будете работать находятся в папке main.

В этой директории будут храниться все ресурсы приложения. К ресурсам относятся картинки, аудиозаписи, иконки, layout, strings и т.д. Об этом более подробно вы узнаете в следующих уроках.

Директория assets

В Android имеется еще один каталог, в котором могут храниться файлы - assets.

Этот каталог необходим для того, чтобы хранить в нем различные файлы для нашего Android-проекта, типа: .properties, .txt, .pdf и т.д. **assets** не существует по умолчанию в структуре нашего проекта, его нужно будет создать самим через меню Android Studio: **File -> New -> Folder -> Assets Folder**.

Этот каталог находится на том же уровне, что и **res**. Для файлов, располагающихся в assets, не генерируются идентификаторы ресурсов. Для их считывания необходимо указать путь к файлу. Путь к файлу является относительным и начинается с **assets**. Этот каталог, в отличие от **res**, позволяет задавать произвольную глубину подкаталогов и произвольные имена файлов и подкаталогов. Также, запомните, что в **assets** нельзя записывать(перезаписывать) никакие файлы, у вас это просто не получится.

Чтобы создать директорию assets, в меню нужно выбрать **File -> New -> Folder -> Assets Folder**.

AssetManager

Чтобы получить доступ к assets нам нужно обратиться к AssetManager:

```
AssetManager assetManager = getAssets();
```

Метод **getAssets()** есть у любого объекта класса Context.

С помощью AssetManager мы можем получить доступ к нашим файлам, которые хранятся в assets:

```
AssetManager assetManager = getAssets();
```

```
try {  
    InputStream is = assetManager.open("icon.png");  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

Метод **open(String fileName)** вернет нам объект класса **InputStream**. Данный **InputStream** мы можем преобразовать в файл и сделать все, что посчитаем нужным.