



## 4. What Are Resources? (HelloRes)

- Вместо съдържание:
- Какво са ресурсите и за какво ги ползваме?
- Базови и как да ги осигуряваме?
- Как да ги достъпваме?
- Типове ресурси (a looong list this is ... ;) )





# Ресурси (Resources)

- Вече се запознахме с .xml „начина“, да опишем UI елементите на нашето приложение и да ги достъпваме (управляваме) през код;
- Основната ни цел днес е да придобием представа, какво още можем да моделираме в .xml, къде в нашият Android проект се намира и до колко ни улеснява /или затруднява ;)/ да го използваме;
- Вместо definition: Външни за кода на нашето приложение единици (entities), които Android или ние, можем да „управляваме“;
- Външни в смисъл че могат да се поддържат (сменят, редактират), без да е необходимо да се променя кода на приложението;
- „Android или ние“ - в смисъл че entities се управляват от Android за нас или ние се грижим за това да ги достъпим, изведем (или скрием) в зависимост от действията на потребителя;



## Ресурси (Resources) cont...

- Примери за ресурси: изображения, низови литерали, първични файлове (.mp3) или дори UI компоненти;
- Основната причина, поради която искаме (или биваме принудени) да разглеждаме ресурсите като външни за кода на приложението е възможността да предоставяме алтернативи на определен ресурс.
- Това е изключително важно, имайки в предвид голямото разнообразие от у-ва поддържани от Android и различните комбинации от версия на системата и hardware device vendor, които могат да се окажат runtime среда за нашето приложение;
- Ресурсите на нашето Android приложение, подреждаме в res/ директорията на проекта му; res/ може да има доста сложна структура от поддиректории, с чиято помощ организираме ресурсите по тип и конфигурация;
- За всеки вид (тип) ресурс, разполагаме с начин да опишем негов default вариант и една или повече алтернативи;
- Алтернативите на ресурс са всъщност начина за справяне с многообразието от у-ва;



„brief“ пример за ресурс – във  
вид на look && feel;





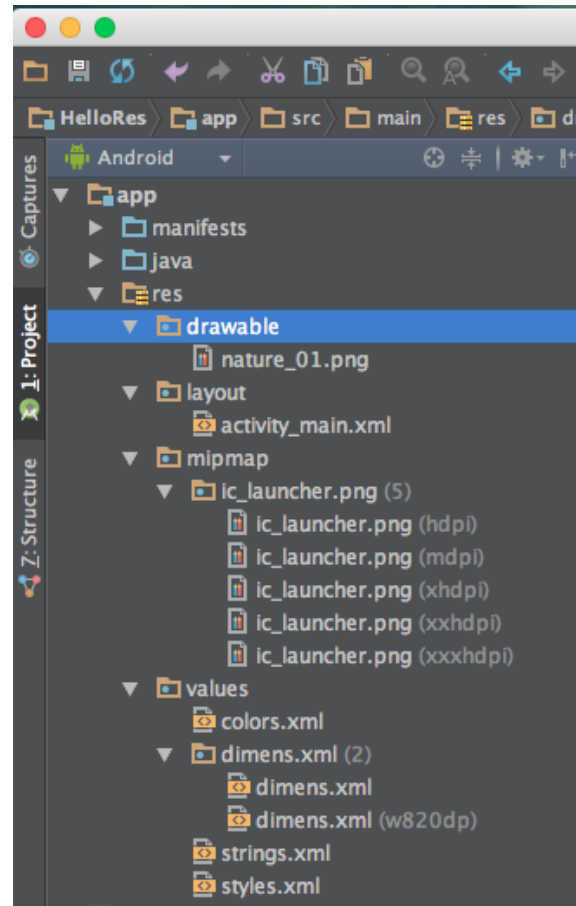
## „brief“ пример за ресурс - обясненията;

- Layouts, за нашите Activities (Views) са ресурси (например LinearLayout или Relative Layout);
- Първото изображение ни показва как би изглеждало приложение, което не предоставя алтернатива за своя layout на две различни у-ва (с различни големина на екрана);
- Второто изображение - приложение, предоставящо алтернатива на своя layout;
- Съответно, първото разполага само с res/layout директория, докато второто разполага с още една - res/layout-land, в която е описана алтернативата (RelativeLayout в този случай);



# Предоставяне на ресурси (./1)

- Външните за кода ресурси групираме в `res/` директорията на проекта;
- За всеки тип (вид) ресурс е отделена поддиректория с определено име;
- `res/` от demo проекта:





## Предоставяне на ресурси (./2)

- Основните (базовите) типове ресурси и съответните им поддиректории в `res/` са изведени във файла `base-res-subdirs.pdf`
- Правилото за подреждане на ресурсите (ресурсните файлове) в поддиректориите – всеки файл (`.xml` или друг) представлява отделен ресурс, видим под съответният „namespace“ - `R.<resource-subdir>.<resource-name>`
- След добавянето на ресурс в някоя от поддиректориите, Android Studio се грижи за това да имаме коректни стойности за `R`; При добавяне на ресурс, който не се поддържа, със сигурност получаваме предупреждение и грешка;
- Горното правило, разбира се, си има и изключение; Това е `values/` поддиректорията на `res/`; За всеки файл в тази директория, всеки елемент на `<resources>`, определя отделен ресурс;



## Предоставяне на ресурси (./3)

- Освен базовите ресурси с които разполагаме, имаме и възможност да ги комбинираме или по друг начин казано – да създаваме алтернативи на ресурс;
- Пълният списък с алтернативи, поддържан от Android се намира във файла: `res-alternatives-labels.pdf` (доооста е дълъг);
- Създаваме алтернативен ресурс, като създадем нова директория със име `<resource_name>-<qualifier>`;
- Това се отнася за базови ресурси (тези от `base-res-subdirs.pdf`);
- Стойностите на Qualifiers са от `res-alternatives-labels.pdf`;



## Предоставяне на ресурси (. /4)

- Правилата, по които подреждаме qualifiers са следните:
- Можем да укажем повече от един qualifier, за даден базов ресурс; Всички те трябва да са разделени със тире;
- Редът на qualifiers трябва да е този описан в res-alternatives-labels.pdf;
- Не можем да влягаме алтернативни ресурси;
- В имената на ресурсите не се отличават големи и малки букви; След компилация, всички те се конвертират до малки букви;
- Всеки qualifier може да участва само веднъж в дадена алтернатива на ресурс; Ако искаме да използваме един и същ ресурс например за en и fr не можем да напишем <resource\_name>-en-fr;
- Трябва да направим две отделни директории: <resource\_name>-en и <resource\_name>-fr;
- Повече информация за базовите типове ресурси и техните алтернативи, можем да намерим тук:  
<http://developer.android.com/guide/topics/resources/providing-resources.html>



# Достъпване на ресурси (./1)

- След като добавим ресурси в нашето приложение, можем да ги достъпваме/прилагаме/;
- За целта използваме техните Ids; Всички те са дефинирани за нас в класа R; Това се прави автоматично от Android Studio, с помощта на програмата aapt;
- R класа съдържа IDs, за всички ресурси в res/ директорията; За всеки тип ресурс (поддиректория в res/) се съпоставя подклас в R; И за всяко entity в дадена поддиректория, разполагаме с int в съответният подклас; (R.drawable.nature\_01 например);
- Въпреки че R съдържа IDs на всички ресурси които сме добавили в приложението, не трябва да го използваме него за да открием (намерим) дадено ID;
- Resource ID винаги се състои от:
- Тип (resource type) – например string, drawable, layout (тоест всяка поддиректория е тип);
- Име (resource name) – това е или името на файла в съответната поддиректория (без разширението) или стойността на атрибута android:name, ако ресурса е от тип values/, например string;



## Достъпване на ресурси (./2)

- Достъпваме ресурси по два основни начина:
- В кода на приложението, използвайки статичните Ids, от подкласовете на R; Например `R.string.some_string`;
- В `.xml` – например `@string/some_string`; Очаквано `string` е типа на ресурса а `some_string` е неговото име; Навсякъде в `.xml` където се очаква стойност на ресурс можем да използваме този синтаксис;
- Пълният синтаксис за да достъпим ресурс през кода на приложението е:
- `[<package_name>.]R.<resource_type>.<resource_name>`
- Пълният синтаксис за да достъпим ресурс през `.xml` е:
- `@ [<package_name>:]<resource_type>/<resource_name>`



## Достъпване на ресурси (./3)

- Няколко примера за достъпване на ресурси през кода на приложението:

```
// a drawable as a background;  
getWindow().setBackgroundDrawableResource(R.drawable.my_background_image) ;  
  
// our activity title;  
getWindow().setTitle(getResources().getText(R.string.main_title));  
  
// our custom layout;  
setContentView(R.layout.main_screen);  
  
// setting predefined text inside text object;  
TextView msgTextView = (TextView) findViewById(R.id.msg);  
msgTextView.setText(R.string.hello_message);
```



## Достъпване на ресурси (./4)

- Няколко примера за достъпване на ресурси през .xml:

```
<Button  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/submit" />
```

```
<EditText xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:textColor="@color/opaque_red"  
    android:text="@string/hello" />
```



## Типове ресурси (the long list from beggining)

- Пълен списък на всички типове ресурси поддържани от Android можем да намерим тук:
- <http://developer.android.com/guide/topics/resources/available-resources.html>
- За всеки от базовите типове е описан .xml синтаксиса и основните use cases за които е мислен този тип;
- Въпреки многото информация, това е добро refernce място, което можем да използваме, ако искаме да задълбаем (се запознаем) с даден ресурсен тип;



# HelloRes

- Приложението днес – HelloRes.zip
- Минимално количество код;
- Няколко различни типа ресурси;
- И все пак някаква динамика;
- С други думи – wire-framing our app gui with resources;