

Anotacion StringRes

El "@StringRes" permite obtener un string desde la ubicación de los recursos. La documentación de esta anotación la veremos en el siguiente enlace:

<https://github.com/androidannotations/androidannotations/wiki/Resources#stringres>

Crear un nuevo proyecto con una *Empty Activity*.

Añadiendo los Strings

Debe agregar estos Strings en res >> values >> strings.xml:

```
Key = the_avengers
    Value = The Avengers

Key = iron_man
    Value = Iron Man
Key = iron_man_actor
    Value = Robert Downey Jr.

Key = viuda_negra
    Value = Viuda Negra
Key = viuda_negra_actor
    Value = Scarlett Johansson

Key = capitan_america
    Value = Capitán América
Key = capitan_america_actor
    Value = Chris Evans

Key = thor
    Value = Thor
Key = thor_actor
    Value = Chris Hemsworth

Key = hulk
    Value = Hulk
Key = hulk_actor
    Value = Mark Ruffalo

Key = sprider_man
    Value = Sprider-Man
Key = sprider_man_actor
    Value = Tom Holland
```

Inicializando los recursos en el Java

Para inicializar un String Java desde un recurso, debe agregar @StringRes y hay

tres formas:

- Utilizando el ID de recurso
- Directamente
- Directamente con más de una variable.

Usando el ID de recurso

El identificador del recurso debe estar en el parámetro del "@StringRes":

```
@StringRes (R.string.the_avengers)
String avengers;
```

Directamente

El identificador será el nombre de la variable.

```
@StringRes
String iron_man;
```

Directamente con más de una variable.

La misma filosofía que la opción previa pero con más de una variable.

```
@StringRes
String iron_man, iron_man_b;
```

Recuperando todos los recursos

A continuación agregaremos este código en la clase MainActivity:

```
@StringRes(R.string.the_avengers)
String avengers;
@StringRes
String iron_man;
@StringRes
String iron_man_actor;
@StringRes
String viuda_negra, viuda_negra_actor;
@StringRes
String capitan_america, capitan_america_actor;
@StringRes
String thor, thor_actor;
@StringRes
String hulk, hulk_actor;
@StringRes
String sprider_man, sprider_man_actor;
```

Y posteriormente añade este método:

```
private String generateAvengerInfo(String actor, String heroe, int c  
    return actor + " as " + heroe + " \n\n";  
}
```

Finalmente vamos a crear un nuevo método anotado con `@AfterViews` y dentro del método mostraremos con el debugger todos los recursos que llaman al método anterior (`generateAvengerInfo`).