

# facebook / fresco

Android image library






# Co je to fresco?

- Pomocná knihovna pro zobrazování obrázků v Android aplikacích
- Efektivní práce s pamětí
- Malá náročnost na CPU
- Podpora od Androidu 2.3
- JPEG Streaming - rychlé načítání náhledů
- Animace - podpora GIFů a WebPs
- Použito v React Native, Facebook, Messenger, Wikipedia a další..
- Malá část v C++, většina Java

# Další výhody

- Placeholders
- Automatické škálování obrazu
- Zobrazení načítání
- RAM a disk cache
- API pro vlastní filtry (nebo jiné knihovny např. [wasabeef/fresco-processors](#))

# Informace o vývoji

-  Watch ▾ 796
  -  Star 8,486
  -  Fork 2,456
- 6 hlavních vývojářů (5 z facebooku)
  - Založeno v březnu 2015

# Vývojový proces

1. Pull request ověřen fb zaměstnancem na githubu
2. Pull request poslán do interního repozitáře fb. Rozsáhlé automatické testy.
3. Mergnuto do master branch a vyexportováno zpět na github

# Současné API

Nutnost pokaždé ověřit nenulovost url. Jinak vyjímka.

```
String url = "https://www.gstatic.com/webp/gallery/2.sm.webp";  
if(url != null){  
    mStaticWebView.setImageURI(Uri.parse(url));  
}
```

# Moje úprava

Nová metoda `SimpleDraweeView.setImageURI(String)`, ve které je kontrola nenulovosti.

```
String url = "https://www.gstatic.com/webp/gallery/2.sm.webp";  
mStaticWebpView.setImageURI(url);
```

# Moje úprava

```
/**  
 * Displays an image given by the uri string.  
 *  
 * @param uriString uri string of the image  
 * @param callerContext caller context  
 */  
public void setImageURI(@Nullable String uriString, @Nullable Object  
callerContext) {  
    Uri uri = (uriString != null) ? Uri.parse(uriString) : null;  
    setImageURI(uri, callerContext);  
}
```



# Děkuji za pozornost

<https://github.com/facebook/fresco>