ButterKnife

Butterknife is a view binding tool that uses annotations to generate boilerplate code for us. This tool is developed by Jake Wharton at Square and is essentially used to save typing repetitive lines of code like findViewById(R.id.view) when dealing with views thus making our code look a lot cleaner. To be clear, Butterknife is not a dependency injection library. Butterknife injects code at compile time. It is very similar to the work done by Android Annotations.

LeakCanary

According to Pierre-Yves Ricau from medium.com , some objects have a limited lifetime. When their job is done, they are expected to be garbage collected. If a chain of references holds an object in memory after the end of its expected lifetime, this creates a memory leak. When these leaks accumulate, the app runs out of memory. For instance, after Activity.onDestroy() is called, the activity, its view hierarchy and their associated bitmaps should all be garbage collectable. If a thread running in the background holds a reference to the activity, then the corresponding memory cannot be reclaimed. This eventually leads to an OutOfMemoryError crash. To fix the memory leak first we need to detect the leaks. LeakCanary is a memory leak detection library for Android and Java.

Retrofit

Retrofit is a type-safe HTTP client for Android and Java. The Retrofit class generates an implementation of the GitHubService interface. Each Call from the created GitHubService can make a synchronous or asynchronous HTTP request to the remote webserver.

Dagger

Dagger is a fast dependency injector for Android and Java. The best classes in any application are the ones that do stuff: the BarcodeDecoder, the KoopaPhysicsEngine, and the AudioStreamer. These classes have dependencies; perhaps a BarcodeCameraFinder, DefaultPhysicsEngine, and an HttpStreamer. To contrast, the worst classes in any application are the ones that take up space without doing much at all: the BarcodeDecoderFactory, the CameraServiceLoader, and the MutableContextWrapper. These classes are the clumsy duct tape that wires the interesting stuff together. Dagger is a replacement for these FactoryFactory classes. It allows you to focus on the interesting classes. Declare dependencies, specify how to satisfy them, and ship your app. By building on standard javax.inject annotations (JSR-330), each class is easy to test. You don't need a bunch of boilerplate just to swap the RpcCreditCardService out for a FakeCreditCardService. Dependency injection isn't just for testing. It also makes it easy to create reusable , interchangeable modules. You can share the same AuthenticationModule across all of your apps. And you can run DevLoggingModule during development and ProdLoggingModulein production to get the right behavior in each situation.

Glide

Glide is a fast and efficient open source media management and image loading framework for Android that wraps media decoding, memory and disk caching, and resource pooling into a simple and easy to use interface.

Glide supports fetching, decoding, and displaying video stills, images, and animated GIFs. Glide includes a flexible API that allows developers to plug in to almost any network stack. By default Glide uses a custom HttpUrlConnection based stack, but also includes utility libraries plug in to Google's Volley project or Square's OkHttp library instead.

Glide's primary focus is on making scrolling any kind of a list of images as smooth and fast as possible, but Glide is also effective for almost any case where you need to fetch, resize, and display a remote image.

References

http://stackoverflow.com/documentation/android/1072/butterknife#t=20170426205116855513 2

https://medium.com/square-corner-blog/leakcanary-detect-all-memory-leaks-875ff8360745

http://square.github.io/retrofit/

http://square.github.io/dagger/

https://github.com/bumptech/glide