

Challenge

Task

Imagine you are on the Pinterest iOS team and you are working with some colleagues on the pinboard (the scrolling wall of images), you split up the tasks among each other and your task is to create an image loading library that will be used to asynchronously download the images for the pins on the pinboard when they are needed.

The library will also be useful for all other parts of the app where asynchronous remote image loading is required. The images are available on a publicly accessible URL (like a CDN). The library should be general purpose and not assume anything about the use case, the pinboard is an example but other parts of the app that show images will also use it (e.g. a user's profile pic on the profile screen).

One of your colleagues will also want to use the library for loading JSON documents, and you just know that your boss and colleagues will love your library so much that they will ask you to support other data types in the future as well, so your design should not limit support to a particular data type.

The purpose of the library is to abstract the downloading (images, pdf, zip, etc) and caching of remote resources (images, JSON, XML, etc) so that client code can easily "swap" a URL for any kind of files (JSON, XML, etc) without worrying about any of the details. Resources which are reused often should not be continually re-downloaded and should be cached, but the library cannot use infinite memory.

Requirements

1. Use the following url for loading data: <http://pastebin.com/raw/wgkJgazE>
2. Images and JSON should be cached efficiently (in-memory only, no need for caching to disk);
3. The cache should have a configurable max capacity and should evict images not recently used;
4. An image load may be cancelled;
5. The same image may be requested by multiple sources simultaneously (even before it has loaded), and if one of the sources cancels the load, it should not affect the remaining requests;
6. Multiple distinct resources may be requested in parallel;
7. You can work under the assumption that the same URL will always return the same resource;
8. The library should be easy to integrate into new iOS project / apps;
9. You are supposed to build a solid structure and use the needed programming design patterns;
10. Think that the list of item returned by the API can reach 100 items or even more. At a time, you should only load 10 items, and load more from the API when the user reach the end of the list;
11. Usage of size classes is an advantage;
12. Usage of cool animations and transitions is an advantage;
13. Adding "pull to refresh" is an advantage.