

List of issues with the app

Things that don't work as well as they should:

- When a new bid is received, the items in the Search screen are not updated (only on the Watchlist screen). That's OK because the search data is only temporary anyway.
- There is a potential race condition with `cullBids()`. If `cullBids()` happens while a refresh is taking place, the new data from the server will still include the old bids. So the networking completion handler should really cull again. (Will not solve that for this demo app; after the rewrites we'll be doing, this culling is no longer necessary anyway.)
- After adding a new bid, the number of bids (and potentially the largest bid amount) has changed. In the starter project, this properly updates the Watchlist screen, but not:
 - the search screen. It only works when the search screen still shows the same Item instance that is also in the watch list. After a new search it will have a different Item instance for the same item ID and the update won't have any effect.
 - the activity screen. The new bid won't appear until you pull-to-refresh or the automatic polling brings in the new bid.
 - any open Item Detail screens. These should add the new bid to their table views.
- After you start watching an item, the bids for this item are not immediately added to the Activity screen. You need to wait until the server is polled again.
- The Item objects in the search screen and in the watchlist are not always the same instances even though they may have the same ID. This can cause an item to be updated in the watchlist, but not in the search screen (when you or someone else adds a new bid). The solution is to "sync" these two lists, so that the same ID always gives you the same instance (what Core Data calls "uniquing").
- Network spinner in status bar can conflict between different view controllers. Example: if search is still busy but the "latest" polling finishes earlier. This is something that the networking layer can manage because networking requests are no longer independent of each other.

