// Ping API for Channels (roll/feature?segment=ipad_standard)
[ShelbyAPIClient getAllChannels];

// On Success, store channels on background thread using instance of CoreDataUtility
[dataUtility storeChannels:JSON];

/* Parse JSON for ChannelRolls and ChannelDashboards in CoreDataUtility, and store ChannelRoll and ChannelDashboard meta-data on Roll and Dashboard NSManagedObjects, respectively. Also, store rolls/dashboard in the order in which they appeared in the JSON results using the displayTag property. */
// Rolls
[self storeChannelRolls:rollsArray withInitialTag:displayTag];
[ShelbyAPIClient getChannelRoll:rollID];
[dataUtility storeFrames:JSON forChannelRoll:rollID];
// Dashboards
[self storeChannelDashboard:dashboardArray withInitialTag:displayTag];
[ShelbyAPIClient getChannelDashboardEntries:dashboardID];
[dataUtility storeDashboardEntries:JSON forDashboard:channelID];

// Sync CoreData ChannelRolls and ChannelDashboards with results from web.
/ * For example, let's say the first time the app is launched 3 rolls (R1, R2, R3) and 3 Dashboards (D1, D2, D3) are returned. Let's say a week later, the API throws back R2, R3, R4 and D3, D4, D5. After running the storeChannelRolls: and storeChannelDashboard: methods, Core Data will have R1, R2, R3, R4 and D1, D2, D3, D4, D5. The sync method makes sure that Core Data removed rolls and dashboard that no longer exist, which in this case would be R1, D1, D2.
[self syncChannels:resultsDictionary];
[self syncChannelRolls:webRollIDsArray];
[self syncChannelDashboards:webChannelIDsArray];

// Fetch older ChannelDashboard entries when swiping to the end of a collectionView
[ShelbyAPIClient getMoreDashboardEntries:skipParam forChannelDashboard:dashbaordID];
[dataUtility storeDashboardEntries:JSON forDashboard:dashboardID];

// Fetch older ChannelRoll frames when swiping to the end of a collectionView
[ShelbyAPIClient getMoreFrames:skipParam forChannelRoll:rollID];
[dataUtility storeRollFrames:JSON forGroupType:GroupType_ChannelRoll];