```
// Stream
// Get Stream from API
[ShelbyAPIClient getStream];
// Store Stream Meta Data as Dashboard NSManagedObject
[dataUtility storeStream:JSON];
// Store dashboardEntries from Stream in Core Data. Use user object’s userID as dashboardID
(this connects each dashabordEntry to the dashboard that it belongs to, in this case the Stream).
[self storeDashboardEntries:resultsDictionary forDashboard:user.userID];
```

```
// Stream - Get older stream entries when reaching the end of a collectionView. Takes one
parameter, the amount the API should skip before returning new results. The skip value is
calculated by grabbing the current number of entries for a dashboard
[ShelbyAPIClient getMoreFramesInStream:skipParam];
[dataUtility storeStream:JSON];
[self storeDashboardEntries:resultsDictionary forDashboard:user.userID];
```

```
// Personal Roll
// Fetch Personal Roll from API
[ShelbyAPIClient getPersonalRoll];
// Store Roll-Frames for Personal Roll
[dataUtility storeRollFrames:JSON forGroupType:GroupType_PersonalRoll];
// Sync Personal Rolls with Web. If user had Frame X in personal roll at time t1, but at a later date,
they deleted Frame X on Shelby.tv Web, then Core Data should remove Frame X
[NSTimer scheduledTimerWithTimeInterval:2.0f target:[ShelbyAPIClient class]
selector:@selector(getPersonalRollForSync) userInfo:nil repeats:NO];
```

```
// Personal Roll - Get older personal roll entries.
[ShelbyAPIClient getMoreFramesInPersonalRoll:skipParam];
[dataUtility storeRollFrames:JSON forGroupType:GroupType_PersonalRoll];
[NSTimer scheduledTimerWithTimeInterval:2.0f target:[ShelbyAPIClient class]
selector:@selector(getPersonalRollForSync) userInfo:nil repeats:NO];
```

```
// Likes (logged-in user)
// Fetch Likes from API
[ShelbyAPIClient getLikes];
// Store Roll-Frames for Likes
[dataUtility storeRollFrames:JSON forGroupType:GroupType_Likes];
// Sync Likes with Web. Works the same way as personal roll syncing.
[ShelbyAPIClient getLikesForSync];
```

```
// Likes (logged-in user) - Get older likes entries.
[ShelbyAPIClient getMoreFramesInLikes:skipParam]
[dataUtility storeRollFrames:JSON forGroupType:GroupType_Likes];
[ShelbyAPIClient getLikesForSync];
```

```
// Likes (logged-out user)
// Store frame as Liked frame even though user isn't logged out. This is achieved by
changing a boolean flag from NO to YES on a frame that is liked by a logged-out user;
frame.isStoredForLoggedOutUser == YES;
CoreDataUtility *dataUtility = [[CoreDataUtility alloc]
 initWithRequestType:DataRequestType_StoreLoggedOutLike];
[dataUtility storeFrameInLoggedOutLikes:frame];
// Also, PUT an anonymous like on that frame on web.
[ShelbyAPIClient postFrameToLikes:frame.frameID];
```