## Goals

Display content (from gdrive, box, dropbox, wiki’s, etc) related to people a user is following.

## Account Table

This table has all people in the system. Some accounts are provisioned and have enough information to extract content from the service they represent. Some accounts only have profile information for the account they represent.

|  |  |
| --- | --- |
| **Field** | **Description** |
| objectId | The Parse Object Id |
| serviceId | The id of the service { “gdrive” | “dropbox” | … } |
| userId | The service specific id of the user represented |
| refreshToken | The OAuth refresh token |
| userEmail | the email address of the user |
| userDisplayName | the display name of the user |
| userPictureURL | the url to a picture of the user |
| watchState | the service specific watch state used for notifications |
| following | array of objectId’s of the accounts this user is following |
| syncState | the sync state |

## AccountVector Table

All the internal and external accounts a given account can see is represented by this table.

|  |  |
| --- | --- |
| **Field** | **Description** |
| objectId | The Parse Object Id |
| account | The account that can see the target account |
| targetAccount | The account that can be seen by the owner account |

## 

## 

## File Table

## This table contains file metadata.

|  |  |
| --- | --- |
| Field | Description |
| objectId | The Parse Object Id |
| fileId | the service specific id of the file |
| account | the pointer to account object that owns this file |
| mimeType | the mime type of the file |
| tipRevision | the last revision id of the file |
| fileName | the name of the file |
| thumbnailFile | a thumbnail representation of this file |
| modified | timestamp the file was modified |
| shareDate | timestamp the file was shared |
| modifiedByMe | timestamp the file was last modified by this account |
| viewDate | timestamp the file was last viewed by this account |
| sortDate | last modified timestamp adjusted for the edits by the account |

## 

## MostRecentActivity Table

This table contains a subset of the entire activity history for all content. It specifically contains the most recent activity for each content item and each user.

|  |  |
| --- | --- |
| **Field** | **Description** |
| objectId | The Parse Object Id |
| file | the pointer to file object which this activity is for |
| account | the pointer to account object that can see this activity |
| activityAccountId | the id of the account that made this activity |
| activityType | the type of activity (created, updated, comment, etc) |
| activityDate | the date this activity occurred |

## ActivityVector Table

This table contains the map for each user to the activities.

|  |  |
| --- | --- |
| **Field** | **Description** |
| objectId | The Parse Object Id |
| account | the pointer to account object that can see this activity |
| file | the pointer to file object |
| activity | the pointer to activity object that belongs to users the account is following |

## Client Interaction

To paint the “*conversation view*” the client does the following parse query

var account = [ account1, account2, account3 ];

function getActivityVectors(accountArray) {

return new Parse.Query('ActivityVector')

.containedIn('account', accountArray)

.find();

}

function getActivities(activityVectors) {

return \_.pluck(activityVectors, 'activity');

}

function sortActivities(activities) {

return \_.sortBy(activities, function(activity) {

return activity.activityDate;

});

}

var activities = getActivityVectors(account)

.then(getActivities)

.then(sortActivities);

## Outstanding Questions

1. How is all this stuff populated and updated and kept in sync
2. How does the client make people recommendations quickly