Pensieve header: Make a facebook friends graph following http://mathematica.stackexchange.com/questions/11673/how-to-play-with-facebook-data-inside-mathematica.

Get facebook token from http://developers.facebook.com/tools/explorer/?method=GET&path=me and put it below.

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\2012-10"];
PDFFileName = "MyFacebookGraph@150912.pdf";
token = InputString["Enter FB token:"]

(*Get friends name and facebook code*)getFriendsList[] :=
 Rule @@@ Import["https://graph.facebook.com/me/friends?access_token=" <> token,
    "JSON"][[1, 2, All, All, 2]]

(*Get your friends photo link and sex*)
{cName, cPhotoLink, cSex, cCode} = Range[4];
getFriendsData[fList_] := Module[{url, query, friendsData, fListString},
  fListString = conv2StringList[fList[[All, 1]]];
  url = "https://api.facebook.com/method/fql.query?access_token=" <>
    token <> "&query=";
  query = url <> "SELECT uid, name, sex,pic_square FROM user WHERE uid in " <>
    fListString <> "&format=JSON";
  query = StringReplace[query, " " → "%20"];
  friendsData = Import[query, "JSON"][[All, All, -1]]]

(*Get friends pairs connections*)
getFriendsPairsPart[fList1_, fList2_] :=
 Module[{url, query1, friendsPairs, friendsStr1, firendsStr2},
  friendsStr1 = conv2StringList[fList1];
  firendsStr2 = conv2StringList[fList2];
  url = "https://api.facebook.com/method/fql.query?access_token=" <>
    token <> "&query=";
  query1 = url <> "SELECT uid1, uid2 FROM friend WHERE uid1 in " <>
    friendsStr1 <> "and uid2 in" <> firendsStr2 <> "&format=JSON";
  query1 = StringReplace[query1, " " → "%20"];
  friendsPairs = Import[query1, "JSON"];
  friendsPairs = (Sort /@ friendsPairs) // Union]

getFriendsPairs[friendsList_] :=
 Module[{groupsComb, groupsCompLen, maxUsers = 3, friendsPairs, i = 1},
  SetSharedVariable[i];
  groupsComb = Partition[friendsList[[All, 1]], maxUsers, maxUsers, 1, {}];
  groupsComb = Subsets[groupsComb, {2}];
  groupsCompLen = Length[groupsComb];
  Print["Extracting Connections"];
```

```
  Print[Dynamic@mrtProgressBar[i, groupsCompLen]];
  friendsPairs = Flatten[ParallelMap[(i++;
       getFriendsPairsPart@@#) &, groupsComb], 1];
  friendsPairs = UndirectedEdge@@@friendsPairs[[All, All, 2]];
  friendsPairs = Union[Sort /@ friendsPairs];
  Print[Row@{"Connections number: ", Length@friendsPairs}];
  friendsPairs]

(*Get friends photos*)
getFriendsPhotos[friendsData_] :=
 Module[{append, friends, photos, page, i = 1, tabImg = {}}, SetSharedVariable[i];
  SetSharedVariable[tabImg];
  Print["Extracting user pictures:"];
  Print[Dynamic@mrtProgressBar[i, Length[friendsData]]];
  Print[Dynamic@GraphicsGrid[If[Length[tabImg] == 0,
      {{""}}, Partition[tabImg, 10, 10, 1, {}]], ImageSize → 200]];
  append[data_Image] := Module[{}, If[Length[tabImg] > 100, tabImg = {}];
    AppendTo[tabImg, data];
    data];
  (*CloseKernels[];
  LaunchKernels[8];*)photos = ParallelMap[(i++;
      {append[Import[#[[cPhotoLink]]]], ToString[#[[cCode]]]}) &, friendsData];
  Print[Row@{"Photo's number: ", Length[photos]}];
  photos]

adjustPhotos[friendsPhotos_, friendsPairs_] :=
 Module[{friends, photosSel, graph, page}, graph = Graph@friendsPairs;
  friends = VertexList@graph;
  page = PageRankCentrality[graph, 0.1];
  page = Rescale[page, {0, Max[page]}, {0.1, 0.9}];
  page = Rule@@@Transpose[{friends, page}];
  photosSel = Select[friendsPhotos, MemberQ[friends, #[[2]]] &];
  (#[[2]] → Hyperlink[Magnify[#[[1]], #[[2]]] /. page,
       "http://www.facebook.com/profile.php?id=" <> #[[2]]]) & /@ photosSel]

(*Plot Facebook graph*)
createGraph[friendsPairs_, friendsPhotosForVertex_] :=
 Module[{g1, g2, g3, label}, g1 = Graph[friendsPairs,
    VertexShape → friendsPhotosForVertex, VertexSize → 5, EdgeStyle → Opacity[0]];
  g2 = Graph[friendsPairs, VertexSize → 0, EdgeStyle → Thickness[0.0001]];
  label = Graphics[{Style[Text["by Rodrigo Murta\nwww.rodrigomurta.com"], Blue]},
    ImageSize → 300];
  g3 = Show[g2, g1, label, ImageSize → 1000]]
```

```
(*Execute code*)
createMyFacebookPDF[] :=
 Module[{myFacebookGraph}, SetDirectory[NotebookDirectory[]];
   Print["Extracting friends data"];
   friendsList = getFriendsList[];
   (* friendsList=DeleteCases[friendsList, "100000344024829"→_]; *)
   Print[friendsList];
   friendsData = getFriendsData[friendsList];
   friendsPairs = getFriendsPairs[friendsList];
   friendsPhotos = getFriendsPhotos[friendsData];
   photosForVertice = adjustPhotos[friendsPhotos, friendsPairs];
   Print["Creating GraphPlot"];
   myFacebookGraph = createGraph[friendsPairs, photosForVertice];
   Print["Creating PDF"];
   Export[PDFFileName, myFacebookGraph];
   Print["PDF Created!"];] // Quiet
(*quiet to avoid uni core msg*)

(*Other Funcitons*)
mrtProgressBar[var_, total_] := Row[{ProgressIndicator[var, {0, total}],
   " ", Row[{NumberForm[100. var / total, {∞, 2}], "%"}], "% ", var}]
conv2StringList[list_] := StringReplace[ToString[list],
  {"{" → "(", "}" → ")", " " → ""}]

createMyFacebookPDF[]
```

```
Extracting friends data
$Failed[[1, 2, All, All, 2]]
Extracting Connections
mrtProgressBar[i$504, groupsCompLen$504]
```

Part::partd : Part specification $Failed[[1, 2, All, All, 2]] is longer than depth of object.

Part::partd : Part specification All[[1]] is longer than depth of object.

Part::partd : Part specification $Failed[[1, 2, All, All, 2]] is longer than depth of object.

Part::partd : Part specification All[[1]] is longer than depth of object.

StringJoin::string : String expected at position 2 in https://api.facebook.com/method/fql.query?access_token= <> token <> &query=.

StringJoin::string : String expected at position 2 in https://api.facebook.com/method/fql.query?access_token= <> token <> &query=SELECT uid1, uid2 FROM friend WHERE uid1 in Alland uid2 in1&format=JSON.

StringJoin::string : String expected at position 2 in https://api.facebook.com/method/fql.query?access_token= <> token <> &query=SELECT uid1, uid2 FROM friend WHERE uid1 in Alland uid2 in1&format=JSON.

General::stop : Further output of StringJoin::string will be suppressed during this calculation.

StringReplace::strse : String or list of strings expected at position 1 in StringReplace[https://api.facebook.com/method/fql.query?access_token= <> token <> &query=SELECT uid1, uid2 FROM friend WHERE uid1 in Alland uid2 in1&format=JSON, → %20].

Import::chtype : First argument StringReplace[https://api.facebook.com/method/fql.query?access_token= <> token <> &query=SELECT uid1, uid2 FROM friend WHERE uid1 in Alland uid2 in1&format=JSON, → %20] is not a valid file, directory, or URL specification.

Union::normal : Nonatomic expression expected at position 1 in Union[$Failed].

```
Connections number: 3
Extracting user pictures:
mrtProgressBar[i$1318, 0]
```

```
Photo's number: 0
Creating GraphPlot
Creating PDF
PDF Created!
```