```
 1 var fs = require('fs');
 2 var request = require('request');
 3 var async = require('async');
 4 var _ = require('lodash');
 5 require('dotenv').load();
 6
 7 var apiRoot = "https://api.github.com";
 8 var API_KEY = process.env.OAUTH_TOKEN;
 9 var oAuthAppend = "?access_token=" + API_KEY;
10 var standardHeaders = {
11   "User-Agent": "benkahle"
12 };
13
14 var storedReposByOwnerFile = "./reposByOwner.json";
15 var storedCollabsByPersonFile = "./collabsByPerson.json";
16 var storedBDCollabsByPersonFile = "./bdCollabsByPerson.json";
17
18 var namesList = fs.readFileSync("./githubnames.txt", "utf-8").split("\n").slice(0,-1);
19
20 //Users -> Repos
21 var reposByOwner = {};
22 //Repos -> Collaborators(users)
23 var collabsByPerson= {};
24
25 // Get repositories by a github username
26 var getUserRepos = (userName, callback) => {
27   var options = {
28     url: `${apiRoot}/users/${userName}/repos${oAuthAppend}`,
29     headers: standardHeaders
30   };
31 Â
32   request.get(options, (err, res, body) => {
33     if (!err && res.statusCode === 200) {
34       var info = JSON.parse(body);
35       // Grab just the list of repository names from the response
36       var repoNames = info.map(repo => repo.name);
37       callback(null, repoNames);
38     }
39   });
40 };
41
42 // Get list of collaborators from a user's repository
```

```
var getCollabsFromUserRepo = (userName, repo, callback) => {
  console.log(userName, repo);
  var options = {
    url: `${apiRoot}/repos/${userName}/${repo}/contributors${oAuthAppend}`,
    headers: standardHeaders
  };

  request.get(options, (err, res, body) => {
    if (err) {
      console.log(err);
      callback(err);
    } else if (res.statusCode !== 200) {
      //empty repositories result in 204:
      if (res.statusCode !== 204) {
        // No idea what went wrong, log to investigate
        console.log(res);
      }
      // Send error that no data was found
      callback("no data");
    } else {
      var info = JSON.parse(body);
      // Get just the collaborators' usernames
      var collabs = info.map(collab => collab.login);
      // Remove yourself from list of collaborators
      if (collabs.indexOf(userName) !== -1) {
        collabs.splice(collabs.indexOf(userName), 1);
      }
      callback(null, collabs);
    }
  });
};

//NOTE: Generate Repos by Owners
async.forEach(namesList, (name, cb) => {
  getUserRepos(name, (err, repos) => {
    reposByOwner[name] = repos;
    cb();
  });
}, (err) => {
  fs.writeFileSync(storedReposByOwnerFile, JSON.stringify(reposByOwner));
});
// Use cached data instead of querying Github again
// reposByOwner = JSON.parse(fs.readFileSync(storedReposByOwnerFile, "utf-8"));
```

```
 86
 87 //NOTE: Count repositories (number of github requests to make)
 88 // var count = 0;
 89 // Object.keys(reposByOwner).forEach(owner => {
 90 //   reposByOwner[owner].forEach(repo => count++)
 91 // });
 92 // console.log(count);
 93
 94
 95 // NOTE: Generate Collabs
 96 async.forEach(Object.keys(reposByOwner), (repoOwner, outerCallback) => {
 97   collabsByPerson[repoOwner] = {};
 98   async.forEach(reposByOwner[repoOwner], (repo, innerCallback) => {
 99
100     getCollabsFromUserRepo(repoOwner, repo, (err, collabs) => {
101       if (err){
102         console.error(err);
103       } else{
104         collabs.forEach((collab) => {
105           if (collabsByPerson[repoOwner][collab]){
106             collabsByPerson[repoOwner][collab] += 1;
107             if(collabsByPerson[collab]){
108               if(collabsByPerson[collab][repoOwner]){
109
110                 delete collabsByPerson[collab][repoOwner];
111               }
112             }
113           } else {
114             collabsByPerson[repoOwner][collab] = 1;
115             if(collabsByPerson[collab]){
116               if(collabsByPerson[collab][repoOwner]){
117
118                 delete collabsByPerson[collab][repoOwner];
119               }
120             }
121           }
122         })
123       }
124       innerCallback();
125     });
126   }, (err) => {
127     outerCallback();
128   });
```

```
}, (err) => {
  console.log("writing");
  fs.writeFileSync(storedCollabsByPersonFile, JSON.stringify(collabsByPerson));
});
// Use cached data instead of querying Github again
// collabsByPerson = JSON.parse(fs.readFileSync(storedCollabsByPersonFile, "utf-8"));

//NOTE: Make collaborator data bi-directional
Object.keys(collabsByPerson).forEach(person => {
  Object.keys(collabsByPerson[person]).forEach(collab => {
    if (collabsByPerson[collab]) {
      if (collabsByPerson[collab][person]){
        collabsByPerson[collab][person] += collabsByPerson[person][collab];
        collabsByPerson[person][collab] = collabsByPerson[collab][person];
      } else {
        collabsByPerson[collab][person] = collabsByPerson[person][collab];
      }
    } else {
      collabsByPerson[collab] = {};
      collabsByPerson[collab][person] = collabsByPerson[person][collab];
    }
  });
});
fs.writeFileSync(storedBDCollabsByPersonFile, JSON.stringify(collabsByPerson));

//NOTE: Find the most collaborations between any two people
var max = 0;
Object.keys(collabsByPerson).forEach(person => {
  Object.keys(collabsByPerson[person]).forEach(collab => {
    if(collabsByPerson[person][collab] > max){
      max = collabsByPerson[person][collab];
    }
  });
});
console.log(max);
```