

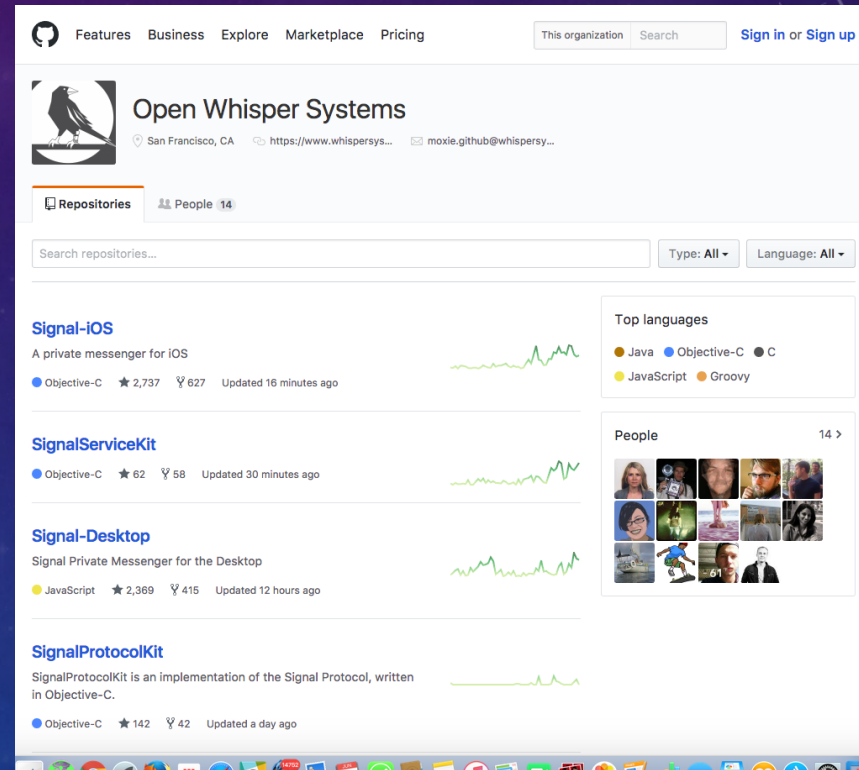
The background features a dark blue gradient with a subtle pattern of white dots. On the left side, there are several overlapping circular elements. A prominent one is a large circle with a scale around its perimeter, marked with numbers from 140 to 260 in increments of 10. Other circles are partially visible, some with dashed lines and arrows, suggesting a technical or data-related theme.

AUTOMATICALLY UPDATE REPOSITORIES FROM GITHUB

AN ARRAY OF LINUX UTILITIES AND TECHNIQUES

OBJECTIVE

- I wanted to automatically and periodically clone and pull GitHub repositories
- Requires a variety of techniques covered in the following slides



The screenshot shows the GitHub organization page for "Open Whisper Systems". The page header includes navigation links for "Features", "Business", "Explore", "Marketplace", and "Pricing", along with a search bar and "Sign in" or "Sign up" options. The organization's profile includes a logo of a crow, the name "Open Whisper Systems", and location information: "San Francisco, CA", "https://www.whispersys...", and "moxie.github@whispersy...". Below the profile, there are tabs for "Repositories" (selected) and "People" (14). A search bar for repositories is present, along with filters for "Type: All" and "Language: All". The main content area displays a list of repositories with their descriptions, languages, star counts, and update times. Each repository entry includes a small green line graph showing activity over time. The "Top languages" section on the right lists Java, Objective-C, C, JavaScript, and Groovy. The "People" section shows a grid of 14 user avatars.

Repository Name	Description	Language	Stars	Forks	Updated
Signal-iOS	A private messenger for iOS	Objective-C	2,737	627	Updated 16 minutes ago
SignalServiceKit		Objective-C	62	58	Updated 30 minutes ago
Signal-Desktop	Signal Private Messenger for the Desktop	JavaScript	2,369	415	Updated 12 hours ago
SignalProtocolKit	SignalProtocolKit is an implementation of the Signal Protocol, written in Objective-C.	Objective-C	142	42	Updated a day ago

GITHUB API

REST API v3

Reference Guides Libraries Webhooks

Overview

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests please contact [support](#).

- [Current Version](#)
- [Schema](#)
- [Parameters](#)
- [Root Endpoint](#)
- [Client Errors](#)
- [HTTP Redirects](#)
- [HTTP Verbs](#)
- [Authentication](#)
- [Hypermedia](#)
- [Pagination](#)
- [Rate Limiting](#)
- [User Agent Required](#)
- [Conditional requests](#)
- [Cross-Origin Resource Sharing](#)
- [JSON-P Callbacks](#)
- [Timezones](#)

Current Version

By default, all requests to `https://api.github.com` receive the v3 version of the REST API. We encourage you to explicitly request this version via the `Accept` header.

```
Accept: application/vnd.github.v3+json
```

For information about GitHub's GraphQL API v4, see the [v4 documentation](#).

Schema

- GitHub has an API
- Available via REST
 - Check out <https://developer.github.com/v3/>
 - Returns JSON formatted information
- Some useful verbs
 - `curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems/repos" # Authenticate and obtain information on the Whisper Systems organization`
 - `curl -u "username:password" --silent "https://api.github.com/users/Anonabox/repos" # Authenticate and obtain information on the Anonabox user`

JQ

- jq is a handy utility to parse JSON formatted information
- Examples
 - `curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems" | jq .public_repos --raw-output` # Obtain the number of repositories
 - `curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems/repos" | jq .[].name --raw-output` # Obtain the list of repositories

```
pi@jcb-github:~/bin $ curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems/repos" | jq .[].name --raw-output
WhisperYAFFS
Signal-Android
maven
Signal-Server
BitHub
whispersystems.org
Signal-Desktop
Signal-iOS
gradle-witness
SignalProtocolKit
libpastelog
Flock
WebSocket-Resources
jobmanager
PushServer
libsignal-protocol-java
ShortcutBadger
gcm-sender-async
curve25519-java
JSQMessagesViewController
CLAServer
libaxolotl-j2me
libsignal-service-java
dropwizard-simpleauth
dropwizard-wavefront
libsignal-protocol-c
SignalServiceKit
libsignal-protocol-javascript
SocketRocket
OpenSSL-Pod
pi@jcb-github:~/bin $
```

GITHUB PAGINATION

```
# Determine the number of repos
REPO_COUNT=`curl -u "username:password" --silent "https://api.github.com/orgs/{ORGANIZATION}" | jq .public_repos --raw-output`
echo This organization has ${REPO_COUNT} repos

# Determine the number of pages
PAGES=$(((${REPO_COUNT}+99)/100)
echo This organization has ${PAGES} pages of repos

# For each page of repos...
for CURRENT_PAGE in `seq 1 ${PAGES}`;
do
    # For each repository owned by that organization
```

- GitHub will return at most 100 results in a response.
- But, some organizations and users have more than 100 repositories
- So, must paginate the results

GIT

- Use git clone to create a repository that isn't yet present
- Use git pull to update a repository that has already been cloned

```
for CURRENT_PAGE in `seq 1 ${PAGES}`;
do
  # For each repository owned by that organization
  for PROJECT in `curl -u "username:password" --silent "https://api.github.com/orgs/${ORGANIZATION}/repos?page=${CURRENT_PAGE}&per_page=100" | jq .[].name --raw-output`
  do
    echo Project: ${PROJECT}

    # if the folder doesn't exist...
    if [ ! -d "${PROJECT}" ]; then
      # Clone the project
      git clone https://github.com/${ORGANIZATION}/${PROJECT}

      FILE_COUNT=`ls ${PROJECT} | wc -l`
      echo Repository has ${FILE_COUNT} files
      if [ ${FILE_COUNT} -eq 0 ]; then
        echo Repository is empty... deleting it
        rm -rf ${PROJECT}
      fi
    else
      # Change into the project directory
      cd ${PROJECT}
      echo Current Working Directory: `pwd`

      # Update the repository
      git pull

      cd ..

      FILE_COUNT=`ls ${PROJECT} | wc -l`
      echo Repository has ${FILE_COUNT} files
      if [ ${FILE_COUNT} -eq 0 ]; then
        echo Repository is empty... deleting it
        rm -rf ${PROJECT}
      fi
    fi
  done
done
```

NILFS

```
# Take a NILFS snapshot
sync
sleep 1
sync
sleep 1
chcp ss `lscp -rn 1 | sed -n 2p | sed -e 's/^[ \t]*//' | cut -f 1 -d ' '`
```

- After cloning/updating all the repositories of interest, create a snapshot of the file system
- A series of commands
 - `chcp ss` # change a file system checkpoint into a snapshot
 - `lscp -rn 1` # get the most recent checkpoint
 - `sed -n 2p` # filter to eliminate the header line
 - `sed -e 's/^[\t]*//'` # convert consecutive spaces to a tab to help with parsing
 - `cut -f 1 -d ' '` # get only the checkpoint number

CRON

- To update the repositories each day, rely on cron
- Create a shell script for use by cron
- Create an link in /etc/crontab.daily to this script

```
[pi@jcb-github:~/bin $ sudo cat grabgit.cron.sh
#!/bin/bash

cd ~/pi/bin
./grabgit.sh >> grabgit.log 2>&1
pi@jcb-github:~/bin $
```

```
[pi@jcb-github:/etc/cron.daily $ ls -l
total 44
-rwxr-xr-x 1 root root 15000 Dec 13 16:38 apt
-rwxr-xr-x 1 root root 314 Jun 15 2014 aptitude
-rwxr-xr-x 1 root root 355 Oct 28 2014 bsdmainutils
-rwxr-xr-x 1 root root 1597 Jun 5 2016 dpkg
lrwxrwxrwx 1 root root 28 Jun 3 20:40 grabgit -> /home/pi/bin/grabgit.cron.sh
-rwxr-xr-x 1 root root 89 Jan 28 2014 logrotate
-rwxr-xr-x 1 root root 1293 Jan 5 2015 man-db
-rwxr-xr-x 1 root root 1110 Jul 25 2016 ntp
-rwxr-xr-x 1 root root 249 Feb 24 16:49 passwd
pi@jcb-github:/etc/cron.daily $
```


LOGROTATE

```
pi@jcb-github:/etc/logrotate.d $ cat grabgit
/home/pi/bin/*log {
    rotate 30
    daily
    compress
    delaycompress
    missingok
    notifempty
}
pi@jcb-github:/etc/logrotate.d $
```

- To keep the log file from growing indefinitely, rely on logrotate