Archived:Accessing public Qik streams using PySymbian

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

All PySymbian articles have been archived. PySymbian is no longer maintained by Nokia and is not guaranteed to work on more recent Symbian devices. It is not possible to submit apps to Nokia Store.

This PySymbian code example shows you how to use Qik service to stream video live from your cell phone to the web.

Introduction

Qik is a new and innovative service that allows you stream video live from your cell phone to the web. These videos can be shared with your friends and they are available for viewing and download. Qik has a straight relationship with other on-line services, like twitter, Youtube, Facebook and wordpress. In this article an API would be presented for browsing public streams for some Qik account and an S60 application for displaying these videos, both written in Python. Even though it is not complete, this API is an excellent start point for new services and programs based on Qik.

Qik API

Current Qik API is based on JSON-RPC and REST (only available for the Qik-ly API and not used in this article). JSON-RPC Qik API uses HTTP for transfer data between client and Qik engine. For instance, the public streams of Qik user marcelobarrosalmeida can be retrieved with the following HTTP data:

```
POST /api/jsonrpc?apikey=YOUR_API_KEY HTTP/1.0
Content-Length: 80
Host: engine.qik.com
Content-Type: application/json; charset=UTF-8

{"method": "qik.stream.public_user_streams","params": ["marcelobarrosalmeida"]}
```

The response is below:

```
HTTP/1.1 200 OK
Server: nginx/0.7.59
Date: Fri, 07 Aug 2009 14:31:06 GMT
Content-Type: text/json
Connection: close
Content-Length: 1706
X-Qik-Origin: 229

[[{"url": "http://qik.com/video/2363785", "live": false, "user id": 365150, "small
thumbnail url":
"http://media.qik.com/media.thumbnails.128/2a7c74d9862b4f2caa9ea6d7296b9225.jpg",
"title": "Untitled", "duration": 54, "created at": "2009-07-31 13:23:30", "views": 7,
"id": 2363785}, {"url": "http://qik.com/video/2366002", "live": false, "user id":
365150, "small thumbnail url":
"http://media.qik.com/media.thumbnails.128/93befe9eb93c464aadbd92d8ea16ac77.jpg",
"title": "Pint da Guinness", "duration": 30, "created at": "2009-07-31 18:15:29",
"views": 18, "id": 2366002}, {"url": "http://qik.com/video/2363998", "live": false,
"user id": 365150, "small thumbnail url":
"http://media.qik.com/media.thumbnails.128/566d3f87915b46479ad72eff0c0a21ca.jpg",
"title": "Trafalgar square", "duration": 41, "created at": "2009-07-31 14:01:39",
"views": 11, "id": 2363998}, {"url": "http://qik.com/video/2357498", "live": false,
"user id": 365150, "small thumbnail url":
"http://media.qik.com/media.thumbnails.128/cd4e173774884249968e757d364fe34d.jpg",
"title": "Untitled", "duration": 55, "created at": "2009-07-30 21:06:58", "views": 5,
"id": 2357498}, {"url": "http://qik.com/video/2356796", "live": false, "user id":
365150, "small thumbnail url":
"http://media.qik.com/media.thumbnails.128/2053be143c2c4e708b9bcd51d62a7359.jpg",
"title": "Untitled", "duration": 97, "created at": "2009-07-30 19:54:10", "views": 7,
"id": 2356796}, {"url": "http://qik.com/video/2363619", "live": false, "user id":
365150, "small thumbnail url":
"http://media.qik.com/media.thumbnails.128/d230dcf8cf3e4f9fb6980f512b66265c.jpg",
"title": "Untitled", "duration": 61, "created at": "2009-07-31 12:54:22", "views": 5,
"id": 2363619}]]
```

Responses uses JSON as well and it will vary depending on required command. Check Qik API for details and this article for a good introduction about object serialization and JSON.

Implementing Qik API for Symbian devices

Using simplejson and urllib it is possible to implement an API for accessing Qik services. Since it is necessary to include special HTTP headers, instead using urllib.urlopen, we need to create an URL opener (using urllib.URLopener), adding required headers to it:

```
urlopener = urllib.URLopener()
urlopener.addheaders = [('Host','engine.qik.com'), ('Content-Type','application/json;
charset=UTF-8')]
```

The desired remote procedure call can be reached using the `open()` method:

```
data = json.dumps(data)
url = 'http://engine.qik.com/api/jsonrpc?apikey=YOUR_API_KEY'
f = urlopener.open(url,data)
```

Finally, using the file descriptor returned by `open()`, it is possible to get the response and decode it:

```
data = json.loads(f.read())[0]
```

The API code is below and it can be run in PySymbian 1.4.x or 1.9.x and probably in any version of Python for PC with urllib. But you will need an API key for using it. Request it at Qik API page.

```
# -*- coding: utf-8 -*-
# Marcelo Barros de Almeida
# marcelobarrosalmeida (at) gmail.com
# License: GPL3

import simplejson as json
import urllib

class QikApi(object):
    """ Simple class for Qik videos proxy support
    """
    def __init__(self, api_key, qik_usr):
        """ Create a new Qik API instance with given API key and user name
```

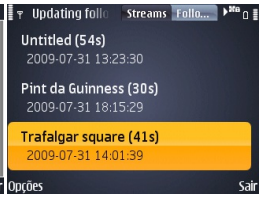

(C) Copyright Nokia 2014. All rights reserved.

```
        """
        self.qik_url = 'http://engine.qik.com/api/jsonrpc?apikey=' + api_key
        self.qik_usr = qik_usr
        self.qik_id = -1

    def __urlopener(self):
        """ Return an urlopener with Qik required headers already set
        """
        urlopener = urllib.URLopener()
        urlopener.addheaders = [('Host','engine.qik.com'),
                                ('Content-Type','application/json; charset=UTF-8')]
        return urlopener

    def __open(self,url,params=""):
        """ Open a given URL using GET or POST and Qik headers
        """
        if params:
            f = self.__urlopener().open(url,params) #post
        else:
            f = self.__urlopener().open(url) #get

        return f

    def __qik_request(self,data):
        """ Qik request. Encode data in json format, do the request and
            decode json response
        """
        data = json.dumps(data)
        f = self.__open(self.qik_url,data)
        res = json.loads(f.read())[0]
        return res

    def __check_id(self,qik_id):
        """ Check if user ID was retrieved or not. If not, download it
        """
        if qik_id == -1:
            if self.qik_id == -1:
                self.qik_id = self.get_user_public_profile()[u'id']
            qik_id = self.qik_id
        return qik_id

    def get_public_user_streams(self,usr=''):
        """ Return all public stream for a given user
            (or for the current user, if it not provided)
        """
        if not usr:
            usr = self.qik_usr
        data = {'method': 'qik.stream.public_user_streams','params': [usr]}
        return self.__qik_request(data)

    def get_user_public_profile(self,usr=''):
        """ Return public profile for a given user
            (or for the current user, if it not provided)
        """
        if not usr:
            usr = self.qik_usr
        data = {'method': 'qik.user.public_profile','params': [usr]}
        return self.__qik_request(data)

    def get_user_public_detailed_profile(self,usr=''):
        """ Return detailed public profile for a given user
            (or for the current user, if it not provided)
        """
        if not usr:
            usr = self.qik_usr
        data = {'method': 'qik.user.public_detailed_profile','params': [usr]}
        return self.__qik_request(data)

    def get_user_followers(self,qik_id=-1):
        """ Return the list of followers for a given user
            (or for the current user, if it not provided)
        """
        qik_id = self.__check_id(qik_id)
        data = {'method': 'qik.user.followers','params': [qik_id]}
        return self.__qik_request(data)

    def get_user_following(self,qik_id=-1):
        """ Return the list of following for a given user
            (or for the current user, if it not provided)
        """
        qik_id = self.__check_id(qik_id)
        data = {'method': 'qik.user.following','params': [qik_id]}
        return self.__qik_request(data)

    def get_public_stream_info(self,vid_id):
        """ Get detailed information about some public video
        """
        data = {'method': 'qik.stream.public_info','params': [vid_id]}
        return self.__qik_request(data)
```

Demo application and source code

Using the Archived:PySymbian basic user interface app framework it is simple to create a demo application for showing user streams and their Qik following/followers, each one in a different tab. For playing the videos (Flash-lite capable device is required) a local HTML with links to the embedded video is created (see QIK_TEMPLATE variable) and the native web browser is called to show it.

You can see this application in action in the following video. Check the program repository for newer versions and updates. For running it, just copy all files (window.py, qikapi.py, qikview.py, simplejson.py) to your memory card (into e:\Python) and use the Python interpreter to execute qikview.py. A zip package is available here.



(C) Copyright Nokia 2014. All rights reserved.

```
# -*- coding: utf-8 -*-
# Marcelo Barros de Almeida
# marcelobarrosalmeida (at) gmail.com
# License: GPL3

import sys
sys.path.append('e:\\Python')

import window
from appuifw import *
from qikapi import QikApi
import time

API_KEY = 'YOUR_API_KEY'

QIK_TEMPLATE = u"""
<html><head><meta http-equiv="Content-Type" content="application/vnd.wap.xhtml+xml;
charset=utf-8" /><title>__TITLE__</title></head><body>
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0"
width="220" height="185" id="player" align="middle">
<param name="movie" value="http://qik.com/swfs/qik_player_lite.swf?
file=http://qik.com/flv/__FILENAME__.flv&amp;thumbnail=http://qik.com/redir/__FILENAME__.jpg&amp;size=false&amp;aplay=true&amp;autorew=false&amp;layout=small&amp;title=__TITLE__"/>
<param name="menu" value="false" />
<param name="quality" value="high" />
<param name="bgcolor" value="#999999" />
<embed src="http://qik.com/swfs/qik_player_lite.swf?
file=http://qik.com/flv/__FILENAME__.flv&amp;thumbnail=http://qik.com/redir/__FILENAME__.jpg&amp;size=false&amp;aplay=true&amp;autorew=false&amp;layout=small&amp;title=__TITLE__"
menu="false" quality="high" bgcolor="#999999" width="220" height="185" name="player"
align="middle" allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
pluginspage="http://get.adobe.com/flashplayer/otherversions/"/>
</object></body></html>
""".encode('utf-8')

class QikView(window.Application):
    def __init__(self):
        self.qik_usr = u""
        self.qik_api = None
        self.data = {'profile':[], 'streams':[], 'followers':[], 'following':[]}
        # menus
        streams_menu = [(u"Show stream",self.show_video)]
        common_menu = [(u"Update",self.update),
                       (u"Setup",self.setup),
                       (u"About",self.about)]
        # bodies
        self.streams = Listbox([(u"Please, setup and update",u"")],self.show_video)
        self.following = Listbox([(u"Please, setup and update",u"")])
        self.followers = Listbox([(u"Please, setup and update",u"")])

        window.Application.__init__(self,
                             u"Qik View",
                             [(u"Streams",self.streams,streams_menu),
                              (u"Following",self.following,[]),
                              (u"Followers",self.followers,[])],
                             common_menu)

    def update(self):
        if not self.qik_usr or not self.qik_api:
            note(u"Please, setup the Qik user",u"error")
        else:
            self.lock_ui()
            try:
                self.set_title(u"Updating profile...")
                self.data['profile'] = self.qik_api.get_user_public_profile()
                self.set_title(u"Updating streams...")
                self.data['streams'] = self.qik_api.get_public_user_streams()
                self.set_title(u"Updating followers...")
                self.data['followers'] = self.qik_api.get_user_followers()
                self.set_title(u"Updating following...")
                self.data['following'] = self.qik_api.get_user_following()
            except:
                note(u"Network error. Please, try again","error")
            else:
                self.update_bodies()
            self.set_title(u"Qik View")
            self.unlock_ui()
            self.refresh()

    def update_bodies(self):
        streams = []
        followers = []
        following = []

        for s in self.data['streams']:
            h1 = s['title'] + (u" (%ds)" % s['duration'])
            h2 = s['created_at']
            streams.append((h1,h2))

        for f in self.data['followers']:
            followers.append((f[u'username'],f[u'full_name']))

        for f in self.data['following']:
            following.append((f[u'username'],f[u'full_name']))

        if streams:
            self.streams.set_list(streams)
        else:
            self.streams.set_list([(u"No streams available",u"")])

        if followers:
            self.followers.set_list(followers)
        else:
            self.followers.set_list([(u"No followers available",u"")])

        if following:
            self.following.set_list(following)
        else:
            self.following.set_list([(u"No following available",u"")])

    def setup(self):
```


(C) Copyright Nokia 2014. All rights reserved.

```
        usr = query(u"Qik user:","text",self.qik_usr)
        if usr is not None:
            self.qik_usr = usr
            self.qik_api = QikApi(API_KEY,self.qik_usr)

    def show_video(self):
        if self.data['streams']:
            # retrieve information about video
            idx = self.streams.current()
            if 'stream_info' not in self.data['streams'][idx]:
                vid = self.data['streams'][idx][u'id']
                self.lock_ui(u"Downloading stream info...")
                try:
                    self.data['streams'][idx]['stream_info'] =
self.qik_api.get_public_stream_info(vid)
                except:
                    note(u"Network error. Please, try again","error")
                    ret = True
                else:
                    ret = False
                self.set_title(u"Qik View")
                self.unlock_ui()
                self.refresh()
                if ret:
                    return
            tit = self.data['streams'][idx]['stream_info'][u'title'].encode('utf-8')
            fn = self.data['streams'][idx]['stream_info'][u'filename'].encode('utf-8')
            html_code = QIK_TEMPLATE.replace('__FILENAME__',fn).replace('__TITLE__',tit)
            html_file = "html_" + time.strftime("%Y%m%d_%H%M%S", time.localtime()) +
".html"
            try:
                fp = open(html_file,"wt")
                fp.write(html_code)
                fp.close()
            except:
                note(u"Could not create HTML file","error")
                return

            viewer = Content_handler(self.refresh)
            try:
                viewer.open(html_file)
            except:
                note(u"Can not open browser","error")

    def about(self):
        note(u"Qik API for PySymbian\nby marcelobarrosalmeida@gmail.com",u"info")

if __name__ == "__main__":
    app = QikView()
    app.run()
```


(C) Copyright Nokia 2014. All rights reserved.