Python flask.request.stream() Examples

The following are code examples for showing how to use *flask.request.stream()*. They are from open source Python projects. You can vote up the examples you like or vote down the ones you don't like.

Example 1

Project: *zwift-offline* Author: *zoffline* File: *zwift_offline.py* GNU General Public License v3.0 6 vo

```
def api_profiles_activities(player_id):
    if request.method == 'POST':
        if not request.stream:
            return '', 400
        activity = activity_pb2.Activity()
        activity.ParseFromString(request.stream.read())
        activity.id = get_id('activity')
        insert_protobuf_into_db('activity', activity)
        return '{"id": %ld}' % activity.id, 200

    # request.method == 'GET'
    activities = activity_pb2.Activities()
    cur = g.db.cursor()
    cur.execute("SELECT * FROM activity WHERE player_id = ?", (str(player_id),))
    for row in cur.fetchall():
        activity = activities.activities.add()
        row_to_protobuf(row, activity, exclude_fields=['fit'])

    return activities.SerializeToString(), 200


# With 64 bit ids Zwift can pass negative numbers due to overflow, which the flask
# converter does not handle so it's a string argument
```

Example 2

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 6 vo

```
def ngfileupload_uploadfactory(content_length=None, content_type=None,
                               uploaded_file=None):
    """Get default put factory.

    If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

    :param content_length: The content length. (Default: ``None``)
    :param content_type: The HTTP Content-Type. (Default: ``None``)
    :param uploaded_file: The upload request. (Default: ``None``)
    :param file_tags_header: The file tags. (Default: ``None``)
    :returns: A tuple containing stream, content length, and empty header.
    """
    if not content_type.startswith('multipart/form-data'):
        abort(422)

    return uploaded_file.stream, content_length, None, parse_header_tags()


#
# Object retrieval
#
```

Example 3

Project: *nametableswapper* Author: *graphicore* File: *main.py* Apache License 2.0 6 vo

```
def changenames():
    result = BytesIO()
    zipf = zipfile.ZipFile(result,  "w")
    i=0
    for desc, font in _unpack(request.stream):
        i += 1
        print('#',i,'got oldname', font['name'].getDebugName(6))
        changefont(desc, font)
        filename = desc['filename']
        print('changed', filename)

        # write the font file to the zip
        fontIO = BytesIO()
        font.save(fontIO)
        fontData = fontIO.getvalue()
        zipf.writestr(filename, fontData)

    zipf.close()
    data = result.getvalue()
    response = make_response(data)
    response.headers['Content-Type'] = 'application/octet-stream'
    response.headers['Content-Disposition'] = 'attachment; filename=fonts-with-cha
    return response
```

Example 4

Project: *wechat_automated_jump_game* Author: *microdog* File: *server.py* Apache License 2.0 6 vo

```
def handler():
    try:
        press_time = solver.solve_from_stream(request.stream)
    except SolverInputException as e:
        abort(400, e.message)
        return

    if press_time is None:
        abort(400, 'jump target not found in image')

    try:
        jitter = request.args.get('jitter', None, float)
    except ValueError:
        abort(400, 'invalid jitter value')
        return

    if jitter is not None:
        app.logger.debug('Applying jitter: %s', jitter)
        press_time = int(
            float(press_time) * random.uniform(1 - jitter, 1 + jitter))
        app.logger.debug('Actual press time: %s', press_time)

    return str(press_time)
```

Example 5

Project: *zwift-offline* Author: *zoffline* File: *zwift_offline.py* GNU General Public License v3.0 5 vo

```
def api_profiles_id(player_id):
    if not request.stream:
        return '', 400
    with open('%s/profile.bin' % STORAGE_DIR, 'wb') as f:
        f.write(request.stream.read())
    return '', 204
```

Example 6

Project: *zwift-offline* Author: *zoffline* File: *zwift_offline.py* GNU General Public License v3.0 5 vo

```
def api_profiles_goals(player_id):
    if request.method == 'POST':
        if not request.stream:
            return '', 400
        goal = goal_pb2.Goal()
        goal.ParseFromString(request.stream.read())
        goal.id = get_id('goal')
        now = datetime.datetime.now()
        goal.created_on = unix_time_millis(now)
        set_goal_end_date(goal, now)
        fill_in_goal_progress(goal, player_id)
        insert_protobuf_into_db('goal', goal)

        return goal.SerializeToString(), 200

    # request.method == 'GET'
    goals = goal_pb2.Goals()
    cur = g.db.cursor()
    cur.execute("SELECT * FROM goal WHERE player_id = ?", (str(player_id),))
    rows = cur.fetchall()
    for row in rows:
        goal = goals.goals.add()
        row_to_protobuf(row, goal)
        end_dt = datetime.datetime.fromtimestamp(goal.period_end_date / 1000)
        now = datetime.datetime.now()
        if end_dt < now:
            set_goal_end_date(goal, now)
            update_protobuf_in_db('goal', goal, goal.id)
        fill_in_goal_progress(goal, player_id)

    return goals.SerializeToString(), 200
```

Example 7

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 5 vo

```
def default_partfactory(part_number=None, content_length=None,
                        content_type=None, content_md5=None):
    """Get default part factory.

    :param part_number: The part number. (Default: ``None``)
    :param content_length: The content length. (Default: ``None``)
    :param content_type: The HTTP Content-Type. (Default: ``None``)
    :param content_md5: The content MD5. (Default: ``None``)
    :returns: The content length, the part number, the stream, the content
        type, MD5 of the content.
    """
    return content_length, part_number, request.stream, content_type, \
        content_md5, None
```

Example 8

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 5 vo

```
def stream_uploadfactory(content_md5=None, content_length=None,
                         content_type=None):
    """Get default put factory.

    If Content-Type is ``'multipart/form-data'`` then the stream is aborted.

    :param content_md5: The content MD5. (Default: ``None``)
    :param content_length: The content length. (Default: ``None``)
    :param content_type: The HTTP Content-Type. (Default: ``None``)
    :returns: The stream, content length, MD5 of the content.
    """
    if content_type.startswith('multipart/form-data'):
        abort(422)

    return request.stream, content_length, content_md5, parse_header_tags()
```

Example 9

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 5 vo

```
def ngfileupload_partfactory(part_number=None, content_length=None,
                             uploaded_file=None):
    """Part factory for ng-file-upload.

    :param part_number: The part number. (Default: ``None``)
    :param content_length: The content length. (Default: ``None``)
    :param uploaded_file: The upload request. (Default: ``None``)
    :returns: The content length, part number, stream, HTTP Content-Type
        header.
    """
    return content_length, part_number, uploaded_file.stream, \
        uploaded_file.headers.get('Content-Type'), None, None
```

Example 10

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 5 vo

```
def ensure_input_stream_is_not_exhausted(f):
    """Make sure that the input stream has not been read already."""
    @wraps(f)
    def decorate(*args, **kwargs):
        if request.content_length and request.stream.is_exhausted:
            raise ExhaustedStreamError()
        return f(*args, **kwargs)
    return decorate


#
# Permission checking
#
```

Example 11

Project: *invenio-files-rest* Author: *inveniosoftware* File: *views.py* MIT License 5 vo

```
def multipart_uploadpart(self, multipart):
        """Upload a part.

        :param multipart: A :class:`invenio_files_rest.models.MultipartObject`
            instance.
        :returns: A Flask response.
        """
        content_length, part_number, stream, content_type, content_md5, tags =\
            current_files_rest.multipart_partfactory()

        if content_length:
            ck = multipart.last_part_size if \
                part_number == multipart.last_part_number \
                else multipart.chunk_size

            if ck != content_length:
                raise MultipartInvalidChunkSize()

        # Create part
        try:
            p = Part.get_or_create(multipart, part_number)
            p.set_contents(stream)
            db.session.commit()
        except Exception:
            # We remove the Part since incomplete data may have been written to
            # disk (e.g. client closed connection etc.) so it must be
            # reuploaded.
            db.session.rollback()
            Part.delete(multipart, part_number)
            raise
        return self.make_response(
            data=p,
            context={
                'class': Part,
            },
            etag=p.checksum
        )
```

Example 12

Project: *curldump* Author: *ledeuns* File: *curldump.py* ISC License 5 vo

```
def postfile():
    rv = []
    for file in request.files.itervalues():
        h = savefile(file.filename, file.stream)
        rv.append(BASE_URL+h+"\n")

    return Response("".join(rv), mimetype="text/uri-list")
```

Example 13

Project: *curldump* Author: *ledeuns* File: *curldump.py* ISC License 5 vo

```
def putfile(filename):
    h = savefile(filename, request.stream)
    return Response(BASE_URL+h+"\n", mimetype="text/uri-list")
```

Example 14

5 vo

Project: *curldump* Author: *ledeuns* File: *curldump.py* ISC License

```
def putstream():
    filename = str(uuid.uuid4())
    h = savefile(filename, request.stream)
    return Response(BASE_URL+h+"\n", mimetype="text/uri-list")
```

Example 15

Project: *nametableswapper* Author: *graphicore* File: *main.py* Apache License 2.0 5 vo

```
def _unpack(stream):
    # L = unsignedlong 4 bytes
    while True:
        head = stream.read(8)
        if not head:
            break
        jsonlen, fontlen = struct.unpack('II', head)
        desc = json.loads(stream.read(jsonlen).decode('utf-8'))
        font = TTFont(BytesIO(stream.read(fontlen)))
        yield (desc, font)
```

Example 16

Project: *c* Author: *rettier* File: *main.py* MIT License 5 vo

```
def post():
    return storage_backend.put(get_key(), request.stream)
```

Example 17

Project: *c* Author: *rettier* File: *main.py* MIT License 5 vo

```
def get():
    key = get_key()
    result = process_custom_command(key)

    if result is False:
        if storage_backend.has_key(key):
            result = storage_backend.get(key)
        else:
            result = empty_gzip

    if isinstance(result, str):
        result = gzip.compress(result.encode("utf-8"))

    return Response(result, content_type="application/octet-stream")
```

Example 18

Project: *zwift-offline* Author: *zoffline* File: *zwift_offline.py* GNU General Public License v3.0 4 vo

```
def api_profiles_activities_id(player_id, activity_id):
    if not request.stream:
        return '', 400
    activity_id = int(activity_id) & 0xffffffffffffffff
    activity = activity_pb2.Activity()
    activity.ParseFromString(request.stream.read())
    update_protobuf_in_db('activity', activity, activity_id)
```

```
    response = '{"id":%s}' % activity_id
    if request.args.get('upload-to-strava') != 'true':
        return response, 200
    try:
        from stravalib.client import Client
    except ImportError:
        logger.warn("stravalib is not installed. Skipping Strava upload attempt.")
        return response, 200
    strava = Client()
    try:
        with open('%s/strava_token.txt' % STORAGE_DIR, 'r') as f:
            client_id = f.readline().rstrip('\n')
            client_secret = f.readline().rstrip('\n')
            strava.access_token = f.readline().rstrip('\n')
            refresh_token = f.readline().rstrip('\n')
            expires_at = f.readline().rstrip('\n')
    except:
        logger.warn("Failed to read %s/strava_token.txt. Skipping Strava upload at
        return response, 200
    try:
        if time.time() > int(expires_at):
            refresh_response = strava.refresh_access_token(client_id=client_id, cl
                                                           refresh_token=refresh_t
            with open('%s/strava_token.txt' % STORAGE_DIR, 'w') as f:
                f.write(client_id + '\n');
                f.write(client_secret + '\n');
                f.write(refresh_response['access_token'] + '\n');
                f.write(refresh_response['refresh_token'] + '\n');
                f.write(str(refresh_response['expires_at']) + '\n');
    except:
        logger.warn("Failed to refresh token. Skipping Strava upload attempt.")
        return response, 200
    try:
        # See if there's internet to upload to Strava
        strava.upload_activity(BytesIO(activity.fit), data_type='fit', name=activi
        # XXX: assume the upload succeeds on strava's end. not checking on it.
    except:
        logger.warn("Strava upload failed. No internet?")
    return response, 200
```

Example 19

Project: *zwift-offline* Author: *zoffline* File: *zwift_offline.py* GNU General Public License v3.0 4 vo

```
def handle_segment_results(request):
    if request.method == 'POST':
        if not request.stream:
            return '', 400
        result = segment_result_pb2.SegmentResult()
        result.ParseFromString(request.stream.read())
        result.id = get_id('segment_result')
        result.world_time = world_time()
        result.finish_time_str = datetime.datetime.now().strftime("%Y-%m-%dT%H:%M:
        result.f20 = 0
        insert_protobuf_into_db('segment_result', result)
        return '{"id": %ld}' % result.id, 200

    # request.method == GET
#    world_id = int(request.args.get('world_id'))
    player_id = request.args.get('player_id')
#    full = request.args.get('full') == 'true'
```

```
    # Require segment_id
    if not request.args.get('segment_id'):
        return '', 422
    segment_id = int(request.args.get('segment_id')) & 0xffffffffffffffff
    only_best = request.args.get('only-best') == 'true'
    from_date = request.args.get('from')
    to_date = request.args.get('to')

    results = segment_result_pb2.SegmentResults()
    results.world_id = 1
    results.segment_id = segment_id

    cur = g.db.cursor()
    where_stmt = "WHERE segment_id = ?"
    where_args = [str(segment_id)]
    if player_id:
        where_stmt += " AND player_id = ?"
        where_args.append(player_id)
    if from_date:
        where_stmt += " AND strftime('%s', finish_time_str) > strftime('%s', ?)"
        where_args.append(from_date)
    if to_date:
        where_stmt += " AND strftime('%s', finish_time_str) < strftime('%s', ?)"
        where_args.append(to_date)
    if only_best:
        where_stmt += " ORDER BY elapsed_ms LIMIT 1"
    cur.execute("SELECT * FROM segment_result %s" % where_stmt, where_args)
    for row in cur.fetchall():
        result = results.segment_results.add()
        row_to_protobuf(row, result, ['f3', 'f4', 'segment_id', 'event_subgroup_id

    return results.SerializeToString(), 200
```

Example 20

Project: *duckpond* Author: *alexmilowski* File: *api.py* Apache License 2.0 4 vo

```
def content_item_resource(id,resource):
   if request.method == 'GET':
      wrap = request.args.get('wrap')
      status_code,data,contentType = model.getContentResource(id,resource);
      if status_code==200:
         if contentType.startswith("text/html") and wrap is not None:
            blob = io.BytesIO()
            for chunk in data:
               blob.write(chunk)
            content = blob.getvalue().decode("utf-8").strip()
            if not content.startswith('<!DOCTYPE'):
               editorConfig = app.config.get('EDITOR_CONFIG')
               header = ''
               bodyStart = ''
               bodyEnd = ''
               if editorConfig is not None and wrap=='preview':
                  wheader = editorConfig.get('wrap-header')
                  pheader = editorConfig.get('preview-wrap-header')
                  if pheader is not None:
                     header = pheader
                  elif wheader is not None:
                     header = wheader
                  wbody = editorConfig.get('wrap-body')
                  pbody = editorConfig.get('preview-body-main')
                  if pbody is not None:
```

```
                    bodyStart = pbody[0]
                    bodyEnd = pbody[1]
                elif wbody is not None:
                    bodyStart = wbody[0]
                    bodyEnd = wbody[1]
            elif editorConfig is not None and wrap=='formatted':
                wheader = editorConfig.get('wrap-header')
                if wheader is not None:
                    header = wheader
                wbody = editorConfig.get('wrap-body')
                if wbody is not None:
                    bodyStart = wbody[0]
                    bodyEnd = wbody[1]
            content = """
<!DOCTYPE html>
<html>
<head><title>""" + resource + '</title>' + header + """
</head>
<body>
""" + bodyStart + content + bodyEnd + '</body></html>'
            return Response(stream_with_context(content),content_type = contentTyp
         else:
            return Response(stream_with_context(data),content_type = contentType)
      else:
         abort(status_code)
   if request.method == 'PUT':
      status_code,data,contentType = model.updateContentResource(id,resource,reque
      if status_code==200 or status_code==201:
         return Response(stream_with_context(data),status=status_code,content_type
      else:
         return Response(status=status)

   if request.method == 'DELETE':
      status = model.deleteContentResource(id,resource)
      return Response(status=status)
```