

Python `flask.request.content_length()` Examples

The following are code examples for showing how to use `flask.request.content_length()`. 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: *padex* Author: *dszakallas* File: *padex.py* GNU General Public License v2.0

6 vc

```
def decrypt():
    if request.content_length != 44:
        abort(400)

    data = b64decode(request.get_data())
    aes = AES.new(key, AES.MODE_CBC, IV=iv)
    mess = aes.decrypt(data)
    padsize = mess[-1]

    if padsize < 1 or padsize > 16:
        abort(403)

    for x in mess[-padsize:-1]:
        if x != padsize:
            abort(403)

    return 'OK', 200
```

Example 2

Project: *AlforEarth-API-Development* Author: *microsoft* File: *ai4e_service.py* MIT License

6 vc

```
def before_request(self):
    # Don't accept a request if SIGTERM has been called on this instance.
    if (self.is_terminating):
        print('Process is being terminated. Request has been denied.')
        abort(503, {'message': 'Service is busy, please try again later.})

    if request.path in self.func_properties:
        if (self.func_request_counts[request.path] + 1 > self.func_properties[
            print('Service is busy. Request has been denied.')
            abort(503, {'message': 'Service is busy, please try again later.})

        if (self.func_properties[request.path][CONTENT_TYPE_KEY_NAME] and not
            print('Invalid content type. Request has been denied.')
            abort(401, {'message': 'Content-type must be ' + self.func_propert

        if (self.func_properties[request.path][CONTENT_MAX_KEY_NAME] and requ
            print('Request is too large. Request has been denied.')
            abort(413, {'message': 'Request content too large (' + str(request
```

Example 3

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

6 vc

```
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 4

Project: *sevilla* Author: *federicotdn* File: *frontend.py* GNU General Public License v3.0

6 vc

```
def upsert_note(note_id):
    if not NotesService.id_is_valid(note_id):
        abort(400)

    if (request.content_length or 0) > current_app.config["MAX_NOTE_LENGTH"]:
        abort(413)

    timestamp_millis = args_int("timestamp")
    seconds = timestamp_millis // 1000
    millis = timestamp_millis % 1000
    timestamp = datetime.utcnow().timestamp() + timedelta(millisecons=millis)
    contents = request.get_data(as_text=True)

    try:
        NotesService.upsert_note(note_id, contents, timestamp)
    except ModelException:
        current_app.logger.exception("Error storing note:")
        abort(500)

    current_app.logger.info("Note ID {} created/updated.".format(note_id))

    return {"id": note_id, "timestamp": timestamp_millis}
```

Example 5

Project: *polyswarmd* Author: *polyswarm* File: *__init__.py* MIT License

6 vc

```
def before_request():
    g.user = User()

    config = app.config['POLYSWARM']

    if not config.require_api_key:
        return

    # Ignore prefix if present
    try:
        api_key = request.headers.get('Authorization').split()[-1]
```

```

except Exception:
    # exception == unauthenticated
    return whitelist_check(request.path)

if api_key:
    g.user = User.from_api_key(api_key)
    if not g.user:
        return whitelist_check(request.path)

size = request.content_length
if size is not None and size > g.user.max_artifact_size * 256:
    return failure('Payload too large', 413)

```

Example 6

Project: *flask-request-logger* Author: *BbsonLin* File: *request_logger.py* MIT License

5 vc

```

def _logging_req_resp(self, response):
    req_log = RequestLog(request.method, request.url, request.content_length,
        self.db.add(req_log)
    self.db.commit()
    res_log = ResponseLog(response.status_code, response.content_length, req_
        self.db.add(res_log)
    self.db.commit()

    return response

```

Example 7

Project: *SempoBlockchain* Author: *teamsempo* File: *__init__.py* GNU General Public License v3.0

5 vc

```

def register_extensions(app):
    db.init_app(app)

    basic_auth.init_app(app)

    @app.before_request
    def enable_form_raw_cache():
        # Workaround to allow unparsed request body to be read from cache
        # This is required to validate a signature on webhooks
        # This MUST go before Sentry integration as sentry triggers form parsing
        if not config.IS_TEST and (
            request.path.startswith('/api/slack/') or request.path.startswith(
                if request.content_length > 1024 * 1024: # 1mb
                    # Payload too large
                    return make_response(jsonify({'message': 'Payload too large'})), 4
            request.get_data(parse_form_data=False, cache=True)

    if not config.IS_TEST:
        sentry.init_app(app, dsn=app.config['SENTRY_SERVER_DSN'])
        # limiter.init_app(app)

    CORS(app, resources={r"/api/*": {"origins": "*"}})

    celery_app.conf.update(app.config)

    print('celery joined on {} at {}'.format(
        app.config['REDIS_URL'], datetime.utcnow()))

```

Example 8

```
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 9

```
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 10

```
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 11

```
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 12

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

5 vc

```
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 13

Project: *curl2share* Author: *cuongnv23* File: *utils.py* MIT License

5 vc

```
def validate_filesize(size):
    """
    Validate if file size is too large or empty
    size: size of file
```

```

'''
if size > config.max_file_size * 1024 * 1024:
    abort(413)
if not request.content_length or not size:
    logger.error('Request {} {} with empty file.'.format(request.method,
                                                           request.path))
    abort(411)

```

Example 14

Project: *topology* Author: *opensciencegrid* File: *webhook_app.py* Apache License 2.0

5 vc

```

def validate_request_signature(request):
    if request.content_length > _max_payload_size:
        app.logger.error("Refusing to read overly-large payload of size %s"
                        % request.content_length)
        return False
    payload_body = request.get_data()
    x_hub_signature = request.headers.get('X-Hub-Signature')
    ret = validate_webhook_signature(payload_body, x_hub_signature)
    if ret or ret is None:
        return True # OK, signature match or secret key not configured
    else:
        app.logger.error("Payload signature did not match for secret key")
        return False

```

Example 15

Project: *soja-box* Author: *itaa* File: *soja_upload_API.py* MIT License

5 vc

```

def upload_zip():
    # param_dict: 存放请求参数
    param_dict = dict.fromkeys(upload_request_param_list, None)
    start_time = time.time()

    file_size = request.content_length

    try:
        # 以表单形式发送数据
        parameter = request.form
        for param in upload_request_param_list:
            param_dict[param] = parameter.get(param)
    except:
        raise Exception

```

Example 16

Project: *prometheus-flask* Author: *thangbn* File: *__init__.py* Apache License 2.0

5 vc

```

def monitor(app):
    def before_request():
        flask.g.start_time = time.time()
        http_concurrent_request_count.inc()
        content_length = request.content_length
        if (content_length):
            http_request_size_bytes.labels(request.method, request.path).observe(c

    def after_request(response):
        request_latency = time.time() - flask.g.start_time
        http_request_latency_ms.labels(request.method, request.path).observe(reque

```

```

    http_concurrent_request_count.dec()

    http_request_count.labels(request.method, request.path, response.status_code)
    http_response_size_bytes.labels(request.method, request.path).observe(response.content_length)
    return response

monitor_host_metrics()

http_request_latency_ms = Histogram('http_request_latency_ms', 'HTTP Request Latency',
                                     ['method', 'endpoint'])

http_request_size_bytes = Histogram('http_request_size_bytes', 'HTTP request size',
                                    ['method', 'endpoint'])

http_response_size_bytes = Histogram('http_response_size_bytes', 'HTTP response size',
                                     ['method', 'endpoint'])

http_request_count = Counter('http_request_count', 'HTTP Request Count', ['method', 'endpoint'])
http_concurrent_request_count = Gauge('http_concurrent_request_count', 'Flask app concurrent request count')
app.before_request(before_request)
app.after_request(after_request)

app.add_url_rule('/metrics', 'prometheus_metrics', view_func=metrics)

```

Example 17

Project: *ras-frontstage* Author: *ONSdigital* File: *upload_survey.py* MIT License

4 vc

```

def upload_survey(session):
    party_id = session['party_id']
    case_id = request.args['case_id']
    business_party_id = request.args['business_party_id']
    survey_short_name = request.args['survey_short_name']
    logger.info('Attempting to upload collection instrument', case_id=case_id, party_id=party_id)

    if request.content_length > app.config['MAX_UPLOAD_LENGTH']:
        return redirect(url_for('surveys_bp.upload_failed',
                                _external=True,
                                case_id=case_id,
                                business_party_id=business_party_id,
                                survey_short_name=survey_short_name,
                                error_info='size'))

    # Check if respondent has permission to upload for this case
    party_controller.is_respondent_enrolled(party_id, business_party_id, survey_short_name)

    # Get the uploaded file
    upload_file = request.files['file']
    upload_filename = upload_file.filename
    upload_file = {
        'file': (upload_filename, upload_file.stream, upload_file.mimetype, {'Expires': 3600})
    }

    try:
        # Upload the file to the collection instrument service
        collection_instrument_controller.upload_collection_instrument(upload_file,
                                error_message="File upload failed")
    except CiUploadError as ex:
        if ".xlsx format" in ex.error_message:
            error_info = "type"
        elif "50 characters" in ex.error_message:
            error_info = "charLimit"

```

```

elif "File too large" in ex.error_message:
    error_info = 'size'
else:
    logger.error('Unexpected error message returned from collection instru
                  status=ex.status_code,
                  error_message=ex.error_message,
                  party_id=party_id,
                  case_id=case_id)
    error_info = "unexpected"
return redirect(url_for('surveys_bp.upload_failed',
                        _external=True,
                        case_id=case_id,
                        business_party_id=business_party_id,
                        survey_short_name=survey_short_name,
                        error_info=error_info))

logger.info('Successfully uploaded collection instrument', party_id=party_id,
return render_template('surveys/surveys-upload-success.html', upload_filename=

```

Example 18

Project: *pydota2_archive* Author: *pydota2* File: *client_connector.py* Apache License 2.0

4 vc

```

def post():
    global post_connected
    global rtt_queue

    #print('IN POST')
    response = {}
    response['status'] = 200

    if request.method == 'POST':
        try:
            data = request.get_json()

            if data == None:
                # this should raise an HTTPException
                abort(400, 'POST Data was not JSON')

            if request.content_length < 2400 and request.content_length != 0:
                #print("Received Post: ", str(data))

                response['Type'] = data['Type']

                if data['Type'] == 'P':

                    rtt_lock.acquire()
                    rtt_queue = data
                    rtt_lock.release()

                    response['Data'] = {}
                    while not post_queue.empty():
                        action_tuple = ClientThread.get_from_post_queue()
                        #print('Action Tuple To Send To Dota: ', action_tuple)
                        if action_tuple:
                            response['Data'][str(action_tuple[0])] = {}
                            response['Data'][str(action_tuple[0])][str(action_
elif data['Type'] == 'X':
                    post_connected = True

                response['Time'] = data['Time']
        else:

```



```

        print("Request too long", request.content_length)
        response = {"status": 413, "content_length": request.content_length}
        return jsonify(response)
    except:
        traceback.print_exc()
        response['status'] = 500
    else:
        response['status'] = 401
        abort(400, 'Request Method is not POST')

    #print('SENDING RESPONSE:\n', response)
    return jsonify(response)

```

Example 19

Project: *flask-gcp-log-groups* Author: *salrashid123* File: *gcp_logging.py* Apache License 2.0

4 vc

```

def init_app(self, app):

    # capture the http_request time
    @app.before_request
    def before_request():
        g.request_start_time = time.time()
        g.request_time = lambda: "%.5fs" % (time.time() - g.request_start_time)

    # always log the http_request@ default INFO
    @app.after_request
    def add_logger(response):
        TRACE = None
        SPAN = None
        if (self.traceHeaderName in request.headers.keys()):
            # trace can be formatted as "X-Cloud-Trace-Context: TRACE_ID/SPAN_ID"
            rawTrace = request.headers.get(self.traceHeaderName).split('/')
            TRACE = rawTrace[0]
            if ( len(rawTrace) > 1 ) :
                SPAN = rawTrace[1].split(';')[0]

        # https://github.com/googleapis/googleapis/blob/master/google/logging/
        REQUEST = {
            'requestMethod': request.method,
            'requestUrl': request.url,
            'status': response.status_code,
            'responseSize': response.content_length,
            'latency': g.request_time(),
            'remoteIp': request.remote_addr,
            'requestSize': request.content_length
        }

        if 'user-agent' in request.headers:
            REQUEST['userAgent'] = request.headers.get('user-agent')

        if request.referrer:
            REQUEST['referer'] = request.referrer

        # find the log level priority sub-messages; apply the max level to the
        if len(self.mLogLevels) == 0:
            severity = logging.getLevelName(logging.INFO)
            if (response.status_code >= 400 ):
                severity = logging.getLevelName(logging.ERROR)
        else:
            severity= min(self.mLogLevels)
        self.mLogLevels=[]

```

```
self.transport_parent.send(
    None,
    timestamp= datetime.datetime.utcnow(),
    severity = severity,
    resource=self.resource,
    labels=self.labels,
    trace=TRACE,
    span_id = SPAN,
    http_request=REQUEST)

#response.headers['x-upstream-service-time'] = g.request_time()
return response
```
