# Python flask.request.query_string() Examples

The following are code examples for showing how to use *flask.request.query_string()*. 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: *roger-api* Author: *rogertalk* File: *admin.py* MIT License 7 vo

```
def _format(self, template, **kwargs):
        qs = dict(urlparse.parse_qsl(request.query_string))
        qs['cursor'] = self.cursor.urlsafe()
        kwargs.setdefault('path', '%s?%s' % (request.path, urllib.urlencode(qs)))
        if self.next_cursor:
            qs['cursor'] = self.next_cursor.urlsafe()
            kwargs.setdefault('next_path', '%s?%s' % (request.path, urllib.urlenco
        if self.processed < len(self.records):
            kwargs.setdefault('count', '%d/%d' % (self.processed, len(self.records
        else:
            kwargs.setdefault('count', str(self.processed))
        kwargs.setdefault('kind', self.record_kind or 'record')
        lines = [template.format(**kwargs)]
        if self.logs:
            lines.append('--')
            lines.extend(cgi.escape(line) for line in self.logs)
        html = u'<pre>{}</pre>'.format(u'\n'.join(lines))
        if self.auto_page:
            html += AUTO_SCRIPT % (int(self.auto_page_delay * 10),)
        return html
```

### Example 2

Project: *torrent-workshops* Author: *zielmicha* File: *tracker.py* GNU General Public License v3.0 6 vo

```
def hello():
    args = urllib_raw.urldecode(request.query_string)
    print(args)
    info_hash = args[b'info_hash']
    peer_id = args[b'peer_id']
    info = (request.remote_addr, int(args[b'port']))
    event = args.get(b'event')

    if event != b'stopped' and peer_id.startswith(b'-TR'):
        data[info_hash][peer_id] = info

    return bencode.encode({
        b'interval': 10,
        b'peers': b''.join([
            ipaddress.IPv4Address(this_info[0]).packed
            + struct.pack('!H', this_info[1])
            for this_peer_id, this_info in data[info_hash].items()
            if this_peer_id != peer_id
        ])
    })
```

### Example 3

6 vo

Project: *flask-oauth2-devices* Author: *greedo* File: *devices.py* MIT License

```
def _verify_request(self, scopes):
        """ verify recieved oauth2 data
        """
        if request.method == 'POST':
            return False

        uri = request.base_url
        if request.query_string:
            uri += '?' + request.query_string.decode('utf-8')

        data = request.form.to_dict()
        headers = dict(request.headers)

        if ['oauth_version', 'oauth_nonce', 'oauth_timestamp\
         ', 'user' 'client'] not in data.keys():
            return False

        return True
```

### Example 4

Project: *melee* Author: *leeyingmu* File: *wsgiapp.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def after_request(self, response):
        if request.endpoint is None:
            return response
        if response is None:
            return response

        g.request_cost = int(time.time()*1000) - g.startms

        if getattr(g, 'response_code', None) is None:
            code = response.status_code
        else:
            code = g.response_code

        # 支持jsonp，解决ajax get 请求跨域问题
        #if g.jsonpcallback:
            #response.response = '%s(%s)' % (g.jsonpcallback, response.response)
        response.headers['Access-Control-Allow-Origin'] = '*'

        self.logger.info('REQUEST', request.remote_addr, request.method, g.request
            '%s?%s' % (request.path, request.query_string), request.headers.get('
            response.status_code, code, response.response, str(response.headers.ge

        return response
```

### Example 5

Project: *rate.sx* Author: *chubin* File: *srv.py* MIT License 5 vo

```
def answer(topic = None):
    """
    Main rendering function, it processes incoming weather queries.
    Depending on user agent it returns output in HTML or ANSI format.

    Incoming data:
        request.args
        request.headers
        request.remote_addr
        request.referrer
        request.query_string
    """

    user_agent = request.headers.get('User-Agent', '').lower()
    html_needed = is_html_needed(user_agent)
    options = parse_query(request.args)
    hostname = request.headers['Host']

    if request.headers.getlist("X-Forwarded-For"):
       ip = request.headers.getlist("X-Forwarded-For")[0]
       if ip.startswith('::ffff:'):
           ip = ip[7:]
    else:
       ip = request.remote_addr
    if request.headers.getlist("X-Forwarded-For"):
       ip = request.headers.getlist("X-Forwarded-For")[0]
       if ip.startswith('::ffff:'):
           ip = ip[7:]
    else:
       ip = request.remote_addr


    if topic is None:
        topic = ":firstpage"

    answer = cmd_wrapper(topic, hostname=hostname, request_options=options, html=i

    if ip not in SKIP_LOGGING_FOR_THIS_IPS:
        log_query(ip, hostname, topic, user_agent)
    return answer
```

### Example 6

Project: *Quiver-alfred* Author: *danielecook* File: *flask_utils.py* MIT License 5 vo

```
def get_current_url():
    if not request.query_string:
        return request.path
    return '%s?%s' % (request.path, request.query_string)
```

### Example 7

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def query(page=0):
    try:
        ans = requestquery(page=page)
        return jsonify(ans)

    except errors.KarpException as e:  # pass on karp exceptions
        _logger.exception(e)
        raise
    except Exception as e:  # catch *all* exceptions and show for user
        _logger.exception(e)
        raise errors.KarpGeneralError(
            str(e), user_msg=str(e), query=request.query_string
        )
```

### Example 8

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def test():
    auth, permitted = validate_user(mode="read")
    try:
        # default
        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)
    except QueryError as e:
        raise errors.KarpQueryError(
            "Parse error", debug_msg=e, query=request.query_string
        )
    return jsonify({"elastic_json_query": elasticq})
```

### Example 9

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def explain():
    auth, permitted = validate_user(mode="read")
    try:
        # default
        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)
    except QueryError as e:
        raise errors.KarpQueryError(
            "Parse error", debug_msg=e, query=request.query_string
        )
    es = conf_mgr.elastic(mode=settings["mode"])
    index, typ = conf_mgr.get_mode_index(settings["mode"])
    ex_ans = es.indices.validate_query(index=index, body=elasticq, explain=True)
    q_ans = requestquery(page=0)
    return jsonify({"elastic_json_query": elasticq, "ans": q_ans, "explain": ex_an
```

### Example 10

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def statistics():
    """ Returns the counts and stats for the query """
    auth, permitted = validate_user(mode="read")
    try:
        mode = parser.get_mode()
        default = {
            "buckets": conf_mgr.searchfield(mode, "statistics_buckets"),
            "size": 100,
            "cardinality": False,
        }
        settings = parser.make_settings(permitted, default)
        exclude = [] if auth else conf_mgr.searchfield(mode, "secret_fields")

        elasticq, more = parser.statistics(settings, exclude=exclude)
        es = conf_mgr.elastic(mode=settings["mode"])
        index, typ = conf_mgr.get_mode_index(settings["mode"])
        is_more = check_bucketsize(more, settings["size"], index, es)

        # TODO allow more than 100 000 hits here?
        _logger.debug("stat body %s", elasticq)
        ans = es.search(
            index=index, body=elasticq, search_type="query_then_fetch", size=0
        )
        ans["is_more"] = is_more
        return jsonify(ans)
    except AuthenticationError as e:
        _logger.exception(e)
        msg = e.message
        raise errors.KarpAuthenticationError(msg)
    except errors.KarpException as e:  # pass on karp exceptions
        _logger.exception(e)
        raise
    except Exception as e:  # catch *all* exceptions
        _logger.exception(e)
        raise errors.KarpGeneralError(
            "Unknown error", debug_msg=e, query=request.query_string
        )
```

### Example 11

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def testquery():
    """ Returns the query expressed in elastics search api """
    auth, permitted = validate_user(mode="read")
    try:
        # default
        settings = parser.make_settings(permitted, {"size": 25, "page": 0})
        elasticq = parser.parse(settings)
        mode = settings["mode"]
        if not settings.get("sort", ""):
            # default: group by lexicon, then sort by score
            sort = conf_mgr.searchfield(mode, "sort_by")
        else:
            sort = settings["sort"]
        start = (
            settings["start"]
            if "start" in settings
            else settings["page"] * settings["size"]
        )
        elasticq = parser.parse()
        return json.dumps(elasticq) + json.dumps(
            {"sort": sort, "_from": start, "size": settings["size"], "version": "t
        )
    except Exception as e:  # catch *all* exceptions
        # TODO only catch relevant exceptions
        _logger.exception(e)
        raise errors.KarpGeneralError(e, request.query_string)
```

### Example 12

Project: *wttr.in* Author: *chubin* File: *proxy.py* Apache License 2.0 5 vo

```
def proxy(path):
    """
    Main proxy function. Handles incoming HTTP queries.
    """

    lang = request.args.get('lang', 'en')
    query_string = request.query_string
    query_string = query_string.replace('sr-lat', 'sr')
    query_string = query_string.replace('lang=None', 'lang=en')
    content, headers = _load_content_and_headers(path, query_string)
    if content is None:
        srv = _find_srv_for_query(path, query_string)
        url = '%s/%s?%s' % (srv, path, query_string)
        print(url)

        attempts = 10
        response = None
        while attempts:
            try:
                response = requests.get(url, timeout=2)
            except requests.ReadTimeout:
                attempts -= 1
                continue
            try:
                json.loads(response.content)
                break
            except ValueError:
                attempts -= 1

        _touch_empty_file(path, query_string, content, headers)
        if response:
            headers = {}
            headers['Content-Type'] = response.headers['content-type']
            content = add_translations(response.content, lang)
            _save_content_and_headers(path, query_string, content, headers)
        else:
            content = "{}"

    return content, 200, headers
```

### Example 13

Project: *benchtracker* Author: *LemonPi* File: *server_db.py* MIT License 5 vo

```
def parse_filters(verbose=False):
    """
    Parse filter from current request query string and return the filtered paramet

    verbose mode returns filters without splitting out the type
    """
    filter_param = None
    filter_method = None
    filters = []
    filter_args = []
    filter_params = []
    for arg in urlparse.parse_qsl(request.query_string):
        if arg[0][0] != 'f':
            continue
        # new filter parameter
        if arg[0] == 'fp':
            # previous filter ready to be built
            if filter_param and filter_method and filter_args:
                filters.append(d.Task_filter(filter_param, filter_method, filter_a
                filter_args = []    # clear arguments; important!
                print("{}: {}".format(filters[-1], filters[-1].args))
                filter_params.append(filter_param)
            # split out the optional type following parameter name
            if verbose:
                filter_param = arg[1]
            else:
                filter_param = sql_escape(strip_last_word(arg[1]))
        if arg[0] == 'fm':
            filter_method = arg[1]
        if arg[0] == 'fa':
            filter_args.append(arg[1])
    # last filter to be added
    if (not filters or filter_param != filters[-1].param) and filter_args:
        filters.append(d.Task_filter(filter_param, filter_method, filter_args))
        print("{}: {}".format(filters[-1], filters[-1].args))
        filter_params.append(filter_param)

    return filter_params,filters
```

### Example 14

Project: *steemrocks* Author: *emre* File: *app.py* MIT License 5 vo

```
def index():
    if request.query_string and request.args.get('account'):
        return redirect('/' + request.args.get('account'))
    return render_template('index.html')
```

### Example 15

Project: *steemrocks* Author: *emre* File: *app.py* MIT License 5 vo

```
def profile(username, page):
    if username.startswith("@"):
        username = username.replace("@", "")

    op_type = None
    if request.query_string and request.args.get('op_type'):
        op_type = request.args.get("op_type")
        if op_type not in op_types:
            op_type = None

    account = Account(username, get_steem_conn()).set_account_deta()
    if not account.account_data:
        abort(404)

    page = page - 1
    start = page * PER_PAGE
    pagination = Pagination(page, PER_PAGE,
                            account.get_operation_count(op_type=op_type))

    operations = account.get_operations(start=start, end=PER_PAGE,
                                        op_type=op_type)

    return render_template(
        'profile.html', account=account,
        operations=operations,
        site_url=SITE_URL, pagination=pagination,
        op_type=op_type, op_types=op_types)
```

### Example 16

Project: *blockexplorer* Author: *GenesisKernel* File: *utils.py* GNU General Public License v2.0 5 vo

```
def get_db_id_from_request():
    logger.debug("request.url: %s" % request.url)
    logger.debug("request.query_string: %s" % request.query_string)
    db_id = None
    if request and hasattr(request, 'url'):
        p = urlparse(request.url)
        logger.debug("path: %s" % p.path)
        m = re.search('^\/(genesis|db-engine)\/database\/([0-9]+)\/.*', p.path)
        logger.debug("m: %s" % m)
        if m:
            try:
                db_id = int(m.group(2))
            except Exception as e:
                pass
    return db_id
```

### Example 17

Project: *fame* Author: *certsocietegenerale* File: *views.py* GNU General Public License v3.0 5 vo

```
def prepare_auth_request(request):
    url_data = urlparse(request.url)
    return {
        "https": 'on',
        'http_host': request.host,
        'server_port': url_data.port,
        'script_name': request.path,
        'get_data': request.args.copy(),
        'post_data': request.form.copy(),
        # Uncomment if using ADFS as IdP, https://github.com/onelogin/python-saml/
        # 'lowercase_urlencoding': True,
        'query_string': request.query_string
    }
```

### Example 18

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vo

```
def get_buffer():
    dict_ = parse_qs(request.query_string)
    kwargs =  dict([(k, int(e[0])) for (k, e) in dict_.items()])
    rbio = RandomBytesIO(**kwargs)
    fsize = rbio.size()
    headers = Headers()
    headers['Content-Length'] = fsize
    filename = make_csv_fifo(rbio)
    file_ = open(filename, 'rb')
    data = wrap_file(request.environ, file_)
    return current_app.response_class(data, mimetype='text/csv', headers=headers,
                                        direct_passthrough=True)
```

### Example 19

Project: *yeti* Author: *yeti-platform* File: *views.py* Apache License 2.0 5 vo

```
def prepare_auth_request(request):
    url_data = urlparse(request.url)
    return {
        "https": 'on',
        'http_host': request.host,
        'server_port': url_data.port,
        'script_name': request.path,
        'get_data': request.args.copy(),
        'post_data': request.form.copy(),
        # Uncomment if using ADFS as IdP, https://github.com/onelogin/python-saml/
        # 'lowercase_urlencoding': True,
        'query_string': request.query_string
    }
```

### Example 20

Project: *yabgp* Author: *smartbgp* File: *utils.py* Apache License 2.0 5 vo

```
def log_request(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        LOG.info('API request url %s', request.url)
        if request.query_string:
            LOG.info('API query string %s', request.query_string)
        LOG.info('API request method %s', request.method)
        if request.method == 'POST':
            LOG.info('API POST data %s', request.json)
        LOG.debug('API request environ %s', request.environ)
        return f(*args, **kwargs)
    return decorated_function
```

### Example 21

Project: *InfraBox* Author: *SAP* File: *saml.py* Apache License 2.0 5 vo

```
def init_saml_auth():
    parsed_url = urlparse(request.url)
    request_data = {
        "https": "on" if request.scheme == "https" else "off",
        "http_host": request.host,
        "server_port": parsed_url.port,
        "script_name": request.path,
        "get_data": request.args.copy(),
        "post_data": request.form.copy(),
        "query_string": request.query_string
        }

    auth = OneLogin_Saml2_Auth(request_data, custom_base_path=get_env("INFRABOX_AC
    return auth
```

### Example 22

Project: *onearth-image-analytics* Author: *nasa-gibs* File: *main.py* Apache License 2.0 5 vo

```
def single_tile(projection, kind, product, date, resolution, tilematrix, x, y, ext
    url = "http://onearth-tile-services" + request.path

    ACCESS_LOG(f"URL: {url}")
    ACCESS_LOG(f"Projection {projection}, kind: {kind}, product: {product}, date:
    ACCESS_LOG(f"Args: {request.query_string}")

    r, status_code, headers = make_request(url, headers=request.headers)

    if status_code == 404:
        ACCESS_LOG("Status_code 404")
        return render_template('404.html'), 404, headers
    if status_code == 304:
        ACCESS_LOG("Status_code 304")
        return "", 304, headers

    try:
        method, arg_dict = parse_args(request.args)
    except Exception as e:
        ACCESS_LOG("Error in arg parsing!")
        error_dict = { "Error" : str(e), "Code" : 404, "Request" : request.path }
        return jsonify(error_dict), 404

    if method is None:
        output = r.content
    else:
        output = method(r, **arg_dict)

    # resp = handle_varnish(r, resp)
    # ACCESS_LOG(str(headers))
    return output, status_code, dict(headers)
```

### Example 23

Project: *geobricks_mapclassify* Author: *geobricks* File: *mapclassify_rest.py* GNU General Public License v2.0 5 vo

```
def proxy():
    try:
        url = request.args.get('urlWMS')
        if url is None:
            raise Exception('Parameter is not set')

        # TODO: add other checks (on all the other parameters)
        #r = requests.get(url + "?" + request.query_string)
        r = urllib2.urlopen(url + "?" + request.query_string).read()
        return Response(r,  content_type='text/plain; charset=utf-8')
    except Exception, e:
        log.error(e)
```

### Example 24

Project: *deresuteme* Author: *marcan* File: *app.py* Apache License 2.0 5 vo

```
def try_get_banner(user_id, sizename, privacy=0):
    if sizename.endswith(".png"):
        sizename = sizename[:-4]
    if sizename not in sizemap:
        abort(404)
    if len(str(user_id)) != 9:
        abort(404)
    size = sizemap[sizename]
    try:
        data, mtime = get_data(user_id)
        key = "%d_p%d" % (user_id, privacy)
        privatize(data, privacy)
        res = get_sized_banner(key, data, mtime, size)
        if request.query_string == "dl":
            res.headers['Content-Disposition'] = 'attachment; filename=%d_p%d_%s.p
        return res
    except APIError as e:
        if e.code == 1457:
            return send_file("static/error_404_%d.png" % size, mimetype="image/png
        elif e.code == 101:
            return send_file("static/error_503_%d.png" % size, mimetype="image/png
        else:
            app.logger.exception("API error for %r/%r/%r" % (user_id, sizename, pr
            return send_file("static/error_%d.png" % size, mimetype="image/png", c
    except Exception as e:
        app.logger.exception("Exception thrown for %r/%r/%r" % (user_id, sizename,
        return send_file("static/error_%d.png" % size, mimetype="image/png", cache
```

### Example 25

Project: *deresuteme* Author: *marcan* File: *app.py* Apache License 2.0 5 vo

```
def try_get_snap(snap, sizename):
    if sizename.endswith(".png"):
        sizename = sizename[:-4]
    if sizename not in sizemap:
        abort(404)
    size = sizemap[sizename]
    data = load_snap(snap)
    key = "s_" + snap
    res = get_sized_banner(key, data, None, size, max_age=None)
    if request.query_string == "dl":
        res.headers['Content-Disposition'] = 'attachment; filename=snap_%s_%s.png'
    return res
```

### Example 26

Project: *aeon-ztps* Author: *Apstra* File: *views.py* Apache License 2.0 5 vo

```
def _get_devices():
    db = aeon_ztp.db.session
    to_json = device_schema

    # ---------------------------------------------------------------
    # if the request has arguments, use these to form an "and" filter
    # and return only the subset of items matching
    # ---------------------------------------------------------------

    if request.args:
        try:
            recs = find_devices(db, request.args.to_dict())
            if len(recs) == 0:
                return jsonify(ok=False,
                               message='Not Found: %s' % request.query_string), 4

            items = [to_json.dump(rec).data for rec in recs]
            return jsonify(count=len(items), items=items)

        except AttributeError:
            return jsonify(ok=False, message='invalid arguments'), 500

    # -------------------------------------------
    # otherwise, return all items in the database
    # -------------------------------------------

    items = [to_json.dump(rec).data for rec in db.query(Device).all()]
    return jsonify(count=len(items), items=items)


# -----------------------------------------------------------------------------
#                                  POST /api/devices
# -----------------------------------------------------------------------------
```

### Example 27

Project: *aeon-ztps* Author: *Apstra* File: *views.py* Apache License 2.0 5 vo

```
def _delete_devices():
    if request.args.get('all'):
        try:
            db = aeon_ztp.db.session
            db.query(Device).delete()
            db.commit()

        except Exception as exc:
            return jsonify(
                ok=False,
                message='unable to delete all records: {}'.format(exc.message)), 4

        return jsonify(ok=True, message='all records deleted')

    elif request.args:
        db = aeon_ztp.db.session
        try:
            recs = find_devices(db, request.args.to_dict())
            n_recs = len(recs)
            if n_recs == 0:
                return jsonify(ok=False,
                               message='Not Found: %s' % request.query_string), 4

            for dev in recs:
                db.delete(dev)
            db.commit()
            return jsonify(
                ok=True, count=n_recs,
                message='{} records deleted'.format(n_recs))

        except AttributeError:
            return jsonify(ok=False, message='invalid arguments'), 500

        except Exception as exc:
            msg = 'unable to delete specific records: {}'.format(exc.message)
            return jsonify(ok=False, message=msg), 500
    else:
        return jsonify(ok=False, message='all or filter required'), 400
```

### Example 28

Project: *zappa-bittorrent-tracker* Author: *Miserlou* File: *track.py* MIT License 5 vo

```
def get_info_hash(request, multiple=False):
    """
    Get infohashes from a QS.
    """
    if not multiple:
        return b2a_hex(cgi.parse_qs(request.query_string)['info_hash'][0])
    else:
        hashes = set()
        for hash in cgi.parse_qs(request.query_string)['info_hash']:
            hashes.add(b2a_hex(hash))
    return hashes
```

### Example 29

Project: *melee* Author: *leeyingmu* File: *callbackpay_view.py* BSD 3-Clause "New" or "Revised" License 5 vc

```
def before_request():
    g.url = request.url
    g.query_string = request.query_string
    headers = {}
    for k, v in request.headers or {}:
      headers[k.lower()] = v
    g.headers = headers
    g.jsondata = json.loads(g.rawdata)
    g.jsondata.update(request.values.to_dict())
```

**Example 30**

Project: *melee* Author: *leeyingmu* File: *wsgiapp.py* BSD 3-Clause "New" or "Revised" License 5 vc

```
def before_request(self):
        self.logger.info('REQUEST', '%s?%s' % (request.path, request.query_string
        g.endpoint = request.endpoint.split('.')[-1] if request.endpoint else None
        g.rawdata = request.data
        g.jsondata = {}
        if request.endpoint is None:
            return
        g.startms = int(time.time()*1000)

        content = request.values.get('content')
        signature = request.values.get('signature', '')
        sig_kv = request.values.get('sig_kv')
        timestamp = request.values.get('timestamp') or 0
        g.jsonpcallback = request.values.get('callback')

        if content:
            if not timestamp or (time.time()*1000)-int(timestamp) > 86400000:
                raise BadRequest(description='request expired %s' % timestamp)

            if not self.verify_signature(sig_kv, signature, content, timestamp):
                raise SignatureError(description='Signature Not Correct.')
            try:
                g.jsondata = json.loads(content)
            except:
                g.jsondata = {}

            if config.appids and g.jsondata.get('appid') not in config.appids:
                raise BadRequest(description='Reqeust appid error')
```

**Example 31**

Project: *BhagavadGita* Author: *gita* File: *utils.py* GNU General Public License v3.0 5 vc

```
def _get_uri_from_request(request):
    """
    The uri returned from request.uri is not properly urlencoded
    (sometimes it's partially urldecoded) This is a weird hack to get
    werkzeug to return the proper urlencoded string uri
    """
    uri = request.base_url
    if request.query_string:
        uri += '?' + request.query_string.decode('utf-8')
    return uri
```

### Example 32

Project: *RSPET* Author: *panagiks* File: *rspet_server_api.py* MIT License 5 vo

```
def get_hosts():
    """Return all hosts."""
    #Check for query string, redirect to endpoint with trailling '/'.
    if request.query_string:
        return redirect(url_for('run_cmd') + '?' + request.query_string)
    hosts = RSPET_API.get_hosts()
    return jsonify({'hosts': [make_public_host(hosts[h_id], h_id) for h_id in host
```

### Example 33

Project: *RSPET* Author: *panagiks* File: *rspet_server_api.py* MIT License 5 vo

```
def get_host(host_id):
    """Return specific host."""
    #Check for query string, redirect to endpoint with trailling '/'.
    if request.query_string:
        return redirect(url_for('run_cmd_host', host_id=host_id) + '?' + request.
    hosts = RSPET_API.get_hosts()
    try:
        host = hosts[host_id]
    except KeyError:
        abort(404)
    return jsonify(make_public_host(host, host_id))
```

### Example 34

Project: *estuary-api* Author: *release-engineering* File: *monitoring.py* GNU General Public License v3.0 5 vo

```
def stop_request_timer(response):
    """
    Stop the request timer.

    :param flask.Response response: the Flask response to stop the timer on
    :return: the Flask response
    :rtype: flask.Response
    """
    resp_time = time.time() - request.start_time
    REQUEST_LATENCY.labels(
        'estuary-api', request.path, request.query_string.decode('utf-8')).observ
    return response
```

### Example 35

Project: *estuary-api* Author: *release-engineering* File: *monitoring.py* GNU General Public License v3.0 5 vo

```
def record_request_metadata(response):
    """
    Record metadata about the request.

    :param flask.Response response: the Flask response to record metadata about
    :return: the Flask response
    :rtype: flask.Response
    """
    REQUEST_COUNT.labels(
        'estuary-api', request.method, request.path, request.query_string,
        response.status_code).inc()
    return response
```

### Example 36

Project: *xdata-feat* Author: *ContinuumIO* File: *feat.py* MIT License 5 vc

```
def edgar_num():
    args = request.args
    url = ES_ENDPOINT + "/edgar-base15/num/_search?" + request.query_string
    res = requests.get(url, timeout=2)

    return jsonify(res.json())
```

### Example 37

Project: *Sploits* Author: *iDuronto* File: *rspet_server_api.py* MIT License 5 vc

```
def get_hosts():
    """Return all hosts."""
    #Check for query string, redirect to endpoint with trailling '/'.
    if request.query_string:
        return redirect(url_for('run_cmd') + '?' + request.query_string)
    hosts = RSPET_API.get_hosts()
    return jsonify({'hosts': [make_public_host(hosts[h_id], h_id) for h_id in host
```

### Example 38

Project: *Sploits* Author: *iDuronto* File: *rspet_server_api.py* MIT License 5 vc

```
def get_host(host_id):
    """Return specific host."""
    #Check for query string, redirect to endpoint with trailling '/'.
    if request.query_string:
        return redirect(url_for('run_cmd_host', host_id=host_id) + '?' + request.
    hosts = RSPET_API.get_hosts()
    try:
        host = hosts[host_id]
    except KeyError:
        abort(404)
    return jsonify(make_public_host(host, host_id))
```

### Example 39

Project: *FlaskBackend* Author: *iamrajhans* File: *auth_required.py* MIT License 4 vc

```
def auth_decorator(func):

    @wraps(func)
    def decorator_func(*args,**kwargs):
        user          = request.headers.get('user')
        api_key       = request.headers.get('api_key')
        # api_secret   = request.headers.get('api_secret')
        user_hash     = request.headers.get('hash')
        user_timestamp    = request.headers.get('timestamp')

        if not user or not api_key :
            return jsonify("Error: Invalid Request"),412
        if not hash or not user_timestamp or not user_hash:
            return jsonify("Error: Invalid Request"), 412

        server_key = get_key(api_key,user)

        if not server_key:
            return jsonify("key not found"),412

        timestamp_hash = generate_hmac(str(server_key), str(user_timestamp))
        #for get request

        if request.method == 'GET':
            url = request.path + '?' + request.query_string if request.query_stri
            server_hash = generate_hmac(str(timestamp_hash), str(url))
            if hmac.compare_digest(server_hash, user_hash):
                return func(*args, **kwargs)
            else:
                return jsonify("Error : HMAC is not matched"), 412
            #change with the hmac
            # server_hash = base64.base64encode(str(server_key),str(url))
            # if user_hash == server_hash:
            #     return func(*args,**kwargs)

            # else :
                # return jsonify("Error: HMAC is not matched"),412


        if request.method == 'POST':
            #check for file upload
            data = request.data.decode('utf-8')
            server_hash = generate_hmac(str(timestamp_hash),data)
            if hmac.compare_digest(server_hash,user_hash):
                return func(*args, **kwargs)
            else:
                return jsonify("Error : HMAC is not matched"), 412
    return decorator_func
```

### Example 40

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 4 vo

```
def querycount(page=0):
    # TODO error if buckets is used here
    # TODO validate_user is also done once in requestquery
    # but since we need the permitted dict, it is called
    # here as well
    auth, permitted = validate_user(mode="read")
    try:
        # TODO buckets should be gathered from some config
        stat_size = request.args.get("statsize", conf_mgr.app_config.MAX_PAGE)
        default = {"buckets": ["lexiconOrder", "lexiconName"], "size": stat_size}
        settings = parser.make_settings(permitted, default)
        q_ans = requestquery(page=page)

        # raise the size for the statistics call
        count_elasticq, more = parser.statistics(
            settings,
            order={"lexiconOrder": ("_key", "asc")},
            show_missing=False,
            force_size=stat_size,
        )
        mode = settings["mode"]
        es = conf_mgr.elastic(mode=mode)
        index, typ = conf_mgr.get_mode_index(mode)
        _logger.debug("|querycount| Will ask %s", count_elasticq)
        count_ans = es.search(
            index=index,
            body=count_elasticq,
            search_type="query_then_fetch",
            # raise the size for the statistics call
            size=25,  # stat_size
        )
        _logger.debug("ANNE: count_ans: %s\n", count_ans)
        distribution = count_ans["aggregations"]["q_statistics"]["lexiconOrder"][
            "buckets"
        ]
    except errors.KarpException as e:  # pass on karp exceptions
        _logger.exception(e)
        raise

    except (elasticsearch.RequestError, elasticsearch.TransportError) as e:
        _logger.exception(e)
        raise errors.KarpElasticSearchError("ElasticSearch failure. Message: %s.\r

    except Exception as e:  # catch *all* exceptions
        # Remember that 'buckets' is not allowed here! %s"
        _logger.exception(e)
        raise errors.KarpQueryError(
            "Could not parse data", debug_msg=e, query=request.query_string
        )
    return jsonify({"query": q_ans, "distribution": distribution})
```

### Example 41

Project: *karp-backend* Author: *spraakbanken* File: searching.py MIT License 4 vc

```
def minientry():
    """ Returns the counts and stats for the query """
    max_page = conf_mgr.app_config.MINIENTRY_PAGE
    auth, permitted = validate_user(mode="read")
    try:
        mode = parser.get_mode()
        default = {"show": conf_mgr.searchfield(mode, "minientry_fields"), "size":
        settings = parser.make_settings(permitted, default)
        elasticq = parser.parse(settings)
        show = settings["show"]
        if not auth:
            # show = show - exclude
            exclude = conf_mgr.searchfield(mode, "secret_fields")
            show = list(set(show).difference(exclude))

        sort = sortorder(settings, mode, settings.get("query_command", ""))
        start = settings["start"] if "start" in settings else 0
        es = conf_mgr.elastic(mode=settings["mode"])
        index, typ = conf_mgr.get_mode_index(settings["mode"])
        ans = parser.adapt_query(
            settings["size"],
            start,
            es,
            elasticq,
            {
                "index": index,
                "_source": show,
                "from_": start,
                "sort": sort,
                "size": min(settings["size"], max_page),
                "search_type": "dfs_query_then_fetch",
            },
        )
        if settings.get("highlight", False):
            clean_highlight(ans)

        return jsonify(ans)
    except AuthenticationError as e:
        _logger.exception(e)
        msg = e.message
        raise errors.KarpAuthenticationError(msg)
    except QueryError as e:
        raise errors.KarpQueryError(
            "Parse error, %s" % e.message, debug_msg=e, query=request.query_strin
        )
    except errors.KarpException as e:  # pass on karp exceptions
        _logger.exception(e)
        raise
    except Exception as e:  # catch *all* exceptions
        _logger.exception(e)
        raise errors.KarpGeneralError(
            "Unknown error", debug_msg=e, query=request.query_string
        )
```

### Example 42

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 4 vo

```
def statlist():
    """ Returns the counts and stats for the query """
    auth, permitted = validate_user(mode="read")
    try:
        mode = parser.get_mode()
        _logger.debug("mode is %s", mode)
        default = {
            "buckets": conf_mgr.searchfield(mode, "statistics_buckets"),
            "size": 100,
            "cardinality": False,
        }
        settings = parser.make_settings(permitted, default)

        exclude = [] if auth else conf_mgr.searchfield(mode, "secret_fields")
        elasticq, more = parser.statistics(settings, exclude=exclude, prefix="STAT
        es = conf_mgr.elastic(mode=settings["mode"])
        index, typ = conf_mgr.get_mode_index(settings["mode"])
        is_more = check_bucketsize(more, settings["size"], index, es)
        # TODO allow more than 100 000 hits here?
        size = settings["size"]
        ans = es.search(
            index=index, body=elasticq, search_type="query_then_fetch", size=0
        )
        tables = []
        for key, val in list(ans["aggregations"]["q_statistics"].items()):
            if key.startswith("STAT_"):
                tables.extend(generate_table(val, []))
        # the length of tables might be longer than size, so truncate it
        # generating shorter tables is not faster than generating all of it
        # and then truncating
        if size:
            tables = tables[:size]
        return jsonify({"stat_table": tables, "is_more": is_more})

    except AuthenticationError as e:
        _logger.exception(e)
        msg = e.message
        raise errors.KarpAuthenticationError(msg)
    except errors.KarpException as e:  # pass on karp exceptions
        _logger.exception(e)
        raise
    except Exception as e:  # catch *all* exceptions
        # raise
        _logger.exception(e)
        raise errors.KarpGeneralError(
            "Unknown error", debug_msg=e, query=request.query_string
        )
```