Python flask.request.args() Examples

The following are code examples for showing how to use *flask.request.args()*. 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

```python
def create_message(session):
    """Creates and sends a message outside of the context of an existing conversat
    survey = request.args['survey']
    ru_ref = request.args['ru_ref']
    party_id = session['party_id']
    form = SecureMessagingForm(request.form)
    if request.method == 'POST' and form.validate():
        logger.info("Form validation successful", party_id=party_id)
        sent_message = _send_new_message(party_id, survey, ru_ref)
        thread_url = url_for("secure_message_bp.view_conversation",
                             thread_id=sent_message['thread_id']) + "#latest-messa
        flash(Markup('Message sent. <a href={}>View Message</a>'.format(thread_url
        return redirect(url_for('secure_message_bp.view_conversation_list'))

    else:
        return render_template('secure-messages/secure-messages-view.html',
                               ru_ref=ru_ref, survey=survey,
                               form=form, errors=form.errors, message={})
```

Example 2

```python
def download_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 download collection instrument', case_id=case_id, p

    # Check if respondent has permission to download for this case
    case = case_controller.get_case_by_case_id(case_id)
    party_controller.is_respondent_enrolled(party_id, business_party_id, survey_sh

    collection_instrument, headers = collection_instrument_controller.download_col

    logger.info('Successfully downloaded collection instrument', case_id=case_id,

    return collection_instrument, 200, headers
```

Example 3

```python
def access_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']
collection_instrument_type = request.args['ci_type']

if collection_instrument_type == 'EQ':
    logger.info('Attempting to redirect to EQ', party_id=party_id, case_id=cas
    return redirect(case_controller.get_eq_url(case_id, party_id, business_par

logger.info('Retrieving case data', party_id=party_id, case_id=case_id)
referer_header = request.headers.get('referer', {})

case_data = case_controller.get_case_data(case_id, party_id, business_party_id

logger.info('Successfully retrieved case data', party_id=party_id, case_id=cas
return render_template('surveys/surveys-access.html', case_id=case_id,
                       collection_instrument_id=case_data['collection_instrume
                       collection_instrument_size=case_data['collection_instru
                       survey_info=case_data['survey'],
                       collection_exercise_info=case_data['collection_exercise
                       business_info=case_data['business_party'],
                       referer_header=referer_header)
```

Example 4

Project: *cloudygo* Author: *sethtroisi* File: *serve.py* Apache License 2.0   6 vc

```
def game_view(bucket, model_name, filename):
    view_type = request.args.get('type')
    if not view_type:
        path = re.search(r'/(clean|full|eval)/', request.base_url)
        if path:
            view_type = path.group(1)
        else:
            view_type = 'clean'
    assert view_type in ('clean', 'eval', 'full'), view_type

    data, game_view = cloudy.get_game_data(
        bucket, model_name, filename, view_type)

    render_sorry = game_view != view_type

    # HACK: we'd like all eval games to be full in the future
    is_full_eval = 'cc-evaluator' in filename

    return render_game(bucket, model_name, data,
        filename=filename,
        force_full=is_full_eval,
    )
```

Example 5

Project: *cloudygo* Author: *sethtroisi* File: *serve.py* Apache License 2.0   6 vc

```
def position_comparison(bucket, model_name_a, model_name_b):
    model_a, _ = cloudy.load_model(bucket, model_name_a)
    model_b, _ = cloudy.load_model(bucket, model_name_b)
    if model_a is None or model_b is None:
        return 'Model {} or {} not found'.format(model_name_a, model_name_b)

    rule_group = 'policy' if '/policy/' in request.url_rule.rule else 'pv'
    arg_group = request.args.get('group', None)
    group = arg_group or rule_group
```

```
        count, data = cloudy.get_position_sgfs(bucket, [model_a[0], model_b[0]])

        return render_template('position-comparison.html',
                               bucket=bucket,
                               model_a=model_a,
                               model_b=model_b,
                               group=group,
                               sgfs=data,
                               )
```

Example 6

```
def __init__(self, controller: ProxyController):
        self.blueprint = Blueprint('proxy_blueprint', __name__)

        @self.blueprint.route('/', defaults={'path': ''})
        @self.blueprint.route('/<path:path>', methods=['GET', 'OPTIONS', 'HEAD', '
        def proxy(path):
            return controller.proxy(
                path,
                Request(
                    method=request.method,
                    args=request.args,
                    headers=request.headers,
                    raw_content=request.stream.read(),
                ),
                session.get(SESSION_TOKEN)
            )
```

Example 7

```
def get_dig_info():
    if 'host' not in request.args:
        return "Required param 'host' is missing", 400
    host = request.args['host']

    temp_file = generate_random_file_name()
    try:
        with open(temp_file, 'w') as f:
            call(['dig', host, '+short'], stdout=f)

        with open(temp_file, 'r') as f:
            content = f.read()
    except Exception as e:
        content = "Error: {0}".format(e)
    finally:
        if os.path.isfile(temp_file):
            os.remove(temp_file)
    return content
```

Example 8

```
def related():
    """Get related documents or entities."""
    if request.args['type'] == 'document':
        service = 'related_documents'
    elif request.args['type'] == 'entity':
        service = 'related_entities'
    else:
        return Response('Invalid type %s' % request.args['type'])

    if not _globals['gransk'].pipeline.get_service(service):
        return Response('{"error": "service not found"}', status=200, mimetype='appl

    result = _globals['gransk'].pipeline.get_service(service).get_related_to(
        request.args['id'])

    return Response(json.dumps(result), status=200, mimetype='application/json')
```

Example 9

```
def setup(args, pipeline, runmod, injector):
    """Load configuration"""
    logging.basicConfig(
        format='[%(asctime)s] [%(levelname)s] %(name)s: %(message)s',
        level=logging.INFO,
        datefmt='%Y-%m-%d %H:%M:%S')

    _globals['gransk'] = gransk.api.API(injector)
    _globals['config'] = _globals['gransk'].config

    if pipeline:
        _globals['gransk'].pipeline = pipeline

    if _globals['gransk'].pipeline.get_service('related_entities'):
        _globals['gransk'].pipeline.get_service('related_entities').load_all(_globals[

    if _globals['gransk'].pipeline.get_service('related_documents'):
        _globals['gransk'].pipeline.get_service('related_documents').load_all(_globals
```

Example 10

```
def listRemove(self):
    cpe = request.args.get('cpe', type=str)
    cpe = urllib.parse.quote_plus(cpe).lower()
    cpe = cpe.replace("%3a", ":")
    cpe = cpe.replace("%2f", "/")
    lst = request.args.get('list', type=str)
    if cpe and lst:
        result=wl.removeWhitelist(cpe) if lst.lower()=="whitelist" else bl.removeBla
        status = "removed_from_list" if (result > 0) else "already_removed_from_list
    else:
        status = "invalid_cpe"
    returnList = db.getWhitelist() if lst=="whitelist" else db.getBlacklist()
    return jsonify({"status":status, "rules":returnList, "listType":lst.title()})


    # /admin/editInList
```

Example 11

```python
def listEdit(self):
    old = request.args.get('oldCPE')
    new = request.args.get('cpe')
    lst = request.args.get('list')
    CPEType = request.args.get('type')
    if old and new:
      result = wl.updateWhitelist(old, new, CPEType) if lst=="whitelist" else bl.u
      status = "cpelist_updated" if (result) else "cpelist_update_failed"
    else:
      status = "invalid_cpe"
    returnList = list(db.getWhitelist()) if lst=="whitelist" else list(db.getBlack
    return jsonify({"rules":returnList, "status":status, "listType":lst})


  # /admin/listmanagement/<vendor>/<product>
  # /admin/listmanagement/<vendor>
  # /admin/listmanagement
```

Example 12

```python
def check_request(request):

    headers = dict(request.headers)
    if 'Orcid-Authorization' not in headers:
        raise Exception('Header Orcid-Authorization is missing')
    h = {
        'Accept': 'application/json',
        'Authorization': headers['Orcid-Authorization'],
        'Content-Type': 'application/json'
        }
    # transfer headers from the original
    #for x in ['Content-Type']:
    #    if x in headers:
    #        h[x] = headers[x]

    if 'Content-Type' in headers \
        and 'application/json' in headers['Content-Type'] \
        and request.method in ('POST', 'PUT'):
        payload = request.json
    else:
        payload = dict(request.args)
        payload.update(dict(request.form))

    return (payload, h)
```

Example 13

```python
def index():

    # Displays the shown string above the user entered text
    header_review = "Review:"
```

```python
        # Displays the show string above the model determined sentiment
        header_sentiment = "Sentiment:"

    print(request. args )

        # Contains a dictionary containing the parsed contents of the query string
        if(request. args ):

            # Passes contents of query string to the prediction function contained in
            x_input, prediction = predict_sentiment(request. args ['text_in'])
            print(prediction[0]['prob'])

            # Indexes the returned dictionary for the sentiment probability
            if((prediction[0]['prob']) > 0.5):
                prediction = "Positive"
                return flask.render_template('index.html', text_in=x_input, prediction
            else:
                prediction = "Negative"
                return flask.render_template('index.html', text_in=x_input, prediction

        # If the parsed query string does not contain anything then return index page
        else:
            return flask.render_template('index.html')
```

Example 14

```python
def response_text_basic_rewrite(* args , **kwargs):   # coverage: exclude
    """本函数在v0.28.3被移除，对本函数的调用会被映射出去
    如果需要查看本函数代码，请查看git历史到 v0.28.3 以前
    """
    from warnings import warn
    warn("This function is deprecated since v0.28.3, use response_text_basic_mirro
    return response_text_basic_mirrorlization(* args , **kwargs)
```

Example 15

```python
def authorized(self, allowed_roles, resource, method):
        authorized = False

        if request.authorization:
            auth = request.authorization
            authorized = self.check_auth(auth.username, auth.password,
                                         allowed_roles, resource, method)
        else:
            try:
                access_token = request. args ['access_token']
            except KeyError:
                access_token = request.headers.get('Authorization', '').partition(
            authorized = self.check_token(access_token, allowed_roles, resource, m

        return authorized
```

Example 16

```python
def authenticate(self):
    """
    Indicate to the client that it needs to authenticate via a 401.
    """
    if request.headers.get('Authorization') or request.args.get('access_token
        realm = 'Bearer realm="%s", error="invalid_token"' % __package__
    else:
        realm = 'Bearer realm="%s"' % __package__
    resp = Response(None, 401, {'WWW-Authenticate': realm})
    abort(401, description='Please provide proper credentials', response=resp)
```

Example 17

```python
def requires_token(self, audiences=None, allowed_roles=None):
    """
    Decorator for functions that will be protected with token authentication.

    Token must be provvided either through access_token parameter or Authoriza
    header.

    See check_token() method for further details.
    """
    def requires_token_wrapper(f):
        @wraps(f)
        def decorated(*args, **kwargs):
            try:
                token = request.args['access_token']
            except KeyError:
                token = request.headers.get('Authorization', '').partition(' '

            if not self._perform_verification(token, audiences, allowed_roles)
                abort(401)

            return f(*args, **kwargs)
        return decorated
    return requires_token_wrapper
```

Example 18

```python
def receivesms():
    '''Handles and processes all messages coming from Kannel and going towards the

    NOTE: See the enclosed sample configuration file in kannel/ for knowing wha
    and the name of the arguments
    '''
    try: #TODO: Better exception handling!
        app.logger.debug("Received data %s", request.args)
        #TODO: Support GET as well as POST requests equally well

        msg = {}
        msg['from'] = request.args['from']
        msg['text'] = request.args['text']
        msg['args'] = request.args

        #get the ip address of the kannel server so that we can identify it and us
        #if request.remote_addr
        msg['host']    = get_kannel_server(request)
```

```
        app.logger.debug("Identified! This message came from %s Kannel server", ms

        if msg['host'] is False: #if we can't get the IP of the origin of request,
            raise Exception("Cannot retrieve IP from the request to recognize the

        send_to_rapidpro.apply_async(kwargs = {'msg': msg}, serializer = 'json')
        #we will NOT return any text because whatever is returned will be sent as
        #we return in the format (response, status, headers) so that Kannel knows
        return ('',200,[])

    except Exception as e:
        #TODO: Send an email when unrecoverable exceptions occur, instead of just
        app.logger.debug("Exception %s occurred", e)
        raise e
```

Example 19

Project: *pnp*   Author: *HazardDede*   File: *http.py*   MIT License   5 vo

```
def _create_app(self):
        that = self
        flask = load_optional_module('flask', self.EXTRA)
        app = flask.Flask(__name__)

        if self.server_impl == 'flask':
            # We need to register a shutdown endpoint, to end the serving if using
            # development server
            @app.route('/_shutdown', methods=['DELETE'])
            def shutdown():  # pylint: disable=unused-variable
                from flask import request
                func = request.environ.get('werkzeug.server.shutdown')
                if func is None:
                    raise RuntimeError('Not running with the Werkzeug Server')  #
                func()
                return json.dumps({'success': True}), 200, {'ContentType': 'applic

        @app.route('/', defaults={'path': '/'}, methods=self.allowed_methods)
        @app.route('/<path:path>', methods=self.allowed_methods)
        def catch_all(path):  # pylint: disable=unused-variable
            from flask import request
            data = request.get_json(force=True, silent=True)
            if data is None:  # No valid json in request body > fallback to data
                data = request.data if request.data != b'' else None

            payload = dict(
                endpoint=path,
                levels=["/"] if path == "/" else path.split('/'),
                method=request.method,
                query=self._flatten_query_args(dict(request.args)),
                data=data,
                is_json=isinstance(data, dict),
                url=request.url,
                full_path=request.full_path,
                path=request.path
            )
            that.notify(payload)

            return json.dumps({'success': True}), 200, {'ContentType': 'applicatio

        return app
```

Example 20

```python
def _flatten_query_args(args):
    """Iterates through query args and transforms any one-element lists to si

    Examples:

        >>> Server._flatten_query_args({'key': 'value'})  # Return as is
        {'key': 'value'}
        >>> Server._flatten_query_args({'key': ['value']})  # One item list ->
        {'key': 'value'}
        >>> # multiple items list -> no flatten
        >>> Server._flatten_query_args({'key': ['value1', 'value2']})
        {'key': ['value1', 'value2']}
        >>> Server._flatten_query_args({'key': ['']})  # Empty string -> None
        {'key': None}
        >>> Server._flatten_query_args({'key': []})  # Empty list -> None
        {'key': None}
        >>> # Multiple Empty string -> Multiple None's
        >>> Server._flatten_query_args({'key': ['', '']})
        {'key': [None, None]}
        >>> Server._flatten_query_args("notadict")  # Argument has to be a dic
        Traceback (most recent call last):
        ...
        TypeError: Argument 'args' is expected to be a (<class 'dict'>,), but

    """
    def _make_flat(item):
        if not item:
            return None
        if not isinstance(item, list):
            return item
        # item -> list
        if len(item) == 1:
            return item[0] if item[0] else None  # Empty string -> None
        return [x if x else None for x in item]

    Validator.is_instance(dict, args=args)
    res = dict()
    for key, val in args.items():
        res[key] = _make_flat(val)
    return res
```

Example 21

```python
def do_tag():
    f = retrieve(request.args['plugin'])
    if f is not None:
        project_id = request.args['project_id']
        f.do_tag(project_id)
    return 'ok', 200
```

Example 22

```python
def upload_failed(session):
    case_id = request.args.get('case_id')
    business_party_id = request.args['business_party_id']
    survey_short_name = request.args['survey_short_name']
    party_id = session['party_id']
    error_info = request.args.get('error_info', None)

    case_data = case_controller.get_case_data(case_id, party_id, business_party_id

    # Select correct error text depending on error_info
    if error_info == "type":
        error_info = {'header': "Error uploading - incorrect file type",
                      'body': 'The spreadsheet must be in .xls or .xlsx format'}
    elif error_info == "charLimit":
        error_info = {'header': "Error uploading - file name too long",
                      'body': 'The file name of your spreadsheet must be '
                              'less than 50 characters long'}
    elif error_info == "size":
        error_info = {'header': "Error uploading - file size too large",
                      'body': 'The spreadsheet must be smaller than 20MB in size'}
    else:
        error_info = {'header': "Something went wrong",
                      'body': 'Please try uploading your spreadsheet again'}

    return render_template('surveys/surveys-upload-failure.html', business_info=ca
                           collection_exercise_info=case_data['collection_exercise
```

Example 23

```python
def callback(self):
        if 'code' not in request.args:
            return (None, None)
        # The following two commands pass **kwargs to requests.
        oauth_session = self.service.get_auth_session(
                data={'code': request.args['code'],
                      'grant_type': 'authorization_code',
                      'redirect_uri': self.get_callback_url()
                     },
                decoder = json.loads
        )
        me = oauth_session.get('').json()
        return (me['name'],
                me['email'])
```

Example 24

```python
def change():
    connection = connect.AWS()
    connection.session()
    identity_vault_client = connection.identity_vault_client()

    user_profile = request.get_json(silent=True)
    if isinstance(user_profile, str):
        user_profile = json.loads(user_profile)

    user_id = request.args.get("user_id", user_profile["user_id"]["value"])
    logger.info("A json payload was received for user: {}".format(user_id), extra=
```

```python
    vault = profile.Vault(sequence_number=None, profile_json=user_profile, **reque

    if request.method in ["POST", "PUT", "GET"]:
        vault.identity_vault_client = identity_vault_client
        result = vault.put_profile(user_profile)
        logger.info(
            "The result of publishing for user: {} is: {}".format(user_id, result)
            extra={"user_id": user_id, "result": result},
        )
    if config("allow_delete", namespace="cis", default="false") == "true":
        if request.method in ["DELETE"]:
            vault.identity_vault_client = identity_vault_client
            result = vault.delete_profile(user_profile)
            logger.info(
                "A delete operation was performed for user: {}".format(user_id),
                extra={"user_id": user_id, "result": result},
            )
    return jsonify(result)
```

Example 25

```python
def status():
    sequence_number = request.args.get("sequenceNumber")
    status = profile.Status(sequence_number)
    result = status.all
    return jsonify(result)
```

Example 26

```python
def get_bool_arg(name, args):
    value = args.get(name, 'false').lower()
    return value not in ('f', 'false')
```

Example 27

```python
def render_game(bucket, model_name, data, filename="",
                force_full=False, render_sorry=False):
    is_raw = get_bool_arg('raw', request.args)
    if is_raw:
        if request.args.get('raw', '') == 'sgf':
            return Response(data, mimetype='application/x-go-sgf')
        return sgf_utils.pretty_print_sgf(data)

    # 3200 > 500 * 'B[aa];'
    player_evals = []
    if len(data) > 3200:
        try:
            # NOTE: evals are printed ~near~ the move they are for but plus or
            # minus one because of 2*m+1 below.
            _, comments = sgf_utils.raw_game_data(filename, data)
            evals = [comment[2][0] for comment in comments]
            for m, (b_eval, w_eval) in enumerate(zip(evals[::2], evals[1::2])):
                player_evals.append((2 * m + 1, b_eval, w_eval))
        except Exception as e:
```

```
                print("Failed to eval parse:", bucket, model_name)
                print(e)
                pass

        return render_template(
            'game.html',
            bucket=bucket,
            model=model_name,
            data=data,
            player_evals=player_evals,
            filename=filename,
            force_full=force_full or len(player_evals) > 0,
            render_sorry=False,
        )
```

Example 28

```
def __init__(self, controller: AuthenticationController):
        self.blueprint = Blueprint('auth_blueprint', __name__)

        @self.blueprint.route('/api/login')
        def step1_login():
            """
            STEP 1: User navigates to /login. Redirect them to the authentication
            """
            redirect_url, state = controller.login()
            session[SESSION_STATE] = state
            return redirect(redirect_url)

        @self.blueprint.route('/api/authorization-code')
        def step2_authorization_code():
            """
            STEP 2: User went to the authentication backend, entered their credent
            redirected back to this endpoint with some data in args.

            We now have a temporary 'authorization_token' that we can use to fetch
            In order to get this new token, we must call the the auth backend (CAS
            """
            args = request.args
            token, err = controller.get_tokens(
                state=args.get('state'),
                authorization_code=args.get('code'),
                stored_state=session.get(SESSION_STATE),
            )
            if token is None:
                return err, 400

            del session[SESSION_STATE]  # Prevent state reuse.
            session[SESSION_TOKEN] = token

            return str(token)
```

Example 29

```
def get_current_location(request):
    lat = float(request.args['lat'])
```

```
        lng = float(request.args['lng'])
        return np.array([lat, lng])
```

Example 30

```
def hello():
    current_location = get_current_location(request)
    time = request.args['time']
    current_station = get_current_station(current_location)
    return json.dumps(get_response_dict(current_station, time), ensure_ascii=False
```

Example 31

```
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 32

```python
def show_jobs():
    '''获取所有jobs信息'''
    response = {}
    try:
        jid = request.args.get('id')
        if jid == None:
            ret_list = scheduler.get_jobs()
        else:
            ret_list = [scheduler.get_job(jid)]
        inof_list = []
        for ret in ret_list:
            fields = ret.trigger.fields
            cron = {}
            for field in fields:
                cron[field.name] = str(field)
            cron_list = [cron['second'],cron['minute'],cron['hour'],cron['day'],cr
            info = {
                'id':ret.id,
                'next_run_time':ret.next_run_time,
                'cmd':ret.kwargs.get('cmd'),
                #'func':ret.func_ref,
                'status':'running' if ret.next_run_time != None else 'stop',
                'cron':' '.join(cron_list)
            }
            inof_list.append(info)
        response['status'] = 0
        response['data'] = inof_list
        response['count'] = len(inof_list)
    except Exception as e:
        response['msg'] = str(e)
    return json.dumps(response,cls=DateEncoder)
```

Example 33

```python
def job_log():
    '''获取所有job log信息'''
    response = {}
    try:
        ret = get_job_logs(request.args)
        response['status'] = 0
        response['data'] = ret
        response['count'] = len(ret)
    except Exception as e:
        response['msg'] = str(e)
    return json.dumps(response,cls=DateEncoder)
```

Example 34

```python
def movie():
    form = SearchCriteria()
    if form.validate_on_submit():
        search = str(form.search.data)
        films = query.returnFilm(search)
        if len(films) == 0:
            msg = "No results found for %s" % (search)
```

```
            return render_template("main.html", form=form)
        return render_template("main.html", films=films, form=form)

    movieid = int(request.args["id"])
    film = query.returnOneFilm(movieid)
    cast = query.returnCast(movieid)
    crew = query.returnCrew(movieid)

    ratings = query.returnRatings(movieid)
    try:
        rating = round(ratings[0]["rating"])
    except:
        rating = 0
    stream = streaming(film[0]["title"])
    rent = stream["rent"]
    buy = stream["buy"]

    '''
    like = Liked()
    if like.validate_on_submit():
        query.insert(userid, movieid, "liked")

    return render_template("movie.html", like = like, rent = rent, buy = buy, form
    '''

    return render_template("movie.html", rent=rent, buy=buy, form=form, film=film,
```

Example 35

```
def repos_for_keyword():
    """
  /repos?keyword=<keyword>

  Finds all repos which contain the given keyword in the name, readme, or descript
    if "keyword" not in request.args:
        return "", 400

    keyword = request.args["keyword"]
    future = _repos_for_keyword(keyword)
    repos = loop.run_until_complete(future)
    return jsonify(repos)
```

Example 36

```
def users_for_repo(user, repo_name):
    """
  /users/<user>/repo/<repo_name>[?oldest-age=<age in weeks>]

  Returns list of users who have commited in the resource user/repo in the last gi
  weeks """

    oldest_age = (
        55 if "oldest-age" not in request.args else request.args["oldest-age"]
    )
    future = _users_for_repo(user, repo_name, oldest_age=oldest_age)
    users = loop.run_until_complete(future)
    return jsonify(users)
```

Example 37

```python
def users_for_keyword():
    """
    /users?keyword=<keyword>[?oldest-age=<age in weeks>]

    Find the top users who have commited in repositories matching the keyword in the
    if "keyword" not in request.args:
        return "", 400

    keyword = request.args["keyword"]
    oldest_age = (
        55 if "oldest-age" not in request.args else request.args["oldest-age"]
    )

    repos_future = _repos_for_keyword(keyword)
    repos = loop.run_until_complete(repos_future)

    # gather futures for getting users from each repo
    users_futures = []
    users = set()
    for repo in repos:
        user, repo_name = repo.split("/")
        users_futures.append(
            _users_for_repo(user, repo_name, oldest_age=oldest_age)
        )

    # barrier on all the users futures
    users_results = loop.run_until_complete(asyncio.wait(users_futures))

    # gather the results
    for users_result in users_results:
        for task in users_result:
            if task.result():
                users.update(set(task.result()))

    return jsonify(list(users))
```

Example 38

```python
def geo():
    if not request.args:
        return no_args_waring
    else:
        ip = request.args.get("ip")

        ret_dict = {
            "stats": "Fail",
            "ip": ip,
            "loc": "Not Find IP address." if ip is None else None
        }

        ret_dict.update(ip_query.searchIp(ip))

        return jsonify(ret_dict)
```

Example 39

```python
def get_environment_varable():
    if 'var' not in request.args:
        return "Required param 'var' is missing", 400

    var = request.args['var']
    if var not in os.environ:
        return "Not found '{0}' in environment variables".format(var), 404
    return str(os.environ[var])
```

Example 40

```python
def proxy():
    url = request.args.get('url')
    link = request.args.get('link')
    port = request.args.get('port')
    path = request.args.get('path')

    if link is not None and port is not None and path is not None:
        link = link.upper()
        dest_port = os.environ.get(link + "_PORT_" + port + "_TCP_PORT")
        dest_host = os.environ.get(link + "_PORT_" + port + "_TCP_ADDR")
        err_msg = "Not found '{0}' in environment variables"
        if dest_port is None:
            return err_msg.format(dest_port), 404
        if dest_host is None:
            return err_msg.format(dest_host), 404
        url = 'http://{0}:{1}/{2}'.format(dest_host, dest_port, path)

    if url is None:
        return ("Required param missing: Either 'url', or all params "
                "'link', 'port' and 'path' are required"), 400
    try:
        response = requests.get(url=url)
    except Exception as e:
        return "Error: {0}".format(e), 400
    if not response.ok:
        return response.content, response.status_code
    return response.content, 200
```

Example 41

```python
def authorized(self):
        resp = self.auth.authorized_response()
        if resp is None or resp.get('access_token') is None:
            return 'Access denied: reason=%s error=%s resp=%s' % (
                request.args['error'],
                request.args['error_description'],
                resp
            )
        response = redirect(self.internal_url_for('/'))
        response.set_cookie('token', resp['access_token'])
        return response
```

Example 42

```python
def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth = request.authorization
        if not auth or not check_auth(auth.username, auth.password):
            if _globals.get('test'):
                return f(*args, **kwargs)
            return authenticate()
        return f(*args, **kwargs)
    return decorated
```

Example 43

```python
def get_file():
  """Get original file."""
  filename = document.secure_path(request.args['filename'])
  ext = document.secure_path(request.args['ext'])
  mediatype = request.args['mediatype']

  root = os.path.join(_globals['gransk'].config[helper.DATA_ROOT], 'files')
  file_path = os.path.join(root, ext, filename)

  if not os.path.exists(file_path):
    abort(404)

  with open(file_path, 'rb') as inp:
    return Response(inp.read(), mimetype=mediatype, status=200)
```

Example 44

```python
def search():
  query = json.loads(request.args['q'])
  if 'type' in query:
    url = 'http://%s:9200/gransk/%s/_search?' % (_globals['config']['es_host'][0],
  else:
    url = 'http://%s:9200/gransk/_search' % _globals['config']['es_host'][0]

  r = requests.get(url, data=json.dumps(query['body']))

  return Response(r.text, status=200, mimetype='application/json')
```

Example 45

```python
def picture():
  """Get document content as picture."""
  name = document.secure_path(request.args['name'])
  mediatype = request.args['mediatype']

  root = os.path.join(_globals['gransk'].config[helper.DATA_ROOT], 'pictures')
  image_path = os.path.join(root, name)

  if not os.path.exists(image_path):
```

```
        abort(404)

    with open(image_path, 'rb') as fp:
        return Response(fp.read(), mimetype=mediatype, status=200)
```

Example 46

```
def filter_logic(self, filters, skip, limit=None):
    query = self.generate_full_query(filters)
    limit = limit if limit else self.args['pageLength']
    cve   = db.getCVEs(limit=limit, skip=skip, query=query)
    # marking relevant records
    if current_user.is_authenticated():
        if filters['whitelistSelect'] == "on":   cve = self.list_mark('white', cve
        if filters['blacklistSelect'] == "mark": cve = self.list_mark('black', cve
    self.plugManager.mark(cve, **self.pluginArgs)
    cve = list(cve)
    return cve
```

Example 47

```
def _get_cve_actions(self):
    cve = request.args.get('cve', type=str)
    if not current_user.is_authenticated(): # Don't show actions requiring auth if
        actions = [x for x in self.plugManager.getCVEActions(cve, **self.pluginArgs)
    else:
        actions = self.plugManager.getCVEActions(cve, **self.pluginArgs)
    return jsonify({"actions": actions})


    # /plugin/<plugin>
```

Example 48

```
def openPlugin(self, plugin):
    if self.plugManager.requiresAuth(plugin) and not current_user.is_authenticated
        return render_template("requiresAuth.html")
    else:
        page, args = self.plugManager.openPage(plugin, **self.pluginArgs)
        if page:
            try:
                return render_template(page, ** args)
            except jinja2.exceptions.TemplateSyntaxError: return render_template("errc
            except jinja2.exceptions.TemplateNotFound:     return render_template("errc
        else: abort(404)


    # /plugin/<plugin>/subpage/<page>
```

Example 49

```
def openPluginSubpage(self, plugin, page):
    if self.plugManager.requiresAuth(plugin) and not current_user.is_authenticated
      return render_template("requiresAuth.html")
    else:
      page, args = self.plugManager.openSubpage(plugin, page, **self.pluginArgs)
      if page:
        try:
          return render_template(page, ** args)
        except jinja2.exceptions.TemplateSyntaxError: return render_template("erro
        except jinja2.exceptions.TemplateNotFound:    return render_template("erro
      else: abort(404)


  # /plugin/<plugin>/_cve_action/<action>
```

Example 50

```
def change_pass(self):
    current_pass = request.args.get('current_pass')
    new_pass     = request.args.get('new_pass')
    if current_user.authenticate(current_pass):
      if new_pass:
        db.changePassword(current_user.id , new_pass)
        return jsonify({"status": "password_changed"})
      return jsonify({"status": "no_password"})
    else:
      return jsonify({"status": "wrong_user_pass"})

  # /admin/request_token
```