Python flask.request.view_args() Examples

The following are code examples for showing how to use *flask.request.view_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

Project: *python-flask-restful-api* Author: *akashtalole* File: *event_locations.py* MIT License 6 vo

```
def get(self, *args, **kwargs):
        qs = QSManager(request.args, self.schema)
        popular_locations = db.session.query(Event.searchable_location_name, func.
            .group_by(Event.searchable_location_name) \
            .order_by(desc('counts')) \
            .limit(6)
        locations = []
        for location, _ in popular_locations:
            if location is not None:
                new_location = EventLocation(location)
                new_location.id = len(locations)
                locations.append(new_location)
        schema = EventLocationSchema()
        result = schema.dump(locations, many=True).data
        view_kwargs = request.view_args if getattr(self, 'view_kwargs', None) is
        add_pagination_links(result,
                             len(locations),
                             qs,
                             url_for(self.view, **view_kwargs))
        result.update({'meta': {'count': len(locations)}})
        return result
```

Example 2

Project: *cookiecutter-flask-restful* Author: *karec* File: *pagination.py* MIT License 6 vo

```
def paginate(query, schema):
    page = request.args.get('page', DEFAULT_PAGE_NUMBER)
    per_page = request.args.get('page_size', DEFAULT_PAGE_SIZE)
    page_obj = query.paginate(page=page, per_page=per_page)
    next = url_for(
        request.endpoint,
        page=page_obj.next_num if page_obj.has_next else page_obj.page,
        per_page=per_page,
        **request.view_args
    )
    prev = url_for(
        request.endpoint,
        page=page_obj.prev_num if page_obj.has_prev else page_obj.page,
        per_page=per_page,
        **request.view_args
    )

    return {
        'total': page_obj.total,
        'pages': page_obj.pages,
        'next': next,
        'prev': prev,
```

```
        'results': schema.dump(page_obj.items).data
    }
```

Example 3

Project: *osm-wikidata* Author: *EdwardBetts* File: *__init__.py* GNU General Public License v3.0 6 vo

```
def log_exception(self, exc_info):
        self.logger.error("""
Path:                 %s
HTTP Method:          %s
Client IP Address:    %s
User Agent:           %s
User Platform:        %s
User Browser:         %s
User Browser Version: %s
GET args:             %s
view args:            %s
URL:                  %s
""" % (
            request.path,
            request.method,
            request.remote_addr,
            request.user_agent.string,
            request.user_agent.platform,
            request.user_agent.browser,
            request.user_agent.version,
            dict(request.args),
            request.view_args,
            request.url
        ), exc_info=exc_info)
```

Example 4

Project: *flump* Author: *rolepoint* File: *__init__.py* MIT License 6 vo

```
def __init__(self, *args, **kwargs):
        super(FlumpBlueprint, self).__init__(*args, **kwargs)

        register_error_handlers(self)

        if kwargs.pop('logging', False):
            @self.before_request
            def do_logging():
                logger = logging.getLogger('flump.view.{}'.format(self.name))

                debug_string = (
                    "%s request made for resource type %s with kwargs: %s "
                    "and data: %s"
                )

                logger.debug(debug_string, request.method, request.view_args,
                             request.data)
```

Example 5

Project: *eq-survey-runner* Author: *ONSdigital* File: *questionnaire.py* MIT License 6 vo

```
def before_questionnaire_request():
    metadata = get_metadata(current_user)
```

```
    if not metadata:
        raise NoTokenException(401)

    logger.bind(tx_id=metadata['tx_id'])

    values = request.view_args

    if check_multiple_survey(metadata, values):
        raise MultipleSurveyError

    logger.bind(eq_id=values['eq_id'], form_type=values['form_type'],
                ce_id=values['collection_id'])
    logger.info('questionnaire request', method=request.method, url_path=request.f

    session_store = get_session_store()
    session_data = session_store.session_data

    language_code = request.args.get('language_code')
    if language_code:
        session_data.language_code = language_code
        session_store.save()

    g.schema = load_schema_from_session_data(session_data)
```

Example 6

Project: *eq-survey-runner* Author: *ONSdigital* File: *questionnaire.py* MIT License 6 vo

```
def before_post_submission_request():
    session_store = get_session_store()
    if not session_store or not session_store.session_data:
        raise NoTokenException(401)

    session_data = session_store.session_data
    g.schema = load_schema_from_session_data(session_data)

    logger.bind(tx_id=session_data.tx_id)

    values = request.view_args
    logger.bind(eq_id=values['eq_id'], form_type=values['form_type'])
    logger.info('questionnaire request', method=request.method, url_path=request.f

    metadata_from_session_data = {
        'tx_id': session_data.tx_id,
        'eq_id': session_data.eq_id,
        'form_type': session_data.form_type,
    }
    if check_multiple_survey(metadata_from_session_data, values):
        raise NoTokenException(401)
```

Example 7

Project: *notifications-admin* Author: *alphagov* File: *test_permissions.py* MIT License 6 vo

```
def _test_permissions(
    client,
    usr,
    permissions,
    will_succeed,
    kwargs=None,
):
```

```
    request.view_args.update({'service_id': 'foo'})
    if usr:
        client.login(usr)

    decorator = user_has_permissions(*permissions, **(kwargs or {}))
    decorated_index = decorator(index)

    if will_succeed:
        decorated_index()
    else:
        try:
            if (
                decorated_index().location != '/sign-in?next=%2F' or
                decorated_index().status_code != 302
            ):
                pytest.fail("Failed to throw a forbidden or unauthorised exceptior
        except (Forbidden, Unauthorized):
            pass
```

Example 8

Project: *notifications-admin* Author: *alphagov* File: *test_permissions.py* MIT License 6 vo

```
def test_user_has_permissions_for_organisation(
    client,
    mocker,
):
    user = _user_with_permissions()
    user['organisations'] = ['org_1', 'org_2']
    mocker.patch('app.user_api_client.get_user', return_value=user)
    client.login(user)

    request.view_args = {'org_id': 'org_2'}

    @user_has_permissions()
    def index():
        pass

    index()
```

Example 9

Project: *notifications-admin* Author: *alphagov* File: *test_permissions.py* MIT License 6 vo

```
def test_platform_admin_can_see_orgs_they_dont_have(
    client,
    platform_admin_user,
    mocker,
):
    platform_admin_user['organisations'] = []
    mocker.patch('app.user_api_client.get_user', return_value=platform_admin_user)
    client.login(platform_admin_user)

    request.view_args = {'org_id': 'org_2'}

    @user_has_permissions()
    def index():
        pass

    index()
```

Example 10

Project: *notifications-admin* Author: *alphagov* File: *test_permissions.py* MIT License 6 vo

```
def test_cant_use_decorator_without_view_args(
    client,
    platform_admin_user,
    mocker,
):
    mocker.patch('app.user_api_client.get_user', return_value=platform_admin_user)
    client.login(platform_admin_user)

    request.view_args = {}

    @user_has_permissions()
    def index():
        pass

    with pytest.raises(NotImplementedError):
        index()
```

Example 11

Project: *notifications-admin* Author: *alphagov* File: *test_permissions.py* MIT License 6 vo

```
def test_user_doesnt_have_permissions_for_organisation(
    client,
    mocker,
):
    user = _user_with_permissions()
    user['organisations'] = ['org_1', 'org_2']
    mocker.patch('app.user_api_client.get_user', return_value=user)
    client.login(user)

    request.view_args = {'org_id': 'org_3'}

    @user_has_permissions()
    def index():
        pass

    with pytest.raises(Forbidden):
        index()
```

Example 12

Project: *flask-monitor* Author: *fraoustin* File: *main.py* GNU General Public License v2.0 5 vo

```
def _dict(self):
        mydict = {}
        # manage timing
        mydict['timing'] = {}
        mydict['timing']['delta'] = self.timing
        mydict['timing']['start'] = self.request._stats_start_event
        mydict['timing']['asctime'] = asctime(gmtime(self.request._stats_start_eve
        # manage flask
        mydict['flask'] = {}
        mydict['flask']['secret_key'] = current_app.config['SECRET_KEY']
        mydict['flask']['server_name'] = current_app.config['SERVER_NAME']
        mydict['flask']['session_cookie_name'] = current_app.config['SESSION_COOKI
        mydict['flask']['session_cookie_domain'] = current_app.config['SESSION_COO
        mydict['flask']['session_cookie_path'] = current_app.config['SESSION_COOKI
```

```
        mydict['flask']['session_cookie_httponly'] = current_app.config['SESSION_C
        mydict['flask']['session_cookie_secure'] = current_app.config['SESSION_COC
        mydict['flask']['session_refresh_each_request'] = current_app.config['SESS
        # manage request
        mydict['request'] = {}
        mydict['request']['url'] = request.url
        mydict['request']['args'] = {arg: request.args.get(arg) for arg in request
        mydict['request']['view_args'] = request.view_args
        mydict['request']['path'] = request.path
        mydict['request']['method'] = request.method
        mydict['request']['remote_addr'] = request.remote_addr
        try:
            mydict['request']['rule'] = request.url_rule.rule
        except:
            mydict['request']['rule'] = ''
        #manage response
        mydict['response'] = {}
        mydict['response']['status_code'] = self.response.status_code
        mydict['response']['headers'] = { i:j for i,j in self.response.headers}
        return mydict
```

Example 13

Project: *flask_restapi* Author: *iwwxiong* File: *views.py* MIT License 5 vo

```
def get_obj(self):
        args = request.args.to_dict()
        args.update({self.pk_field: request.view_args[self.pk_url_kwarg]})
        self.builder = PeeweeQueryBuilder(self.model, args)
        try:
            obj = self.builder.build().get()
        except self.model.DoesNotExist:
            obj = None
        return obj
```

Example 14

Project: *sqlalchemy-flux-serializer* Author: *alexkuz* File: *flask_metadata_provider.py* MIT License 5 vo

```
def get_metadata(next_model_id):
        if not next_model_id:
            return None

        url_args = {}
        url_args.update(request.view_args)
        url_args.update(request.args)
        url_args['after'] = next_model_id

        next_url = url_for(request.endpoint, **url_args)

        return {
            'next': next_url
        }
```

Example 15

Project: *vanilla* Author: *hamdielhamdi* File: *main.py* MIT License 5 vo

```
def providerdetails(section):
    section = request.view_args['section']
```

```
    with open('descript_data.json') as f:
        meta = json.load(f)
    if section == 'facebook':
        meta = meta['facebook']
    else:
        meta = meta['webpage']
    return render_template('providerdetails.html', title=section, meta=meta)
```

Example 16

Project: *prometheus_flask_exporter* Author: *rycus86* File: *test_metrics.py* MIT License 5 vo

```
def test_gauge(self):
        metrics = self.metrics()

        @self.app.route('/test/1')
        @metrics.gauge('gauge_1', 'Gauge 1')
        def test1():
            self.assertMetric('gauge_1', '1.0')

            return 'OK'

        self.client.get('/test/1')

        self.assertMetric('gauge_1', '0.0')

        @self.app.route('/test/<int:a>')
        @metrics.gauge('gauge_2', 'Gauge 2', labels={
            'uri': lambda: request.path,
            'a_value': lambda: request.view_args['a']
        })
        def test2(a):
            self.assertMetric(
                'gauge_2', '1.0',
                ('uri', '/test/2'), ('a_value', 2)
            )

            return 'OK: %d' % a

        self.client.get('/test/2')

        self.assertMetric(
            'gauge_2', '0.0',
            ('uri', '/test/2'), ('a_value', 2)
        )
```

Example 17

Project: *flask-restalchemy* Author: *ESSS* File: *test_query_callback.py* MIT License 5 vo

```
def sample_property_query(parent_query, model):
    """
    Query callback to get the collection of colleagues but not it self
    :param model:
    :param parent_query:
    :return:
    """
    self_id = request.view_args["relation_id"]
    if parent_query:
        query = parent_query.filter(model.id != self_id)
    else:
```

```
        query = model.query.filter_by(model.id != self_id)
    return query
```

Example 18

Project: *fastapi* Author: *zhangnian* File: *cache.py* MIT License 5 vo

```
def user_cache_key():
    """从path paramaters中获取userid
    """
    return 'cache::user::{}'.format(request.view_args['id'])
```

Example 19

Project: *SuperOcto* Author: *mcecchi* File: *views.py* GNU Affero General Public License v3.0 5 vo

```
def _check_etag_and_lastmodified_for_i18n():
        locale = request.view_args["locale"]
        domain = request.view_args["domain"]

        etag_ok = util.flask.check_etag(_compute_etag_for_i18n(request.view_args[

        lastmodified = _compute_date_for_i18n(locale, domain)
        lastmodified_ok = lastmodified is None or util.flask.check_lastmodified(la

        return etag_ok and lastmodified_ok
```

Example 20

Project: *eq-survey-runner* Author: *ONSdigital* File: *errors.py* MIT License 5 vo

```
def csrf_error(error=None):
    metadata = get_metadata(current_user)
    if metadata and check_multiple_survey(metadata, request.view_args):
        log_exception(error, 200)
        return render_template('multiple_survey.html')

    log_exception(error, 401)
    return render_template('session-expired.html'), 401
```

Example 21

Project: *flask-rest-jsonapi* Author: *miLibris* File: *resource.py* MIT License 5 vo

```
def get(self, *args, **kwargs):
        """Retrieve a collection of objects"""
        self.before_get(args, kwargs)

        qs = QSManager(request.args, self.schema)

        objects_count, objects = self.get_collection(qs, kwargs)

        schema_kwargs = getattr(self, 'get_schema_kwargs', dict())
        schema_kwargs.update({'many': True})

        self.before_marshmallow(args, kwargs)

        schema = compute_schema(self.schema,
                                schema_kwargs,
```

```
                                qs,
                                qs.include)

        result = schema.dump(objects).data

        view_kwargs = request.view_args if getattr(self, 'view_kwargs', None) is
        add_pagination_links(result,
                             objects_count,
                             qs,
                             url_for(self.view, _external=True, **view_kwargs))

        result.update({'meta': {'count': objects_count}})

        final_result = self.after_get(result)

        return final_result
```

Example 22

Project: *fava* Author: *beancount* File: *application.py* MIT License 5 vo

```
def url_for_current(**kwargs):
    """URL for current page with updated request args."""
    if not kwargs:
        return url_for(request.endpoint, **request.view_args)
    args = request.view_args.copy()
    args.update(kwargs)
    return url_for(request.endpoint, **args)
```

Example 23

Project: *fava* Author: *beancount* File: *application.py* MIT License 5 vo

```
def _incognito(response):
    """Replace all numbers with 'X'."""
    if app.config.get("INCOGNITO") and response.content_type.startswith(
        "text/html"
    ):
        is_editor = (
            request.endpoint == "report"
            and request.view_args["report_name"] == "editor"
        )
        if not is_editor:
            original_text = response.get_data(as_text=True)
            response.set_data(replace_numbers(original_text))
    return response
```

Example 24

Project: *flasgger* Author: *flasgger* File: *base.py* MIT License 5 vo

```
def __init__(self, *args, **kwargs):
        view_args = kwargs.pop('view_args', {})
        self.config = view_args.get('config')
        super(APIDocsView, self).__init__(*args, **kwargs)
```

Example 25

Project: *dodotable* Author: *spoqa* File: *flask.py* MIT License 5 vo

```
def build_url(self, **kwargs):
        arg = request.args.copy()
        view_args = request.view_args
        arg.update(view_args)
        for attr in kwargs.keys():
            if attr in arg:
                arg.pop(attr)
        arg.update(kwargs.items())
        rule = request.url_rule
        result = rule.build(arg)
        return result[1]
```

Example 26

Project: *vulncode-db* Author: *google* File: *app_factory.py* Apache License 2.0 5 vo

```
def register_custom_helpers(app):
    def url_for_self(**args):
        return url_for(request.endpoint, **dict(request.view_args, **args))

    app.jinja_env.globals['url_for_self'] = url_for_self
```

Example 27

Project: *scfcli* Author: *tencentyun* File: *local_service.py* Apache License 2.0 5 vo

```
def _generate_api_event(request):
        req_context = {
            'path': PathConverter.convert_path_to_api_gateway(request.endpoint),
            'httpMethod': request.method,
            'requestId': str(uuid.uuid1()),
            'sourceIp': request.remote_addr,
            'stage': 'prod',
        }

        headers = dict(request.headers)

        body = request.get_data()
        if body:
            body = body.decode('utf-8')

        queries = LocalService._generate_query_dict(request)

        event = ApigwEvent(method=request.method,
                           path=request.path,
                           path_paras=request.view_args,
                           body=body,
                           headers=headers,
                           req_context=req_context,
                           queries_dict=queries)

        return event.to_str()
```

Example 28

Project: *tcfcli* Author: *tencentyun* File: *local_service.py* Apache License 2.0 5 vo

```
def _generate_api_event(request):
        req_context = {
            'path': PathConverter.convert_path_to_api_gateway(request.endpoint),
```

```
            'httpMethod': request.method,
            'requestId': str(uuid.uuid1()),
            'sourceIp': request.remote_addr,
            'stage': 'prod',
        }

        headers = dict(request.headers)

        body = request.get_data()
        if body:
            body = body.decode('utf-8')

        queries = LocalService._generate_query_dict(request)

        event = ApigwEvent(method=request.method,
                           path=request.path,
                           path_paras=request.view_args,
                           body=body,
                           headers=headers,
                           req_context=req_context,
                           queries_dict=queries)

        return event.to_str()
```

Example 29

Project: *flask-react-spa* Author: *briancappello* File: *decorators.py* MIT License 5 vo

```
def auth_required_same_user(*args, **kwargs):
    """Decorator for requiring an authenticated user to be the same as the
    user in the URL parameters. By default the user url parameter name to
    lookup is 'id', but this can be customized by passing an argument:

    @auth_require_same_user('user_id')
    @bp.route('/users/<int:user_id>/foo/<int:id>')
    def get(user_id, id):
        # do stuff

    Any keyword arguments are passed along to the @auth_required decorator,
    so roles can also be specified in the same was as it, eg:
    @auth_required_same_user('user_id', role='ROLE_ADMIN')

    Aborts with HTTP 403: Forbidden if the user-check fails
    """
    auth_kwargs = {}
    user_id_parameter_name = 'id'
    if not was_decorated_without_parenthesis(args):
        auth_kwargs = kwargs
        if args and isinstance(args[0], str):
            user_id_parameter_name = args[0]

    def wrapper(fn):
        @wraps(fn)
        @auth_required(**auth_kwargs)
        def decorated(*args, **kwargs):
            try:
                user_id = request.view_args[user_id_parameter_name]
            except KeyError:
                raise KeyError('Unable to find the user lookup parameter '
                               f'{user_id_parameter_name} in the url args')
            if not Permission(UserNeed(user_id)).can():
                abort(HTTPStatus.FORBIDDEN)
```

```
            return fn(*args, **kwargs)
        return decorated

    if was_decorated_without_parenthesis(args):
        return wrapper(args[0])
    return wrapper
```

Example 30

Project: *do-portal* Author: *certeu* File: *__init__.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def api_audit_log(response):
    """Saves information about the request in the ``audit_log``

    :param response: Server :class:`~flask.Response`
    :return: :class:`~flask.Response`
    """
    kwargs = {
        'module': api.name,
        'user': current_user.name,
        'email': current_user.email,
        'action': _HTTP_METHOD_TO_AUDIT_MAP[request.method.lower()],
        'data': addslashes(request.data.decode()),
        'url': request.url,
        'endpoint': request.endpoint,
        'ip': request.remote_addr,
        'status': response.status,
        'timestamp': datetime.datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S')
    }
    if not request.view_args and request.method.lower() == 'put':
        kwargs['action'] = _HTTP_METHOD_TO_AUDIT_MAP['post']
    entry = []
    for k, v in kwargs.items():
        entry.append('{0!s}="{1!s}"'.format(k, v))
    entry = ' '.join(entry)
    current_app.audit_log.info('{0!s}'.format(entry))
    return response
```

Example 31

Project: *do-portal* Author: *certeu* File: *__init__.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def cp_audit_log(response):
    """Saves information about the request in the ``audit_log``

    :param response: Server :class:`~flask.Response`
    :return: :class:`~flask.Response`
    """
    try:
        jdata = json.loads(request.data.decode())
        if 'password' in jdata:
            jdata['password'] = '*********'
        jdata_str = json.dumps(jdata)
    except ValueError:
        jdata_str = ''

    kwargs = {
        'module': cp.name,
        'user': g.user.name,
        'email': g.user.email,
        'action': _HTTP_METHOD_TO_AUDIT_MAP[request.method.lower()],
```

```
        'data': addslashes(jdata_str),
        'url': request.url,
        'endpoint': request.endpoint,
        'ip': request.remote_addr,
        'status': response.status,
        'timestamp': datetime.datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S')
    }
    if not request.view_args and request.method.lower() == 'put':
        kwargs['action'] = _HTTP_METHOD_TO_AUDIT_MAP['post']
    entry = []
    for k, v in kwargs.items():
        entry.append('{0!s}="{1!s}"'.format(k, v))
    entry = ' '.join(entry)
    current_app.audit_log.info('{0!s}'.format(entry))
    return response
```

Example 32

Project: *invenio-deposit* Author: *inveniosoftware* File: *links.py* MIT License 4 vo

```
def deposit_links_factory(pid):
    """Factory for record links generation.

    The dictionary is formed as:

    .. code-block:: python

        {
            'files': '/url/to/files',
            'publish': '/url/to/publish',
            'edit': '/url/to/edit',
            'discard': '/url/to/discard',
            ...
        }

    :param pid: The record PID object.
    :returns: A dictionary that contains all the links.
    """
    links = default_links_factory(pid)

    def _url(name, **kwargs):
        """URL builder."""
        endpoint = '.{0}_{1}'.format(
            current_records_rest.default_endpoint_prefixes[pid.pid_type],
            name,
        )
        return url_for(endpoint, pid_value=pid.pid_value, _external=True,
                       **kwargs)

    links['files'] = _url('files')

    ui_endpoint = current_app.config.get('DEPOSIT_UI_ENDPOINT')
    if ui_endpoint is not None:
        links['html'] = ui_endpoint.format(
            host=request.host,
            scheme=request.scheme,
            pid_value=pid.pid_value,
        )

    deposit_cls = Deposit
    if 'pid_value' in request.view_args:
        deposit_cls = request.view_args['pid_value'].data[1].__class__
```

```
    for action in extract_actions_from_class(deposit_cls):
        links[action] = _url('actions', action=action)
    return links
```

Example 33

Project: *prometheus_flask_exporter* Author: *rycus86* File: *test_metrics.py* MIT License 4 vo

```
def test_histogram(self):
        metrics = self.metrics()

        @self.app.route('/test/1')
        @metrics.histogram('hist_1', 'Histogram 1')
        def test1():
            return 'OK'

        self.client.get('/test/1')

        self.assertMetric('hist_1_count', '1.0')
        self.assertMetric('hist_1_bucket', '1.0', ('le', '2.5'))

        @self.app.route('/test/2')
        @metrics.histogram('hist_2', 'Histogram 2', labels={
            'uri': lambda: request.path,
            'code': lambda r: r.status_code
        })
        def test2():
            return 'OK'

        self.client.get('/test/2')

        self.assertMetric(
            'hist_2_count', '1.0',
            ('uri', '/test/2'), ('code', 200)
        )
        self.assertMetric(
            'hist_2_bucket', '1.0',
            ('le', '1.0'), ('uri', '/test/2'), ('code', 200)
        )

        @self.app.route('/test/<int:x>/<int:y>')
        @metrics.histogram('hist_3', 'Histogram 3', labels={
            'x_value': lambda: request.view_args['x'],
            'y_value': lambda: request.view_args['y']
        }, buckets=(0.7, 2.9))
        def test3(x, y):
            return 'OK: %d/%d' % (x, y)

        self.client.get('/test/3/4')

        self.assertMetric(
            'hist_3_count', '1.0',
            ('x_value', '3'), ('y_value', '4')
        )
        self.assertMetric(
            'hist_3_bucket', '1.0',
            ('le', '0.7'), ('x_value', '3'), ('y_value', '4')
        )
```

Example 34

Project: *puppenc* Author: *Jeoffreybauvin* File: *puppenc.py* Apache License 2.0 4 vo

```
def after_request(response):
        resp_time = PuppencResource.stop_timer(response)

        timestamp = time.strftime('%Y-%b-%d %H:%M')

        if(response.status_code != 200):
            message = str(response.data)
        else:
            message = False

        if not 'user' in g:
            user = False
        else:
            if g.user:
                user = g.user.name
            else:
                user = False

        if request.headers.getlist("X-Forwarded-For"):
            client_ip = request.headers.getlist("X-Forwarded-For")[0]
        else:
            client_ip = request.remote_addr

        log = {
            "timestamp": timestamp,
            "user": user,
            "method": request.method,
            "remote_addr": client_ip,
            "endpoint": request.endpoint,
            "path": request.path,
            "full_path": request.full_path,
            "return_code": response.status_code,
            "view_args": request.view_args,
            "time": float(resp_time),
            "message": message
        }

        if(response.status_code != 200):
            app.logger.warning(log)
        else:
            app.logger.info(log)
        return response
```

Example 35

Project: *flask-unchained* Author: *briancappello* File: *auth_required_same_user.py* MIT License 4 vo

```
def auth_required_same_user(*args, **kwargs):
    """
    Decorator for requiring an authenticated user to be the same as the
    user in the URL parameters. By default the user url parameter name to
    lookup is ``id``, but this can be customized by passing an argument::

        @auth_require_same_user('user_id')
        @bp.route('/users/<int:user_id>/foo/<int:id>')
        def get(user_id, id):
            # do stuff

    Any keyword arguments are passed along to the @auth_required decorator,
    so roles can also be specified in the same was as it, eg::
```

```
        @auth_required_same_user('user_id', role='ROLE_ADMIN')

    Aborts with ``HTTP 403: Forbidden`` if the user-check fails.
    """
    auth_kwargs = {}
    user_id_parameter_name = 'id'
    if not (args and callable(args[0])):
        auth_kwargs = kwargs
        if args and isinstance(args[0], str):
            user_id_parameter_name = args[0]

    def wrapper(fn):
        @wraps(fn)
        @auth_required(**auth_kwargs)
        def decorated(*args, **kwargs):
            try:
                user_id = request.view_args[user_id_parameter_name]
            except KeyError:
                raise KeyError('Unable to find the user lookup parameter '
                               f'{user_id_parameter_name} in the url args')
            if not Permission(UserNeed(user_id)).can():
                abort(HTTPStatus.FORBIDDEN)
            return fn(*args, **kwargs)
        return decorated

    if args and callable(args[0]):
        return wrapper(args[0])
    return wrapper
```

Example 36

Project: *flasgger* Author: *flasgger* File: *base.py* MIT License 4 vo

```
def register_views(self, app):
        """
        Register Flasgger views
        """

        # Wrap the views in an arbitrary number of decorators.
        def wrap_view(view):
            if self.decorators:
                for decorator in self.decorators:
                    view = decorator(view)
            return view

        if self.config.get('swagger_ui', True):
            uiversion = self.config.get('uiversion', 3)
            blueprint = Blueprint(
                self.config.get('endpoint', 'flasgger'),
                __name__,
                url_prefix=self.config.get('url_prefix', None),
                subdomain=self.config.get('subdomain', None),
                template_folder=self.config.get(
                    'template_folder', 'ui{0}/templates'.format(uiversion)
                ),
                static_folder=self.config.get(
                    'static_folder', 'ui{0}/static'.format(uiversion)
                ),
                static_url_path=self.config.get('static_url_path', None)
            )
```

```
            blueprint.add_url_rule(
                self.config.get('specs_route', '/apidocs/'),
                'apidocs',
                view_func=wrap_view(APIDocsView().as_view(
                    'apidocs',
                    view_args=dict(config=self.config)
                ))
            )

            # backwards compatibility with old url style
            blueprint.add_url_rule(
                '/apidocs/index.html',
                view_func=lambda: redirect(url_for('flasgger.apidocs'))
            )
        else:
            blueprint = Blueprint(
                self.config.get('endpoint', 'flasgger'),
                __name__
            )

        for spec in self.config['specs']:
            self.endpoints.append(spec['endpoint'])
            blueprint.add_url_rule(
                spec['route'],
                spec['endpoint'],
                view_func=wrap_view(APISpecsView.as_view(
                    spec['endpoint'],
                    loader=partial(
                        self.get_apispecs, endpoint=spec['endpoint'])
                ))
            )

        app.register_blueprint(blueprint)
```

Example 37

Project: *Publ* Author: *PlaidWeb* File: *rendering.py* MIT License 4 vo

```
def render_exception(error):
    """ Catch-all renderer for the top-level exception handler """

    LOGGER.debug("render_exception %s %s", type(error), error)

    # Effectively strip off the leading '/', so map_template can decide
    # what the actual category is
    category = request.path[1:]

    qsize = index.queue_length()
    if isinstance(error, http_error.NotFound) and qsize:
        retry = max(5, qsize / 5)
        return render_error(
            category, "Site reindex in progress", 503,
            exception={
                'type': 'Service Unavailable',
                'str': "The site's contents are not fully known; please try again
                'qsize': qsize
            },
            headers={
                **NO_CACHE,
                'Retry-After': retry,
                'Refresh': retry
            })
```

```
if isinstance(error, http_error.Unauthorized):
    from flask import current_app as app

    force_ssl = config.auth.get('AUTH_FORCE_HTTPS')
    if force_ssl and request.scheme != 'https':
        return redirect(utils.secure_link(request.endpoint,
                                          **request.view_args,
                                          **request.args))

    flask.g.needs_token = True
    if 'token_error' in flask.g:
        flask.flash(flask.g.token_error)
    return app.authl.render_login_form(destination=utils.redir_path()), 401

if isinstance(error, http_error.HTTPException):
    return render_error(category, error.name, error.code, exception={
        'type': type(error).__name__,
        'str': error.description,
        'args': error.args
    })

return render_error(category, "Exception occurred", 500, exception={
    'type': type(error).__name__,
    'str': str(error),
    'args': error.args
})
```