

Python `flask.request.url_rule()` Examples

The following are code examples for showing how to use `flask.request.url_rule()`. 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: *SayluaLegacy* Author: *saylua* File: [request_hooks.py](#) GNU Affero General Public License v3.0

5 vc

```
def load_user():
    # Make sure not to run function for static files
    if request.url_rule and '/static/' in request.url_rule:
        return

    # Load user
    session_id = request.cookies.get('session_id')
    user_id = request.cookies.get('user_id')

    if session_id and user_id:
        try:
            user_id = int(user_id)
        except ValueError:
            return

        session = db.session.query(LoginSession).get((session_id, user_id))

        if session:
            user = session.user

            if user and session.active:
                g.logged_in = True
                g.user = user

                # Update user's last_action timestamp if it's been at least
                # LAST_ACTION_OFFSET minutes.
                current = datetime.datetime.now()
                mins_ago = current - datetime.timedelta(
                    minutes=app.config['LAST_ACTION_OFFSET'])

                if g.user and g.user.last_action < mins_ago:
                    g.user.last_action = current
                    db.session.commit()

            return

    g.logged_in = False
    g.user = None
    return
```

Example 2

Project: *enjoliver* Author: *JulienBalestra* File: [api.py](#) MIT License

5 vc

```
def submit_lifecycle_ignition(request_raw_query):
    """
    Lifecycle Ignition
    ---
```

```

tags:
    - lifecycle
responses:
    200:
        description: A JSON of the ignition status
        """
try:
    machine_ignition = json.loads(request.get_data())
except ValueError:
    app.logger.error("%s have incorrect content" % request.path)
    return jsonify({"message": "FlaskValueError"}), 406
req = requests.get("%s/ignition?%s" % (EC.matchbox_uri, request_raw_query))
try:
    matchbox_ignition = json.loads(req.content)
    req.close()
except ValueError:
    app.logger.error("%s have incorrect matchbox return" % request.path)
    return jsonify({"message": "MatchboxValueError"}), 406

@smartdb.cockroach_transaction
def op(caller=request.url_rule):
    with SMART.new_session() as session:
        try:
            inject = crud.InjectLifecycle(session, request_raw_query=request_r
            if json.dumps(machine_ignition, sort_keys=True) == json.dumps(matc
                inject.refresh_lifecycle_ignition(True)
                return jsonify({"message": "Up-to-date"}), 200
            else:
                inject.refresh_lifecycle_ignition(False)
                return jsonify({"message": "Outdated"}), 210
        except AttributeError:
            return jsonify({"message": "Unknown"}), 406

return op(caller=request.url_rule)

```

Example 3

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License

5 vc

```

def lifecycle_rolling_delete(request_raw_query):
    """
    Lifecycle Rolling Update
    Disable the current policy for a given machine by UUID or MAC
    ---
    tags:
        - lifecycle
    parameters:
        - name: request_raw_query
          in: path
          description: Pass the mac as 'mac=<mac>'
          required: true
          type: string
    responses:
        200:
            description: Rolling Update is not enable
            schema:
                type: dict
    """
    app.logger.info("%s %s" % (request.method, request.url))

@smartdb.cockroach_transaction
def op(caller=request.url_rule):

```

```

        with SMART.new_session() as session:
            life = crud.InjectLifecycle(session, request_raw_query)
            life.apply_lifecycle_rolling(False, None)
            return jsonify({"enable": False, "request_raw_query": request_raw_query})

    return op(caller=request.url_rule)

```

Example 4

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License

5 vc

```

def report_lifecycle_coreos_install(status, request_raw_query):
    """
    Lifecycle CoreOS Install
    Report the status of a CoreOS install by MAC
    ---
    tags:
      - lifecycle
    responses:
      200:
        description: CoreOS Install report
        schema:
          type: dict
    """
    app.logger.info("%s %s" % (request.method, request.url))
    if status.lower() == "success":
        success = True
    elif status.lower() == "fail":
        success = False
    else:
        app.logger.error("%s %s" % (request.method, request.url))
        return "success or fail != %s" % status.lower(), 403

    @smartdb.cockroach_transaction
    def op(caller=request.url_rule):
        with SMART.new_session() as session:
            inject = crud.InjectLifecycle(session, request_raw_query=request_raw_query)
            inject.refresh_lifecycle_coreos_install(success)

    op(caller=request.url_rule)
    repositories.machine_state.update(
        mac=tools.get_mac_from_raw_query(request_raw_query),
        state=MachineStates.installation_succeed if success else MachineStates.installation_fail)
    return jsonify({"success": success, "request_raw_query": request_raw_query}), 200

```

Example 5

Project: *noobotkit* Author: *nazroll* File: *__init__.py* MIT License

5 vc

```

def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)

```

Example 6

Project: *xuemc* Author: *skycucumber* File: *__init__.py* GNU General Public License v2.0

5 vc

```
def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)
```

Example 7

Project: *scanomatic* Author: *Scan-o-Matic* File: *flask_prometheus.py* GNU General Public License v3.0

5 vc

```
def _request_counting(self, response):
    REQUEST_COUNTER.labels(
        request.url_rule if request.url_rule else request.url,
        response.status_code, request.method
    ).inc()
    return response
```

Example 8

Project: *miniDocker-server* Author: *WhiteVermouth* File: *app.py* MIT License

5 vc

```
def check_token():
    if str(request.url_rule) == "/auth_server":
        return
    else:
        args = request.get_json()
        if args is not None and "token" in args and args["token"] == app.config["T
            return
    return jsonify({
        "warn": "You have no permission"
    })
```

Example 9

Project: *url_shortener* Author: *martydill* File: *__init__.py* MIT License

5 vc

```
def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)
```

Example 10

Project: *tasks* Author: *alcarithemad* File: *task.py* GNU General Public License v3.0

5 vc

```
def before_request():
    route = request.url_rule
```

```
if ('logged' not in session):
    if request.path != '/login' and not re.match('/static/', request.path):
        return redirect(url_for('login'))
```

Example 11

Project: *arxiv-search* Author: *arXiv* File: *context_processors.py* MIT License

5 vc

```
def url_for_page_builder() -> Dict[str, Callable]:
    """Add a page URL builder function to the template context."""
    def url_for_page(page: int, size: int) -> str:
        """Build an URL to for a search result page."""
        rule = request.url_rule
        parts = urlparse(url_for(rule.endpoint)) # type: ignore
        args = request.args.copy()
        args['start'] = (page - 1) * size
        parts = parts._replace(query=urlencode(list(args.items(multi=True))))
        url: str = urlunparse(parts)
        return url
    return dict(url_for_page=url_for_page)
```

Example 12

Project: *arxiv-search* Author: *arXiv* File: *context_processors.py* MIT License

5 vc

```
def current_url_sans_parameters_builder() -> Dict[str, Callable]:
    """Add a function to strip GET parameters from the current URL."""
    def current_url_sans_parameters(*params_to_remove: str) -> str:
        """Get the current URL with ``param`` removed from GET parameters."""
        if request.url_rule is None:
            raise ValueError('No matching URL rule for this request (oddly)')
        rule = request.url_rule
        parts = urlparse(url_for(rule.endpoint))
        args = request.args.copy()
        for param in params_to_remove:
            args.pop(param, None)
        parts = parts._replace(query=urlencode(list(args.items(multi=True))))
        url: str = urlunparse(parts)
        return url
    return dict(current_url_sans_parameters=current_url_sans_parameters)
```

Example 13

Project: *pipa-pay-server* Author: *davidvon* File: *__init__.py* Apache License 2.0

5 vc

```
def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)
```

Example 14

Project: *opentelemetry-python* Author: *open-telemetry* File: *__init__.py* Apache License 2.0

5 vc

```

def _before_flask_request():
    environ = flask_request.environ
    span_name = flask_request.endpoint or otel_wsgi.get_default_span_name(
        environ
    )
    parent_span = propagators.extract(
        otel_wsgi.get_header_from_environ, environ
    )

    tracer = trace.tracer_source().get_tracer(__name__, __version__)

    attributes = otel_wsgi.collect_request_attributes(environ)
    if flask_request.url_rule:
        # For 404 that result from no route found, etc, we don't have a url_rule.
        attributes["http.route"] = flask_request.url_rule.rule
    span = tracer.start_span(
        span_name,
        parent_span,
        kind=trace.SpanKind.SERVER,
        attributes=attributes,
        start_time=environ.get(_ENVIRON_STARTTIME_KEY),
    )
    activation = tracer.use_span(span, end_on_exit=True)
    activation.__enter__()
    environ[_ENVIRON_ACTIVATION_KEY] = activation
    environ[_ENVIRON_SPAN_KEY] = span

```

Example 15

Project: *apm-agent-python* Author: *elastic* File: *__init__.py* [BSD 3-Clause "New" or "Revised"](#)

5 vc

[License](#)

```

def request_started(self, app):
    if not self.app.debug or self.client.config.debug:
        trace_parent = TraceParent.from_headers(request.headers)
        self.client.begin_transaction("request", trace_parent=trace_parent)
        elasticapm.set_context(
            lambda: get_data_from_request(
                request,
                capture_body=self.client.config.capture_body in ("transactions", "all"),
                capture_headers=self.client.config.capture_headers,
            ),
            "request",
        )
        rule = request.url_rule.rule if request.url_rule is not None else ""
        rule = build_name_with_http_method_prefix(rule, request)
        elasticapm.set_transaction_name(rule, override=False)

```

Example 16

Project: *junior_project* Author: *tishq* File: *__init__.py* [MIT License](#)

5 vc

```

def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)

```

Example 17

Project: *junior_project* Author: *tishq* File: *__init__.py* MIT License

5 vc

```
def _has_fr_route(self):
    """Encapsulating the rules for whether the request was to a Flask endpoint
    # 404's, 405's, which might not have a url_rule
    if self._should_use_fr_error_handler():
        return True
    # for all other errors, just check if FR dispatched the route
    if not request.url_rule:
        return False
    return self.owns_endpoint(request.url_rule.endpoint)
```

Example 18

Project: *IntegraTI-API* Author: *discentes-imd* File: *middlewares.py* GNU General Public License v3.0

5 vc

```
def verify_route():
    """
    Verify if the route is in a list of routes that need have the authorization header
    """
    for route in routes:
        if route[0] == str(request.url_rule) and request.method in route[1] and '.
            abort(403, 'Authorization header missing')

# TODO: Implementar a validação dos modelos de entrada
```

Example 19

Project: *IntegraTI-API* Author: *discentes-imd* File: *middlewares.py* GNU General Public License v3.0

5 vc

```
def encrypt_password():
    """
    Verify if the route is for password reset or user create, then encrypts the password
    """
    if request.json is None or not 'password' in request.json:
        return
    if str(request.url_rule) == '/auth/user/' and request.method == 'POST' \
    or str(request.url_rule) == '/auth/user/resetpassword/' and request.method == 'POST':
        request.json['password'] = generate_password_hash(request.json['password'])
```

Example 20

Project: *docker* Author: *getavalon* File: *flask.py* MIT License

5 vc

```
def before_request(self, *args, **kwargs):
    self.last_event_id = None

    if request.url_rule:
        self.client.transaction.push(request.url_rule.rule)

    try:
        self.client.http_context(self.get_http_info(request))
    except Exception as e:
        self.client.logger.exception(to_unicode(e))

    try:
        self.client.user_context(self.get_user_info(request))
```

```
except Exception as e:
    self.client.logger.exception(to_unicode(e))
```

Example 21

Project: *docker* Author: *getavalon* File: *flask.py* MIT License

5 vc

```
def after_request(self, sender, response, *args, **kwargs):
    if self.last_event_id:
        response.headers['X-Sentry-ID'] = self.last_event_id
    self.client.context.clear()
    if request.url_rule:
        self.client.transaction.pop(request.url_rule.rule)
    return response
```

Example 22

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

5 vc

```
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 23

Project: *gitmostwanted.com* Author: *kkamkou* File: *user_oauth.py* MIT License

5 vc

```
def load_user_from_session():
    if str(request.url_rule) in ['/logout']:
        return None
    g.user = User.query.get(session['user_id']) if 'user_id' in session else None
```

@todo #2:15min move after_request method to a general place or a middleware

Example 24

Project: *serverless-ping* Author: *nickromano* File: *flask.py* MIT License

5 vc

```
def before_request(self, *args, **kwargs):
    self.last_event_id = None

    if request.url_rule:
        self.client.transaction.push(request.url_rule.rule)

    try:
        self.client.http_context(self.get_http_info(request))
    except Exception as e:
        self.client.logger.exception(to_unicode(e))
    try:
        self.client.user_context(self.get_user_info(request))
```



```
except Exception as e:
    self.client.logger.exception(to_unicode(e))
```

Example 25

Project: *serverless-ping* Author: *nickromano* File: *flask.py* MIT License

5 vc

```
def after_request(self, sender, response, *args, **kwargs):
    if self.last_event_id:
        response.headers['X-Sentry-ID'] = self.last_event_id
    self.client.context.clear()
    if request.url_rule:
        self.client.transaction.pop(request.url_rule.rule)
    return response
```

Example 26

Project: *python-deepweb* Author: *serfer2* File: *main.py* GNU General Public License v3.0

5 vc

```
def index():
    items = []
    with open('data.json', 'r') as f:
        items = json.loads(f.read())
    print(request.url_rule)
    fuck = False
    if str(request.url_rule).find('/fuck/') != -1:
        fuck = {
            'ip': request.remote_addr,
            'headers': [{'k': k, 'v': v} for k, v in request.headers.items()]
        }
    return render_template('main.html', items=items, fuck=fuck)
```

Example 27

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License

4 vc

```
def change_lifecycle_rolling(request_raw_query):
    """
    Lifecycle Rolling Update
    Change the current policy for a given machine by MAC
    ---
    tags:
      - lifecycle
    parameters:
      - name: request_raw_query
        in: path
        description: Pass the mac as 'mac=<mac>'
        required: true
        type: string
    responses:
      200:
        description: Rolling Update is enable
        schema:
          type: dict
      401:
        description: Mac address is not in database
        schema:
          type: dict
    """
```

```

app.logger.info("%s %s" % (request.method, request.url))
try:
    strategy = json.loads(request.get_data())["strategy"]
    app.logger.info("%s %s rolling strategy: setting to %s" % (request.method,
except (KeyError, ValueError):
    # JSONDecodeError is a subclass of ValueError
    # Cannot use JSONDecodeError because the import is not consistent between
    app.logger.info("%s %s rolling strategy: setting default to kexec" % (request.method,
    strategy = "kexec"

@smartdb.cockroach_transaction
def op(caller=request.url_rule):
    with SMART.new_session() as session:
        try:
            life = crud.InjectLifecycle(session, request_raw_query)
            life.apply_lifecycle_rolling(True, strategy)
            return jsonify({"enable": True, "request_raw_query": request_raw_query})
        except AttributeError:
            return jsonify({"enable": None, "request_raw_query": request_raw_query})

return op(caller=request.url_rule)

```

Example 28

Project: *eclogue* Author: *eclogue* File: *jwt.py* GNU General Public License v3.0

4 v

```

def get_claims():
    try:
        authorization = request.headers.get('Authorization', None)
        if not authorization:
            return 0

        parts = authorization.split()
        if len(parts) < 2 or parts[0] != 'Bearer':
            return 0

        token = parts[1]
        claims = jwt.verify(token)
        url_rule = str(request.url_rule)
        if config.api.get('force_check_binding'):
            found = _force_check_menu_apis(url_rule)
            if not found:
                return -1

        if claims is False:
            return 0

        if claims.get('is_admin'):
            return claims

        method = request.method.lower()
        if url_rule in routes.get('Default'):
            return claims

        username = claims.get('username')
        user_id = claims.get('user_id')
        user = User()
        if not user_id:
            user_info = User.find_one({'username': username})
            user_id = str(user_info['_id'])

        menus, roles = user.get_permissions(user_id)

```

```
if not menus:
    return -1

is_allow = -1
for menu in menus:
    apis = menu.get('apis')
    actions = menu.get('actions')
    if not apis:
        continue

    if url_rule in apis and method in actions:
        is_allow = 1
        break

return claims if is_allow == 1 else is_allow
except JWTErrors:
    return False
```
