Python flask.request._get_current_object() Examples

The following are code examples for showing how to use *flask.request._get_current_object()*. 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: *nplusone* Author: *jmcarp* File: *flask_sqlalchemy.py* MIT License 5 vo

```
def get_worker():
    try:
        return request._get_current_object()
    except RuntimeError:
        return None
```

Example 2

Project: *partycrasher* Author: *naturalness* File: *api_utils.py* GNU General Public License v3.0 5 vo

```
def determine_user_agent_facing_host():
    """
    Determines the host for the active request as seen by the User-Agent
    (client), assuming proxies along the way have been being truthful.
    """

    # Request is a proxy object, and cannot be weakly-referenced; instead,
    # get a reference to true object.
    true_request = request._get_current_object()
    if true_request in HOST_CACHE:
        return HOST_CACHE[true_request]
    else:
        host = calculate_user_agent_facing_host()
        HOST_CACHE[true_request] = host
        return host
```

Example 3

Project: *partycrasher* Author: *naturalness* File: *api_utils.py* GNU General Public License v3.0 5 vo

```
def determine_user_agent_facing_host():
    """
    Determines the host for the active request as seen by the User-Agent
    (client), assuming proxies along the way have been being truthful.
    """

    # Request is a proxy object, and cannot be weakly-referenced; instead,
    # get a reference to true object.
    true_request = request._get_current_object()
    if true_request in HOST_CACHE:
        return HOST_CACHE[true_request]
    else:
        host = calculate_user_agent_facing_host()
        HOST_CACHE[true_request] = host
        return host
```

Example 4

Project: *aws-xray-sdk-python* Author: *aws* File: *middleware.py* Apache License 2.0 5 vo

```
def _before_request(self):
        headers = request.headers
        xray_header = construct_xray_header(headers)
        req = request._get_current_object()

        name = calculate_segment_name(req.host, self._recorder)

        sampling_req = {
            'host': req.host,
            'method': req.method,
            'path': req.path,
            'service': name,
        }
        sampling_decision = calculate_sampling_decision(
            trace_header=xray_header,
            recorder=self._recorder,
            sampling_req=sampling_req,
        )

        if self.in_lambda_ctx:
            segment = self._recorder.begin_subsegment(name)
        else:
            segment = self._recorder.begin_segment(
                name=name,
                traceid=xray_header.root,
                parent_id=xray_header.parent,
                sampling=sampling_decision,
            )

        segment.save_origin_trace_header(xray_header)
        segment.put_http_meta(http.URL, req.base_url)
        segment.put_http_meta(http.METHOD, req.method)
        segment.put_http_meta(http.USER_AGENT, headers.get('User-Agent'))

        client_ip = headers.get('X-Forwarded-For') or headers.get('HTTP_X_FORWARDE
        if client_ip:
            segment.put_http_meta(http.CLIENT_IP, client_ip)
            segment.put_http_meta(http.X_FORWARDED_FOR, True)
        else:
            segment.put_http_meta(http.CLIENT_IP, req.remote_addr)
```

Example 5

Project: *zeus* Author: *getsentry* File: *nplusone.py* Apache License 2.0 5 vo

```
def get_worker():
    try:
        return request._get_current_object()

    except RuntimeError:
        return None
```

Example 6

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

```
def get_session(self):
        ctx = request._get_current_object()
        try:
            session = ctx._current_session
        except AttributeError:
            return None
        else:
            return session
```

Example 7

Project: *laboratoria* Author: *almeidaw* File: *middleware.py* Apache License 2.0 5 vo

```
def _before_request(self):
        headers = request.headers
        xray_header = construct_xray_header(headers)
        req = request._get_current_object()

        name = calculate_segment_name(req.host, self._recorder)

        sampling_req = {
            'host': req.host,
            'method': req.method,
            'path': req.path,
            'service': name,
        }
        sampling_decision = calculate_sampling_decision(
            trace_header=xray_header,
            recorder=self._recorder,
            sampling_req=sampling_req,
        )

        segment = self._recorder.begin_segment(
            name=name,
            traceid=xray_header.root,
            parent_id=xray_header.parent,
            sampling=sampling_decision,
        )

        segment.save_origin_trace_header(xray_header)
        segment.put_http_meta(http.URL, req.base_url)
        segment.put_http_meta(http.METHOD, req.method)
        segment.put_http_meta(http.USER_AGENT, headers.get('User-Agent'))

        client_ip = headers.get('X-Forwarded-For') or headers.get('HTTP_X_FORWARDE
        if client_ip:
            segment.put_http_meta(http.CLIENT_IP, client_ip)
            segment.put_http_meta(http.X_FORWARDED_FOR, True)
        else:
            segment.put_http_meta(http.CLIENT_IP, req.remote_addr)
```

Example 8

Project: *laboratoria* Author: *almeidaw* File: *middleware.py* Apache License 2.0 5 vo

```
def _before_request(self):
        headers = request.headers
        xray_header = construct_xray_header(headers)
        req = request._get_current_object()

        name = calculate_segment_name(req.host, self._recorder)

        sampling_req = {
            'host': req.host,
            'method': req.method,
            'path': req.path,
            'service': name,
        }
        sampling_decision = calculate_sampling_decision(
            trace_header=xray_header,
            recorder=self._recorder,
            sampling_req=sampling_req,
        )

        segment = self._recorder.begin_segment(
            name=name,
            traceid=xray_header.root,
            parent_id=xray_header.parent,
            sampling=sampling_decision,
        )

        segment.save_origin_trace_header(xray_header)
        segment.put_http_meta(http.URL, req.base_url)
        segment.put_http_meta(http.METHOD, req.method)
        segment.put_http_meta(http.USER_AGENT, headers.get('User-Agent'))

        client_ip = headers.get('X-Forwarded-For') or headers.get('HTTP_X_FORWARDE
        if client_ip:
            segment.put_http_meta(http.CLIENT_IP, client_ip)
            segment.put_http_meta(http.X_FORWARDED_FOR, True)
        else:
            segment.put_http_meta(http.CLIENT_IP, req.remote_addr)
```