Python flask.request.scheme() Examples

The following are code examples for showing how to use *flask.request.scheme()*. 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: *invenio-records-rest* Author: *inveniosoftware* File: *links.py* MIT License 6 vo

```
def default_links_factory_with_additional(additional_links):
    """Generate a links generation factory with the specified additional links.

    :param additional_links: A dict of link names to links to be added to the
           returned object.
    :returns: A link generation factory.
    """
    def factory(pid, **kwargs):
        links = default_links_factory(pid)
        for link in additional_links:
            links[link] = additional_links[link].format(pid=pid,
                                                        scheme=request.scheme,
                                                        host=request.host)
        return links

    return factory
```

Example 2

Project: *track-scanner* Author: *skyderby* File: *logging.py* GNU Affero General Public License v3.0 5 vo

```
def after_request(response):
    # This IF avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.
    if response.status_code != 500:
        logger.error(
            '%s %s %s %s %s %s',
            strftime('[%Y-%m-%d %H:%M:%S %z]'),
            request.remote_addr,
            request.method,
            request.scheme,
            request.full_path,
            response.status
        )
    return response
```

Example 3

Project: *shorty* Author: *PadamSethia* File: *app.py* MIT License 5 vo

```
def after_request(response):
        timestamp = strftime('[%Y-%b-%d %H:%M]')
        logger.error('%s %s %s %s %s %s',timestamp , request.remote_addr , \
                                request.method , request.scheme , request.full_pa
        return response
```

Example 4

Project: *shorty* Author: *PadamSethia* File: *app.py* MIT License 5 vo

```
def exceptions(e):
        tb = traceback.format_exc()
        timestamp = strftime('[%Y-%b-%d %H:%M]')
        logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',
        timestamp, request.remote_addr, request.method,
        request.scheme, request.full_path, tb)
        return make_response(e , 405)
```

Example 5

Project: *hellogithub.com* Author: *521xueweihan* File: *__init__.py* GNU Affero General Public License v3.0 5 vo

```
def after_request(response):
    logger.info('%s %s %s %s %s', request.method,
                request.environ.get('HTTP_X_REAL_IP', request.remote_addr),
                request.scheme, request.full_path, response.status)
    return response
```

Example 6

Project: *hellogithub.com* Author: *521xueweihan* File: *__init__.py* GNU Affero General Public License v3.0 5 vo

```
def exceptions(e):
    tb = traceback.format_exc()
    tb = tb.decode('utf-8')
    logger.error('%s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',
                 request.environ.get('HTTP_X_REAL_IP', request.remote_addr),
                 request.method, request.scheme, request.full_path, tb)
    return '500 INTERNAL SERVER ERROR', 500
```

Example 7

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

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

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

Example 8

Project: *ArguminSci* Author: *anlausch* File: *api.py* MIT License 5 vo

```
def after_request(response):
    """ Logging after every request. """
```

```
    # This avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.
    if response.status_code != 500:
        ts = strftime('[%Y-%b-%d %H:%M]')
        logger.error('%s %s %s %s %s %s',
                     ts,
                     request.remote_addr,
                     request.method,
                     request.scheme,
                     request.full_path,
                     response.status)
    return response
```

Example 9

Project: *ArguminSci* Author: *anlausch* File: *api.py* MIT License 5 vo

```
def exceptions(e):
    """ Logging after every Exception. """
    ts = strftime('[%Y-%b-%d %H:%M]')
    tb = traceback.format_exc()
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',
                 ts,
                 request.remote_addr,
                 request.method,
                 request.scheme,
                 request.full_path,
                 tb)
    return "Internal Server Error", 500
```

Example 10

Project: *PyHub* Author: *521xueweihan* File: *__init__.py* MIT License 5 vo

```
def after_request(response):
    logger.info('%s %s %s %s %s', request.remote_addr, request.method,
                request.scheme, request.full_path, response.status)
    return response
```

Example 11

Project: *PyHub* Author: *521xueweihan* File: *__init__.py* MIT License 5 vo

```
def exceptions(e):
    tb = traceback.format_exc()
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR\n%s',
        request.remote_addr, request.method,
        request.scheme, request.full_path, tb)
    return e.status_code
```

Example 12

Project: *flask-gopher* Author: *michael-lazar* File: *flask_gopher.py* GNU General Public License v3.0 5 vo

```
def _add_gopher_error_handler(self, app):
        """
        Intercept all errors for GOPHER requests and replace the default
        HTML error document with a gopher compatible text document.
        """
        def handle_error(error):
```

```
            if request.scheme != 'gopher':
                # Pass through the error to the default handler
                return error

            code = getattr(error, 'code', 500)
            name = getattr(error, 'name', 'Internal Server Error')
            desc = getattr(error, 'description', None)
            if desc is None and self.show_stack_trace:
                desc = traceback.format_exc()
            elif desc is None:
                desc = 'An internal error has occurred'
            body = [menu.error(code, name), '', self.formatter.wrap(desc)]

            # There's no way to know if the client has requested a gopher
            # menu, a text file, or a binary file. But we can make a guess
            # based on if the request path has a file extension at the end.
            ext = os.path.splitext(request.path)[1]
            if ext:
                return '\r\n'.join(body)
            else:
                return self.render_menu(*body)

        # Attach this handler to all of the builtin flask exceptions
        for cls in HTTPException.__subclasses__():
            app.register_error_handler(cls, handle_error)
```

Example 13

Project: *AIOPS_PLATFORM* Author: *kylechenoO* File: *WebApp.py* MIT License 5 vo

```
def after_request(response):
    if response.status_code != 500:
        ts = strftime('[%Y-%b-%d %H:%M]')
        logger.info('%s %s %s %s %s %s',
                    ts,
                    request.remote_addr,
                    request.method,
                    request.scheme,
                    request.full_path,
                    response.status)
    return(response)
```

Example 14

Project: *AIOPS_PLATFORM* Author: *kylechenoO* File: *WebApp.py* MIT License 5 vo

```
def exceptions(e):
    """ Logging after every Exception. """
    ts = strftime('[%Y-%b-%d %H:%M]')
    logger.error('%s %s %s %s %s 5xx INTERNAL SERVER ERROR',
                 ts,
                 request.remote_addr,
                 request.method,
                 request.scheme,
                 request.full_path)
    return("Internal Server Error", 500)
```

Example 15

Project: *dpr-api* Author: *openknowledge-archive* File: *controllers.py* MIT License 5 vo

```
def auth0_login():
    """
    Login through external auth provider
    ---
    tags:
        - auth
    """
    callback_url = request.scheme + '://' + request.headers['Host'] + '/api/auth/
    return app.config['github'].authorize(callback=callback_url)
```

Example 16

Project: *SnowAlert* Author: *snowflakedb* File: *app.py* Apache License 2.0 5 vo

```
def error_handler(ex):
    logger.exception(
        'An error has occurred! ({} {} {} {})'.format(
            request.remote_addr, request.method, request.scheme, request.full_pat
        )
    )
    return 'Internal Server Error', 500
```

Example 17

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: *Dr0p1t_Server.py* MIT License 5 vo

```
def after_request(response):
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    f = open("server.log","a").write( "\n"+"--"*10+"\n"+'%s %s %s %s %s %s'%(times
    return response
```

Example 18

Project: *Dr0p1t-Framework* Author: *Exploit-install* File: *Dr0p1t_Server.py* MIT License 5 vo

```
def exceptions(e):
    tb = traceback.format_exc()
    timestamp = strftime('[%Y-%b-%d %H:%M]')
    f = open("server.log","a").write( "\n"+"--"*10+"\n"+'%s %s %s %s %s 5xx INTERN
    return abort(500)
```

Example 19

Project: *summarize-webpage* Author: *akashp1712* File: *app.py* MIT License 5 vo

```
def after_request(response):
    """ Logging after every request. """
    # This avoids the duplication of registry in the log,
    # since that 500 is already logged via @app.errorhandler.
    if response.status_code != 500:
        ts = strftime('[%Y-%b-%d %H:%M]')
        app.logger.info('%s %s %s %s %s %s',
                        ts,
                        request.remote_addr,
                        request.method,
                        request.scheme,
                        request.full_path,
                        response.status)
    return response
```

Example 20

Project: *summarize-webpage* Author: *akashp1712* File: *app.py* MIT License 5 vo

```
def exceptions(exception):
    """ Logging after every Exception. """
    ts = strftime('[%Y-%b-%d %H:%M]')
    tb = traceback.format_exc()
    app.logger.error('%s %s %s %s %s ERROR:%s \n%s',
                     ts,
                     request.remote_addr,
                     request.method,
                     request.scheme,
                     request.full_path,
                     str(exception),
                     tb)

    return make_response(jsonify({'error': str(exception)}))
```

Example 21

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 22

Project: *flask-gopher* Author: *michael-lazar* File: *flask_gopher.py* GNU General Public License v3.0 4 vo

```
def save_session(self, app, session, response):
        """
        Normally the session is saved by adding a cookie header to the
        response object. However, in this case, because were using a
        query param we need to insert the session into every internal
        link that's returned in the response body. Unfortunately there's
        no easy way to do this, so for now I'm using a regex search
        that looks for gopher internal menu links and appends the _session
        query param to the end of each link selector.
        """
        if not session or response.direct_passthrough:
            # Don't bother trying to save the session if there's nothing to save,
            # or if the response is a static file or streaming file.
            return None

        s = self.get_gopher_signing_serializer(app)
        session_str = s.dumps(dict(session))

        # Build the regex pattern that searches for internal gopher menu links
        host = request.environ['SERVER_NAME']
        port = request.environ['SERVER_PORT']
        url_pattern = '^(?P<type>[^i])(?P<desc>.+)\t(?P<selector>.*)\t%s\t%s\r$'
        url_pattern = url_pattern % (re.escape(host), re.escape(port))

        def on_match(matchobj):
            """
            This function is called on every regex match. It takes an
            existing gopher link, extracts the path and the query string,
            adds the _session param to it, and rebuilds the link.
            """
            url_parts = urlsplit(matchobj.group('selector'))
            query = parse_qs(url_parts.query)
            query['_session'] = [session_str]
            new_query = urlencode(query, doseq=True)
            new_url = urlunsplit([
                url_parts.scheme, url_parts.netloc, url_parts.path,
                new_query, url_parts.fragment])
            new_line = '%s%s\t%s\t%s\t%s\r' % (
                matchobj.group('type'), matchobj.group('desc'), new_url, host, por
            return new_line

        data = bytes.decode(response.data)
        new_data = re.sub(url_pattern, on_match, data, flags=re.M)
        response.data = new_data.encode()
```

Example 23

Project: *trackupdates* Author: *ZhuPeng* File: *server.py* MIT License 4 vo

```
def init_route(self):
        app = self.dash.server
        settings = self.sched.settings

        @app.route('/')
        def index():
            return send_from_directory(os.path.join(dir_path, '../web/dist'), 'inc

        @app.route('/dash/')
        def dash():
            return appdash.app.index()

        @app.route('/<path>')
        def static_file(path):
            return send_from_directory(os.path.join(dir_path, '../web/dist'), patl

        @app.route('/api')
        def api_index():
            base_url = "{}://{}/api".format(request.scheme, request.host)
            results = {}
            for config in settings.get_all_job_configs():
                name = config['name']
                results[name] = {
                    'url': '{}/items?jobname={}'.format(base_url, name),
                    'name': config.get('view', name),
                    'cron': config.get('cron', ''),
                }
            basic_info = {
                "yaml_config": '{}/_{}'.format(base_url, 'yaml'),
                "items": results,
            }
            return jsonify(basic_info)

        @app.route('/api/_yaml')
        def get_yaml():
            with open(settings.path, 'r') as stream:
                yaml_dict = yaml.load(stream)
            return jsonify(yaml_dict)

        @app.route('/api/items')
        def get_job_items():
            jobname = request.args.get('jobname')
            fmt = request.args.get('format', 'json')
            job = self.sched.jobs.get(jobname, None)
            if job is None:
                abort(404)
            originitems = job.store.iter(**request.args.to_dict())
            if len(originitems):
                if getattr(originitems[0], fmt, None) is None:
                    for f in ['html', 'markdown', 'json']:
                        if getattr(originitems[0], f, None) is not None:
                            fmt = f
                            break

            items = [getattr(i, fmt)() for i in originitems]
            columns = [c.key for c in job.store.item_class.__table__.columns if c.
            if fmt == 'markdown':
                # Markdown donot support open new tab
                items = [markdown2.markdown(i).replace('href=', 'target="_blank" h
            return jsonify({'columns': columns, 'data': items, 'format': fmt, 'yam
```

Example 24

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
    })
```