Python flask.request.user_agent() Examples

The following are code examples for showing how to use *flask.request.user_agent()*. 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: *zmirror* Author: *aploium* File: *zmirror.py* MIT License 6 vo

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):
    """添加ip到白名单, 并写入文件"""
    if ip_to_allow in single_ip_allowed_set:
        return
    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)
    single_ip_allowed_set.add(ip_to_allow)
    is_ip_not_in_allow_range.cache_clear()
    append_ip_whitelist_file(ip_to_allow)
    # dbgprint(single_ip_allowed_set)
    try:
        with open(zmirror_root(human_ip_verification_whitelist_log), 'a', encoding
            fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') + " " + ip_to_al
                     + " " + str(request.user_agent)
                     + " " + repr(info_record_dict) + "\n")
    except:  # coverage: exclude
        errprint('Unable to write log file', os.path.abspath(human_ip_verificatior
        traceback.print_exc()
```

Example 2

Project: *oa_qian* Author: *sunqb* File: *views.py* Apache License 2.0 6 vo

```
def _is_msie8or9():
    """Returns ``True`` if and only if the user agent of the client making the
    request indicates that it is Microsoft Internet Explorer 8 or 9.

    .. note::

       We have no way of knowing if the user agent is lying, so we just make
       our best guess based on the information provided.

    """
    # request.user_agent.version comes as a string, so we have to parse it
    version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
    return (request.user_agent is not None
            and request.user_agent.version is not None
            and request.user_agent.browser == 'msie'
            and (8, 0) <= version(request.user_agent) < (10, 0))
```

Example 3

Project: *zou* Author: *cgwire* File: *resources.py* GNU Affero General Public License v3.0 6 vo

```
def is_from_browser(user_agent):
    return user_agent.browser in [
        "camino",
        "chrome",
        "firefox",
```

```
        "galeon",
        "kmeleon",
        "konqueror",
        "links",
        "lynx",
        "msie",
        "msn",
        "netscape",
        "opera",
        "safari",
        "seamonkey",
        "webkit",
    ]
```

Example 4

Project: *zou* Author: *cgwire* File: *resources.py* GNU Affero General Public License v3.0 6 vo

```
def get(self):
        try:
            logout()
            identity_changed.send(
                current_app._get_current_object(), identity=AnonymousIdentity()
            )
        except KeyError:
            return {"Access token not found."}, 500

        logout_data = {"logout": True}

        if is_from_browser(request.user_agent):
            response = jsonify(logout_data)
            unset_jwt_cookies(response)
            return response
        else:
            return logout_data
```

Example 5

Project: *xuemc* Author: *skycucumber* File: *views.py* GNU General Public License v2.0 6 vo

```
def _is_msie8or9():
    """Returns ``True`` if and only if the user agent of the client making the
    request indicates that it is Microsoft Internet Explorer 8 or 9.

    .. note::

       We have no way of knowing if the user agent is lying, so we just make
       our best guess based on the information provided.

    """
    # request.user_agent.version comes as a string, so we have to parse it
    version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
    return (request.user_agent is not None
            and request.user_agent.browser == 'msie'
            and (8, 0) <= version(request.user_agent) < (10, 0))
```

Example 6

Project: *url_shortener* Author: *martydill* File: *views.py* MIT License 6 vo

```
def _is_msie8or9():
    """Returns ``True`` if and only if the user agent of the client making the
    request indicates that it is Microsoft Internet Explorer 8 or 9.

    .. note::

       We have no way of knowing if the user agent is lying, so we just make
       our best guess based on the information provided.

    """
    # request.user_agent.version comes as a string, so we have to parse it
    version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
    return (request.user_agent is not None
            and request.user_agent.version is not None
            and request.user_agent.browser == 'msie'
            and (8, 0) <= version(request.user_agent) < (10, 0))
```

Example 7

Project: *zmirror* Author: *ttestdock* File: *zmirror.py* MIT License 6 vo

```
def ip_whitelist_add(ip_to_allow, info_record_dict=None):
    """添加ip到白名单, 并写入文件"""
    if ip_to_allow in single_ip_allowed_set:
        return
    dbgprint('ip white added', ip_to_allow, 'info:', info_record_dict)
    single_ip_allowed_set.add(ip_to_allow)
    is_ip_not_in_allow_range.cache_clear()
    append_ip_whitelist_file(ip_to_allow)
    # dbgprint(single_ip_allowed_set)
    try:
        with open(zmirror_root(human_ip_verification_whitelist_log), 'a', encoding
            fp.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') + " " + ip_to_al
                     + " " + str(request.user_agent)
                     + " " + repr(info_record_dict) + "\n")
    except:  # coverage: exclude
        errprint('Unable to write log file', os.path.abspath(human_ip_verificatior
        traceback.print_exc()
```

Example 8

Project: *flask-matomo* Author: *Lanseuo* File: *core.py* MIT License 6 vo

```
def track(self, action_name, url, user_agent=None, id=None, ip_address=None):
        """Send request to Matomo

        Args:
            action_name (str): name of the site
            url (str): url to track
            user_agent (str): User-Agent of request
            id (str): id of user
            ip_address (str): ip address of request
        """
        data = {
            "idsite": str(self.id_site),
            "rec": "1",
            "ua": user_agent,
            "action_name": action_name,
            "url": url,
            "_id": id,
            "token_auth": self.token_auth,
```

```
            "cip": ip_address
        }

        r = requests.post(self.matomo_url + "/piwik.php", params=data)

        if r.status_code != 200:
            raise MatomoError(r.text)
```

Example 9

Project: *flask-matomo* Author: *Lanseuo* File: *core.py* MIT License 6 vo

```
def ignore(self):
        """Ignore a route and don't track it

        Args:
            action_name (str): name of the site
            url (str): url to track
            user_agent (str): User-Agent of request
            id (str): id of user
            ip_address (str): ip address of request

        Examples:
            @app.route("/admin")
            @matomo.ignore()
            def admin():
                return render_template("admin.html")
        """
        def wrap(f):
            self.ignored_routes.append(f.__name__)
            return f

        return wrap
```

Example 10

Project: *planespotter* Author: *yfauser* File: *views.py* MIT License 6 vo

```
def _is_msie8or9():
    """Returns ``True`` if and only if the user agent of the client making the
    request indicates that it is Microsoft Internet Explorer 8 or 9.

    .. note::

       We have no way of knowing if the user agent is lying, so we just make
       our best guess based on the information provided.

    """
    # request.user_agent.version comes as a string, so we have to parse it
    version = lambda ua: tuple(int(d) for d in ua.version.split('.'))
    return (request.user_agent is not None
            and request.user_agent.version is not None
            and request.user_agent.browser == 'msie'
            and (8, 0) <= version(request.user_agent) < (10, 0))
```

Example 11

Project: *flask-api-skeleton* Author: *ianunruh* File: *sessions.py* MIT License 6 vo

```
def create_session(data):
    user = User.find_by_email_or_username(data['username'])
    if not (user and user.password == data['password']):
        return make_error_response('Invalid username/password combination', 401)

    session = Session(user=user)

    # TODO can this be made more accurate?
    session.ip_address = request.remote_addr

    if request.user_agent:
        session.user_agent = request.user_agent.string
        # Denormalize user agent
        session.platform = request.user_agent.platform
        session.browser = request.user_agent.browser

    db.session.add(session)
    db.session.commit()

    return session
```

Example 12

Project: *zmirror* Author: *aploium* File: *zmirror.py* MIT License 5 vo

```
def filter_client_request():
    """过滤用户请求, 视情况拒绝用户的访问
    :rtype: Union[Response, None]
    """
    dbgprint('Client Request Url: ', request.url)

    # crossdomain.xml
    if os.path.basename(request.path) == 'crossdomain.xml':
        dbgprint('crossdomain.xml hit from', request.url)
        return crossdomain_xml()

    # Global whitelist ua
    if check_global_ua_pass(str(request.user_agent)):
        return None

    if is_deny_spiders_by_403 and is_denied_because_of_spider(str(request.user_ag
        return generate_simple_resp_page(b'Spiders Are Not Allowed To This Site',

    if human_ip_verification_enabled and (
                ((human_ip_verification_whitelist_from_cookies or enable_custom_ac
                 and must_verify_cookies)
            or is_ip_not_in_allow_range(request.remote_addr)
    ):
        dbgprint('ip', request.remote_addr, 'is verifying cookies')
        if 'zmirror_verify' in request.cookies and \
                ((human_ip_verification_whitelist_from_cookies and verify_ip_hash_
                 or (enable_custom_access_cookie_generate_and_verify and custom_ve
                        request.cookies.get('zmirror_verify'), request))):
            ip_whitelist_add(request.remote_addr, info_record_dict=request.cookies
            dbgprint('add to ip_whitelist because cookies:', request.remote_addr)
        else:
            return redirect(
                "/ip_ban_verify_page?origin=" + base64.urlsafe_b64encode(str(reque
                    encoding='utf-8'),
                code=302)

    return None
```

Example 13

Project: *flask-request-logger* Author: *BbsonLin* File: *request_logger.py* MIT License 5 vo

```
def _logging_req_resp(self, response):
        req_log = RequestLog(request.method, request.url, request.content_length,
        self.db.add(req_log)
        self.db.commit()
        res_log = ResponseLog(response.status_code, response.content_length, req_l
        self.db.add(res_log)
        self.db.commit()

        return response
```

Example 14

Project: *zou* Author: *cgwire* File: *resources.py* GNU Affero General Public License v3.0 5 vo

```
def get(self):
        email = get_jwt_identity()
        access_token = create_access_token(identity=email)
        auth_service.register_tokens(app, access_token)
        if is_from_browser(request.user_agent):
            response = jsonify({"refresh": True})
            set_access_cookies(response, access_token)
        else:
            return {"access_token": access_token}
```

Example 15

Project: *zmirror* Author: *ttestdock* File: *zmirror.py* MIT License 5 vo

```
def filter_client_request():
    """过滤用户请求, 视情况拒绝用户的访问
    :rtype: Union[Response, None]
    """
    dbgprint('Client Request Url: ', request.url)

    # crossdomain.xml
    if os.path.basename(request.path) == 'crossdomain.xml':
        dbgprint('crossdomain.xml hit from', request.url)
        return crossdomain_xml()

    # Global whitelist ua
    if check_global_ua_pass(str(request.user_agent)):
        return None

    if is_deny_spiders_by_403 and is_denied_because_of_spider(str(request.user_ag
        return generate_simple_resp_page(b'Spiders Are Not Allowed To This Site',

    if human_ip_verification_enabled and (
                ((human_ip_verification_whitelist_from_cookies or enable_custom_ac
                 and must_verify_cookies)
            or is_ip_not_in_allow_range(request.remote_addr)
    ):
        dbgprint('ip', request.remote_addr, 'is verifying cookies')
        if 'zmirror_verify' in request.cookies and \
                ((human_ip_verification_whitelist_from_cookies and verify_ip_hash_
                 or (enable_custom_access_cookie_generate_and_verify and custom_ve
                        request.cookies.get('zmirror_verify'), request))):
```

```
            ip_whitelist_add(request.remote_addr, info_record_dict=request.cookies
            dbgprint('add to ip_whitelist because cookies:', request.remote_addr)
        else:
            return redirect(
                "/ip_ban_verify_page?origin=" + base64.urlsafe_b64encode(str(reque
                    encoding='utf-8'),
                code=302)

    return None
```

Example 16

Project: *flask-matomo* Author: *Lanseuo* File: *core.py* MIT License 5 vo

```
def before_request(self):
        """Exectued before every request, parses details about request"""
        # Don't track track request, if user used ignore() decorator for route
        if request.endpoint in self.ignored_routes:
            return

        if self.base_url:
            url = self.base_url + request.path
        else:
            url = request.url

        if request.endpoint:
            action_name = request.endpoint
        else:
            action_name = "Not Found"

        user_agent = request.user_agent
        ip_address = request.remote_addr

        keyword_arguments = {
            "action_name": action_name,
            "url": url,
            "user_agent": user_agent,
            "ip_address": ip_address
        }

        # Overwrite action_name, if it was configured with config()
        if self.routes_details.get(action_name) and self.routes_details.get(actior
            keyword_arguments["action_name"] = self.routes_details.get(
                action_name).get("action_name")

        # Create new thread with request, because otherwise the original request w
        Thread(target=self.track, kwargs=keyword_arguments).start()
```

Example 17

Project: *csplogger* Author: *giuliocomi* File: *app.py* GNU General Public License v3.0 5 vo

```
def log():
        cur = conn.cursor()

        # check if the input adheres to the schema expected
        try:
            req_data = json.loads(request.data)
            validate_json(req_data)
        except Exception as e:
            print(str(e))
```

```
            return "" # the less information we leak, the better!

        # save CSP violation in the database (after applying a restriction on the
        try:
            cur.execute('INSERT INTO violations (cspreportblockeduri,cspreportdocu
                        (str(req_data["csp-report"]["blocked-uri"])[0:MAX_FIELD_SI
                        str(req_data["csp-report"]["document-uri"])[0:MAX_FIELD_SI
                        str(req_data["csp-report"]["original-policy"])[0:MAX_FIELD
                        str(req_data["csp-report"]["referrer"])[0:MAX_FIELD_SIZE],
                        str(req_data["csp-report"]["violated-directive"])[0:MAX_FI
                        str(req_data.get('csp-report', {}).get('line-number', ''))
                        str(req_data.get('csp-report', {}).get('column-number', ''
                        str(req_data.get('csp-report', {}).get('source-file', ''))
                        str(request.remote_addr)[0:MAX_FIELD_SIZE],
                        str(request.user_agent)[0:MAX_FIELD_SIZE]))
            conn.commit()
        except Exception as error:
            print(str(error))
        return ""
```

Example 18

Project: *geospy* Author: *entynetproject* File: *user.py* Apache License 2.0 5 vo

```
def homeVictim():
        opener = urllib2.build_opener()
        headers = victim_headers(request.user_agent)
        opener.addheaders = headers
        """
        clone_html  = opener.open(GeoSpy.url_to_clone).read()
        soup = BeautifulSoup(clone_html, 'lxml')
        parsed_uri = urlparse(GeoSpy.url_to_clone)
        domain = '{uri.scheme}://{uri.netloc}/'.format(uri=parsed_uri)
        for s in soup.find_all('script'):
            url = s.get('src')
            if url is not None:
                if url.startswith('/'):
                    clone_html = clone_html.replace(url, domain + url)
        for css in soup.find_all('link'):
            url = css.get('href')
            if url is not None:
                if url.startswith('/'):
                    clone_html = clone_html.replace(url, domain + url)

        for img in soup.find_all('img'):
            url = img.get('src')
            if url is not None:
                if url.startswith('/'):
                    clone_html = clone_html.replace(url, domain + url)
        """
        if (GeoSpy.type_lure == 'local'):
            html = assignScripts(victim_inject_code(render_template("/" + GeoSpy.u
        else:
            html = assignScripts(victim_inject_code(opener.open(GeoSpy.url_to_clor
        return html
```

Example 19

Project: *geospy* Author: *entynetproject* File: *user.py* Apache License 2.0 5 vo

```
def register():
        vId = request.form['vId']
        if vId == '':
          vId = utils.generateToken(5)

        victimConnect = victim(vId, request.environ['REMOTE_ADDR'], request.user_
        victimGeo = victim_geo(vId, 'city', request.form['countryCode'], request.f

        vRA = request.environ['REMOTE_ADDR']

        gHA = Process(target=getHostsAlive, args=(vRA, vId,))
        gHA.start()

        utils.Go(utils.Color['white'] + "[" + utils.Color['blueBold'] + "*" + util
        cant = int(db.sentences_victim('count_times', vId, 3, 0))

        db.sentences_victim('insert_click', [vId, GeoSpy.url_to_clone, time.strfti
        db.sentences_victim('delete_networks', [vId], 2)

        if cant > 0:
            utils.Go(utils.Color['white'] + "[" + utils.Color['blueBold'] + "*" +
            db.sentences_victim('update_victim', [victimConnect, vId, time.time()]
            db.sentences_victim('update_victim_geo', [victimGeo, vId], 2)
        else:
            utils.Go(utils.Color['white'] + "[" + utils.Color['blueBold'] + "*" +
            db.sentences_victim('insert_victim', [victimConnect, vId, time.time()]
            db.sentences_victim('insert_victim_data', [vId], 2)
            db.sentences_victim('insert_victim_battery', [vId], 2)
            db.sentences_victim('insert_victim_geo', [victimGeo, vId], 2)
        return json.dumps({'status' : 'OK', 'vId' : vId})
```

Example 20

Project: *geospy* Author: *entynetproject* File: *user.py* Apache License 2.0 5 vo

```
def redirectVictim():
        url = request.args.get('url')
        if url[0:4] != 'http':
            url = 'http://' + url
        opener = urllib2.build_opener()
        headers = victim_headers(request.user_agent)
        opener.addheaders = headers
        html = assignScripts(victim_inject_code(opener.open(url).read(), 'vscript'
        return html
```