Python flask.request.is_secure() Examples

The following are code examples for showing how to use *flask.request.is_secure()*. 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: *flasky* Author: *RoseOu* File: *flask_moment.py* MIT License 6 vo

```
def include_moment(version = '2.3.1'):
        if version is not None:
            if request.is_secure:
                protocol = 'https'
            else:
                protocol = 'http'
            js = '<script src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo
        return Markup('''%s<script>
function flask_moment_render(elem) {
    $(elem).text(eval('moment("' + $(elem).data('timestamp') + '").' + $(elem).dat
    $(elem).removeClass('flask-moment');
}
function flask_moment_render_all() {
    $('.flask-moment').each(function() {
        flask_moment_render(this);
        if ($(this).data('refresh')) {
            (function(elem, interval) { setInterval(function() { flask_moment_renc
        }
    })
}
$(document).ready(function() {
    flask_moment_render_all();
});
</script>''' % js)
```

Example 2

Project: *flasky* Author: *RoseOu* File: *flask_sslify.py* MIT License 6 vo

```
def redirect_to_ssl(self):
        """Redirect incoming requests to HTTPS."""
        # Should we redirect?
        criteria = [
            request.is_secure,
            current_app.debug,
            request.headers.get('X-Forwarded-Proto', 'http') == 'https'
        ]

        if not any(criteria) and not self.skip:
            if request.url.startswith('http://'):
                url = request.url.replace('http://', 'https://', 1)
                code = 302
                if self.permanent:
                    code = 301
                r = redirect(url, code=code)
                return r
```

Example 3

Project: *jbox* Author: *jpush* File: *csrf.py* MIT License 6 vo

```
def protect(self):
        if request.method not in self._app.config['WTF_CSRF_METHODS']:
            return

        if not validate_csrf(self._get_csrf_token()):
            reason = 'CSRF token missing or incorrect.'
            return self._error_response(reason)

        if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
                reason = 'Referrer checking failed - no Referrer.'
                return self._error_response(reason)

            good_referrer = 'https://%s/' % request.host
            if not same_origin(request.referrer, good_referrer):
                reason = 'Referrer checking failed - origin does not match.'
                return self._error_response(reason)

        request.csrf_valid = True  # mark this request is csrf valid
```

Example 4

Project: *RSSNewsGAE* Author: *liantian-cn* File: *csrf.py* Apache License 2.0 6 vo

```
def protect(self):
        if request.method not in current_app.config['WTF_CSRF_METHODS']:
            return

        try:
            validate_csrf(self._get_csrf_token())
        except ValidationError as e:
            logger.info(e.args[0])
            self._error_response(e.args[0])

        if request.is_secure and current_app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
                self._error_response('The referrer header is missing.')

            good_referrer = 'https://{0}/'.format(request.host)

            if not same_origin(request.referrer, good_referrer):
                self._error_response('The referrer does not match the host.')

        g.csrf_valid = True  # mark this request as CSRF valid
```

Example 5

Project: *chihu* Author: *yelongyu* File: *csrf.py* GNU General Public License v3.0 6 vo

```
def protect(self):
        if request.method not in self._app.config['WTF_CSRF_METHODS']:
            return

        if not validate_csrf(self._get_csrf_token()):
            reason = 'CSRF token missing or incorrect.'
            return self._error_response(reason)

        if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
```

```
                reason = 'Referrer checking failed - no Referrer.'
                return self._error_response(reason)

            good_referrer = 'https://%s/' % request.host
            if not same_origin(request.referrer, good_referrer):
                reason = 'Referrer checking failed - origin does not match.'
                return self._error_response(reason)

        request.csrf_valid = True  # mark this request is csrf valid
```

Example 6

Project: *Mocha* Author: *mardix* File: *ext.py* MIT License 6 vo

```
def init_app(self, app):
        delivery_method = app.config.get("ASSETS_DELIVERY_METHOD")
        if delivery_method and delivery_method.upper() in ["S3", "CDN"]:
            #with app.app_context():
            is_secure = False #request.is_secure

            if delivery_method.upper() == "CDN":
                domain = app.config.get("ASSETS_DELIVERY_DOMAIN")
                if "://" in domain:
                    domain_parsed = utils.urlparse(domain)
                    is_secure = domain_parsed.scheme == "https"
                    domain = domain_parsed.netloc
                app.config.setdefault("S3_CDN_DOMAIN", domain)

            app.config["FLASK_ASSETS_USE_S3"] = True
            app.config["FLASKS3_ACTIVE"] = True
            app.config["FLASKS3_URL_STYLE"] = "path"
            app.config.setdefault("FLASKS3_USE_HTTPS", is_secure)
            app.config.setdefault("FLASKS3_ONLY_MODIFIED", True)
            app.config.setdefault("FLASKS3_GZIP", True)
            app.config.setdefault("FLASKS3_BUCKET_NAME", app.config.get("AWS_S3_BU

            super(self.__class__, self).init_app(app)
```

Example 7

Project: *WRGameVideos-API* Author: *thundernet8* File: *flask_moment.py* GNU General Public License v2.0 6 vo

```
def include_moment(version = '2.3.1'):
        if version is not None:
            if request.is_secure:
                protocol = 'https'
            else:
                protocol = 'http'
            js = '<script src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo
        return Markup('''%s<script>
function flask_moment_render(elem) {
    $(elem).text(eval('moment("' + $(elem).data('timestamp') + '").' + $(elem).dat
    $(elem).removeClass('flask-moment');
}
function flask_moment_render_all() {
    $('.flask-moment').each(function() {
        flask_moment_render(this);
        if ($(this).data('refresh')) {
            (function(elem, interval) { setInterval(function() { flask_moment_renc
        }
```

```
    })
}
$(document).ready(function() {
    flask_moment_render_all();
});
</script>''' % js)
```

Example 8

Project: *WRGameVideos-API* Author: *thundernet8* File: *flask_sslify.py* GNU General Public License v2.0 6 vo

```
def redirect_to_ssl(self):
        """Redirect incoming requests to HTTPS."""
        # Should we redirect?
        criteria = [
            request.is_secure,
            current_app.debug,
            request.headers.get('X-Forwarded-Proto', 'http') == 'https'
        ]

        if not any(criteria) and not self.skip:
            if request.url.startswith('http://'):
                url = request.url.replace('http://', 'https://', 1)
                code = 302
                if self.permanent:
                    code = 301
                r = redirect(url, code=code)
                return r
```

Example 9

Project: *plataforma-livre-dados-abertos* Author: *pbaesse* File: *csrf.py* GNU General Public License v3.0 6 vo

```
def protect(self):
        if request.method not in current_app.config['WTF_CSRF_METHODS']:
            return

        try:
            validate_csrf(self._get_csrf_token())
        except ValidationError as e:
            logger.info(e.args[0])
            self._error_response(e.args[0])

        if request.is_secure and current_app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
                self._error_response('The referrer header is missing.')

            good_referrer = 'https://{0}/'.format(request.host)

            if not same_origin(request.referrer, good_referrer):
                self._error_response('The referrer does not match the host.')

        g.csrf_valid = True  # mark this request as CSRF valid
```

Example 10

Project: *zeus* Author: *getsentry* File: *ssl.py* Apache License 2.0 6 vo

```
def redirect_to_ssl(self):
        """
        Redirect incoming requests to HTTPS.
        """
        criteria = [
            request.is_secure,
            current_app.debug,
            current_app.testing,
            request.headers.get("X-Forwarded-Proto", "http") == "https",
        ]

        if (
            request.headers.get("User-Agent", "")
            .lower()
            .startswith(self.exclude_user_agents)
        ):
            return

        if not any(criteria):
            if request.url.startswith("http://"):
                url = request.url.replace("http://", "https://", 1)
                r = redirect(url, code=301)
                return r
```

Example 11

Project: *webapp* Author: *superchilli* File: *flask_moment.py* MIT License 6 vo

```
def include_moment(version = '2.3.1'):
        if version is not None:
            if request.is_secure:
                protocol = 'https'
            else:
                protocol = 'http'
            js = '<script src="%s://cdnjs.cloudflare.com/ajax/libs/moment.js/%s/mo
        return Markup('''%s<script>
function flask_moment_render(elem) {
    $(elem).text(eval('moment("' + $(elem).data('timestamp') + '").' + $(elem).dat
    $(elem).removeClass('flask-moment');
}
function flask_moment_render_all() {
    $('.flask-moment').each(function() {
        flask_moment_render(this);
        if ($(this).data('refresh')) {
            (function(elem, interval) { setInterval(function() { flask_moment_renc
        }
    })
}
$(document).ready(function() {
    flask_moment_render_all();
});
</script>''' % js)
```

Example 12

Project: *webapp* Author: *superchilli* File: *csrf.py* MIT License 6 vo

```
def protect(self):
        if request.method not in current_app.config['WTF_CSRF_METHODS']:
            return
```

```
        try:
            validate_csrf(self._get_csrf_token())
        except ValidationError as e:
            logger.info(e.args[0])
            self._error_response(e.args[0])

        if request.is_secure and current_app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
                self._error_response('The referrer header is missing.')

            good_referrer = 'https://{0}/'.format(request.host)

            if not same_origin(request.referrer, good_referrer):
                self._error_response('The referrer does not match the host.')

        g.csrf_valid = True  # mark this request as CSRF valid
```

Example 13

Project: *WRGameVideos-Server* Author: *thundernet8* File: *csrf.py* GNU General Public License v2.0 6 vo

```
def protect(self):
        if request.method not in self._app.config['WTF_CSRF_METHODS']:
            return

        if not validate_csrf(self._get_csrf_token()):
            reason = 'CSRF token missing or incorrect.'
            return self._error_response(reason)

        if request.is_secure and self._app.config['WTF_CSRF_SSL_STRICT']:
            if not request.referrer:
                reason = 'Referrer checking failed - no Referrer.'
                return self._error_response(reason)

            good_referrer = 'https://%s/' % request.host
            if not same_origin(request.referrer, good_referrer):
                reason = 'Referrer checking failed - origin does not match.'
                return self._error_response(reason)

        request.csrf_valid = True  # mark this request is csrf valid
```

Example 14

Project: *flasky* Author: *RoseOu* File: *__init__.py* MIT License 5 vo

```
def include_pagedown(self):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('''
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
'''.format(protocol))
```

Example 15

Project: *flasky* Author: *RoseOu* File: *flask_moment.py* MIT License 5 vo

```
def include_jquery(version = '1.10.1'):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js"></scrip
```

Example 16

Project: *flasky* Author: *RoseOu* File: *flask_sslify.py* MIT License 5 vo

```
def set_hsts_header(self, response):
        """Adds HSTS header to each response."""
        # Should we add STS header?
        if request.is_secure and not self.skip:
            response.headers.setdefault('Strict-Transport-Security', self.hsts_hea
        return response
```

Example 17

Project: *flasky* Author: *RoseOu* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 18

Project: *ibart* Author: *jbech-linaro* File: *ibart.py* MIT License 5 vo

```
def restart_page(pr_id, pr_sha1):
    worker.user_add(pr_id, pr_sha1)
    # if request.is_secure:
    #     if request.referrer:
    #         return redirect(request.referrer)
    return redirect(request.referrer)
```

Example 19

Project: *ibart* Author: *jbech-linaro* File: *ibart.py* MIT License 5 vo

```
def stop_page(pr_id, pr_sha1):
    worker.cancel(pr_id, pr_sha1)
    # if request.is_secure:
    #     if request.referrer:
    #         return redirect(request.referrer)
    return redirect(request.referrer)


# logs/jbech-linaro/
```

Example 20

Project: *circleci-demo-python-flask* Author: *CircleCI-Public* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 21

Project: *qis* Author: *quru* File: *flask_util.py* GNU Affero General Public License v3.0 5 vo

```
def get_port(request):
    """
    Returns the port number in use on a Flask/Werkzeug request object.
    """
    sep_idx = request.host.find(':')
    if sep_idx == -1:
        return 443 if request.is_secure else 80
    else:
        return parse_int(request.host[sep_idx + 1:])
```

Example 22

Project: *qis* Author: *quru* File: *views_util.py* GNU Affero General Public License v3.0 5 vo

```
def _check_ssl_request(request, from_web):
    """
    A low-level component implementing a request checker that tests for HTTPS
    and returns a Flask redirect if required (or a JSON error response if not
    from_web), but otherwise returns None.
    """
    if not request.is_secure:
        if from_web:
            to_url = request.url.replace('http:', 'https:', 1)
            return redirect(to_url)
        else:
            return make_api_error_response(AuthenticationError(
                'HTTPS must be used to access this function'
            ), logger)
    return None
```

Example 23

Project: *Flashcards* Author: *KevDi* File: *users.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 24

Project: *Python-OpenCV-Webserver* Author: *iandowling* File: *app.py* MIT License 5 vo

```
def ssl_wrapper(req):
    @wraps(req)
    def redirect_ssl(*args, **argsv):
        if current_app.config.get("SSL"):
            if request.is_secure:
                return req(*args, **argsv)
            else:
                return redirect(request.url.replace("http://", "https://"))

        return req(*args, **argsv)

    return redirect_ssl
```

Example 25

Project: *chihu* Author: *yelongyu* File: *models.py* GNU General Public License v3.0 5 vo

```
def gravatar(self, size=50, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://cdn.v2ex.com/gravatar/'    # 换成国内的源，不然你懂的
        else:
            url = 'http://cn.gravatar.com/avatar'
        hash = hashlib.md5(self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}'.format(url=url, hash=hash, size=size)
```

Example 26

Project: *flasky-appengine* Author: *russomi* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 27

Project: *RPGOne* Author: *RTHMaK* File: *models.py* Apache License 2.0 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 28

Project: *flask-now* Author: *richgieg* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or self.generate_avatar_hash()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 29

Project: *elibrarian* Author: *frank-u* File: *models.py* GNU General Public License v3.0 5 vo

```
def gravatar(self, size=100, default='identicon', rating='x'):
        """Get link pointing to user's gravatar"""
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        av_hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=av_hash, size=size, default=default, rating=rating)
```

Example 30

Project: *roger-api* Author: *rogertalk* File: *__init__.py* MIT License 5 vo

```
def enforce_https():
    if not request.is_secure:
        return 'Try again with HTTPS.', 403
```

Example 31

Project: *Oyster-app* Author: *XzAmrzs* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
                self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
                url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 32

Project: *WRGameVideos-API* Author: *thundernet8* File: *__init__.py* GNU General Public License v2.0 5 vo

```
def include_pagedown(self):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('''
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
```

```
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
'''.format(protocol))
```

Example 33

Project: *WRGameVideos-API* Author: *thundernet8* File: *flask_moment.py* GNU General Public License v2.0 5 vo

```
def include_jquery(version = '1.10.1'):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js"></scrip
```

Example 34

Project: *WRGameVideos-API* Author: *thundernet8* File: *flask_sslify.py* GNU General Public License v2.0 5 vo

```
def set_hsts_header(self, response):
        """Adds HSTS header to each response."""
        # Should we add STS header?
        if request.is_secure and not self.skip:
            response.headers.setdefault('Strict-Transport-Security', self.hsts_hea
        return response
```

Example 35

Project: *database_project* Author: *HughWen* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 36

Project: *Blog* Author: *CharlesZhong* File: *models.py* GNU General Public License v2.0 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 37

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

```
def set_hsts_header(self, response):
        """
        Adds HSTS header to each response.
        """
        if request.is_secure:
            response.headers.setdefault("Strict-Transport-Security", self.hsts_hea
        return response
```

Example 38

Project: *FlaskPyrezAPI* Author: *luissilva1044894* File: *__init__.py* MIT License 5 vo

```
def check_redirects(app):
        #@app.before_request
        @app.before_first_request
        def do_before_request():
                if isinstance(app, flask.Flask):
                        from flask import request, g
                else:
                        from quart import request, g
                scheme = request.headers.get('X-Forwarded-Proto')
                # https://stackoverflow.com/questions/32237379/python-flask-redire
                # if not request.is_secure and app.env != 'development':
                if scheme and scheme == 'http' and request.url.startswith('http://
                        return redirect(request.url.replace('http://', 'https://',
                g.__cookies__ = []
                from utils.file import read_file
                for _ in (read_file('data/redirects.json', is_json=True) or {}).ge
                        if _.get('path') and _.get('path').lower() == request.path
                                if isinstance(app, flask.Flask):
                                        from flask import redirect, url_for
                                else:
                                        from quart import redirect, url_for
                                return redirect(url_for(_.get('for')))
                '''#redirect_old
                for _ in __kwargs__:
                        for __ in __kwargs__[_]:
                                if request.path == __: #request.full_path
                                        from flask import redirect, url_for
                                        _split = __.split('/')[1:]
                                        return redirect(url_for(f'{_split[0]}.{_}.
                '''
```

Example 39

Project: *confidant* Author: *lyft* File: *userauth.py* Apache License 2.0 5 vo

```
def log_in(self):
        response = flask.make_response()
        result = self.authomatic.login(
            WerkzeugAdapter(request, response),
            'google',
            session=session,
            session_saver=lambda: app.save_session(session, response),
            secure_cookie=(True if request.is_secure else False)
        )
        if result:
            if result.error:
                msg = 'Google auth failed with error: {0}'
                logging.error(msg.format(result.error))
                return abort(403)
```

```
            # successful login
            if result.user:
                result.user.update()
                user = result.user
                self.set_expiration()
                self.set_current_user(email=user.email,
                                      first_name=user.first_name,
                                      last_name=user.last_name)
                # TODO: find a way to save the angular args?
                # authomatic adds url params google auth has stripped the
                # angular args anyway, so let's just redirect back to the
                # index.
                resp = self.redirect_to_index()
                self.set_csrf_token(resp)
                return resp

        # Authomatic will have put a redirect in our response here.
        return response
```

Example 40

Project: *material-girl* Author: *bobcolner* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 41

Project: *webapp* Author: *superchilli* File: *__init__.py* MIT License 5 vo

```
def include_pagedown(self):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('''
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
<script type="text/javascript" src="{0}://cdnjs.cloudflare.com/ajax/libs/pagedown/
'''.format(protocol))
```

Example 42

Project: *webapp* Author: *superchilli* File: *flask_moment.py* MIT License 5 vo

```
def include_jquery(version = '1.10.1'):
        if request.is_secure:
            protocol = 'https'
        else:
            protocol = 'http'
        return Markup('<script src="%s://code.jquery.com/jquery-%s.min.js"></scrip
```

Example 43

Project: *webapp* Author: *superchilli* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 44

Project: *flasky-first-edition* Author: *miguelgrinberg* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 45

Project: *NiceBlog* Author: *SheHuan* File: *models.py* Apache License 2.0 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        """
        生成用户头像地址
        :param size:图片大小
        :param default:指定图片生成器
        :param rating:图片级别
        :return:
        """
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or self.gravatar_hash()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(url=url, hash
                                                                    rating=rating
```

Example 46

Project: *PilosusBot* Author: *pilosus* File: *test_models.py* MIT License 5 vo

```
def test_is_testing(self):
        self.assertTrue(current_app.config['TESTING'])
        self.assertFalse(current_app.config['SSL_DISABLE'])
        self.assertFalse(request.is_secure)
```

Example 47

Project: *PilosusBot* Author: *pilosus* File: *test_models.py* MIT License 5 vo

```
def test_user_gravatar(self):
        user_role = Role.query.filter_by(name='User').first()
        user = User(email=forgery_py.internet.email_address(),
                    username=forgery_py.internet.user_name(),
                    password='old_password',
                    avatar_hash=None,
                    role=user_role,
                    confirmed=True)
        db.session.add(user)
        db.session.commit()

        https_url = 'https://secure.gravatar.com/avatar'
        http_url = 'http://www.gravatar.com/avatar'
        size = 150
        default = 'identicon'
        rating = 'g'
        hash = hashlib.md5(user.email.encode('utf-8')).hexdigest()

        http_gravatar = user.gravatar(size=size, default=default, rating=rating)


        self.assertEqual(http_gravatar,
                         '{url}/{hash}?s={size}&d={default}&r={rating}'.
                         format(url=http_url, hash=hash, size=size, default=defaul
                                rating=rating))
        self.assertNotEqual(http_gravatar,
                            '{url}/{hash}?s={size}&d={default}&r={rating}'.
                            format(url=https_url, hash=hash, size=size, default=de
                                   rating=rating))

        # 'PilosusBot.models.request' cannot be patched like this:
        #  with patch('PilosusBot.models.request.is_secure', new_callable=Propert
        #      mock_sec.return_value = True
        #      request.is_secure # returns True now
        #
        # so there's no way to test HTTPS gravatar url other than
        # having fun with HTTP headers probably (?)
```

Example 48

Project: *PilosusBot* Author: *pilosus* File: *models.py* MIT License 5 vo

```
def gravatar(self, size=100, default='identicon', rating='g'):
        if request.is_secure:
            url = 'https://secure.gravatar.com/avatar'
        else:
            url = 'http://www.gravatar.com/avatar'
        hash = self.avatar_hash or hashlib.md5(
            self.email.encode('utf-8')).hexdigest()
        return '{url}/{hash}?s={size}&d={default}&r={rating}'.format(
            url=url, hash=hash, size=size, default=default, rating=rating)
```

Example 49

Project: *openshift3-webhook-proxy* Author: *GrahamDumpleton* File: *app.py* BSD 2-Clause "Simplified" License 4 vo

```
def webhook_travis_ci(cluster, project, application):
    debug = os.environ.get('DEBUG', '').lower() in ('1', 'true')

    authorization = request.headers['Authorization']
```

```
    fields = json.loads(request.form['payload'])

    if debug:
        print('inbound-headers:', request.headers, file=sys.stderr)
        print('inbound-authorization:', authorization, file=sys.stderr)
        print('inbound-payload:', fields, file=sys.stderr)

    if fields['status'] not in (0, None):
        return ''

    url = generic_url % dict(cluster=cluster, project=project,
            application=application, authorization=authorization)

    payload = {}

    payload['type'] = 'git'

    payload['git'] = dict(
        uri=fields['repository']['url'],
        refs='refs/heads/'+fields['branch'],
        commit=fields['commit'],
        author=dict(
            name=fields['author_name'],
            email=fields['author_email']
        ),
        committer=dict(
            name=fields['committer_name'],
            email=fields['committer_email']
        ),
        message=fields['message']
    )

    headers = {}
    headers['Content-Type'] = 'application/json'

    data = json.dumps(payload)

    if os.environ.get('SSL_NO_VERIFY'):
        verify = not(os.environ.get('SSL_NO_VERIFY', '').lower() in ('1', 'true'))
    else:
        verify = request.is_secure

    if debug:
        print('outbound-url:', url, file=sys.stderr)
        print('outbound-payload:', payload, file=sys.stderr)
        print('outbound-verify:', verify, file=sys.stderr)

    try:
        response = requests.post(url, verify=verify, headers=headers, data=data)

    except Exception as e:
        print(e, file=sys.stderr)

        raise

    return ''
```

Example 50

Project: *pipa-pay-server* Author: *davidvon* File: *csrf.py* Apache License 2.0 4 vo

```
def init_app(self, app):
        app.jinja_env.globals['csrf_token'] = generate_csrf
        strict = app.config.get('WTF_CSRF_SSL_STRICT', True)
        csrf_enabled = app.config.get('WTF_CSRF_ENABLED', True)

        @app.before_request
        def _csrf_protect():
            # many things come from django.middleware.csrf
            if not csrf_enabled:
                return

            if request.method in ('GET', 'HEAD', 'OPTIONS', 'TRACE'):
                return

            if self._exempt_views:
                if not request.endpoint:
                    return

                view = app.view_functions.get(request.endpoint)
                if not view:
                    return

                dest = '%s.%s' % (view.__module__, view.__name__)
                if dest in self._exempt_views:
                    return

            csrf_token = None
            if request.method in ('POST', 'PUT', 'PATCH'):
                # find the ``csrf_token`` field in the subitted form
                # if the form had a prefix, the name will be ``{prefix}-csrf_toker
                for key in request.form:
                    if key.endswith('csrf_token'):
                        csrf_token = request.form[key]
            if not csrf_token:
                # You can get csrf token from header
                # The header name is the same as Django
                csrf_token = request.headers.get('X-CSRFToken')
            if not csrf_token:
                # The header name is the same as Rails
                csrf_token = request.headers.get('X-CSRF-Token')
            if not validate_csrf(csrf_token):
                reason = 'CSRF token missing or incorrect.'
                return self._error_response(reason)

            if request.is_secure and strict:
                if not request.referrer:
                    reason = 'Referrer checking failed - no Referrer.'
                    return self._error_response(reason)

                good_referrer = 'https://%s/' % request.host
                if not same_origin(request.referrer, good_referrer):
                    reason = 'Referrer checking failed - origin not match.'
                    return self._error_response(reason)

            request.csrf_valid = True  # mark this request is csrf valid
```