Python flask.request.is_xhr() Examples

The following are code examples for showing how to use *flask.request.is_xhr()*. 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: *sarna* Author: *rsrdesarrollo* File: *app.py* MIT License 6 vo

```
def error_handler(err):
    if request.is_xhr:
        return str(err), err.code

    if isinstance(err, ValueError):
        err = exceptions.BadRequest()
    elif isinstance(err, NoResultFound):
        err = exceptions.NotFound()

    try:
        context = dict(
            code=err.code,
            error=err.name,
            description=err.description,
        )
    except AttributeError:
        context = dict(
            code=500,
            error='Internal Server Error',
            description=str(err),
        )

    return render_template('error.html', **context), context['code']
```

Example 2

Project: *redash-stmo* Author: *mozilla* File: *remote_user_auth.py* Mozilla Public License 2.0 6 vo

```
def redirect_login():
    """Automatically redirects from /login to /remote_user/login.
    """
    login_path = get_login_url(external=False, next=None)
    if (
        settings.REMOTE_USER_LOGIN_ENABLED
        and not request.is_xhr
        and request.path.startswith(login_path)
    ):
        org_slug = current_org.slug
        index_url = url_for("redash.index", org_slug=org_slug)
        unsafe_next_path = request.args.get("next", index_url)
        next_path = get_next_path(unsafe_next_path)
        remote_login_url = url_for(
            "remote_user_auth.login", next=next_path, org_slug=org_slug
        )
        return redirect(remote_login_url)
```

Example 3

5 vo

Project: *calibre-web* Author: *janeczku* File: *web.py* GNU General Public License v3.0

```
def remote_login_required(f):
    @wraps(f)
    def inner(*args, **kwargs):
        if config.config_remote_login:
            return f(*args, **kwargs)
        if request.is_xhr:
            data = {'status': 'error', 'message': 'Forbidden'}
            response = make_response(json.dumps(data, ensure_ascii=False))
            response.headers["Content-Type"] = "application/json; charset=utf-8"
            return response, 403
        abort(403)

    return inner
```

Example 4

Project: *calibre-web* Author: *janeczku* File: *oauth_bb.py* GNU General Public License v3.0 5 vo

```
def oauth_required(f):
    @wraps(f)
    def inner(*args, **kwargs):
        if config.config_login_type == constants.LOGIN_OAUTH:
            return f(*args, **kwargs)
        if request.is_xhr:
            data = {'status': 'error', 'message': 'Not Found'}
            response = make_response(json.dumps(data, ensure_ascii=False))
            response.headers["Content-Type"] = "application/json; charset=utf-8"
            return response, 404
        abort(404)

    return inner
```

Example 5

Project: *calibre-web* Author: *janeczku* File: *shelf.py* GNU General Public License v3.0 5 vo

```
def remove_from_shelf(shelf_id, book_id):
    shelf = ub.session.query(ub.Shelf).filter(ub.Shelf.id == shelf_id).first()
    if shelf is None:
        log.error("Invalid shelf specified: %s", shelf_id)
        if not request.is_xhr:
            return redirect(url_for('web.index'))
        return "Invalid shelf specified", 400

    # if shelf is public and use is allowed to edit shelfs, or if shelf is private
    # allow editing shelfs
    # result   shelf public   user allowed    user owner
    #   false        1             0             x
    #   true         1             1             x
    #   true         0             x             1
    #   false        0             x             0

    if (not shelf.is_public and shelf.user_id == int(current_user.id)) \
            or (shelf.is_public and current_user.role_edit_shelfs()):
        book_shelf = ub.session.query(ub.BookShelf).filter(ub.BookShelf.shelf == s
                                                           ub.BookShelf.book_id ==

        if book_shelf is None:
            log.error("Book %s already removed from %s", book_id, shelf)
```

```
            if not request.is_xhr:
                return redirect(url_for('web.index'))
            return "Book already removed from shelf", 410

        ub.session.delete(book_shelf)
        ub.session.commit()

        if not request.is_xhr:
            flash(_(u"Book has been removed from shelf: %(sname)s", sname=shelf.na
            return redirect(request.environ["HTTP_REFERER"])
        return "", 204
    else:
        log.error("User %s not allowed to remove a book from %s", current_user, sh
        if not request.is_xhr:
            flash(_(u"Sorry you are not allowed to remove a book from this shelf:
                  category="error")
            return redirect(url_for('web.index'))
        return "Sorry you are not allowed to remove a book from this shelf: %s" %
```

Example 6

Project: *platform* Author: *syncloud* File: *public.py* GNU General Public License v3.0 5 vo

```
def _callback():
    if request.is_xhr:
        return 'Unauthorised', 401
    else:
        return redirect(html_prefix + '/login.html')
```

Example 7

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

```
def jsonify_resource(resource):
    indent = None
    if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \
            and not request.is_xhr:
        indent = 2
    return current_app.response_class(json.dumps(resource, indent=indent),
                                      mimetype='application/json')
```

Example 8

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

```
def jsonify_resource(resource):
    indent = None
    if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \
            and not request.is_xhr:
        indent = 2
    return current_app.response_class(json.dumps(resource, indent=indent),
                                      mimetype='application/json')
```

Example 9

Project: *alerta-ui* Author: *itnihao* File: *utils.py* Apache License 2.0 5 vo

```
def jsonify(*args, **kwargs):
    return current_app.response_class(json.dumps(dict(*args, **kwargs), cls=DateEr
                                                 indent=None if request.is_xhr el
```

Example 10

Project: *zspider* Author: *Zephor5* File: *utils.py* MIT License 5 vo

```
def acquire_xhr(f):
    @functools.wraps(f)
    def wrap(*args, **kwargs):
        if not request.is_xhr:
            abort(403)
        return f(*args, **kwargs)
    return wrap
```

Example 11

Project: *kael* Author: *360skyeye* File: *common.py* Apache License 2.0 5 vo

```
def to_json(content):
    """Converts content to json while respecting config options."""
    indent = None
    separators = (',', ':')
    try:
        if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] and not request.is_x
            indent = 2
            separators = (', ', ': ')
    except:
        pass
    return json.dumps(content, indent=indent, separators=separators, cls=JsonEncoc
```

Example 12

Project: *lab5* Author: *zlotus* File: *views.py* MIT License 5 vo

```
def test_instance_attachment_collection_service(test_id):
    resp = flask.Response(json.dumps({'status': 'failed'}))
    if request.method == 'POST':
        if request.is_xhr:
            file = request.files['attachments']
            if file and allowed_file(file.filename):
                filename = secure_filename(file.filename)
                dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
                os.makedirs(dirpath, exist_ok=True)
                filepath = os.path.join(dirpath, filename)
                file.save(filepath)

                # db: update the attachment for the test where test.id = test_id
                test = Test.query.get(test_id)
                test.test_attachment.append(TestAttachment(name=filename, attachme
                db.session.commit()

                resp = flask.Response(json.dumps({'status': 'success', 'url': file
    elif request.method == 'DELETE':
        if request.is_json:
            filename = secure_filename(request.json['removedFile'])
            dirpath = os.path.join(UPLOAD_FOLDER, str(test_id), 'attachments')
            filepath = os.path.join(dirpath, filename)
            try:
                os.remove(filepath)
                # db: delete the attachment for the test where test.id = test_id
                TestAttachment.query.filter(
                    (TestAttachment.test_id == test_id) &
                    (TestAttachment.name == filename)
```

```
                ).delete()
                db.session.commit()

                resp = flask.Response(json.dumps({'status': 'success', 'url': file
            except FileNotFoundError:
                print('FileNotFound: ', filepath)
                resp = flask.Response(json.dumps({'status': 'failed', 'url': filep

    return set_debug_response_header(resp)
```

Example 13

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

```
def __dumps(*args, **kwargs):
    """ Serializes `args` and `kwargs` as JSON. Supports serializing an array
    as the top-level object, if it is the only argument.
    """
    indent = None
    if (current_app.config.get('JSONIFY_PRETTYPRINT_REGULAR', False)
            and not request.is_xhr):
        indent = 2
    return json.dumps(args[0] if len(args) is 1 else dict(*args, **kwargs),
                      indent=indent)
```

Example 14

Project: *redash-x* Author: *olivetree123* File: *__init__.py* BSD 2-Clause "Simplified" License 5 vo

```
def redirect_to_login():
    if request.is_xhr or '/api/' in request.path:
        response = jsonify({'message': "Couldn't find resource. Please login and t
        response.status_code = 404
        return response

    login_url = get_login_url(next=request.url, external=False)

    return redirect(login_url)
```

Example 15

Project: *hypergen* Author: *runekaagaard* File: *_hypergen.py* GNU General Public License v3.0 5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs):
    from flask import request
    return hypergen(
        func,
        *args,
        as_deltas=request.is_xhr,
        auto_id=True,
        id_prefix=request.get_json()["id_prefix"] if request.is_xhr else "",
        liveview=True,
        **kwargs)
```

Example 16

Project: *hypergen* Author: *runekaagaard* File: *hypergen.py* GNU General Public License v3.0 5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs):
    from flask import request
    return hypergen(
        func,
        *args,
        as_deltas=request.is_xhr,
        auto_id=True,
        id_prefix=request.get_json()["id_prefix"] if request.is_xhr else "",
        liveview=True,
        **kwargs)
```

Example 17

Project: *hypergen* Author: *runekaagaard* File: *hypergen.old.py* GNU General Public License v3.0 5 vo

```
def flask_liveview_hypergen(func, *args, **kwargs):
    from flask import request
    return hypergen(
        func,
        *args,
        as_deltas=request.is_xhr,
        auto_id=True,
        liveview=True,
        **kwargs)
```

Example 18

Project: *--Awesome-Python--* Author: *JoMingyu* File: *8. Other Data - request.headers & request.uri & etc.py* GNU General Public License v3.0 5 vo

```
def index():
    # request 객체에는 수많은 속성들이 존재한다
    # werkzeug.wrappers.BaseRequest에서 @cached_property나 @property로 처리된 프로퍼티들
    print(request.host, request.remote_addr)
    print(request.method, request.uri, request.full_url)
    print(request.headers)
    print(request.is_xhr)

    return 'hello'
```

Example 19

Project: *mlapi.io* Author: *Ermlab* File: *app.py* MIT License 5 vo

```
def save_request(response, data_type=None, data=None):
    '''Function invoking request saving in database
    '''
    try:
        dbc.save_request(
                    request_type=request.method,
                    request_url=request.url,
                    response = response,
                    user_id = current_identity['user_id'] if current_identity else
                    is_xhr = request.is_xhr,
                    headers = str(request.headers),
                    data_type = data_type,
                    data = data)
    except:
        logging.warning("There was an error while saving request data to DB. {}".f
```

Example 20

Project: *marvin* Author: *BotDevGroup* File: *web.py* MIT License 5 vo

```
def unauthorized():
    if not request.is_xhr:
        return redirect(url_for('marvinbot.login'))
    return jsonify({}), 403
```

Example 21

Project: *renrenBackup* Author: *whusnoopy* File: *web.py* MIT License 5 vo

```
def render_template(template_name, **kwargs):
    if request.is_xhr:
        return jsonify(success=1, **kwargs)

    return flask_render(template_name, **kwargs)
```

Example 22

Project: *scioncc* Author: *scionrep* File: *service_gateway.py* BSD 2-Clause "Simplified" License 5 vo

```
def json_response(self, response_data):
        """Private implementation of standard flask jsonify to specify the use of
        """
        resp_obj = json_dumps(response_data, default=encode_ion_object, indent=Nor
        resp = self.response_class(resp_obj, mimetype=CONT_TYPE_JSON)
        if self.develop_mode and (self.set_cors_headers or ("api_key" in request.a
            self._add_cors_headers(resp)
        self._log_request_response(CONT_TYPE_JSON, resp_obj, len(resp_obj))
        return resp
```

Example 23

Project: *calibre-web* Author: *janeczku* File: *web.py* GNU General Public License v3.0 4 vo

```
def show_book(book_id):
    entries = db.session.query(db.Books).filter(db.Books.id == book_id).filter(com
    if entries:
        for index in range(0, len(entries.languages)):
            try:
                entries.languages[index].language_name = LC.parse(entries.language
                    get_locale())
            except UnknownLocaleError:
                entries.languages[index].language_name = _(
                    isoLanguages.get(part3=entries.languages[index].lang_code).nam
        cc = get_cc_columns()
        book_in_shelfs = []
        shelfs = ub.session.query(ub.BookShelf).filter(ub.BookShelf.book_id == boo
        for entry in shelfs:
            book_in_shelfs.append(entry.shelf)

        if not current_user.is_anonymous:
            if not config.config_read_column:
                matching_have_read_book = ub.session.query(ub.ReadBook).\
                    filter(and_(ub.ReadBook.user_id == int(current_user.id), ub.Re
                have_read = len(matching_have_read_book) > 0 and matching_have_rea
            else:
                try:
```

```
                    matching_have_read_book = getattr(entries, 'custom_column_'+st
                    have_read = len(matching_have_read_book) > 0 and matching_have
                except KeyError:
                    log.error("Custom Column No.%d is not exisiting in calibre dat
                    have_read = None

        else:
            have_read = None

        entries.tags = sort(entries.tags, key=lambda tag: tag.name)

        entries = order_authors(entries)

        kindle_list = check_send_to_kindle(entries)
        reader_list = check_read_formats(entries)

        audioentries = []
        for media_format in entries.data:
            if media_format.format.lower() in constants.EXTENSIONS_AUDIO:
                audioentries.append(media_format.format.lower())

        return render_title_template('detail.html', entry=entries, audioentries=au
                                     is_xhr=request.is_xhr, title=entries.title,
                                     have_read=have_read, kindle_list=kindle_list,
    else:
        flash(_(u"Error opening eBook. File does not exist or file is not accessib
        return redirect(url_for("web.index"))
```

Example 24

Project: *calibre-web* Author: *janeczku* File: *shelf.py* GNU General Public License v3.0 4 vo

```
def add_to_shelf(shelf_id, book_id):
    shelf = ub.session.query(ub.Shelf).filter(ub.Shelf.id == shelf_id).first()
    if shelf is None:
        log.error("Invalid shelf specified: %s", shelf_id)
        if not request.is_xhr:
            flash(_(u"Invalid shelf specified"), category="error")
            return redirect(url_for('web.index'))
        return "Invalid shelf specified", 400

    if not shelf.is_public and not shelf.user_id == int(current_user.id):
        log.error("User %s not allowed to add a book to %s", current_user, shelf)
        if not request.is_xhr:
            flash(_(u"Sorry you are not allowed to add a book to the the shelf: %(
                  category="error")
            return redirect(url_for('web.index'))
        return "Sorry you are not allowed to add a book to the the shelf: %s" % sh

    if shelf.is_public and not current_user.role_edit_shelfs():
        log.info("User %s not allowed to edit public shelves", current_user)
        if not request.is_xhr:
            flash(_(u"You are not allowed to edit public shelves"), category="erro
            return redirect(url_for('web.index'))
        return "User is not allowed to edit public shelves", 403

    book_in_shelf = ub.session.query(ub.BookShelf).filter(ub.BookShelf.shelf == sh
                                         ub.BookShelf.book_id == book_id).first()
    if book_in_shelf:
        log.error("Book %s is already part of %s", book_id, shelf)
        if not request.is_xhr:
            flash(_(u"Book is already part of the shelf: %(shelfname)s", shelfname
```

```
            return redirect(url_for('web.index'))
        return "Book is already part of the shelf: %s" % shelf.name, 400

    maxOrder = ub.session.query(func.max(ub.BookShelf.order)).filter(ub.BookShelf.
    if maxOrder[0] is None:
        maxOrder = 0
    else:
        maxOrder = maxOrder[0]

    ins = ub.BookShelf(shelf=shelf.id, book_id=book_id, order=maxOrder + 1)
    ub.session.add(ins)
    ub.session.commit()
    if not request.is_xhr:
        flash(_(u"Book has been added to shelf: %(sname)s", sname=shelf.name), cat
        if "HTTP_REFERER" in request.environ:
            return redirect(request.environ["HTTP_REFERER"])
        else:
            return redirect(url_for('web.index'))
    return "", 204
```

Example 25

Project: *white* Author: *whiteclover* File: *patch.py* GNU General Public License v2.0 4 vo

```
def jsonify(value):
    """Creates a :class:`~flask.Response` with the JSON representation of
    the given arguments with an `application/json` mimetype.  The arguments
    to this function are the same as to the :class:`dict` constructor.

    Example usage::

        from flask import jsonify

        class User(object):
            def __json__(self):
                return dict(username=g.user.username,
                            email=g.user.email,
                            id=g.user.id)

        @app.route('/_get_current_user')
        def get_current_user():
            return jsonify(user)

    This will send a JSON response like this to the browser::

        {
            "username": "admin",
            "email": "admin@localhost",
            "id": 42
        }

    For security reasons only objects are supported toplevel.  For more
    information about this, have a look at :ref:`json-security`.

    This function's response will be pretty printed if it was not requested
    with ``X-Requested-With: XMLHttpRequest`` to simplify debugging unless
    the ``JSONIFY_PRETTYPRINT_REGULAR`` config parameter is set to false.

    """
    indent = None
    if current_app.config['JSONIFY_PRETTYPRINT_REGULAR'] \
            and not request.is_xhr:
```

```
    indent = 2
return current_app.response_class(dumps(value,
                                        indent=indent),
                                  mimetype='application/json')
```