Python flask.request.sid() Examples

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

```
def spectate_draft(message):
    room_to_join = message['room_id']
    if room_to_join not in room_map:
        print "'{}' doesn't exist as a room".format(room_to_join)
        emit('spectate_error',
             {
                 'message': 'Room {} does not exist'.format(room_to_join)
             })
        return
    room = room_map[room_to_join]
    room.spectator_list.append(request.sid)
    emit('spectate_join_success', {
        'room_id': room_to_join,
        'sid': request.sid
    })
    broadcast_to_spectator(request.sid, room.get_spectator_data())
```

Example 2

Project: *slurk* Author: *clp-research* File: *events.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def connect():
    current_user.session_id = request.sid
    log_event("connect", current_user)
    db.session.commit()
    if current_user.rooms.count() == 0:
        current_user.rooms.append(current_user.token.room)
    for room in current_user.rooms:
        join_room(room.name)
        if room not in current_user.current_rooms:
            current_user.current_rooms.append(room)
        socketio.emit('status', {
            'type': 'join',
            'user': {
                'id': current_user.id,
                'name': current_user.name,
            },
            'room': room.name,
            'timestamp': timegm(datetime.now().utctimetuple())
        }, room=room.name)
        log_event("join", current_user, room)
    db.session.commit()
```

Example 3

Project: *vantage* Author: *IKNL* File: *__init__.py* Apache License 2.0 6 vo

```
def read_and_forward_pty_output(fd, sid, child):
    max_read_bytes = 1024 * 20

    # fd = session.fd
    timeout_sec = 0.01

    while assert_running_interpreter(child=child):
        socketio.sleep(timeout_sec)

        (rs, ws, es) = select.select([fd], [], [], timeout_sec)

        for r in rs:
            output = os.read(r, max_read_bytes).decode()
            socketio.emit(
                "pty-output",
                {"output": output},
                namespace="/pty",
                room=sid,
            )
```

Example 4

Project: *flask-real-time-map* Author: *hedderich* File: *views.py* MIT License 6 vo

```
def send_initial_data():
    """
    When a new client connects, send the latest valid waypoint of every vehicle
    that showed up in the last 10 minutes back to this client.

    """
    since = datetime.utcnow() - timedelta(minutes=10)
    entries = models.VehicleLocationLog.get_latest_entries(since)

    for entry in entries:
        socketio.emit('update_location',
                      {'vehicle_id': entry.vehicle.vehicle_uuid,
                       'lat': entry.lat,
                       'lng': entry.lng},
                      namespace='/vehicles',
                      room=request.sid)
```

Example 5

Project: *full-stack-flask-smorest* Author: *ssfdust* File: *decorators.py* Apache License 2.0 6 vo

```
def auth_socket(func):
    """
    验证socketio的连接是否合法

    通过md5加密的连接来连接
    """

    @wraps(func)
    def decorated_function(*args, **kwargs):
        user_id = request.args.get("user_id", None)
        if user_id is None:
            disconnect(sid=request.sid)
        else:
            session = SessionManager(user_id)
            status = session.check_session()
            if not status:
                disconnect(request.sid)

        return func(*args, **kwargs)

    return decorated_function
```

Example 6

Project: *generals.io_copy* Author: *mcfx0* File: *server.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def on_change_game_conf(data):
        tmp={}
        tmp['width_ratio']=chkfloat(data['width_ratio'],0,1)
        tmp['height_ratio']=chkfloat(data['height_ratio'],0,1)
        tmp['city_ratio']=chkfloat(data['city_ratio'],0,1)
        tmp['mountain_ratio']=chkfloat(data['mountain_ratio'],0,1)
        tmp['swamp_ratio']=chkfloat(data['swamp_ratio'],0,1)
        tmp['speed']=chkfloat(data['speed'],0.25,16)
        tmp['custom_map']=unicode(data['custom_map'])
        if gr_id.has_key(request.sid) and len(tmp['custom_map'])>=0 and len(tmp['
                gid=gr_id[request.sid]
                ioroom=getval(gid)
                mess_q=[]
                if gr_players[gid][0][0]==request.sid:
                        for i in tmp:
                                if tmp[i]!=gr_conf[gid][i]:
                                        mess_q.append(i)
                        gr_conf[gid]=tmp
                emit('room_update',gen_game_conf(gid),room='game_'+ioroom)
                for i in mess_q:
                        send_system_message(ioroom,gr_players[gid][0][1]+' changed
```

Example 7

Project: *dino* Author: *thenetcircle* File: *api.py* Apache License 2.0 5 vo

```
def on_disconnect() -> (int, None):
    """
    when a client disconnects or the server no longer gets a ping response from th

    :return json if ok, {'status_code': 200}
    """
    user_id = str(environ.env.session.get(SessionKeys.user_id.value))
    try:
        sid = request.sid
    except Exception as e:
        logger.error('could not get sid from request: {}'.format(str(e)))
        logger.exception(traceback.format_exc())
        environ.env.capture_exception(sys.exc_info())
        sid = ''

    data = {
        'verb': 'disconnect',
        'actor': {
            'id': user_id,
            'content': sid
        }
    }
    if not environ.env.config.get(ConfigKeys.TESTING):
        # only used for single-session restrictions
        if environ.env.connected_user_ids.get(user_id) == sid:
            del environ.env.connected_user_ids[user_id]

    activity = as_parser(data)
    environ.env.observer.emit('on_disconnect', (data, activity))
    return ECodes.OK, None
```

Example 8

Project: *dashboard* Author: *pujansrt* File: *app.py* MIT License 5 vo

```
def test_disconnect():
    print('Client disconnected', request.sid)
```

Example 9

Project: *dashboard* Author: *pujansrt* File: *app_namespace.py* MIT License 5 vo

```
def on_disconnect(self):
        print('Client disconnected', request.sid)
```

Example 10

Project: *blockexplorer* Author: *GenesisKernel* File: *socketio_namespace.py* GNU General Public License v2.0 5 vo

```
def on_disconnect(self):
        print('Client disconnected', request.sid)
```

Example 11

Project: *blockexplorer* Author: *GenesisKernel* File: *socketio.py* GNU General Public License v2.0 5 vo

```
def on_disconnect(self):
        print('Client disconnected', request.sid)
```

Example 12

Project: *xp-game* Author: *codeselfstudy* File: *app.py* GNU General Public License v3.0 5 vo

```
def handle_connect():
    # Authentication can go here
    log.game_event(f'client_connected: {request.sid}')
```

Example 13

Project: *xp-game* Author: *codeselfstudy* File: *app.py* GNU General Public License v3.0 5 vo

```
def handle_disconnect():
    log.game_event(f'client_disconnected: {request.sid}')
    ticker.enqueue_client_message({'kind': 'Despawn'}, request.sid)
```

Example 14

Project: *xp-game* Author: *codeselfstudy* File: *app.py* GNU General Public License v3.0 5 vo

```
def handle_event(event_dict):
    """
    Events are of the format { kind: str, data: {...}}
    """
    global client_names
    event = from_dict(event_dict, ClientEvent)

    if event and event.kind == ClientEvent.LOGIN_EVENT_KIND:
        # TODO-- get from_dict to parse recursively
        client_names[request.sid] = event.detail['character_name']
        ticker.enqueue_client_message({'kind': 'Spawn'}, request.sid)
```

Example 15

Project: *xp-game* Author: *codeselfstudy* File: *app.py* GNU General Public License v3.0 5 vo

```
def handle_action(action):
    log.game_event(f'action: {action} by {request.sid}')
    ticker.enqueue_client_message(action, request.sid)
```

Example 16

Project: *xp-game* Author: *codeselfstudy* File: *app.py* GNU General Public License v3.0 5 vo

```
def handle_chat(incoming):
    """Respond to `chat` message from the frontend.

    `incoming` is `{'body': 'the message content'}`.
    """
    trimmed_message = incoming['body'].strip()
    if trimmed_message:
        outgoing = {
            'id': client_names.get(request.sid),
            'body': sanitize(trimmed_message),
        }
        log.game_event(f'chat_message: {outgoing}')
        socketio.emit('chat', outgoing)
```

Example 17

Project: *JumpAwake* Author: *nickwu241* File: *app.py* MIT License 5 vo

```
def join(user):
    print("[WS] {} connected via join".format(request.sid))
    clients[request.sid] = {
        'user': user,
        'data': models.User(user).data
    }
    if len(clients) >= 2:
        __emit_jumps()
```

Example 18

Project: *JumpAwake* Author: *nickwu241* File: *app.py* MIT License 5 vo

```
def leave(data):
    print("[WS] {} disconnected via leave".format(request.sid))
    clients.pop(request.sid, None)
```

Example 19

Project: *JumpAwake* Author: *nickwu241* File: *app.py* MIT License 5 vo

```
def connect():
    print("[WS] {} connected".format(request.sid))
```

Example 20

Project: *JumpAwake* Author: *nickwu241* File: *app.py* MIT License 5 vo

```
def disconnect():
    print("[WS] {} disconnected".format(request.sid))
    clients.pop(request.sid, None)

# Backend Routing
```

Example 21

Project: *SpyPartyDraft* Author: *LtHummus* File: *SpyPartyDraft.py* MIT License 5 vo

```
def test_disconnect():
    print('Client disconnected', request.sid)
```

Example 22

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vo

```
def register_module(self, path, sid):
        "Register a module with a specified path"
        if sid in self._run_number_for_sid:
            self._run_number_for_sid[sid] = 0
            return
        print('Register module:', path, 'sid:', sid)
        self._run_number_for_sid[sid] = 0
        if path in self._sids_for_path:
            sids = self._sids_for_path[path]
            sids.add(sid)
        else:
            self._sids_for_path[path] = set([sid])
```

Example 23

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vo

```
def unregister_module(self, sid):
        "Unregister a specified path"
        if sid in self._run_number_for_sid:
            del self._run_number_for_sid[sid]
        for sids in self._sids_for_path.values():
            if sid in sids:
                sids.remove(sid)
                return
```

Example 24

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vo

```
def sids_for_path(self, path):
        "Get the sid list from a path"
        return self._sids_for_path.get(path, set())
```

Example 25

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def sid_run_number(self, sid):
        "Return the last run_number sent for the specified sid"
        return self._run_number_for_sid.get(sid, 0)
```

Example 26

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def _prevent_tick(self, sid, run_number, ack):
        if ack:
            self._run_number_for_sid[sid] = run_number
        else:
            logging.debug('Ack not well received')
            print('Preventing ticks for', sid)
```

Example 27

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def emit_tick(self, path, run_number, payload=None):
        "Emit a tick unless it has not been acknowledged"
        sids = self.sids_for_path(path)
        for sid in sids:
            if self._run_number_for_sid[sid] == 0:
                #print('Emiting tick for', sid, 'in path', path)
                json_ = {'run_number': run_number}
                if payload is not None: json_['payload'] = payload
                socketio.emit('tick', json_, room=sid,
                              callback=partial(self._prevent_tick, sid, run_numbe
            #else:
            #    #print('No tick for', sid, 'in path', path)
        time.sleep(0) # yield thread
```

Example 28

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def _on_connect():
    print('socketio connect ', request.sid)
```

Example 29

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def _on_disconnect():
    progressivis_bp.unregister_module(request.sid)
    print('socketio disconnect ', request.sid)
```

Example 30

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vc

```
def _on_scheduler(short=False):
    scheduler = progressivis_bp.scheduler
    #print('socketio scheduler called')
    progressivis_bp.register_module('scheduler', request.sid)
    #print(progressivis_bp._sids_for_path)
    assert request.sid in progressivis_bp.sids_for_path('scheduler')
    return scheduler.to_json(short)
```

Example 31

Project: *progressivis* Author: *jdfekete* File: *app.py* BSD 2-Clause "Simplified" License 5 vo

```
def _on_module_get(path):
    module = path_to_module(path)
    if module is None:
        return {'status': 'failed',
                'reason': 'unknown module %s'%path}
    progressivis_bp.register_module(module.name, request.sid)
    module.set_end_run(progressivis_bp.tick_module) # setting it multiple time is
    #print('on_module_get', path)
    return module.to_json()
```

Example 32

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def __init__(self, socketio, sid):
        self._sock = socketio
        self._sid = sid
        self._emit('started', {})
```

Example 33

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def map_kill(sid):
    if sid in sid_program_map:
        sid_program_map[sid].kill()
        del sid_program_map[sid]
```

Example 34

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def compile(msg):
    sid = request.sid
    print("compile:", sid)
    prog_dir = os.path.join(settings.sessions_dir, sid)
    reset_dir(prog_dir)
    prog = compiler.Program(msg, prog_dir, Callbacks(socketio, sid))
    prog.spawn_bg()
    map_kill(sid)
    sid_program_map[sid] = prog
```

Example 35

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def kill(msg):
    sid = request.sid
    print("kill:", sid)
    map_kill(sid)
```

Example 36

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def stdin(data):
    sid = request.sid
    print("stdin:", sid)
    if sid in sid_program_map:
        sid_program_map[sid].stdin(data.encode('utf-8'))
```

Example 37

Project: *java-play2* Author: *johncf* File: *server.py* MIT License 5 vo

```
def disconnect():
    sid = request.sid
    print("disconnected:", sid)
    map_kill(sid)
```

Example 38

Project: *visualizer* Author: *nextgenevoting* File: *__init__.py* GNU Affero General Public License v3.0 5 vo

```
def on_join(data):
    from app.api.syncService import SyncType, fullSync

    electionID = data['election']
    for room in rooms():
        if room != request.sid:
            leave_room(room)
    join_room(electionID)

    from app.api.syncService import emitToClient, SyncType

    fullSync(electionID, SyncType.SENDER_ONLY)

    emitToClient('joinAck', electionID, SyncType.SENDER_ONLY)
```

Example 39

Project: *visualizer* Author: *nextgenevoting* File: *syncService.py* GNU Affero General Public License v3.0 5 vo

```
def emitToClient(messageName, payload, syncType, room = None):
    if syncType == SyncType.ROOM:
        socketio.emit(messageName, payload, room=room)
    elif syncType == SyncType.BROADCAST:
        socketio.emit(messageName, payload, broadcast=True)
    else:
        socketio.emit(messageName, payload, room=request.sid)

# LISTENERS
```

Example 40

Project: *PlayChess* Author: *neverwannafly* File: *routes.py* MIT License 5 vo

```
def handle_connection(message):
    USER_DICT['current_user_' + session['username']].sessionid = request.sid
    emit('user_connect', "Hello!")
```

Example 41

Project: *hacks* Author: *misakar* File: *events.py* MIT License 5 vo

```
def handle_connected():
    """
    connect: 连接事件
    """
    sockets.append(request.sid)
```

Example 42

Project: *hacks* Author: *misakar* File: *events.py* MIT License 5 vo

```
def handle_disconnected():
    """
    disconnect: 连接断开事件
    """
    sockets.remove(request.sid)
```

Example 43

Project: *hacks* Author: *misakar* File: *events.py* MIT License 5 vo

```
def handle_connected():
    # clients.append(request.namespace)
    sockets.append(request.sid)
```

Example 44

Project: *hacks* Author: *misakar* File: *events.py* MIT License 5 vo

```
def handle_disconnected():
    # clients.remove(request.namespace)
    sockets.remove(request.sid)
```

Example 45

Project: *slurk* Author: *clp-research* File: *events.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def ready():
    for room in current_user.current_rooms:
        socketio.emit("joined_room", dict(user=current_user.id, room=room.name), r
```

Example 46

Project: *vantage* Author: *IKNL* File: *__init__.py* Apache License 2.0 5 vo

```
def start_interpreter():
    # create child process attached to a pty we can read from and write to
    if TERMINAL_AVAILABLE:
        env = app.config['environment']
        cmd = ['ipython', '-m', 'pytaskmanager.server.shell', '-i', '--', env]

        log.debug("opening pty")
        master_fd, slave_fd = pty.openpty()

        log.debug("starting process")
        child = subprocess.Popen(
            cmd,
            stdin=slave_fd,
            stdout=slave_fd,
            stderr=slave_fd
        )

        log.debug("adding process details to session")
        session.child = child
        session.fd = master_fd
        session.master_fd = master_fd
        session.slave_fd = slave_fd

        log.debug("setting window size")
        set_winsize(master_fd, 50, 50)

        log.debug("starting background task")
        socketio.start_background_task(
            read_and_forward_pty_output,
            fd=master_fd,
            sid=request.sid,
            child=child,
        )
        log.debug("ipython terminal backend started")
    else:
        log.debug("ipython terminal not available")
```

Example 47

Project: *vantage* Author: *IKNL* File: *__init__.py* Apache License 2.0 5 vo

```
def disconnect_pty():
    print(f'Client {request.sid} disconnected')
    # app.config['socket_connections'].remove(request.sid)
    # session["child"].kill()
    try:
        session.child.kill()
    except Exception as e:
        log.error("Could not kill interpreter backend!?")
        log.exception(e)
```

Example 48

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

```
def addDevice(message):
    try:
        global devices
        deviceClassName = message['classname']
        deviceTitle = message['title']
        devicePort = None
        deviceCategory = (message.has_key('category')) and message['category'] or
        # if port is larger than 5000, then it is an remote device,
        # we should create an room based on its sid
        # if port is allready used, then update the device and make it accessable
        if message.has_key('port'):
            devicePort = message['port']
            for i in range(len(devices)):
                if 'port' in dir(devices[i]):
                    if devicePort > 5000:
                        join_room(request.sid)
                    if devices[i].port == devicePort:
                        # update device
                        deviceID = devices[i].id
                        devices[i] = Device.updateDevice(deviceID, deviceClassName
                        emit('remove', {'id':deviceID}, broadcast = True)
                        emit('add', devices[i].description(), broadcast = True)
                        return
        # add device as usual
        deviceID = devices[len(devices) - 1].id + 1
        message['id'] = deviceID
        device = Device.addDevice(deviceID, deviceClassName, deviceTitle, devicePo
        devices.append(device)
        emit('add', device.description(), broadcast = True)
    except Exception as e:
        logging.debug(e)

# Remove a device and notify all clients
```

Example 49

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

```
def connect():
    global tokens
    if (request.args.get('token', '') not in tokens):
        disconnect()
        return
    global thread
    if thread is None:
        thread = socketio.start_background_task(target=background_thread)
    # session['thread'] = socketio.start_background_task(background_thread, sessic
    # emit('response', {'data': 'Connected'})
    print('Client connected: ', request.sid)


# Disconnect the client and remove device if it has one
# Current: no accessable funtion
```

Example 50

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

```
def test_disconnect():
    # here we have to use the sid to locate the the disconnected device
    # and if it exists, remove its room and set it as unaceessable
    try:
        close_room(request.sid)
    except Exception as e:
        logging.debug(e)

    # Remove token if exist
    global tokens
    token = request.args.get('token', '')
    if token in tokens:
        tokens.remove(token)
        logging.debug("Token removed:" + token)
    # disconnect()
    print('Client disconnected: ', request.sid)


# @app.route('/')
# def index():
#     return render_template('index.html', async_mode=socketio.async_mode)


# @socketio.on('request', namespace = mynamespace)
# def broadcast_message(message):
#     message = {'data': {'message': 'I am the message'}}
#     emit('response',
#          {'data': message['data']['message']},
#          broadcast=True)
```