Python flask.redirect() Examples

The following are code examples for showing how to use *flask.redirect()*. 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: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0 7 vo

```
def register():
    """注册：第一步：输入学号"""
    if request.method == 'GET':
        return render_template('user/register.html')
    else:
        if not request.form.get("xh", None):  # 表单为空
            flash(MSG_EMPTY_USERNAME)
            return redirect(url_for("user.register"))

        _session_save_student_to_register_(request.form.get("xh", None))

        # 如果输入的学号已经注册，跳转到登录页面
        if User.exist(session[SESSION_STUDENT_TO_REGISTER].sid_orig):
            flash(MSG_ALREADY_REGISTERED)
            return redirect(url_for('user.login'))

        return redirect(url_for('user.register_choice'))
```

Example 2

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *signals.py* Apache License 2.0 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 3

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *signals.py* Apache License 2.0 6 vo

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')

        recorded = []
        def record(sender, message, category):
            recorded.append((message, category))

        flask.message_flashed.connect(record, app)
        try:
            client = app.test_client()
            with client.session_transaction():
                client.get('/')
                self.assert_equal(len(recorded), 1)
                message, category = recorded[0]
                self.assert_equal(message, 'This is a flash message')
                self.assert_equal(category, 'notice')
        finally:
            flask.message_flashed.disconnect(record, app)
```

Example 4

Project: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0 6 vo

```
def register_by_email():
    """注册：第三步：使用邮箱验证注册"""
    if not session.get(SESSION_STUDENT_TO_REGISTER, None):  # 步骤异常，跳回第一步
        return redirect(url_for('user.register'))

    sid_orig = session[SESSION_STUDENT_TO_REGISTER].sid_orig

    if User.exist(sid_orig):
        return render_template("common/error.html", message=MSG_ALREADY_REGISTERE

    request_id = IdentityVerification.new_register_request(sid_orig, "email", ID_

    with tracer.trace('send_email'):
        try:
            rpc_result = Auth.register_by_email(request_id, sid_orig)
        except Exception as e:
            return handle_exception_with_error_page(e)

    if rpc_result['acknowledged']:
        return render_template('user/emailSent.html', request_id=request_id)
    else:
        return render_template('common/error.html', message=MSG_INTERNAL_ERROR)
```

Example 5

Project: *pysos* Author: *xmnlab* File: *views.py* GNU General Public License v3.0 6 vo

```
def register_view(self):
        form = RegistrationForm(request.form)
        if helpers.validate_form_on_submit(form):
            user = User()

            form.populate_obj(user)
            # we hash the users password to avoid saving it as plaintext in the db
            # remove to use plain text:
            user.password = generate_password_hash(form.password.data)

            db_session.add(user)
            db_session.commit()

            login.login_user(user)
            return redirect(url_for('.index'))
        link = '<p>Already have an account? <a href="' + url_for('.login_view') +
        self._template_args['form'] = form
        self._template_args['link'] = link
        return super(PySOSAdminView, self).index()
```

Example 6

Project: *zmirror* Author: *aploium* File: *custom_func.sample.py* MIT License 6 vo

```
def custom_prior_redirect_func(request, parse):
    """
    用于在 prior_request_redirect 阶段的自定义重定向

    若返回一个 flask.Response 对象, 则执行重定向, 直接返回这个 Response
    若返回None, 则不进行重定向

    不应该修改parse变量 (添加头和cookie除外)

    详见 `config_default.py` 中 `Custom Redirection` 部分

    :param request: flask request object
    :type request: Request
    :param parse: the zmirror parse variable
    :type parse: ZmirrorThreadLocal
    :rtype: Union[Response, None]
    """
    print(request.url, parse.remote_url)

    from flask import redirect

    # 如果你想重定向, 请使用这句
    # return redirect("/location/you/want/redirect/to")

    return None  # 不进行自定义重定向
```

Example 7

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

```
def test_flash_signal(self):
        app = flask.Flask(__name__)
        app.config['SECRET_KEY'] = 'secret'

        @app.route('/')
        def index():
            flask.flash('This is a flash message', category='notice')
            return flask.redirect('/other')
```

```
    recorded = []
    def record(sender, message, category):
      recorded.append((message, category))

    flask.message_flashed.connect(record, app)
    try:
      client = app.test_client()
      with client.session_transaction():
        client.get('/')
        self.assert_equal(len(recorded), 1)
        message, category = recorded[0]
        self.assert_equal(message, 'This is a flash message')
        self.assert_equal(category, 'notice')
    finally:
      flask.message_flashed.disconnect(record, app)
```

Example 8

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 9

Project: *Nurevam* Author: *Maverun* File: *custom commands.py* MIT License 6 vo

```
def edit_customcmd(cog,server_id, name):
  new_name = request.form.get("cmd_name")
  content = request.form.get("cmd_content")
  brief = request.form.get("cmd_brief")
  if new_name == "":
    flash("Name cannot be blank!","warning")
  else:
    # delete old database
    db.hdel("{}:Customcmd:content".format(server_id), name)
    db.hdel("{}:Customcmd:brief".format(server_id), name)
    db.hdel("{}:Customcmd:owner".format(server_id), name)
    db.srem("{}:Customcmd:name".format(server_id), name)
    # adding, if there is a way to rename them in hash, that would be great...
    db.hset("{}:Customcmd:content".format(server_id), new_name, content)
    db.hset("{}:Customcmd:brief".format(server_id), new_name, brief)
    db.hset("{}:Customcmd:owner".format(server_id), new_name, utils.session["u
    db.sadd("{}:Customcmd:name".format(server_id), new_name)
    db.sadd("{}:Customcmd:update_delete".format(server_id), new_name)
```

```
        flash("Update from {} to {}!".format(name,new_name), "success")
    return redirect(url_for("customcmd.customcmd", server_id=server_id, cog="cust
```

Example 10

Project: *Nurevam* Author: *Maverun* File: *utils.py* MIT License 6 vo

```
def require_auth(f):
    @wraps(f)
    def wrapper(*args, **kwargs):
        # Does the user have an api_token?
        api_token = session.get('api_token')
        if api_token is None:
            log.info("No api token, redirect to login")
            return redirect(url_for('login'))

        # Does his api_key is in the db?
        user_api_key = db.get('user:{}:api_key'.format(api_token['user_id']))
        if user_api_key != api_token['api_key']:
            log.info("API key dont match, refresh them")
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return wrapper
```

Example 11

Project: *Nurevam* Author: *Maverun* File: *profile.py* MIT License 6 vo

```
def update_profile(): #Update a setting.
    list_point = dict(request.form)
    list_point.pop('_csrf_token',None)
    path = "Profile:{}".format(session['user']['id'])
    warning = False
    warning_msg = "One of those have failed, Please double check {} "
    warning_list =[]
    for  x in list_point:
        print(x)
        if request.form.get(x) == "":
            db.hdel(path,x)
            continue
        elif x == "osu":
            results = osu_api.get_user(request.form.get(x))
            if results == []:
                warning = True
                warning_list.append(x)
                continue
        db.hset(path,x,request.form.get(x))
    if warning:
        flash(warning_msg.format(",".join(warning_list)), 'warning')
    else:
        flash('Settings updated!', 'success')
    return redirect(url_for('profile.profile'))
```

Example 12

Project: *Nurevam* Author: *Maverun* File: *profile.py* MIT License 6 vo

```
def anilist_request():
    code = request.args.get("code")
    header = {'Content-Type': 'application/json','Accept': 'application/json'}
    r = requests.post("https://anilist.co/api/v2/oauth/token",json = {
        'client_id':str(utils.data_info.anilist_id),
        'client_secret':utils.data_info.anilist_token,
        'redirect_uri':url_for('profile.anilist_request',_external=True),
        'grant_type': 'authorization_code',
        'code':code},headers=header)
    data =r.json()
    user = session['user']

    db.hmset("Profile:{}:Anilist".format(user["id"]),data)
    print("Successfully create token for ",user["id"]," - ",user["username"])
    flash("Anilist update!","success")
    return redirect(url_for('profile.profile'))
```

Example 13

Project: *flask-template* Author: *pwgraham91* File: *login_view.py* MIT License 5 vo

```
def dev_login(user_id):
    if ENVIRONMENT == 'dev':
        login_user(db.session.query(User).get(user_id))
    return redirect(url_for('index'))
```

Example 14

Project: *flask-template* Author: *pwgraham91* File: *login_view.py* MIT License 5 vo

```
def logout():
    logout_user()
    return flask.redirect(flask.url_for('index'))
```

Example 15

Project: *flaskit* Author: *elston* File: *views.py* MIT License 5 vo

```
def login():
    # ..
    form = LoginForm(request.form)
    if request.method == 'POST':
        if form.validate_on_submit():
            login_user(form.user)
            redirect_url = request.args.get('next') or url_for('admin.index')
            return redirect(redirect_url)
        else:
            flash_errors(form)
    # ..
    return render_template('accounts/login.html', form=form)
```

Example 16

Project: *flaskit* Author: *elston* File: *views.py* MIT License 5 vo

```
def logout():
    logout_user()
    flash('You are logged out.', 'info')
    return redirect(url_for('admin.index'))
```

Example 17

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *testing.py* Apache License 2.0 5 vo

```
def test_redirect_keep_session(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testing'

        @app.route('/', methods=['GET', 'POST'])
        def index():
            if flask.request.method == 'POST':
                return flask.redirect('/getsession')
            flask.session['data'] = 'foo'
            return 'index'

        @app.route('/getsession')
        def get_session():
            return flask.session.get('data', '<missing>')

        with app.test_client() as c:
            rv = c.get('/getsession')
            assert rv.data == b'<missing>'

            rv = c.get('/')
            assert rv.data == b'index'
            assert flask.session.get('data') == 'foo'
            rv = c.post('/', data={}, follow_redirects=True)
            assert rv.data == b'foo'

            # This support requires a new Werkzeug version
            if not hasattr(c, 'redirect_client'):
                assert flask.session.get('data') == 'foo'

            rv = c.get('/getsession')
            assert rv.data == b'foo'
```

Example 18

Project: *Flask-Python-GAE-Login-Registration* Author: *orymeyer* File: *testing.py* Apache License 2.0 5 vo

```
def test_redirect_keep_session(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testing'

        @app.route('/', methods=['GET', 'POST'])
        def index():
            if flask.request.method == 'POST':
                return flask.redirect('/getsession')
            flask.session['data'] = 'foo'
            return 'index'

        @app.route('/getsession')
        def get_session():
            return flask.session.get('data', '<missing>')

        with app.test_client() as c:
            rv = c.get('/getsession')
            assert rv.data == b'<missing>'

            rv = c.get('/')
            assert rv.data == b'index'
            assert flask.session.get('data') == 'foo'
            rv = c.post('/', data={}, follow_redirects=True)
            assert rv.data == b'foo'
            # This support requires a new Werkzeug version
            if not hasattr(c, 'redirect_client'):
                assert flask.session.get('data') == 'foo'

            rv = c.get('/getsession')
            assert rv.data == b'foo'
```

Example 19

Project: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0 5 vc

```
def register_choice():
    """注册：第二步：选择注册方式"""
    if not session.get(SESSION_STUDENT_TO_REGISTER, None):  # 步骤异常，跳回第一步
        return redirect(url_for('user.register'))
    return render_template('user/registerChoice.html')
```

Example 20

Project: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0 5 vc

```
def register_by_password_success():
    """验证成功后跳转到用户首页"""
    return redirect(url_for("user.main"))
```

Example 21

Project: *everyclass-server* Author: *everyclass* File: *views.py* Mozilla Public License 2.0 5 vc

```
def reset_calendar_token():
    """重置日历订阅令牌"""
    CalendarToken.reset_tokens(session[SESSION_CURRENT_USER].sid_orig)
    flash("日历订阅令牌重置成功")
    return redirect(url_for("user.main"))
```

Example 22

Project: *pysos* Author: *xmnlab* File: *views.py* GNU General Public License v3.0 5 vc

```
def index(self):
        if not login.current_user.is_authenticated():
            return redirect(url_for('.login_view'))
        return super(PySOSAdminView, self).index()
```

Example 23

Project: *pysos* Author: *xmnlab* File: *views.py* GNU General Public License v3.0 5 vc

```
def login_view(self):
        # handle user login
        form = LoginForm(request.form)
        if helpers.validate_form_on_submit(form):
            user = form.get_user()
            login.login_user(user)

        if login.current_user.is_authenticated():
            return redirect(url_for('.index'))
        link = '<p>Don\'t have an account? <a href="' + url_for('.register_view')
        self._template_args['form'] = form
        self._template_args['link'] = link
        return super(PySOSAdminView, self).index()
```

Example 24

Project: *pysos* Author: *xmnlab* File: *views.py* GNU General Public License v3.0 5 vo

```
def logout_view(self):
        login.logout_user()
        return redirect(url_for('.index'))
```

Example 25

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_age
        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 26

Project: *Autoline* Author: *zjh1218* File: *views.py* Apache License 2.0 5 vo

```
def login():
    email = request.form["email"]
    password = request.form["password"]
    user = User.query.filter_by(email=email).first()

    if user is not None and user.verify_password(password):
        login_user(user, True)

        return redirect(url_for("main.dashboard"))

    return render_template('index.html', user=user)
```

Example 27

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

```
def test_redirect_keep_session(self):
        app = flask.Flask(__name__)
        app.secret_key = 'testing'

        @app.route('/', methods=['GET', 'POST'])
        def index():
            if flask.request.method == 'POST':
                return flask.redirect('/getsession')
            flask.session['data'] = 'foo'
            return 'index'

        @app.route('/getsession')
        def get_session():
            return flask.session.get('data', '<missing>')

        with app.test_client() as c:
            rv = c.get('/getsession')
            assert rv.data == b'<missing>'

            rv = c.get('/')
            assert rv.data == b'index'
            assert flask.session.get('data') == 'foo'
            rv = c.post('/', data={}, follow_redirects=True)
            assert rv.data == b'foo'

            # This support requires a new Werkzeug version
            if not hasattr(c, 'redirect_client'):
                assert flask.session.get('data') == 'foo'

            rv = c.get('/getsession')
            assert rv.data == b'foo'
```

Example 28

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

```
def before_request():
    if current_user.is_authenticated:
        current_user.ping()
        if not current_user.confirmed \
                and request.endpoint[:5] != 'auth.' \
                and request.endpoint != 'static':
            return redirect(url_for('auth.unconfirmed'))
```

Example 29

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

```
def unconfirmed():
    if current_user.is_anonymous or current_user.confirmed:
        return redirect(url_for('main.index'))
    return render_template('auth/unconfirmed.html')
```

Example 30

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

```
def logout():
    logout_user()
    flash('You have been logged out.')
    return redirect(url_for('main.index'))
```

Example 31

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

```
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        user = User(email=form.email.data,
                    username=form.username.data,
                    password=form.password.data)
        db.session.add(user)
        db.session.commit()
        token = user.generate_confirmation_token()
        send_email(user.email, 'Confirm Your Account',
                   'auth/email/confirm', user=user, token=token)
        flash('A confirmation email has been sent to you by email.')
        return redirect(url_for('auth.login'))
    return render_template('auth/register.html', form=form)
```

Example 32

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

```
def confirm(token):
    if current_user.confirmed:
        return redirect(url_for('main.index'))
    if current_user.confirm(token):
        flash('You have confirmed your account. Thanks!')
    else:
        flash('The confirmation link is invalid or has expired.')
    return redirect(url_for('main.index'))
```

Example 33

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

```
def resend_confirmation():
    token = current_user.generate_confirmation_token()
    send_email(current_user.email, 'Confirm Your Account',
               'auth/email/confirm', user=current_user, token=token)
    flash('A new confirmation email has been sent to you by email.')
    return redirect(url_for('main.index'))
```

Example 34

Project: *flasky* Author: *RoseOu* File: *views.py* MIT License 5 vc

```
def change_password():
    form = ChangePasswordForm()
    if form.validate_on_submit():
        if current_user.verify_password(form.old_password.data):
            current_user.password = form.password.data
            db.session.add(current_user)
            flash('Your password has been updated.')
            return redirect(url_for('main.index'))
        else:
            flash('Invalid password.')
    return render_template("auth/change_password.html", form=form)
```

Example 35

Project: *flasky* Author: *RoseOu* File: *views.py* MIT License 5 vc

```
def password_reset(token):
    if not current_user.is_anonymous:
        return redirect(url_for('main.index'))
    form = PasswordResetForm()
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first()
        if user is None:
            return redirect(url_for('main.index'))
        if user.reset_password(token, form.password.data):
            flash('Your password has been updated.')
            return redirect(url_for('auth.login'))
        else:
            return redirect(url_for('main.index'))
    return render_template('auth/reset_password.html', form=form)
```

Example 36

Project: *flasky* Author: *RoseOu* File: *views.py* MIT License 5 vc

```
def change_email_request():
    form = ChangeEmailForm()
    if form.validate_on_submit():
        if current_user.verify_password(form.password.data):
            new_email = form.email.data
            token = current_user.generate_email_change_token(new_email)
            send_email(new_email, 'Confirm your email address',
                       'auth/email/change_email',
                       user=current_user, token=token)
            flash('An email with instructions to confirm your new email '
                  'address has been sent to you.')
            return redirect(url_for('main.index'))
        else:
            flash('Invalid email or password.')
    return render_template("auth/change_email.html", form=form)
```

Example 37

Project: *flasky* Author: *RoseOu* File: *views.py* MIT License 5 vc

```
def change_email(token):
    if current_user.change_email(token):
        flash('Your email address has been updated.')
    else:
        flash('Invalid request.')
    return redirect(url_for('main.index'))
```

Example 38

Project: *Nurevam* Author: *Maverun* File: *app.py* MIT License 5 vo

```
def login():
    data_info.last_path = request.referrer
    log.info("User is logging in")
    scope = ['identify', 'guilds','email'] #email is for discourse rank purpose.
    discord = utils.make_session(scope=scope)
    authorization_url, state = discord.authorization_url(
        data_info.AUTHORIZATION_BASE_URL,
        access_type="offline"
    )
    session['oauth2_state'] = state
    return redirect(authorization_url)
```

Example 39

Project: *Nurevam* Author: *Maverun* File: *app.py* MIT License 5 vo

```
def confirm_login():
    log.info("Checking login....")
    # Check for state and for 0 errors
    state = session.get('oauth2_state')
    if not state or request.values.get('error'):
        return redirect(url_for('index'))

    # Fetch token
    discord = utils.make_session(state=state)
    discord_token = discord.fetch_token(
        data_info.TOKEN_URL,
        client_secret=data_info.OAUTH2_CLIENT_SECRET,
        authorization_response=request.url)
    if not discord_token:
        log.info("Not clear, returning")
        return redirect(url_for('index'))

    # Fetch the user
    user = utils.get_user(discord_token)
    # Generate api_key from user_id
    serializer = JSONWebSignatureSerializer(app.config['SECRET_KEY'])
    api_key = str(serializer.dumps({'user_id': user['id']}))
    # Store api_key
    db.set('user:{}:api_key'.format(user['id']), api_key)
    # Store token
    db.set('user:{}:discord_token'.format(user['id']), json.dumps(discord_token))
    # Store api_token in client session
    api_token = {
        'api_key': api_key,
        'user_id': user['id']
    }

    session.permanent = True
    session['api_token'] = api_token
    log.info("Clear, redirect...")
    if data_info.last_path and data_info.last_path != request.url_root: #if if it
        path = data_info.last_path
        data_info.last_path = None
        return redirect(path)
    data_info.last_path = None
    return redirect(url_for('after_login'))
```

Example 40

Project: *Nurevam* Author: *Maverun* File: *app.py* MIT License 5 vo

```
def logout():
    session.clear()
    return redirect(url_for('index'))
#Website Route
```

Example 41

Project: *Nurevam* Author: *Maverun* File: *app.py* MIT License 5 vo

```
def select_server():
    guild_id = request.args.get('guild_id')
    if guild_id:
        log.info("Got guild ID, {}".format(guild_id))
        return redirect(url_for('dashboard', server_id=int(guild_id)))

    user = utils.get_user(session['api_token'])
    guilds = utils.get_user_guilds(session['api_token'])
    user_servers = sorted(utils.get_user_managed_servers(user, guilds),key=lambda
    log.info("User servers: {}".format(user_servers))
    return render_template('select-server.html', user=user, user_servers=user_serv

#Core
```

Example 42

Project: *Nurevam* Author: *Maverun* File: *app.py* MIT License 5 vo

```
def update_core(server_id):
    config_delete = db.hgetall("{}:Config:Delete_MSG".format(server_id))
    print(config_delete)
    for x in config_delete:
        print(request.form.get(x))
        db.hset("{}:Config:Delete_MSG".format(server_id),x,request.form.get(x))
    db.set("{}:Config:Whisper".format(server_id),request.form.get("whisper"))
    db.set("{}:Config:CMD_Prefix".format(server_id),request.form.get("command_pref
    flash('Settings updated!', 'success')
    return redirect(url_for('core', server_id=server_id))
```

Example 43

Project: *Nurevam* Author: *Maverun* File: *welcome.py* MIT License 5 vo

```
def update_welcome(server_id):
    welcome_message = request.form.get('message')
    channel = request.form.get('channel')
    whisper_options = request.form.get('whisper')
    role_options = request.form.get("role")
    role_id = request.form.get("assign_role").split(',')
    delete_msg = request.form.get('delete_msg')
    delete_options = request.form.get("enable_delete")
    enable_msg = request.form.get("enable_message")
    if len(welcome_message) >= 2000 or welcome_message == "":
        flash("The welcome message need to be between 1-2000!", 'warning')
    else:
        try:
            delete_msg = int(delete_msg)
        except ValueError:
            flash('The delete message that you provided isn\'t an integer!', 'warr
            return redirect(url_for('plugin_welcome', server_id=server_id))
        db.hset('{}:Welcome:Message'.format(server_id), 'message', welcome_message
        db.hset('{}:Welcome:Message'.format(server_id), 'channel', channel)
        db.hset('{}:Welcome:Message'.format(server_id), 'whisper', whisper_options
        db.hset('{}:Welcome:Message'.format(server_id), 'delete_msg', delete_msg)
        db.hset('{}:Welcome:Message'.format(server_id), 'enable_delete', delete_op
        db.hset('{}:Welcome:Message'.format(server_id), 'enable_message', enable_m
        flash('Settings updated!', 'success')
    db.hset('{}:Welcome:Message'.format(server_id), 'role', role_options)
    db.delete("{}:Welcome:Assign_role".format(server_id))
    if len(role_id) > 0:
        db.sadd("{}:Welcome:Assign_Roles".format(server_id), *role_id)
    return dashboard(server_id=server_id)
```

Example 44

Project: *Nurevam* Author: *Maverun* File: *memes.py* MIT License 5 vo

```
def add_memes(cog, server_id):
    name = request.form.get("meme_name")
    link = request.form.get("meme_link")
    status = utils.check_link(link)
    if status == 0:  # if is true
        if name in db.smembers("{}:Memes:name".format(server_id)):
            flash("This name already exists!", "warning")
        else:
            db.hset("{}:Memes:link".format(server_id), name, link)
            db.sadd("{}:Memes:name".format(server_id), name)
            flash("You have add a new memes!", "success")
    return redirect(url_for("memes.memes", server_id=server_id, cog="memes"))
```

Example 45

Project: *Nurevam* Author: *Maverun* File: *memes.py* MIT License 5 vo

```
def edit_memes(server_id, name):
    new_name = request.form.get("meme_name")
    link = request.form.get("meme_link")
    status = utils.check_link(link)
    if status == 0:
        # delete old database
        db.hdel("{}:Memes:link".format(server_id), name)
        db.srem("{}:Memes:name".format(server_id), name)
        # adding, if there is a way to rename them in hash, that would be great..
        db.hset("{}:Memes:link".format(server_id), new_name, link)
        db.sadd("{}:Memes:name".format(server_id), new_name)
        flash("Update data!", "success")
    return redirect(url_for("memes.memes", server_id=server_id, cog="memes"))
```

Example 46

Project: *Nurevam* Author: *Maverun* File: *memes.py* MIT License 5 vo

```
def delete_memes(server_id, name):
    # Deleting data
    db.hdel("{}:Memes:link".format(server_id), name)
    db.srem("{}:Memes:name".format(server_id), name)
    return redirect(url_for("memes.memes", server_id=server_id, cog="Memes"))
```

Example 47

Project: *Nurevam* Author: *Maverun* File: *discourse.py* MIT License 5 vo

```
def update_category(server_id):
    try:
        data = dict(request.form)
        data.pop("_csrf_token")
        data = dict([[key,values[0]] for key,values in data.items()])
        db.delete("{}:Discourse:Category".format(server_id))
        db.hmset("{}:Discourse:Category".format(server_id),data)
        flash("Update!","success")
    except Exception as e:
        log.info("There is error\n{}".format(e))
    return redirect(url_for("discourse.category",server_id = server_id))
```

Example 48

Project: *Nurevam* Author: *Maverun* File: *discourse.py* MIT License 5 vo

```
def update_user_account_link(server_id):
    name = request.form.get("discourse_user")
    config = db.hgetall("{}:Discourse:Config".format(server_id))
    #we are checkig if it confirm.
    data = discourse(config["domain"],config["api_key"],config["username"],"users/
    is_pass = False
    discord_email = utils.session["user"]["email"]
    if data is None:
        flash("There is problem, I cannot check, there is likely a issue, please r
        return redirect(url_for("discourse.discourse_link", server_id=server_id,c
    elif discord_email == data["email"] or discord_email in data["secondary_emails
        is_pass = True
    elif discord_email in [x["description"] for x in data["associated_accounts"]]:
        is_pass = True

    if is_pass is False:
        get_id = db.hget("{}:Discourse:Trust_User".format(server_id),utils.session
        if(get_id):
            db.hdel("{}:Discourse:Trust_User".format(server_id),utils.session["use
            db.hdel("{}:Discourse:Trust_User_ID".format(server_id),get_id)
        flash("Your email doesn't match up with discord email!","warning")
        return redirect(url_for("discourse.discourse_link", server_id=server_id,c

    #since we got email, we should also get their ID
    discourse_id = discourse(config["domain"], config["api_key"], config["username
    discourse_id = discourse_id['user']['id']
    db.hset("{}:Discourse:Trust_User".format(server_id),utils.session["user"]["id"
    db.hset("{}:Discourse:Trust_User_ID".format(server_id),discourse_id,utils.sess
    flash("Update done!","success")
    return redirect(url_for("discourse.discourse_link",server_id=server_id))
```

```
#/groups/trust_level_0/members.json
```

Example 49

Project: *Nurevam* Author: *Maverun* File: *level.py* MIT License 5 vo

```
def reset_player(server_id, player_id):
    log.info("Reset that player's data")
    db.delete('{}:Level:Player:{}'.format(server_id, player_id))
    db.delete("{}:Level:{}:xp:check".format(server_id, player_id))  # just in case
    db.srem('{}:Level:Player'.format(server_id), player_id)
    return redirect(url_for('level.levels', server_id=server_id))
```

Example 50

Project: *Nurevam* Author: *Maverun* File: *level.py* MIT License 5 vo

```
def reset_all_players(server_id):
    log.info("Someone must be insane reset everything?")
    for player_id in db.smembers('{}:Level:Player'.format(server_id)):
        db.delete('{}:Level:Player:{}'.format(server_id, player_id))
        db.delete("{}:Level:{}:xp:check".format(server_id,player_id)) #just in cas
        db.srem('{}:Level:Player'.format(server_id), player_id)

    return redirect(url_for('level.levels', server_id=server_id))
```