Python flask.request.user() Examples

The following are code examples for showing how to use *flask.request.user()*. 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: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def post(self):
        '''
        新建相册
        '''
        post_data = request.data
        user = request.user
        name = post_data.pop('name', None)
        description = post_data.pop('description', None)
        if name is None:
            return HTTP.BAD_REQUEST(message='相册名称不能为空')
        album = Album(name=name, user=user)
        if description is not None:
            album.description = description
        album.save()
        serializer = AlbumSerializer(album)
        return HTTP.OK(data=serializer.data)
```

Example 2

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def get(self):
        '''
        获取图片列表
        '''
        query_dict = request.data
        user = request.user
        page, number = self.page_info
        keys = ['name', 'description']
        order_by = gen_order_by(query_dict, keys)
        filter_dict = gen_filter_dict(query_dict, keys, user=user)
        album = query_dict.pop('album', None)
        if album is not None:
            filter_dict.update(album__id=album)
        images = Image.query.filter_by(
            **filter_dict).order_by(*order_by).paginate(page, number)
        serializer = ImageSerializer(images.items, True)
        pageinfo = PageInfo(images)
        return HTTP.OK(data=serializer.data, pageinfo=pageinfo)
```

Example 3

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def put(self, pk):
        '''
        修改图片信息
        '''
        post_data = request.data
        user = request.user
        name = post_data.pop('name', None)
        description = post_data.pop('description', None)
        image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在')
        if name is not None:
            image.name = name
            image.url = os.path.join(image.path, name)
        if description is not None:
            image.description = description
        image.save()
        serializer = ImageSerializer(image)
        return HTTP.OK(data=serializer.data)
```

Example 4

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def delete(self, pk):
        '''
        删除图片
        '''
        user = request.user
        image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在')
        serializer = ImageSerializer(image)
        img_path = os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],
                                image.url)
        # 删除原图
        if os.path.exists(img_path):
            os.remove(img_path)
        # 删除缩略图
        thumb_path = os.path.join(current_app.config['UPLOAD_FOLDER_ROOT'],
                                  image.url.replace('photo', 'thumb'))
        if os.path.exists(thumb_path):
            os.remove(thumb_path)
        image.delete()
        return HTTP.OK(data=serializer.data)
```

Example 5

Project: *dark-chess* Author: *AHAPX* File: *decorators.py* GNU General Public License v3.0 6 vo

```
def authenticated(f):
    @wraps(f)
    def decorator(*args, **kwargs):
        token = (request.json or {}).get('auth') or \
            request.values.get('auth') or \
            request.cookies.get('auth')
        request.user = None
        request.auth = None
        if token is not None:
            user_id = get_cache(token)
            if user_id:
                try:
                    user = User.get(pk=user_id)
                except User.DoesNotExist:
                    pass
                else:
                    request.user = user
                    request.auth = token
        return f(*args, **kwargs)
    return decorator
```

Example 6

Project: *dark-chess* Author: *AHAPX* File: *decorators.py* GNU General Public License v3.0 6 vo

```
def with_game(f):
    @wraps(f)
    def decorator(token, *args, **kwargs):
        from game import Game

        try:
            game = Game.load_game(token)
        except errors.GameNotStartedError as exc:
            data = {
                'type': consts.TYPES[exc.type]['name'],
                'limit': exc.limit,
            }
            if (exc.token):
                data['invite'] = exc.token
            return send_data(data)
        except errors.GameNotFoundError as exc:
            return send_error(exc.message)
        if game._loaded_by == consts.WHITE:
            if game.model.player_white is not None and game.model.player_white !=
                return send_error('wrong user')
        else:
            if game.model.player_black is not None and game.model.player_black !=
                return send_error('wrong user')
        return f(game, *args, **kwargs)
    return decorator
```

Example 7

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def accept(game_id):
    try:
        pool = GamePool.get(GamePool.pk == game_id)
    except GamePool.DoesNotExist:
        return send_error('Game not found')
    except Exception as e:
        return send_error('Wrong format')
    if pool.user1 and pool.user1 == request.user:
        return send_error('You cannot start game with yourself')
    with config.DB.atomic():
        pool.player2 = generate_token(True)
        pool.user2 = request.user
        pool.is_started = True
        pool.save()
        game = Game.new_game(
            pool.player1, pool.player2, pool.type_game, pool.time_limit,
            white_user=pool.user1, black_user=pool.user2
        )
        delete_cache('wait_{}'.format(pool.player1))
        result = {'game': pool.player2}
        result.update(game.get_info(consts.BLACK))
    return send_data(result)
```

Example 8

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def invited(token):
    try:
        enemy_token, game_type, game_limit = get_cache('invite_{}'.format(token))
    except:
        return send_error('game not found')
    enemy_user = None
    user_id = get_cache('user_{}'.format(enemy_token))
    if user_id:
        try:
            enemy_user = User.get(pk=user_id)
        except User.DoesNotExist:
# TODO: if user not found game will be created with None as white player
            pass
    user_token = generate_token(True)
    game = Game.new_game(
        enemy_token, user_token, game_type, game_limit,
        white_user=enemy_user, black_user=request.user
    )
    delete_cache('wait_{}'.format(enemy_token))
    result = {'game': user_token}
    result.update(game.get_info(consts.BLACK))
    return send_data(result)
```

Example 9

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 6 vo

```
def recover(token):

    @validated(RecoverValidator)
    def _post(user, data):
        user.set_password(data['password'])
        user.save()
        delete_cache(token)
        return send_message('password changed')

    user = User.get_by_token(token)
    if user:
        if request.method == 'GET':
            return send_success()
        elif request.method == 'POST':
            return _post(user)
    return send_error('token not found')
```

Example 10

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def load_game(self, token):
        try:
            game = Game.load_game(token)
        except errors.GameNotStartedError as e:
            data = {
                'type': consts.TYPES[e.type]['name'],
                'limit': e.limit,
            }
            if (e.token):
                data['invite'] = e.token
            return data
        except errors.GameNotFoundError as e:
            raise errors.APIException(e.message)
        if game._loaded_by == consts.WHITE:
            if game.model.player_white is not None and game.model.player_white !=
                raise errors.APIException('wrong user')
        else:
            if game.model.player_black is not None and game.model.player_black !=
                raise errors.APIException('wrong user')
        self.game = game
```

Example 11

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def get(self):
        result = []
        count = 0
        for pool in GamePool.select().where(
            GamePool.is_started == False,
            GamePool.is_lost == False,
            GamePool.player1 is not None,
        ).order_by(GamePool.date_created.desc()):
            if pool.user1 and pool.user1 == request.user:
                continue
            result.append({
                'id': pool.pk,
                'date_created': pool.date_created.isoformat(),
                'user': pool.user1.username if pool.user1 else None,
                'type': consts.TYPES[pool.type_game]['name'],
                'limit': pool.time_limit,
            })
            count += 1
            if count > 9:
                break
        return {'games': result}
```

Example 12

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def post(self, game_id):
        try:
            pool = GamePool.get(GamePool.pk == game_id)
        except GamePool.DoesNotExist:
            raise errors.APINotFound('game')
        except Exception as e:
            raise errors.APIException('wrong format')
        if pool.user1 and pool.user1 == request.user:
            raise errors.APIException('you cannot start game with yourself')
        pool.player2 = generate_token(True)
        pool.user2 = request.user
        pool.is_started = True
        pool.save()
        game = Game.new_game(
            pool.player1, pool.player2, pool.type_game, pool.time_limit,
            white_user=pool.user1, black_user=pool.user2
        )
        delete_cache('wait_{}'.format(pool.player1))
        result = {'game': pool.player2}
        result.update(game.get_info(consts.BLACK))
        return result
```

Example 13

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 6 vo

```
def get(self, token):
        try:
            enemy_token, game_type, game_limit = get_cache('invite_{}'.format(toke
        except:
            raise errors.APINotFound('game')
        enemy_user = None
        user_id = get_cache('user_{}'.format(enemy_token))
        if user_id:
            try:
                enemy_user = User.get(pk=user_id)
            except User.DoesNotExist:
# TODO: if user not found game will be created with None as white player
                pass
        user_token = generate_token(True)
        game = Game.new_game(
            enemy_token, user_token, game_type, game_limit,
            white_user=enemy_user, black_user=request.user
        )
        delete_cache('wait_{}'.format(enemy_token))
        result = {'game': user_token}
        result.update(game.get_info(consts.BLACK))
        return result
```

Example 14

Project: *flask-io* Author: *viniciuschiele* File: *actions.py* MIT License 6 vo

```
def perform_authentication(self):
        """
        Perform authentication on the incoming request.
        """

        if not self.authenticators:
            return

        request.user = None
        request.auth = None

        for authenticator in self.authenticators:
            auth_tuple = authenticator.authenticate()

            if auth_tuple:
                request.user = auth_tuple[0]
                request.auth = auth_tuple[1]
                break
```

Example 15

Project: *maple-blog* Author: *honmaple* File: *filepath.py* GNU General Public License v3.0 6 vo

```
def get(self, bucket):
        data = request.data
        user = request.user
        page, number = self.pageinfo

        bucket = user.buckets.filter_by(
            name=bucket).get_or_404("bucket not found")
        path = request.data.get("path", "/")

        params = filter_maybe(data, {
            "name": "name__contains",
        })
        rootpath = bucket.get_root_path(path)
        paths = rootpath.child_paths.filter_by(**params).paginate(page, number)
        serializer = FilePathSerializer(paths)
        return HTTP.OK(data=serializer.data)
```

Example 16

Project: *flask-stupe* Author: *numberly* File: *auth.py* MIT License 6 vo

```
def auth_required(function):
    """Decorator checking that the request is made by an authenticated user.

    If you want to use that function, you should set a before_request handler
    that authenticate requests when possible. It must then expose a `user`
    attribute on the :obj:`flask.request` object.

    .. code-block:: python

        @app.before_request
        def get_user():
            token = request.args.get("token")
            if verify_token(token):
                request.user = {"username": "toto"}

    A view decorated with :func:`auth_required` will be aborted with a status
    code 401 if the user making the request is not authenticated.
    """
    @functools.wraps(function)
    def __inner(*args, **kwargs):
        if not request.user:
            abort(401)
        return function(*args, **kwargs)
    return __inner
```

Example 17

Project: *flask-stupe* Author: *numberly* File: *auth.py* MIT License 6 vo

```
def test_permission_required_with_user_object(app, client):
    class User(object):
        pass

    @app.before_request
    def set_user():
        request.user = User()
        request.user.permissions = get_permissions()

    @app.route("/foo")
    @permission_required("vip", "secret_stuff")
    def foo():
        return "bar"
    assert client.get("/foo").status_code == 403
    assert client.get("/foo?permissions=vip").status_code == 200
```

Example 18

Project: *grouporder* Author: *ErnstHaagsman* File: *users.py* MIT License 6 vo

```
def post(self):
        parser = reqparse.RequestParser()
        parser.add_argument('username', type=str, required=True,
                            help='The desired username. Should be unique '
                                 'within the system')
        parser.add_argument('password', type=str, required=True,
                            help='Password, please pick something secure')
        parser.add_argument('fullname', type=str, required=True,
                            help='Your full name')
        parser.add_argument('email', type=str, required=True,
                            help='Your email address')
        args = parser.parse_args()

        try:
            new_user = User.create(args['username'],
                                   args['fullname'],
                                   args['email'],
                                   args['password'])
            return {
                        'username': new_user.username,
                        'fullname': new_user.fullname,
                        'email': new_user.email
                   }, 201
        except DuplicateUserError:
            abort(409, message='A user with this username already exists')
```

Example 19

Project: *grouporder* Author: *ErnstHaagsman* File: *users.py* MIT License 6 vo

```
def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        auth_string = request.headers['Authorization']

        if not auth_string.lower().startswith('bearer '):
            abort(403)

        token = auth_string[7:]  # the bit after 'bearer '

        user = User.from_token(token)

        if user is None:
            abort(403)

        request.user = user

        return f(*args, **kwargs)
    return decorated_function
```

Example 20

Project: *BhagavadGita* Author: *gita* File: *oauth1.py* GNU General Public License v3.0 6 vo

```
def tokengetter(self, f):
        """Register a function as the access token getter.

        The function accepts `client_key` and `token` parameters, and it
        returns an access token object contains:

            - client: Client associated with this token
            - user: User associated with this token
            - token: Access token
            - secret: Access token secret
            - realms: Realms with this access token

        Implement the token getter::

            @oauth.tokengetter
            def get_access_token(client_key, token):
                return AccessToken.get(client_key=client_key, token=token)
        """
        self._tokengetter = f
        return f
```

Example 21

Project: *BhagavadGita* Author: *gita* File: *oauth1.py* GNU General Public License v3.0 6 vo

```
def verifiergetter(self, f):
        """Register a function as the verifier getter.

        The return verifier object should at least contain a user object
        which is the current user.

        The implemented code looks like::

            @oauth.verifiergetter
            def load_verifier(verifier, token):
                data = Verifier.get(verifier)
                if data.request_token == token:
                    # check verifier for safety
                    return data
                return data
        """
        self._verifiergetter = f
        return f
```

Example 22

Project: *BhagavadGita* Author: *gita* File: *oauth1.py* GNU General Public License v3.0 6 vo

```
def verifiersetter(self, f):
        """Register a function as the verifier setter.

        A verifier is better together with request token, but it is not
        required. A verifier is used together with request token for
        exchanging access token, it has an expire time, in this case, it
        would be a better design if you put them in a cache.

        The implemented code looks like::

            @oauth.verifiersetter
            def save_verifier(verifier, token, *args, **kwargs):
                data = Verifier(
                    verifier=verifier['oauth_verifier'],
                    request_token=token,
                    user=get_current_user()
                )
                return data.save()
        """
        self._verifiersetter = f
        return f
```

Example 23

Project: *BhagavadGita* Author: *gita* File: *oauth1.py* GNU General Public License v3.0 6 vo

```
def save_access_token(self, token, request):
        """Save access token to database.

        A tokensetter is required, which accepts a token and request
        parameters::

            def tokensetter(token, request):
                access_token = Token(
                    client=request.client,
                    user=request.user,
                    token=token['oauth_token'],
                    secret=token['oauth_token_secret'],
                    realms=token['oauth_authorized_realms'],
                )
                return access_token.save()
        """
        log.debug('Save access token %r', token)
        self._tokensetter(token, request)
```

Example 24

Project: *BhagavadGita* Author: *gita* File: *oauth1.py* GNU General Public License v3.0 6 vo

```
def save_verifier(self, token, verifier, request):
        """Save verifier to database.

        A verifiersetter is required. It would be better to combine request
        token and verifier together::

            def verifiersetter(token, verifier, request):
                tok = Grant.query.filter_by(token=token).first()
                tok.verifier = verifier['oauth_verifier']
                tok.user = get_current_user()
                return tok.save()

        .. admonition:: Note:

            A user is required on verifier, remember to attach current
            user to verifier.
        """
        log.debug('Save verifier %r for %r', verifier, token)
        self._verifiersetter(token=token, verifier=verifier, request=request)
```

Example 25

Project: *BhagavadGita* Author: *gita* File: *oauth2.py* GNU General Public License v3.0 6 vo

```
def tokensetter(self, f):
        """Register a function to save the bearer token.

        The setter accepts two parameters at least, one is token,
        the other is request::

            @oauth.tokensetter
            def set_token(token, request, *args, **kwargs):
                save_token(token, request.client, request.user)

        The parameter token is a dict, that looks like::

            {
                u'access_token': u'6JwgO77PApxsFCU8Quz0pnL9s23016',
                u'token_type': u'Bearer',
                u'expires_in': 3600,
                u'scope': u'email address'
            }

        The request is an object, that contains an user object and a
        client object.
        """
        self._tokensetter = f
        return f
```

Example 26

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def post(self):
        '''
        登陆
        '''
        post_data = request.data
        username = post_data.pop('username', None)
        password = post_data.pop('password', None)
        remember = post_data.pop('remember', True)
        if username and password:
            user = User.query.filter_by(username=username).first()
            if user and user.check_password(password):
                user.login(remember)
                serializer = UserSerializer(user)
                return HTTP.OK(data=serializer.data)
        return HTTP.UNAUTHORIZED(message='用户名或密码错误')
```

Example 27

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def get(self):
        user = request.user
        user.logout()
        return HTTP.OK(message='登出成功')
```

Example 28

Project: *maple-file* Author: *honmaple* File: *router.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def get(self):
        '''
        获取相册列表
        '''
        query_dict = request.data
        user = request.user
        page, number = self.page_info
        keys = ['name', 'description']
        order_by = gen_order_by(query_dict, keys)
        filter_dict = gen_filter_dict(query_dict, keys, user=user)
        albums = Album.query.filter_by(
            **filter_dict).order_by(*order_by).paginate(page, number)
        serializer = AlbumSerializer(albums.items, True)
        pageinfo = PageInfo(albums)
        return HTTP.OK(data=serializer.data, pageinfo=pageinfo)
```

Example 29

Project: *maple-file*   Author: *honmaple*   File: *router.py*   BSD 3-Clause "New" or "Revised" License   5 vo

```
def get(self, pk):
        '''
        获取具体相册
        '''
        user = request.user
        album = Album.query.filter_by(id=pk, user=user).get_or_404('相册不存在')
        serializer = AlbumSerializer(album)
        return HTTP.OK(data=serializer.data)
```

Example 30

Project: *maple-file*   Author: *honmaple*   File: *router.py*   BSD 3-Clause "New" or "Revised" License   5 vo

```
def delete(self, pk):
        '''
        删除具体相册
        '''
        user = request.user
        album = Album.query.filter_by(id=pk, user=user).get_or_404('相册不存在')
        serializer = AlbumSerializer(album)
        album.delete()
        return HTTP.OK(data=serializer.data)
```

Example 31

Project: *maple-file*   Author: *honmaple*   File: *router.py*   BSD 3-Clause "New" or "Revised" License   5 vo

```
def get(self, pk):
        '''
        显示图片
        '''
        user = request.user
        image = Image.query.filter_by(id=pk, user=user).get_or_404('图片不存在')
        serializer = ImageSerializer(image)
        return HTTP.OK(data=serializer.data)
```

Example 32

Project: *dark-chess*   Author: *AHAPX*   File: *decorators.py*   GNU General Public License v3.0   5 vo

```
def login_required(f):
    @wraps(f)
    def decorator(*args, **kwargs):
        if getattr(request, 'user', None):
            return f(*args, **kwargs)
        return send_error('not authorized')
    return decorator
```

Example 33

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 5 vo

```
def new():

    @validated(GameNewValidator)
    def _post(data):
        game_type = data['type']
        game_limit = data['limit']
        token = generate_token(True)
        pool = GamePool.create(
            player1 = token,
            user1 = request.user,
            type_game = game_type,
            time_limit = game_limit,
        )
        set_cache('wait_{}'.format(token), (game_type, game_limit))
        return send_data({'game': token})

    if request.method == 'GET':
        result = []
        count = 0
        for pool in GamePool.select().where(
            GamePool.is_started == False,
            GamePool.is_lost == False,
            GamePool.player1 is not None,
        ).order_by(GamePool.date_created.desc()):
            if pool.user1 and pool.user1 == request.user:
                continue
            result.append({
                'id': pool.pk,
                'date_created': pool.date_created.isoformat(),
                'user': pool.user1.username if pool.user1 else None,
                'type': consts.TYPES[pool.type_game]['name'],
                'limit': pool.time_limit,
            })
            count += 1
            if count > 9:
                break
        return send_data({'games': result})
    elif request.method == 'POST':
        return _post()
```

Example 34

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 5 vo

```
def games():
    from models import Game

    result = {
        'games': {
            'actives': [],
            'ended': [],
        }
    }
    if request.user:
        games = Game.select().where(
            Game.date_end == None,
            (Game.player_white == request.user) | (Game.player_black == request.u
        )
        for game in games:
            if game.player_white == request.user:
                result['games']['actives'].append(game.white)
            else:
                result['games']['actives'].append(game.black)
        games = Game.select().where(
            Game.date_end != None,
            (Game.player_white == request.user) | (Game.player_black == request.u
        ).limit(10)
        for game in games:
            if game.player_white == request.user:
                result['games']['ended'].append(game.white)
            else:
                result['games']['ended'].append(game.black)
    return send_data(result)
```

Example 35

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def register(data):
    username = data['username']
    password = data['password']
    email = data['email']
    user = User.add(username, password, email)
    if email:
        token = user.get_verification()
        data = {
            'username': username,
            'url': urljoin(config.SITE_URL, config.VERIFY_URL),
            'token': token,
        }
        send_mail_template('registration', [email], data=data)
    return send_message('registration successful')
```

Example 36

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def get_verification():
    try:
        token = request.user.get_verification()
    except Exception as exc:
        return send_error(exc.message)
    data = {
        'username': request.user.username,
        'url': urljoin(config.SITE_URL, config.VERIFY_URL),
        'token': token,
    }
    send_mail_template('verification', [request.user.email], data=data)
    return send_success()
```

Example 37

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def verify(token):
    user = User.get_by_token(token)
    if user:
        user.verify()
        delete_cache(token)
        return send_success()
    return send_error('token not found')
```

Example 38

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def authorized():
    if request.user:
        return send_data({'username': request.user.username})
    return send_error('not authorized')
```

Example 39

Project: *dark-chess* Author: *AHAPX* File: *chat.py* GNU General Public License v3.0 5 vo

```
def messages():

    @validated(MessageValidator)
    def _post(data):
        message = ChatMessage.create(user=request.user, text=data['text'])
        result = {'message': MessageSerializer(message).calc()}
        send_ws(result, consts.WS_CHAT_MESSAGE)
        return send_data(result)

    if request.method == 'GET':
        try:
            limit = int(request.args.get('limit', -1))
            offset = int(request.args.get('offset', -1))
            if limit < 0:
                limit = config.DEFAULT_COUNT_MESSAGES
            if offset < 0:
                offset = 0
        except Exception as e:
            log.error(e)
            return send_error('wrong arguments')
        messages = ChatMessage.select()\
                        .where(ChatMessage.chat == None)\
                        .order_by(-ChatMessage.date_created)\
                        .offset(offset)\
                        .limit(limit)
        return send_data({
            'messages': [MessageSerializer(m).calc() for m in messages],
        })
    elif request.method == 'POST':
        return _post()
```

Example 40

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 5 vo

```
def post(self):
        game_type = self.data['type']
        game_limit = self.data['limit']
        if game_type != consts.TYPE_NOLIMIT and not game_limit:
            raise errors.APIException('game limit must be set for no limit game')
        token_game = generate_token(True)
        token_invite = generate_token(True)
        set_cache('invite_{}'.format(token_invite), (token_game, game_type, game_l
        if request.user:
            set_cache('user_{}'.format(token_game), request.user.pk, 3600)
        set_cache('wait_{}'.format(token_game), (game_type, game_limit, token_invi
        return {
            'game': token_game,
            'invite': token_invite,
        }
```

Example 41

Project: *dark-chess* Author: *AHAPX* File: *game.py* GNU General Public License v3.0 5 vo

```
def get(self):
        from models import Game

        result = {
            'games': {
                'actives': [],
                'ended': [],
            }
        }
        if request.user:
            games = Game.select().where(
                Game.date_end == None,
                (Game.player_white == request.user) | (Game.player_black == reque
            )
            for game in games:
                if game.player_white == request.user:
                    result['games']['actives'].append(game.white)
                else:
                    result['games']['actives'].append(game.black)
            games = Game.select().where(
                Game.date_end != None,
                (Game.player_white == request.user) | (Game.player_black == reque
            ).limit(10)
            for game in games:
                if game.player_white == request.user:
                    result['games']['ended'].append(game.white)
                else:
                    result['games']['ended'].append(game.black)
        return result
```

Example 42

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def post(self):
        username = self.data['username']
        password = self.data['password']
        email = self.data['email']
        user = User.add(username, password, email)
        if email:
            token = user.get_verification()
            data = {
                'username': username,
                'url': urljoin(config.SITE_URL, config.VERIFY_URL),
                'token': token,
            }
            send_mail_template('registration', [email], data=data)
        return 'registration successful'
```

Example 43

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def get(self, token):
        user = User.get_by_token(token)
        if user:
            user.verify()
            delete_cache(token)
            return 'verification completed'
        raise APINotFound('token')
```

Example 44

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def get(self):
        if request.user:
            return {'username': request.user.username}
        raise APIUnauthorized
```

Example 45

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def post(self):
        try:
            user = User.get(User.email == self.data['email'])
        except User.DoesNotExist:
            raise APINotFound('hey email')
        token = user.get_reset()
        data = {
            'username': user.username,
            'url': urljoin(config.SITE_URL, config.RECOVER_URL),
            'token': token,
        }
        send_mail_template('reset', [user.email], data=data)
        return 'send recover email'
```

Example 46

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def get(self, token):
        user = User.get_by_token(token)
        if not user:
            raise APINotFound('token')
        return 'token is found'
```

Example 47

Project: *dark-chess* Author: *AHAPX* File: *auth.py* GNU General Public License v3.0 5 vo

```
def post(self, token):
        user = User.get_by_token(token)
        if not user:
            raise APINotFound('token')
        user.set_password(self.data['password'])
        user.save()
        delete_cache(token)
        return 'password changed'
```

Example 48

Project: *dark-chess* Author: *AHAPX* File: *chat.py* GNU General Public License v3.0 5 vo

```
def post(self):
        message = ChatMessage.create(user=request.user, text=self.data['text'])
        result = {'message': MessageSerializer(message).calc()}
        send_ws(result, WS_CHAT_MESSAGE)
        return result
```

Example 49

Project: *flask-maple* Author: *honmaple* File: *middleware.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def preprocess_request(self):
        if current_user is not None:
            request.user = current_user._get_current_object()
        if request.method in ["GET", "DELETE"]:
            request.data = request.args.to_dict()
        else:
            request.data = request.json
            if request.data is None:
                request.data = request.form.to_dict()
```

Example 50

Project: *flask-io* Author: *viniciuschiele* File: *actions.py* MIT License 5 vo

```
def perform_authorization(self):
        """
        Check if the request should be permitted.
        Raises an appropriate exception if the request is not permitted.
        """

        for permission in self.permissions:
            if not permission.has_permission():
                if request.user:
                    raise errors.PermissionDenied()
                else:
                    raise errors.NotAuthenticated()
```