Python flask.request.is_json() Examples

The following are code examples for showing how to use *flask.request.is_json()*. 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: *DancesafeResults* Author: *siorai* File: *DancesafeResults.py* GNU Affero General Public License v3.0 6 vo

```
def api_new_event():
    if request.method == "POST":
        newConnection = DBConn()
        currentSession = newConnection.cursor()
        print(request.is_json)
        content = request.get_json()
        print(content)
        eventName = content["name"]
        eventYear = content["year"]
        eventCity = content["city"]
        eventState = content["state"]
        eventRegion = content["region"]
        eventAuthor = content["author"]
        currentSession.execute(
            "INSERT INTO EVENT (name, year, city, state, region, author) VALUES ('
                eventName, eventYear, eventCity, eventState, eventRegion, eventAut
            )
        )
        newConnection.commit()
        currentSession.close()
        newConnection.close()
        return "JSON posted"
```

Example 2

Project: *byceps* Author: *byceps* File: *views.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def create_archived_attendance():
    """Create an archived attendance of the user at the party."""
    if not request.is_json:
        abort(415)

    schema = CreateArchivedAttendanceRequest()
    try:
        req = schema.load(request.get_json())
    except ValidationError as e:
        abort(400, str(e.normalized_messages()))

    user = user_service.find_user(req['user_id'])
    if not user:
        abort(400, 'User ID unknown')

    party = party_service.find_party(req['party_id'])
    if not party:
        abort(400, 'Party ID unknown')

    attendance_service.create_archived_attendance(user.id, party.id)
```

Example 3

Project: *flask-jwt-extended* Author: *vimalloc* File: *simple.py* MIT License 6 vo

```
def login():
    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    username = request.json.get('username', None)
    password = request.json.get('password', None)
    if not username:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400

    if username != 'test' or password != 'test':
        return jsonify({"msg": "Bad username or password"}), 401

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username)
    return jsonify(access_token=access_token), 200


# Protect a view with jwt_required, which requires a valid access token
# in the request to access.
```

Example 4

Project: *legacy* Author: *kytos* File: *main.py* MIT License 6 vo

```
def save_topology(name):
        """Save a topology layout in a file.

        This method get a json topology from request and puts this in a file.

        Parameters:
            name (string): name of the topology to  be saved or loaded.

        Returns:
            topology (string): topology using json format.
        """
        if not request.is_json:
            return json.dumps('{"error": "gt was not a JSON request"}'), 400
        topology = request.get_json()
        with open(join(settings.TOPOLOGY_DIR, name + '.json'), 'w') as outfile:
            json.dump(topology, outfile)
        return json.dumps({'response': 'Saved'}), 201
```

Example 5

Project: *JustHear.me* Author: *Team-IF* File: *auth.py* GNU Affero General Public License v3.0 6 vo

```
def login() -> JsonResponse:
    try:
        if not request.is_json:
            return common.rerror("invalid json", 400)

        req = request.json
        if not req.get('email') or not req.get("pass"):
            return common.rerror("이메일과 비밀번호를 입력해 주세요.", 400)
```

```
        user = common.User.fromEmail(req.get('email'))
        if not user or user.matchpw(req.get("pass")):
            return common.rerror("잘못된 이메일/비밀번호", 403)

        newtoken = str(uuid4())
        expiredate = datetime.datetime.utcnow()
        expiredate = expiredate + datetime.timedelta(days=14)
        return JsonResponse({
            'token': newtoken,
            'uuid': user.uuid
        })

    except Exception as e:
        return common.rerror(e, 500)
```

Example 6

Project: *envoxy* Author: *habitio* File: *views.py* MIT License 6 vo

```
def dispatch(self, request, _method, _endpoint, *args, **kwargs):
        kwargs.update({ 'endpoint': _endpoint })
        try:
            return getattr(self, _method)(request, *args, **kwargs)

        except Exception as e:

            _error_log_ref = str(uuid.uuid4())
            _code = 0
            _status = 500

            if isinstance(e, ValidationException):
                if 'code' in e.kwargs : _code = e.kwargs['code']
                if 'status' in e.kwargs: _status = e.kwargs['status']

            if request.is_json:
                Log.error(f"ELRC({_error_log_ref}) - Traceback: {traceback.format_
                return make_response(jsonify({"error": f"{e} :: ELRC({_error_log_r

            Log.error(f"ELRC({_error_log_ref}) - Traceback: {traceback.format_exc(
            return FlaskResponse(str(f"error: {e} :: ELRC({_error_log_ref})"), _st
```

Example 7

Project: *safrs* Author: *thomaxxl* File: *demo_jwt.py* MIT License 6 vo

```
def login():
    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username)

    username = request.json.get("username", None)
    password = request.json.get("password", None)
    if not username:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400

    if username != "test" or password != "test":
        return jsonify({"msg": "Bad username or password"}), 401
```

```
    return jsonify(access_token=access_token), 200
```

Example 8

Project: *Flask-Large-Application-Example-Simplified* Author: *JoMingyu* File: *__init__.py* MIT License 6 vo

```
def json_required(*required_keys):
    def decorator(fn):
        if fn.__name__ == 'get':
            print('[WARN] JSON with GET method? on "{}()"'.format(fn.__qualname__)

        @wraps(fn)
        def wrapper(*args, **kwargs):
            if not request.is_json:
                abort(406)

            for required_key in required_keys:
                if required_key not in request.json:
                    abort(400)

            return fn(*args, **kwargs)
        return wrapper
    return decorator
```

Example 9

Project: *iot_fog* Author: *eduarias* File: *runner.py* GNU General Public License v3.0 6 vo

```
def insert_data():
    """
    Get the data to the sensor and save it
    :return: HTTP response
    """
    data_sender = DataSender(config)
    if not request.is_json:
        logging.debug('Input data is not a json')
        return 'Input data must be a json', HTTPStatus.BAD_REQUEST
    else:
        try:
            request_data = request.get_json()
            device_name = request_data['device_name']
            data = request_data['data']
            logging.debug('Received data: \{}'.format(request_data))
        except KeyError:
            return 'Wrong input data', HTTPStatus.BAD_REQUEST
        data_sender.send_data(data, device_name)
        return '', HTTPStatus.NO_CONTENT
```

Example 10

Project: *push-deploy* Author: *mdgreenwald* File: *apiv1.py* Apache License 2.0 6 vo

```
def login():
    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    username = request.json.get('username', None)
    password = request.json.get('password', None)
    if not username:
```

```
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400

    if username != current_app.config['PD_USER'] or password != current_app.config
        return jsonify({"msg": "Bad username or password"}), 401

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username, expires_delta=timedelta(
    return jsonify(access_token=access_token), 200
```

Example 11

Project: *fastapi* Author: *zhangnian* File: *http_util.py* MIT License 6 vo

```
def get_json_arg(name, parser=None, validator=None):
    jdata = request.get_json(force=True, silent=True)

    if not request.is_json:
        raise APIError(ret=1, msg='请求数据格式错误')

    if jdata is None:
        raise APIError(ret=1, msg='请求数据格式错误')

    val = jdata.get(name, None)
    if val is None:
        raise APIError(ret=1, msg='缺少参数:{}'.format(name))

    if parser and callable(parser):
        try:
            val = parser(val)
        except Exception as e:
            raise APIError(ret=1, msg='转换参数:{}失败'.format(name))

    if validator and callable(validator):
        if not validator(val):
            raise APIError(ret=1, msg='参数:{}不合法'.format(name))
    return val
```

Example 12

Project: *fastapi* Author: *zhangnian* File: *http_util.py* MIT License 6 vo

```
def get_json_arg_default(name, default=None, parser=None, validator=None):
    jdata = request.get_json(force=True, silent=True)

    if not request.is_json:
        raise APIError(ret=1, msg='请求数据格式错误')

    if jdata is None:
        raise APIError(ret=1, msg='请求数据格式错误')

    val = jdata.get(name, None)
    if val is None:
        if default is not None:
            return default
        raise APIError(ret=1, msg='缺少参数:{}'.format(name))

    if parser and callable(parser):
```

```
        try:
            val = parser(val)
        except Exception as e:
            raise APIError(ret=1, msg='转换参数:{}失败'.format(name))

    if validator and callable(validator):
        if not validator(val):
            raise APIError(ret=1, msg='参数:{}不合法'.format(name))
    return val
```

Example 13

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

```
def post_layout():
    if not g.current_permissions.layout_create:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    try:
        name = data.get("title")
        if not name:
            name = data.get("subtitle", "Unnamed")
        layout = Layout.from_json_data(name, data)
        db.session.add(layout)
        db.session.commit()
        return jsonify(layout.as_dict())
    except (IntegrityError, StatementError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 14

Project: *flask-unchained* Author: *briancappello* File: *anonymous_user_required.py* MIT License 6 vo

```
def anonymous_user_required(*decorator_args, msg=None, category=None, redirect_url
    """
    Decorator requiring that there is no user currently logged in.

    Aborts with ``HTTP 403: Forbidden`` if there is an authenticated user.
    """
    def wrapper(fn):
        @wraps(fn)
        def decorated(*args, **kwargs):
            if current_user.is_authenticated:
                if request.is_json:
                    abort(HTTPStatus.FORBIDDEN)
                else:
                    if msg:
                        flash(msg, category)
                    return redirect('SECURITY_POST_LOGIN_REDIRECT_ENDPOINT',
                                    override=redirect_url)
            return fn(*args, **kwargs)
        return decorated

    if decorator_args and callable(decorator_args[0]):
        return wrapper(decorator_args[0])
    return wrapper
```

Example 15

Project: *flask-unchained* Author: *briancappello* File: *security_controller.py* MIT License 6 vo

```
def register(self):
        """
        View function to register user. Supports html and json requests.
        """
        form = self._get_form('SECURITY_REGISTER_FORM')
        if form.validate_on_submit():
            user = self.security_service.user_manager.create(**form.to_dict())
            self.security_service.register_user(user)
            if request.is_json:
                return '', HTTPStatus.NO_CONTENT
            return self.redirect('SECURITY_POST_REGISTER_REDIRECT_ENDPOINT')

        elif form.errors and request.is_json:
            return self.errors(form.errors)

        return self.render('register',
                           register_user_form=form,
                           **self.security.run_ctx_processor('register'))
```

Example 16

Project: *flask-unchained* Author: *briancappello* File: *security_controller.py* MIT License 6 vo

```
def send_confirmation_email(self):
        """
        View function which sends confirmation token and instructions to a user.
        """
        form = self._get_form('SECURITY_SEND_CONFIRMATION_FORM')
        if form.validate_on_submit():
            self.security_service.send_email_confirmation_instructions(form.user)
            self.flash(_('flask_unchained.bundles.security:flash.confirmation_requ
                         email=form.user.email), category='info')
            if request.is_json:
                return '', HTTPStatus.NO_CONTENT
            return self.redirect('send_confirmation_email')

        elif form.errors and request.is_json:
            return self.errors(form.errors)

        return self.render('send_confirmation_email',
                           send_confirmation_form=form,
                           **self.security.run_ctx_processor('send_confirmation_em
```

Example 17

Project: *flask-unchained* Author: *briancappello* File: *security_controller.py* MIT License 6 vo

```
def change_password(self):
        """
        View function for a user to change their password.
        Supports html and json requests.
        """
        form = self._get_form('SECURITY_CHANGE_PASSWORD_FORM')
        if form.validate_on_submit():
            self.security_service.change_password(
                current_user._get_current_object(),
                form.new_password.data)
```

```
            self.after_this_request(self._commit)
            self.flash(_('flask_unchained.bundles.security:flash.password_change')
                       category='success')
            if request.is_json:
                return self.jsonify({'token': current_user.get_auth_token()})
            return self.redirect('SECURITY_POST_CHANGE_REDIRECT_ENDPOINT',
                                 'SECURITY_POST_LOGIN_REDIRECT_ENDPOINT')

        elif form.errors and request.is_json:
            return self.errors(form.errors)

        return self.render('change_password',
                           change_password_form=form,
                           **self.security.run_ctx_processor('change_password'))
```

Example 18

Project: *flask_middleware_jwt* Author: *keyloguer* File: *app.py* Apache License 2.0 6 vo

```
def login():
    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    username = request.json.get('username', None)
    password = request.json.get('password', None)
    if not username:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400

    if username != 'test' or password != 'test':
        return jsonify({"msg": "Bad username or password"}), 401

    # Identity can be any data that is json serializable
    access_token = create_access_token(identity=username)
    return jsonify(access_token=access_token), 200
```

Example 19

Project: *--Awesome-Python--* Author: *JoMingyu* File: *5. JSON Payload - request.json.py* GNU General Public License v3.0 6 vo

```
def json():
    # 요청의 Content-Type이 application/json이고, 직렬화된 JSON 문자열이 들어온다면(RFC 표준에
    # request의 json property나 get_json() 메소드를 이용하면 된다

    req = request.json
    # Flask 0.12.3까지 deprecated되어 있었으나, Flask 1.0에선 다시 deprecated가 제거됨

    req = request.get_json()
    # Flask 0.10에서 추가된 메소드

    # 요청의 Content-Type이 application/json이 아니라면 None이 반환되며
    # Content-Type은 application/json으로 설정되었으나 아무 데이터도 전달되지 않으면 status co

    # 요청의 타입이 json인지 확인해 주는 메소드도 있다
    if not request.is_json:
        return 'Please set your content type "application/json"!', 400

    print(type(req))
    # json 프로퍼티는 요청 데이터에 따라 파이썬 고유의 dict 또는 list 타입으로 처리된다
```

```
    return str(req['test_key'])
```

Example 20

Project: *DMS-Backend* Author: *DSM-DMS* File: *__init__.py* MIT License 6 vo

```
def json_required(*required_keys):
    """
    View decorator for JSON validation.

    - If content-type is not application/json : returns status code 406
    - If required_keys are not exist on request.json : returns status code 400

    :type required_keys: str
    """
    def decorator(fn):
        if fn.__name__ == 'get':
            print('[WARN] JSON with GET method? on "{}()"'.format(fn.__qualname__)

        @wraps(fn)
        def wrapper(*args, **kwargs):
            if not request.is_json:
                abort(406)

            for required_key in required_keys:
                if required_key not in request.json:
                    abort(400)

            return fn(*args, **kwargs)
        return wrapper
    return decorator
```

Example 21

Project: *DMS-Backend* Author: *DSM-DMS* File: *__init__.py* MIT License 6 vo

```
def json_required(required_keys):
    def decorator(fn):
        if fn.__name__ == 'get':
            print('[WARN] JSON with GET method? on "{}()"'.format(fn.__qualname__)

        @wraps(fn)
        def wrapper(*args, **kwargs):
            if not request.is_json:
                abort(406)

            for key, typ in required_keys.items():
                if key not in request.json or type(request.json[key]) is not typ:
                    abort(400)
                if typ is str and not request.json[key]:
                    abort(400)

            return fn(*args, **kwargs)
        return wrapper
    return decorator
```

Example 22

Project: *flask-restapi-example* Author: *mgreenw* File: *app.py* MIT License 6 vo

```
def create_review():
    request_json = request.get_json()
    if not request.is_json or 'doctor_id' not in request_json or 'description' no
        return bad_request('Missing required data.')
    doctor_id = request_json['doctor_id']

    # If the doctor_id is invalid, generate the appropriate 400 message
    try:
        review = Review(doctor_id=doctor_id, description=request_json['descriptior
        db.session.add(review)
        db.session.commit()
    except:
        return bad_request('Given doctor_id does not exist.')
    return jsonify({'review': review.serialize}), 201


# Custom Error Helper Functions
```

Example 23

Project: *speid* Author: *cuenca-mx* File: *views.py* MIT License 6 vo

```
def log_posts():
    if request.method != 'POST':
        return
    if request.is_json:
        body = json.dumps(request.json)
    else:
        body = request.data.decode('utf-8') or json.dumps(request.form)
    req = Request(
        method=HttpRequestMethod(request.method),
        path=request.path,
        query_string=request.query_string.decode(),
        ip_address=request.remote_addr,
        headers=dict(request.headers),
        body=body,
    )
    req.save()
```

Example 24

Project: *CS-Py* Author: *Parkkeo1* File: *flask_api_server.py* MIT License 6 vo

```
def post(self):
        if request.is_json:
            print(request.get_json())
            payload = UserDataPayload(request.get_json())

            if payload.is_valid:
                sql_db = sqlite3.connect(cs_py_server.config['DATABASE'])

                if does_user_exist(payload.steamid, sql_db):  # user already exist
                    if not is_duplicate_match(payload, sql_db):
                        insert_match_data(payload, sql_db)
                        update_existing_user(payload.steamid, sql_db)

                else:  # user does not exist
                    insert_match_data(payload, sql_db)
                    add_new_user(payload.steamid, sql_db)

                sql_db.close()
```

```
                return 'Data Accepted', 202

        return 'Invalid Data', 400
```

Example 25

Project: *s3label* Author: *stonethree* File: *endpoints.py* MIT License 6 vo

```
def login():
    engine = current_app.config['engine']

    if not request.is_json:
        return jsonify({"msg": "Missing JSON in request"}), 400

    user_email = request.json.get('email', None)
    user_password = request.json.get('password', None)

    if not user_email:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not user_password:
        return jsonify({"msg": "Missing password parameter"}), 400

    user_id = sql_queries.get_user_id(engine, user_email, user_password)

    if user_id is None:
        return jsonify({"msg": "Bad username or password"}), 401

    # Identity can be any data that is json serializable
    access_token = fje.create_access_token(identity='user_id={}'.format(user_id),

    return jsonify(access_token=access_token), 200
```

Example 26

Project: *microblog.pub* Author: *tsileo* File: *api.py* GNU Affero General Public License v3.0 6 vo

```
def _user_api_arg(key: str, **kwargs) -> Any:
    """Try to get the given key from the requests, try JSON body, form data and qu
    if request.is_json:
        oid = request.json.get(key)
    else:
        oid = request.args.get(key) or request.form.get(key)

    if not oid:
        if "default" in kwargs:
            app.logger.info(f'{key}={kwargs.get("default")}')
            return kwargs.get("default")

        raise ValueError(f"missing {key}")

    app.logger.info(f"{key}={oid}")
    return oid
```

Example 27

Project: *amnisiac* Author: *treethought* File: *resources.py* BSD 3-Clause "New" or "Revised" License 6 vo

```
def login():
    """Creates and returns access and refresh tokens"""
    if not request.is_json:
```

```
        return jsonify({"msg": "Missing JSON in request"}), 400

    username = request.json.get('username', None)
    password = request.json.get('password', None)

    if not username:
        return jsonify({"msg": "Missing username parameter"}), 400
    if not password:
        return jsonify({"msg": "Missing password parameter"}), 400

    user = User.query.filter(User.username == username).scalar()

    if user and bcrypt.check_password_hash(user.password, password):
        access = create_access_token(identity=username)
        refresh = create_refresh_token(identity=username)

        resp = {'access_token': access, 'refresh_token': refresh}
        return jsonify(resp), 200

    else:
        return jsonify({'msg': "Bad username or password"}), 401
```

Example 28

Project: *DancesafeResults* Author: *siorai* File: *DancesafeResults.py* GNU Affero General Public License v3.0 5 vo

```
def api_add_sample():
    if request.method == "POST":
        newConnection = DBConn()
        currentSession = newConnection.cursor()
        print(request.is_json)
        content = request.get_json()
        print(content)
        eventid = content["eventid"]
        shiftLead = content["shiftLead"]
        tester = content["tester"]
        recorder = content["recorder"]
        typeid = content["typeid"]
        initialSuspect = content["initialSuspect"]
        description = content["description"]
        groundscore = content["groundscore"]
        conclusiveResult = content["conclusiveResult"]
        finalConclusion = content["finalConclusion"]
        acquiredOnSite = content["acquiredOnSite"]
        planToIngest = content["planToIngest"]
        currentSession.execute(
            "INSERT INTO SAMPLE (eventid, shiftlead, tester, recorder, typeid, ini
                eventid,
                shiftLead,
                tester,
                recorder,
                typeid,
                initialSuspect,
                description,
                groundscore,
                conclusiveResult,
                finalConclusion,
                acquiredOnSite,
                planToIngest,
            )
        )
```

```
        newConnection.commit()
        currentSession.close()
        newConnection.close()
        return "JSON posted"
```

Example 29

Project: *release-bot* Author: *user-cont* File: *webhooks.py* GNU General Public License v3.0 5 vo

```
def dispatch_request(self):
        self.logger.info(f'New github webhook call from detected')
        if request.is_json:
            celery_app.send_task(name="task.celery_task.parse_web_hook_payload",
                                 kwargs={"webhook_payload": request.get_json()})
        else:
            self.logger.error("This webhook doesn't contain JSON")
        return jsonify(result={"status": 200})
```

Example 30

Project: *byceps* Author: *byceps* File: *views.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def award_badge_to_user():
    """Award the badge to a user."""
    if not request.is_json:
        abort(415)

    schema = AwardBadgeToUserRequest()
    try:
        req = schema.load(request.get_json())
    except ValidationError as e:
        abort(400, str(e.normalized_messages()))

    badge = badge_service.find_badge_by_slug(req['badge_slug'])
    if not badge:
        abort(400, 'Badge slug unknown')

    user = user_service.find_user(req['user_id'])
    if not user:
        abort(400, 'User ID unknown')

    initiator = user_service.find_user(req['initiator_id'])
    if not initiator:
        abort(400, 'Initiator ID unknown')

    _, event = badge_command_service.award_badge_to_user(
        badge.id, user.id, initiator_id=initiator.id
    )

    signals.user_badge_awarded.send(None, event=event)
```

Example 31

Project: *JustHear.me* Author: *Team-IF* File: *profile.py* GNU Affero General Public License v3.0 5 vo

```
def profile_edit(uuid: str) -> JsonResponse:
    try:
        if not request.is_json:
            return common.rerror("invalid json", 400)
```

```
        token = request.headers['x_access_token']

        if not token:
            return common.rerror("로그인을 해주세요.", 401)
        if uuid != auth.token2uuid(token):
            return common.rerror("자신의 프로필만 수정할 수 있습니다.", 403)

        args = request.json
        if 'username' in args:
            pass
        if 'email' in args:
            pass
        if 'phonenumber' in args:
            pass
        if 'birthday' in args:
            pass
        if 'gender' in args:
            pass
        if 'profileImg' in args:
            pass
        if 'profileMusic' in args:
            pass

        return JsonResponse(get_profile(uuid))

    except Exception as e:
        return common.rerror(e, 500)
```

Example 32

Project: *JustHear.me* Author: *Team-IF* File: *auth.py* GNU Affero General Public License v3.0 5 vo

```
def register() -> JsonResponse:
    try:
        if not request.is_json:
            return common.rerror("invalid json", 400)

        req = request.json
        if not common.emailregex.search(req.get('email')):
            return common.rerror("invalid email", 400)

        uuids = (x['_id'] for x in common.db.user_data.find())
        print(uuids)
        uuid = str(uuid4())
        while uuid in uuids:
            uuid = str(uuid4())

        birth = req.get('birthday')
        if not birth:
            birth = None
        else:
            birth = datetime.datetime.strptime(birth, '%Y-%m-%d').date()

        values = (uuid,
                  req.get('username'),
                  req.get('email'),
                  req.get('pass'),
                  req.get('phonenumber'),
                  birth,
                  req.get('gender'),
                  req.get('profileImg'),
                  req.get('profileMusic'))
```

```
        user = common.User(*values)

        common.db.user_data.insert_one(user.toDict())
        return Response(status=204)

    except Exception as e:
        return common.rerror(e, 500)


# delete login token
```

Example 33

Project: *flask-restutils* Author: *closeio* File: *helpers.py* MIT License 5 vo

```
def request_json():
    if not request.is_json:
        raise BadRequest({'errors': [
            'Invalid request: application/json expected.']})
    return request.get_json()
```

Example 34

Project: *renku-python* Author: *SwissDataScienceCenter* File: *decorators.py* Apache License 2.0 5 vo

```
def accepts_json(f):
    """Wrapper which ensures only JSON payload can be in request."""
    # noqa
    @wraps(f)
    def decorated_function(*args, **kwargs):
        """Represents decorated function."""
        if 'Content-Type' not in request.headers:
            return jsonify(
                error={
                    'code': INVALID_HEADERS_ERROR_CODE,
                    'reason': 'invalid request headers'
                }
            )

        header_check = request.headers['Content-Type'] == 'application/json'

        if not request.is_json or not header_check:
            return jsonify(
                error={
                    'code': INVALID_HEADERS_ERROR_CODE,
                    'reason': 'invalid request payload'
                }
            )

        return f(*args, **kwargs)

    return decorated_function
```

Example 35

Project: *Scrummage* Author: *matamorphosis* File: *Scrummage.py* GNU General Public License v3.0 5 vo

```
def api_auth():

    try:
```

```
        if request.is_json:
            Content = request.get_json()

            if 'Username' in Content and 'Password' in Content:
                Current_User_Object = User(Content['Username'], Content['Password'
                Current_User = Current_User_Object.authenticate()
                Current_User_API = Current_User_Object.API_registration()

                if 'API' in Current_User:
                    Message = "Successful API key registration from " + Current_Us
                    app.logger.warning(Message)
                    Create_Event(Message)
                    JSON_Message = "Registration successful. Welcome " + Current_U

                    if Current_User_API:
                        return jsonify({"Message": JSON_Message, "API Key": Curren

                    else:
                        return jsonify({"Error": "Registration Unsuccessful"})

                elif 'Message' in Current_User:
                    return jsonify({"Error": "Registration Unsuccessful."})

            else:
                return jsonify({"Error": "Invalid fields in request."})

        else:
            return jsonify({"Error": "Invalid request format."})

    except Exception as e:
        return jsonify({"Error": "Invalid request format."})
        app.logger.error(e)
```

Example 36

Project: *NI-Jam-Information-System* Author: *gbaman* File: *decorators.py* GNU General Public License v3.0 5 vo

```
def api_key_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if request.is_json:
            token = json.loads(request.get_json())["token"]
        else:
            token = request.values["token"]
        if not token or token not in secrets.config.api_keys:
            print(f"Received API query with token of \"{token}\" for {request.path
            return redirect("505")
        else:
            print(f"Received API query with token starting with {token[0:5]} for {
            return f(*args, **kwargs)
    return decorated_function
```

Example 37

Project: *codechain-proxy* Author: *CodeChain-io* File: *proxy.py* GNU Affero General Public License v3.0 5 vo

```
def proxy():
        raddr = request.remote_addr
```

```
        if not request.is_json:
            log(raddr, 'Received request is not JSON')
            return invalid_request()

        content = request.get_json(silent=True)
        if content is None:
            log(raddr, 'Failed to parse JSON request: {}'.format(request.data))
            return parse_error()

        if 'id' not in content:
            return ""

        if 'method' in content and content['method'] in app.whitelist:
            try:
                forward_addr = 'http://localhost:{}'.format(app.forward)
                r = requests.post(forward_addr, json=content)
                log(raddr, 'Successfully forwarded {} / Response {}'.format(conter
                return r.content
            except Exception as e:
                log(raddr, 'Failed to receive the response from the server: {}'.fo
                return internal_error(content['id'])
        elif 'method' in content:
            log(raddr, 'Filtered {}: {}'.format(content['method'], content))
            return method_not_found(content['id'])
```

Example 38

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

```
def put_layout(id):
    if not g.current_permissions.layout_update:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    layout = Layout.query.get(id)
    if not layout:
        return make_response(jsonify({'error': 'layout not found'}), 404)

    new_layout = Layout.from_json_data("", data)
    if 'css' in data:
        layout.css = new_layout.css
    if 'html' in data:
        layout.html = new_layout.html
    if 'name' in data:
        layout.name = data['name']
    if 'scripts' in data:
        layout.script = new_layout.script
    if 'subtitle' in data:
        layout.subtitle = new_layout.subtitle
    if 'title' in data:
        layout.title = new_layout.title

    try:
        db.session.commit()
        return jsonify(layout.as_dict())
    except (IntegrityError, StatementError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 39

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

```
def post_task():
    if not g.current_permissions.task_create:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    name = data.get('name')
    num_users = data.get('num_users')
    if not name:
        return make_response(jsonify({'error': 'missing parameter: `name`'}, 400))
    if not num_users:
        return make_response(jsonify({'error': 'missing parameter: `num_users`'},
    try:
        num_users = int(num_users)
    except ValueError:
        return make_response(jsonify({'error': 'invalid number: `num_users`'}, 400

    if 'layout' in data and data['layout']:
        layout = Layout.query.get(data['layout'])
        if not layout:
            return make_response(jsonify({'error': 'layout not found'}), 404)
    else:
        layout = Layout.query.filter(Layout.name == "default").first()

    try:
        task = Task(
            name=name,
            num_users=num_users,
            layout=layout,
        )
        db.session.add(task)
        db.session.commit()
        return jsonify(task.as_dict())
    except (IntegrityError, StatementError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 40

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

```
def put_task(id):
    if not g.current_permissions.task_update:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    task = Task.query.get(id)
    if not task:
        return make_response(jsonify({'error': 'room not found'}), 404)

    try:
        if 'num_users' in data:
            try:
                task.num_users = int(data['num_users'])
            except ValueError:
                return make_response(jsonify({'error': 'invalid number: `num_users
```

```
        if 'name' in data:
            task.name = data['name']
        if 'layout' in data and data['layout']:
            layout = Layout.query.get(data['layout'])
            if not layout:
                return make_response(jsonify({'error': 'layout not found'}), 404)
            task.layout = layout

        db.session.commit()
        return jsonify(task.as_dict())
    except (IntegrityError, StatementError, ValueError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 41

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

```
def put_rooms(name):
    if not g.current_permissions.room_update:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    room = Room.query.get(name)
    if not room:
        return make_response(jsonify({'error': 'room not found'}), 404)

    try:
        if 'label' in data:
            room.label = data['label']
        if 'layout' in data and data['layout']:
            layout = Layout.query.get(data['layout'])
            if not layout:
                return make_response(jsonify({'error': 'layout not found'}), 404)
            room.layout = layout
        if 'read_only' in data:
            room.read_only = data['read_only']
        if 'show_users' in data:
            room.show_users = data['show_users']
        if 'show_latency' in data:
            room.show_users = data['show_latency']
        if 'static' in data:
            room.static = data['static']

        db.session.commit()
        return jsonify(room.as_dict())
    except (IntegrityError, StatementError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 42

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

```
def post_user_logs(id):
    if not g.current_permissions.user_log_event:
        return make_response(jsonify({'error': 'insufficient rights'}), 403)

    user = User.query.get(id)
    if not user:
```

```
        return make_response(jsonify({'error': 'user not found'}), 404)

    data = request.get_json(force=True) if request.is_json else None
    if not data:
        return make_response(jsonify({'error': 'bad request'}, 400))

    event = data.get('event')
    if not event:
        return make_response(jsonify({'error': 'missing parameter: `event`'}, 400)

    if 'room' in data:
        room = Room.query.get(data['room'])
        if not room:
            return make_response(jsonify({'error': 'room not found'}), 404)
    else:
        room = None

    try:
        return jsonify(log_event(event, user, room, data.get('data')).as_dict())
    except (IntegrityError, StatementError) as e:
        return make_response(jsonify({'error': str(e)}), 400)
```

Example 43

Project: *Nexmo-Telegram-Bot* Author: *blopa* File: *app.py* MIT License 5 vo

```
def nexmo_webhook():
        if request.is_json:
                text_message = format_message(request.get_json())
                bot.sendMessage(chat_id=TELEGRAM_CHAT_ID, text=text_message, parse
                # pprint(request.get_json())
        else:
                data = dict(request.form) or dict(request.args)
                text_message = format_message(data)
                bot.sendMessage(chat_id=TELEGRAM_CHAT_ID, text=text_message, parse
                # pprint(data)

        return ('', 204)
```

Example 44

Project: *cheetah-api* Author: *marcosflobo* File: *main.py* BSD 3-Clause "New" or "Revised" License 5 vo

```
def authenticate():
    if not request.is_json:
        abort(400)
    dispacher = Dispacher(config)
    response = dispacher.authenticate(request.get_json())
    print(request.get_json())
    return get_json_response(response)
```

Example 45

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

```
def post(self):
        """Authenticate user or node"""
        log.debug("Authenticate user using username and password")

        if not request.is_json:
```

```
            log.warning('Authentication failed because no JSON body was provided!'
            return {"msg": "Missing JSON in request"}, HTTPStatus.BAD_REQUEST

        # Check JSON body
        username = request.json.get('username', None)
        password = request.json.get('password', None)
        if not username and password:
            msg = "Username and/or password missing in JSON body"
            log.error(msg)
            return {"msg": msg}, HTTPStatus.BAD_REQUEST

        log.debug(f"Trying to login {username}")
        user, code = self.user_login(username, password)
        if code is not HTTPStatus.OK:  # login failed
            log.error(f"Incorrect username/password combination for user='{usernam
            return user, code

        token = create_access_token(user)
        ret = {
            'access_token': token,
            'refresh_token': create_refresh_token(user),
            'user_url': server.api.url_for(server.resource.user.User, user_id=user
            'refresh_url': server.api.url_for(RefreshToken),
        }

        log.info(f"Succesfull login from {username}")
        return ret, HTTPStatus.OK, {'jwt-token': token}
```

Example 46

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

```
def post(self):
        """Authenticate as Node."""
        log.debug("Authenticate Node using api key")

        if not request.is_json:
            log.warning('Authentication failed because no JSON body was provided!'
            return {"msg": "Missing JSON in request"}, HTTPStatus.BAD_REQUEST

        # Check JSON body
        api_key = request.json.get('api_key', None)
        if not api_key:
            msg = "api_key missing in JSON body"
            log.error(msg)
            return {"msg": msg}, HTTPStatus.BAD_REQUEST

        node = db.Node.get_by_api_key(api_key)

        if not node:  # login failed
            log.error(f"Api key is not recognised")
            return {"msg": "Api key is not recognised!"}

        token = create_access_token(node)
        ret = {
            'access_token': create_access_token(node),
            'refresh_token': create_refresh_token(node),
            'node_url': server.api.url_for(server.resource.node.Node, id=node.id),
            'refresh_url': server.api.url_for(RefreshToken),
        }
```

```
        log.info("Succesfull login as node '{}' ({})".format(node.id, node.name))
        return ret, HTTPStatus.OK, {'jwt-token': token}
```

Example 47

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 48

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

```
def formulation_instance_service(f_id):
    resp = flask.Response(json.dumps({'status': 'failed'}))
    if request.method == 'PUT':
        if request.is_json:
            p_json_list = request.json['properties']
            formulation = Formulation.query.get(f_id)
            formulation.formulation_property.delete()
            # fp_rs = Formulation.query.get(f_id).formulation_property
            # FormulationProperty.query.filter(FormulationProperty.formulation_id
            for p in p_json_list:
                formulation.formulation_property.append(FormulationProperty(
```

```
                    key=p['keyName'],
                    value=p['valueName']
                ))
            db.session.commit()
            p_list = []
            for p in formulation.formulation_property:
                p_list.append({p.key: p.value})
            resp = flask.Response(json.dumps({'status': 'success',
                                              'formulation_id': formulation.id,
                                              'formulation_properties': p_list}))
    elif request.method == 'DELETE':
        formulation = Formulation.query.get(f_id)
        test_count = formulation.test.count()
        if test_count > 0:
            resp = flask.Response(json.dumps({'status': 'failed',
                                              'error': 'the tests count of formula
                                                       'delete tests first'}))
        else:
            Formulation.query.filter(Formulation.id == f_id).delete()
            db.session.commit()
    return set_debug_response_header(resp)
```

Example 49

Project: *flask-unchained* Author: *briancappello* File: *security_controller.py* MIT License 5 vo

```
def login(self):
        """
        View function to log a user in. Supports html and json requests.
        """
        form = self._get_form('SECURITY_LOGIN_FORM')
        if form.validate_on_submit():
            try:
                self.security_service.login_user(form.user, form.remember.data)
            except AuthenticationError as e:
                form._errors = {'_error': [str(e)]}
            else:
                self.after_this_request(self._commit)
                if request.is_json:
                    return self.jsonify({'token': form.user.get_auth_token(),
                                         'user': form.user})
                self.flash(_('flask_unchained.bundles.security:flash.login'),
                           category='success')
                return self.redirect('SECURITY_POST_LOGIN_REDIRECT_ENDPOINT')
        else:
            # FIXME-identity
            identity_attrs = app.config.SECURITY_USER_IDENTITY_ATTRIBUTES
            msg = f"Invalid {', '.join(identity_attrs)} and/or password."

            # we just want a single top-level form error
            form._errors = {'_error': [msg]}
            for field in form._fields.values():
                field.errors = None

        if form.errors and request.is_json:
            return self.jsonify({'error': form.errors.get('_error')[0]},
                                code=HTTPStatus.UNAUTHORIZED)

        return self.render('login',
                           login_user_form=form,
                           **self.security.run_ctx_processor('login'))
```

Example 50

Project: *flask-unchained* Author: *briancappello* File: *security_controller.py* MIT License 5 vo

```
def logout(self):
        """
        View function to log a user out. Supports html and json requests.
        """
        if current_user.is_authenticated:
            self.security_service.logout_user()

        if request.is_json:
            return '', HTTPStatus.NO_CONTENT

        self.flash(_('flask_unchained.bundles.security:flash.logout'),
                   category='success')
        return self.redirect('SECURITY_POST_LOGOUT_REDIRECT_ENDPOINT')
```