Python flask.request.get_data() Examples

The following are code examples for showing how to use *flask.request.get_data()*. 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: *ssrspeed_backup* Author: *mazhenting* File: *getpostdata.py* GNU General Public License v3.0 7 vo

```
def getPostData():
        #print(request.content_type)
        data = {}
        if (request.content_type.startswith('application/json')):
                data = request.get_data()
                return json.loads(data.decode("utf-8"))
        elif(request.content_type.startswith("application/x-www-form-urlencoded")):
                #print(1)
                #print(urllib.parse.parse_qs(request.get_data().decode("utf-8")))
                return parse_qs_plus(urllib.parse.parse_qs(request.get_data().dec
        else:
                for key, value in request.form.items():
                        if key.endswith('[]'):
                                data[key[:-2]] = request.form.getlist(key)
                        else:
                                data[key] = value
                return data
```

Example 2

Project: *pluralsight* Author: *jamesbannan* File: *app.py* MIT License 6 vo

```
def predict_image_handler():
    try:
        imageData = None
        if ('imageData' in request.files):
            imageData = request.files['imageData']
        else:
            imageData = io.BytesIO(request.get_data())

        #img = scipy.misc.imread(imageData)
        img = Image.open(imageData)
        results = predict_image(img)
        return json.dumps(results)
    except Exception as e:
        print('EXCEPTION:', str(e))
        return 'Error processing image', 500


# Like the CustomVision.ai Prediction service /url route handles url's
# in the body of hte request of the form:
#     { 'Url': '<http url>'}
```

Example 3

Project: *karp-backend* Author: *spraakbanken* File: *helpers.py* MIT License 6 vo

```
def read_data():
    """ Read the incoming data, make sure a message exists
        Raise errors if data is not well-formatted
    """
    try:
        request.get_data()
        data = loads(request.data)
    except ValueError as e:
        raise errors.KarpParsingError(str(e))
    if "message" not in data:
        # fail if message is not there
        raise errors.KarpGeneralError("Input data not ok")
    if not data:
        errstr = "The source is empty. Empty documents not allowed"
        raise errors.KarpParsingError(errstr)
    return data
```

Example 4

Project: *karp-backend* Author: *spraakbanken* File: *suggestions.py* MIT License 6 vo

```
def acceptmodified(lexicon, _id):
    try:
        request.get_data()
        data = loads(request.data)
        modified_doc = data
        ans = savesuggestion(
            lexicon, _id, status="accepted_modified", source=modified_doc
        )
        return jsonify(ans)
    except (esExceptions.RequestError, esExceptions.TransportError) as e:
        _logger.exception(e)
        update.handle_update_error(
            e, {"id": _id, "data": data}, helpers.get_user(), "accept modified"
        )
        raise errors.KarpElasticSearchError(
            "Error during update. Document not saved.", debug_msg=str(e)
        )
    except Exception as e:
        _logger.exception(e)
        update.handle_update_error(
            e, {"id": _id, "data": data}, helpers.get_user(), "accept modified"
        )
        raise errors.KarpGeneralError(str(e))
```

Example 5

Project: *cellphonedb* Author: *Teichlab* File: *web_endpoint_query_autocomplete.py* MIT License 6 vo

```
def post(self):
        parameters = json.loads(request.get_data(as_text=True))

        partial_element = parameters['partial_element']

        if len(partial_element) < 2:
            return flask.jsonify({'success': True, 'result': []})

        try:
            interactions = cellphonedb_app.cellphonedb.query.autocomplete_launcher
            response = {
                'success': True,
```

```
                'result': interactions.to_dict(orient='records')
            }
        except:
            response = {
                'success': False
            }
            print(traceback.print_exc(file=sys.stdout))

        return flask.jsonify(response)
```

Example 6

Project: *cellphonedb* Author: *Teichlab* File: *web_endpoint_query_complex_deconvoluted.py* MIT License 6 vo

```
def post(self):
        parameters = json.loads(request.get_data(as_text=True))

        complex_name = parameters['complex_name']

        deconvoluted = cellphonedb_app.cellphonedb.query.get_complex_deconvoluted(

        if deconvoluted.empty:
            self.attach_error(
                {'code': 'element_not_found', 'title': '%s is not CellPhoneDB Comp
                 'details': '%s is not present in CellPhoneDB complex table' % com
        else:
            self._attach_csv(deconvoluted.to_csv(index=False, sep=','), 'result')

        self._commit_attachments()

        return Response(self._msg.as_string(), mimetype='multipart/form-data; bour
```

Example 7

Project: *cellphonedb* Author: *Teichlab*
File: *web_endpoint_query_find_interactions_by_element.py* MIT License 6 vo

```
def post(self):
        parameters = json.loads(request.get_data(as_text=True))

        receptor = parameters['receptor']

        interactions = cellphonedb_app.cellphonedb.query.find_interactions_by_elem

        if interactions.empty:
            self.attach_error(
                {'code': 'result_not_found', 'title': '%s is not CellPhoneDB inter
                 'details': '%s is not present in CellPhoneDB interactor enabled t
        else:
            self._attach_csv(interactions.to_csv(index=False, sep=','), 'ligands')

        self._commit_attachments()

        return Response(self._msg.as_string(), mimetype='multipart/form-data; bour
```

Example 8

Project: *End_to_end_machine_learning_approach_for_crop_yeild_prediction* Author: *shardulinamdar4* 6 vo

File: *b_server.py* MIT License

```
def bareilly():
    body = request.get_data()
    header = request.headers

    try:
        num1 = int(request.args['pC']) # Previous year crop yeild
        num2 = int(request.args['area']) # Area
        num3 = int(request.args['r1']) # last year rainfall
        num4 = int(request.args['r2']) # this rainfall

        if((num1!=None) and (num2!=None) and (num3!=None) and (num4!=None)):
           res = predict_bareilly(num1, num2,num3,num4) # call the predict_bareill
        else:
           res = {'success': False,
                  'message': 'input proper data'}

    except:
           res = {'success': False,
                  'message': 'unkonwn error'}

    return jsonify(res)
```

Example 9

Project: *End_to_end_machine_learning_approach_for_crop_yeild_prediction* Author: *shardulinamdar4*
File: *b_server.py* MIT License 6 vo

```
def coimbatore():
    body = request.get_data()
    header = request.headers

    try:
        num1 = int(request.args['pC']) # Previous year crop yeild
        num2 = int(request.args['area']) # Area
        num3 = int(request.args['r1']) # last year rainfall
        num4 = int(request.args['r2']) # this year rainfall

        if((num1!=None) and (num2!=None) and (num3!=None) and (num4!=None)):
           res = predict_coimbatore(num1, num2,num3,num4) # call the predict_barei
        else:
           res = {'success': False,
                  'message': 'input proper data'}

    except:
           res = {'success': False,
                  'message': 'unkonwn error'}

    return jsonify(res)
```

Example 10

Project: *End_to_end_machine_learning_approach_for_crop_yeild_prediction* Author: *shardulinamdar4*
File: *b_server.py* MIT License 6 vo

```
def ghazipur():
    body = request.get_data()
    header = request.headers

    try:
```

```
        num1 = int(request.args['pC']) # Previous year crop yeild
        num2 = int(request.args['area']) # Area
        num3 = int(request.args['r1']) # last year rainfall
        num4 = int(request.args['r2']) # this year  rainfall

        if((num1!=None) and (num2!=None) and (num3!=None) and (num4!=None)):
           res = predict_ghazipur(num1, num2,num3,num4) # call the predict_bareill
        else:
           res = {'success': False,
                  'message': 'input proper data'}

    except:
           res = {'success': False,
                  'message': 'unkonwn error'}

    return jsonify(res)
```

Example 11

Project: *End_to_end_machine_learning_approach_for_crop_yeild_prediction* Author: *shardulinamdar4*
File: *b_server.py* MIT License 6 vo

```
def hassan():
    body = request.get_data()
    header = request.headers

    try:
        num1 = int(request.args['pC']) # Previous year crop yeild
        num2 = int(request.args['area']) # Area
        num3 = int(request.args['r1']) # last year rainfall
        num4 = int(request.args['r2']) #this year  rainfall

        if((num1!=None) and (num2!=None) and (num3!=None) and (num4!=None)):
           res = predict_hassan(num1, num2,num3,num4)
        else:
           res = {'success': False,
                  'message': 'input proper data'}

    except:
           res = {'success': False,
                  'message': 'unkonwn error'}

    return jsonify(res)
```

Example 12

Project: *End_to_end_machine_learning_approach_for_crop_yeild_prediction* Author: *shardulinamdar4*
File: *b_server.py* MIT License 6 vo

```
def sindhudurg():
    body = request.get_data()
    header = request.headers

    try:
        num1 = int(request.args['pC']) # Previous year crop yeild
        num2 = int(request.args['area']) # Area
        num3 = int(request.args['r1']) # last year rainfall
        num4 = int(request.args['r2']) # this year rainfall

        if((num1!=None) and (num2!=None) and (num3!=None) and (num4!=None)):
           res = predict_sindhudurg(num1, num2,num3,num4) # call the predict_sindh
```

```
        else:
            res = {'success': False,
                   'message': 'input proper data'}

    except:
            res = {'success': False,
                   'message': 'unkonwn error'}

    return jsonify(res)
```

Example 13

Project: *FXTest* Author: *liwanlei* File: *views.py* MIT License 6 vo

```
def post(self):
        id = request.get_data('id')
        project = id.decode('utf-8')
        if not project:
            return jsonify({'msg': u'没有发送数据', 'code': 38, 'data': ''})
        project_is = Project.query.filter_by(project_name=project).first()
        if not project_is:
            return jsonify(({'msg': u'成功', 'code': 200, 'data': []}))
        testreport = TestResult.query.filter_by(projects_id=project_is.id, status=
        testreportlist = []
        for test in testreport:
            testreportlist.append({'test_num': test.test_num, 'pass_num': test.pas
                                   'fail_num': test.fail_num, 'hour_time': str(tes
                                   'test_rep': test.test_rep, 'test_log': test.tes
                                   'Exception_num': test.Exception_num, 'can_num':
                                   'wei_num': test.wei_num, 'test_time': str(test.
                                   'Test_user_id': test.users.username, 'id': test
                                   'fenshu': test.pass_num / test.test_num})
        return jsonify(({'msg': u'成功', 'code': 200, 'data': (testreportlist)}))
```

Example 14

Project: *sentry-python* Author: *getsentry* File: *test_flask.py* BSD 2-Clause "Simplified" License 6 vo

```
def test_flask_large_json_request(sentry_init, capture_events, app):
    sentry_init(integrations=[flask_sentry.FlaskIntegration()])

    data = {"foo": {"bar": "a" * 2000}}

    @app.route("/", methods=["POST"])
    def index():
        assert request.get_json() == data
        assert request.get_data() == json.dumps(data).encode("ascii")
        assert not request.form
        capture_message("hi")
        return "ok"

    events = capture_events()

    client = app.test_client()
    response = client.post("/", content_type="application/json", data=json.dumps(d
    assert response.status_code == 200

    event, = events
    assert event["_meta"]["request"]["data"]["foo"]["bar"] == {
        "": {"len": 2000, "rem": [["!limit", "x", 509, 512]]}
```

```
    }
    assert len(event["request"]["data"]["foo"]["bar"]) == 512
```

Example 15

Project: *sentry-python* Author: *getsentry* File: *test_flask.py* BSD 2-Clause "Simplified" License 6 vo

```
def test_flask_empty_json_request(sentry_init, capture_events, app, data):
    sentry_init(integrations=[flask_sentry.FlaskIntegration()])

    @app.route("/", methods=["POST"])
    def index():
        assert request.get_json() == data
        assert request.get_data() == json.dumps(data).encode("ascii")
        assert not request.form
        capture_message("hi")
        return "ok"

    events = capture_events()

    client = app.test_client()
    response = client.post("/", content_type="application/json", data=json.dumps(d
    assert response.status_code == 200

    event, = events
    assert event["request"]["data"] == data
```

Example 16

Project: *sentry-python* Author: *getsentry* File: *test_flask.py* BSD 2-Clause "Simplified" License 6 vo

```
def test_flask_medium_formdata_request(sentry_init, capture_events, app):
    sentry_init(integrations=[flask_sentry.FlaskIntegration()])

    data = {"foo": "a" * 2000}

    @app.route("/", methods=["POST"])
    def index():
        assert request.form["foo"] == data["foo"]
        assert not request.get_data()
        assert not request.get_json()
        capture_message("hi")
        return "ok"

    events = capture_events()

    client = app.test_client()
    response = client.post("/", data=data)
    assert response.status_code == 200

    event, = events
    assert event["_meta"]["request"]["data"]["foo"] == {
        "": {"len": 2000, "rem": [["!limit", "x", 509, 512]]}
    }
    assert len(event["request"]["data"]["foo"]) == 512
```

Example 17

Project: *batcomputer* Author: *benc-uk* File: *server.py* MIT License 6 vo

```
def main_api(project=None):
  try:
    request_dict = json.loads(request.get_data().decode('utf-8'))
    results = predictor.predict(request_dict)
    return jsonify(results)

  except KeyError as key_error:
    print('### KEY_ERROR:', str(key_error))
    return Response(json.dumps({'error': 'Value: '+str(key_error)+' not found in m
  except Exception as err:
    print('### EXCEPTION:', str(err))
    return Response(json.dumps({'error': str(err)}), status=500, mimetype='applica


#
# API route - for status/info
#
```

Example 18

Project: *python-deploy-Tesseract-OCR-to-Heroku* Author: *kevin1061517* File: *linebot_ccu.py* MIT License 6 vo

```
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']
    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)
    # handle webhook body
    try:
        handler.handle(body,signature)
    except LineBotApiError as e:
        print("Catch exception from LINE Messaging API: %s\n" % e.message)
        for m in e.error.details:
            print("ERROR is %s: %s" % (m.property, m.message))
        print("\n")
    except InvalidSignatureError:
        abort(400)
    return 'OK'
```

Example 19

Project: *padex* Author: *dszakallas* File: *padex.py* GNU General Public License v2.0 6 vo

```
def decrypt():
    if request.content_length != 44:
        abort(400)

    data = b64decode(request.get_data())
    aes = AES.new(key, AES.MODE_CBC, IV=iv)
    mess = aes.decrypt(data)
    padsize = mess[-1]

    if padsize < 1 or padsize > 16:
        abort(403)

    for x in mess[-padsize:-1]:
        if x != padsize:
            abort(403)
```

```
    return 'OK', 200
```

Example 20

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

```
def __parse_body(self, schema):
        if not request.get_data():
            raise BadRequest('Payload missing.')

        parser, mimetype = self.content_negotiation.select_parser(request, self.de

        if not parser:
            raise UnsupportedMediaType(request.headers['content-type'])

        try:
            decoded_data = parser.parse(request.get_data(), mimetype)
        except:
            raise BadRequest('Malformed request.')

        model, errors = schema.load(decoded_data)

        if errors:
            raise ValidationError(errors, data=request.get_data(), location='body

        return model
```

Example 21

Project: *flask-telegram-relay-bot* Author: *mimicmobile* File: *main.py* MIT License 6 vo

```
def relay():
    with app.app_context():
        muted = current_app.muted
        chats = current_app.chats
    if not muted:
        try:
            request_data = request.get_data().decode('latin-1')
            logger.debug("request data: {}".format(request_data))
            parsed_json = json.loads(request_data, strict=False)
        except:
            traceback.print_exc()
            return "ERROR"
        for chat in chats:
            chat_id = telegram_bot.get_chat(chat).id
            utils.send_message(chat_id=chat_id, text=parsed_json['message'])
        return "OK"
    return "MUTED"
```

Example 22

Project: *imagery* Author: *dibyadas* File: *app.py* GNU Affero General Public License v3.0 6 vo

```
def handle():
        try:
                url_data = request.get_data()
                print(url_data)
                '''Slacks interactive message request payload is in the form of
                application/x-www-form-urlencoded JSON string. Getting first actio
```

```
                from it.'''
                url_data = json.loads(parse_qs(url_data.decode('utf-8'))['payload'
                eph_value = True if url_data['value'] == "yes" else False
                print(url_data['name'] + " : " + url_data['value'] + " : " + str(e
                if eph_value:
                        params = url_data['name'].split('|')
                        user_id = params[1]
                        channel_id = params[2]
                        file_id = params[3]
                        file_permalink = params[4]
                        comment = params[5]
                        timestamp = params[6]
                        i = pool.apply_async(download_file, [file_permalink, file_
                else:
                        print('---No chosen---')
        except Exception as err:
                print(err)
        finally:
                return jsonify({"response_type": "ephemeral",   "replace_original"
```

Example 23

Project: *Raspberry-Docker-Tensorflow-Pillow-Flask* Author: *Ellerbach* File: *app.py* MIT License 6 vo

```
def predict_image_handler(project=None):
    try:
        imageData = None
        if ('imageData' in request.files):
            imageData = request.files['imageData']
        elif ('imageData' in request.form):
            imageData = request.form['imageData']
        else:
            imageData = io.BytesIO(request.get_data())

        img = Image.open(imageData).convert('RGB')
        results = predict_image(img)
        return jsonify(results)
    except Exception as e:
        print('EXCEPTION:', str(e))
        return 'Error processing image', 500


# Like the CustomVision.ai Prediction service /url route handles url's
# in the body of hte request of the form:
#     { 'Url': '<http url>'}
```

Example 24

Project: *notifications-api* Author: *alphagov* File: *post_template.py* MIT License 6 vo

```
def post_template_preview(template_id):
    # The payload is empty when there are no place holders in the template.
    _data = request.get_data(as_text=True)
    if not _data:
        _data = {}
    else:
        _data = get_valid_json()

    _data['id'] = template_id

    data = validate(_data, post_template_preview_request)
```

```
    template = templates_dao.dao_get_template_by_id_and_service_id(
        template_id, authenticated_service.id)

    template_object = get_template_instance(
        template.__dict__, values=data.get('personalisation'))

    check_placeholders(template_object)

    resp = create_post_template_preview_response(template=template,
                                                 template_object=template_object)

    return jsonify(resp), 200
```

Example 25

Project: *fumblechain* Author: *kudelskisecurity* File: *api.py* GNU General Public License v3.0 6 vo

```
def create_transaction():
    """Add and broadcast the given transaction.
    Returns HTTP 400 if the transaction is considered invalid."""
    try:
        # retrieve transaction from request body
        jso = request.get_data(as_text=True)
        tx = Transaction.from_json(jso)

        # add transaction to local blockchain
        success = app.p2p.bc.add_transaction(tx)
        if success:
            # broadcast transaction to p2p network
            app.p2p.broadcast_tx(tx)

            return Response(tx.to_json(), status=HTTP_CREATED)
        else:
            logger.debug("failed to add tx")
            raise BadRequest()
    except BadRequest:
        raise
    except BaseException as e:
        logger.debug(e)
        logger.debug(traceback.format_exc())
        logger.debug(sys.exc_info())
        raise BadRequest()
```

Example 26

Project: *fumblechain* Author: *kudelskisecurity* File: *api.py* GNU General Public License v3.0 6 vo

```
def create_block():
    """Add and broadcast the given block.
    Returns HTTP 400 if the block is considered invalid."""
    try:
        # retrieve block from request body
        jso = request.get_data(as_text=True)
        b = Block.from_json(jso)

        # add block to local blockchain
        success = app.p2p.bc.discard_block(b)

        if success:
            # broadcast block to p2p network
```

```
            app.p2p.broadcast_block(b)

            logger.debug(f"block {b.index} added")
            return Response(b.to_json(), status=HTTP_CREATED)
        else:
            logger.debug("failed to add block (discard)")
            raise BadRequest()
    except BadRequest:
        raise
    except BaseException as e:
        logger.debug(e)
        raise BadRequest()
```

Example 27

Project: *zmirror* Author: *aploium* File: *zmirror.py* MIT License 5 vo

```
def prepare_client_request_data():
    """
    解析出浏览者发送过来的data, 如果是文本, 则进行重写
    如果是文本, 则对文本内容进行重写后返回str
    如果是二进制则, 则原样返回, 不进行任何处理 (bytes)
    :rtype: Union[str, bytes, None]
    """
    data = request.get_data()  # type: bytes

    # 尝试解析浏览器传入的东西的编码
    encoding = encoding_detect(data)

    if encoding is not None:
        try:
            data = data.decode(encoding=encoding)  # type: str
        except:
            # 解码失败, data是二进制内容或无法理解的编码, 原样返回, 不进行重写
            encoding = None
            pass
        else:
            # data是文本内容, 则进行重写, 并返回str
            data = client_requests_text_rewrite(data)  # type: str

    # 下面这个if是debug用代码, 对正常运行无任何作用
    if developer_string_trace:  # coverage: exclude
        if isinstance(data, str):
            data = data.encode(encoding=encoding)
        if developer_string_trace.encode(encoding=encoding) in data:
            infoprint('StringTrace: appears after client_requests_bin_rewrite, coo

    return data, encoding
```

Example 28

Project: *pluralsight* Author: *jamesbannan* File: *app.py* MIT License 5 vo

```
def predict_url_handler():
    try:
        image_url = json.loads(request.get_data())['Url']
        results = predict_url(image_url)
        return json.dumps(results)
    except Exception as e:
```

```
        print('EXCEPTION:', str(e))
        return 'Error processing image'
```

Example 29

Project: *flask-proxy* Author: *mecforlove* File: *__init__.py* BSD 2-Clause "Simplified" License 5 vo

```
def as_view(cls):
        def _view(*args, **kwargs):
            host = cls._get_attr(cls.host)
            scheme = cls._get_attr(cls.scheme, 'http')
            params = cls._get_attr(cls.params)
            port = cls._get_attr(cls.port, 80)
            timeout = cls._get_attr(cls.timeout)
            method = request.method
            uri = request.url.split(cls.prefix, 1)[1]
            base_url = '%s://%s:%s' % (scheme, host, port)
            url = base_url + uri
            headers = dict(request.headers)
            # Change `Host` in request header.
            headers['Host'] = host
            resp = requests.request(
                method,
                url,
                params=params,
                headers=headers,
                data=request.get_data(),
                stream=True,
                timeout=timeout)
            # Remove some response headers.
            excluded_headers = [
                'content-length', 'transfer-encoding', 'connection'
            ]
            for h in excluded_headers:
                if h in resp.headers:
                    resp.headers.pop(h)
            return Response(resp.raw.read(), resp.status_code,
                            dict(resp.headers))

        return _view
```

Example 30

Project: *turboparser-semafor* Author: *ReutersMedia* File: *index.py* GNU General Public License v3.0 5 vo

```
def parse_frames(pipeline):
    if request.method == 'GET':
        # parse from text parameter
        d = request.args.get('t')
        if d == None:
            abort(400)
    else:
        d = request.get_data(as_text=True)
    tstart = time.time()
    try:
        r = proc_input(d,pipeline.upper())
    except:
        LOGGER.exception("Error processing input")
        abort(500)
    return jsonify(r)
```

Example 31

Project: *cloudml-edge-automation* Author: *GoogleCloudPlatform* File: *app.py* Apache License 2.0 5 vo

```
def predict():
    #  Sending without header
    raw = request.get_data()
    req = json.loads(raw)
    print(req)
    result = tf_session.infer(req["path"])
    #  return "OK"
    return jsonify(result)
```

Example 32

Project: *karp-backend* Author: *spraakbanken* File: *searching.py* MIT License 5 vo

```
def formatpost():
    """ Formats the posted data into wanted format
        The data should be a list
        Currently only working for saol
    """
    # get and parse data
    request.get_data()
    data = request.data
    try:
        data = json.loads(data)
    except ValueError as e:
        raise errors.KarpParsingError(str(e))

    # set all allowed lexicons (to avoid authentication exception
    auth, permitted = validate_user(mode="read")
    # find the wanted format
    settings = parser.make_settings(permitted, {"size": 25})
    parser.parse_extra(settings)
    to_format = settings.get("format", "")
    mode = parser.get_mode()
    _logger.debug('mode "%s"', mode)
    index, typ = conf_mgr.get_mode_index(mode)

    if to_format:
        if not isinstance(data, list):
            data = [data]
        errmsg = "Unkown format %s for mode %s" % (settings["format"], mode)
        format_list = conf_mgr.extra_src(
            mode, "format_list", helpers.notdefined(errmsg)
        )
        ok, html = format_list(
            data, conf_mgr.elastic(mode=mode), settings["format"], index
        )
        return jsonify({"all": len(data), "ok": ok, "data": html})

    else:
        raise errors.KarpQueryError("Unkown format %s" % to_format)
```

Example 33

Project: *karp-backend* Author: *spraakbanken* File: *suggestions.py* MIT License 5 vo

```
def savesuggestion(lexicon, _id, status="accepted", source=""):
    sugg_index, typ = conf_mgr.get_lexicon_suggindex(lexicon)
```

```
    es = conf_mgr.elastic(lexicon=lexicon)
    suggestion = es.get(index=sugg_index, doc_type=typ, id=_id)
    auth, permitted = validate_user()
    set_lexicon = suggestion["_source"]["lexiconName"]
    helpers.check_lexiconName(lexicon, set_lexicon, "rejectsuggestion", _id)
    if lexicon not in permitted:
        raise errors.KarpAuthenticationError(
            "You are not allowed to update lexicon %s" % lexicon
        )

    origin = dbselect(lexicon, suggestion=True, _id=_id, max_hits=1)[0]
    origid = origin["origid"]
    request.get_data()
    data = loads(request.data)
    message = data.get("message")
    suggestion["message"] = message
    suggestion["version"] = origin["version"]
    if not source:
        source = suggestion
    # the user log in is checked in add_doc
    # add_doc raises exception if ES
    if origid:
        # update in ES
        ans = update.update_doc(lexicon, origid, data=source, live=False)
    else:
        # add to ES
        ans = update.add_doc(lexicon, live=False, data=source)
        origid = ans.get("_id")
    # mark as accepted
    ok, err = update.modify_db(_id, lexicon, message, status, origid=origid)
    # delete from suggestion index
    suggans = update.delete_entry(lexicon, _id, sql=False, live=False, suggestion=
    ans["sugg_db_loaded"] = ok
    ans["sugg_es_ans"] = suggans
    if not ok:
        _logger.debug(err)
    update.send_notification(origin["user"], message, _id, status)
    return ans
```

Example 34

Project: *FXTest* Author: *liwanlei* File: *views.py* MIT License 5 vo

```
def post(self):
        url = (request.get_data().decode('utf-8'))
        url_base = (url.split('&')[0])
        jobname = url.split('&')[1]
        try:
            log = Conlenct_jenkins().job_bulid_log(url_base, jobname)
            return jsonify({"code": 200, 'data': str(log)})
        except Exception as e:
            return jsonify({'code': 701, 'data': str(e)})
```

Example 35

Project: *FXTest* Author: *liwanlei* File: *views.py* MIT License 5 vo

```
def post(self):
        project=request.get_data('value')
        project=project.decode('utf-8')
        changpr=Project.query.filter_by(project_name=project).first()
```

```
        if not changpr :
            return  jsonify({"code":26,'msg':'项目查询不到','data':''})
        if changpr.status==True:
            return  jsonify({"code":27,'msg':'项目已经删除或者冻结','data':''})
        testevent=Interfacehuan.query.filter_by(projects=changpr,status=False).all
        testeventlist=[]
        for testeven in testevent:
            testeventlist.append({"url":testeven.url})
        return  jsonify({'code':200,'data':testeventlist,'msg':'请求成功'})
```

Example 36

Project: *dbot-server* Author: *ATNIO* File: *decorates.py* MIT License 5 vo

```
def middleware(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        dbot_address = kwargs.get('dbot_address')
        middleware = dbot.get_service(dbot_address).middleware
        request.new_method, request.new_args, request.new_headers, request.new_dat
            request.method, request.args, request.headers, request.get_data())
        return f(*args, **kwargs)
    return decorated
```

Example 37

Project: *dbot-server* Author: *ATNIO* File: *proxy.py* MIT License 5 vo

```
def proxy(dbot_address, uri, proxy_uri=None):
    # proxy requst to api server host
    dbot_service = dbot.get_service(dbot_address)
    if not dbot_service:
        raise InvalidUsage('dbot address not found', status_code=404)
    url = '{}://{}/{}'.format(dbot_service.protocol, dbot_service.api_host, remove
    headers = {key: value for (key, value) in request.headers if key != 'Host'}
    # Pass original Referer for subsequent resource requests
    headers["Referer"] = url

    logger.info("Proxy the API {}: {}, with headers: \n{}".format(request.method,

    # Fetch the URL, and stream it back
    try:
        resp = requests.request(
            url=url,
            method=request.method,
            params=request.args,
            headers=headers,
            #  TODO: Usually it's a bad idea to call get_data() without checking
            #  content length first as a client could send dozens of megabytes or
            #  to cause memory problems on the server.
            data=request.get_data(),
            cookies=request.cookies,
            allow_redirects=False)

        logger.info("Got {} response from {}".format(resp.status_code, url))

        excluded_headers = ['content-encoding', 'content-length', 'transfer-encodi
        headers = [(name, value) for (name, value) in resp.raw.headers.items()
                if name.lower() not in excluded_headers]

        return Response(resp.content, resp.status_code, headers)
```

```
    except Exception as err:
        raise InvalidUsage('Cannot proxy the request.\n{}'.format(err), status_coc
```

Example 38

Project: *flask-stripe* Author: *raicheff* File: *extension.py* MIT License 5 vo

```
def handle_webhook(self):
        """
        https://stripe.com/docs/webhooks
        https://stripe.com/docs/api#event_types
        ---
        https://www.petekeen.net/stripe-webhook-event-cheatsheet
        """
        event_id = request.get_json().get('id')
        logger.info('event_id=%s', event_id)

        signature = request.headers.get('stripe-signature')
        if signature is None:
            abort(BAD_REQUEST)

        try:
            # event = Event.retrieve(event_id)
            event = Webhook.construct_event(request.get_data(as_text=True), signa
            logger.info('event=%s', event)
            namespace.signal(event.type).send(self, object=event.data.object)
        except ValueError as error:
            # Invalid payload
            logger.warning('error=%s', error)
            abort(BAD_REQUEST)
        except SignatureVerificationError as error:
            # Invalid signature
            logger.warning('error=%s', error)
            abort(BAD_REQUEST)

        if event_id == TEST_EVENT_ID:
            return Response(status=OK)

        return Response(status=OK)
```

Example 39

Project: *covador* Author: *baverman* File: *flask.py* MIT License 5 vo

```
def get_form():
    try:
        return request._covador_form
    except AttributeError:
        ctype = request.content_type or ''
        if ctype.startswith('multipart/form-data'):
            form = request.form.to_dict(False)
        elif ctype.startswith('application/x-www-form-urlencoded'):
            form = parse_qs(request.get_data(parse_form_data=False))
        else:
            form = {}
        request._covador_form = form
        return form
```

Example 40

5 vo

Project: *SempoBlockchain* Author: *teamsempo* File: *__init__.py* GNU General Public License v3.0

```
def register_extensions(app):
    db.init_app(app)

    basic_auth.init_app(app)

    @app.before_request
    def enable_form_raw_cache():
        # Workaround to allow unparsed request body to be be read from cache
        # This is required to validate a signature on webhooks
        # This MUST go before Sentry integration as sentry triggers form parsing
        if not config.IS_TEST and (
                request.path.startswith('/api/slack/') or request.path.startswith(
            if request.content_length > 1024 * 1024:  # 1mb
                # Payload too large
                return make_response(jsonify({'message': 'Payload too large'})), 4
            request.get_data(parse_form_data=False, cache=True)

    if not config.IS_TEST:
        sentry.init_app(app, dsn=app.config['SENTRY_SERVER_DSN'])
    # limiter.init_app(app)

    CORS(app, resources={r"/api/*": {"origins": "*"}})

    celery_app.conf.update(app.config)

    print('celery joined on {} at {}'.format(
        app.config['REDIS_URL'], datetime.utcnow()))
```

Example 41

Project: *sentry-python* Author: *getsentry* File: *test_flask.py* BSD 2-Clause "Simplified" License 5 vo

```
def test_flask_too_large_raw_request(sentry_init, input_char, capture_events, app)
    sentry_init(integrations=[flask_sentry.FlaskIntegration()], request_bodies="sm

    data = input_char * 2000

    @app.route("/", methods=["POST"])
    def index():
        assert not request.form
        if isinstance(data, bytes):
            assert request.get_data() == data
        else:
            assert request.get_data() == data.encode("ascii")
        assert not request.get_json()
        capture_message("hi")
        return "ok"

    events = capture_events()

    client = app.test_client()
    response = client.post("/", data=data)
    assert response.status_code == 200

    event, = events
    assert event["_meta"]["request"]["data"] == {
        "": {"len": 2000, "rem": [["!config", "x", 0, 2000]]}
    }
    assert not event["request"]["data"]
```

Example 42

Project: *minemeld-core* Author: *PaloAltoNetworks* File: *statusapi.py* Apache License 2.0 5 vo

```
def sns_wish():
    request.get_data()
    message = request.data
    success = SNS_OBJ.make_wish(message)
    if success:
        return jsonify(result='ok')
    return jsonify(error={'messsage': 'Error sending the message'}), 400
```

Example 43

Project: *howtoacceptcrypto-integrations* Author: *r4victor* File: *api.py* MIT License 5 vo

```
def handle_callback():
    provided_signature = request.headers.get('X-CC-Webhook-Signature')
    expecetd_signarure = hmac.digest(WEBHOOK_SECRET.encode(), request.get_data(),
    if provided_signature != expecetd_signarure:
        abort(401)

    event = request.json['event']
    db.update_invoice(
        event['data']['id'],
        status=event['type'].split(':')[1]
    )
    return 'Thank you, Coinbase Commerce, for the free of charge service!'
```

Example 44

Project: *howtoacceptcrypto-integrations* Author: *r4victor* File: *api.py* MIT License 5 vo

```
def handle_callback():
    provided_signature = request.headers.get('Hmac')
    expecetd_signarure = hmac.digest(IPN_SECRET.encode(), request.get_data(), 'sh
    if provided_signature != expecetd_signarure:
        abort(401)

    data = request.form
    db.update_invoice(
        data['txn_id'],
        status=data['status'],
        status_text=data['status_text']
    )
    return 'Thank you, CoinPayments, for secure API!'
```

Example 45

Project: *bot-line-indonesian-summarizer* Author: *ec2ainun* File: *app.py* MIT License 5 vo

```
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']

    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)

    # handle webhook body
```

```
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)

    return 'OK'
```

Example 46

Project: *facebook-chatbot-python* Author: *hult* File: *server.py* MIT License 5 vo

```
def webhook():
    payload = request.get_data()
    for sender, message in messenger.messaging_events(payload):
        print "Incoming from %s: %s" % (sender, message)

        response = bot.respond_to(message)

        print "Outgoing to %s: %s" % (sender, response)
        messenger.send_message(FACEBOOK_TOKEN, sender, response)

    return "ok"
```

Example 47

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License 5 vo

```
def submit_lifecycle_ignition(request_raw_query):
    """
    Lifecycle Ignition
    ---
    tags:
      - lifecycle
    responses:
      200:
        description: A JSON of the ignition status
    """
    try:
        machine_ignition = json.loads(request.get_data())
    except ValueError:
        app.logger.error("%s have incorrect content" % request.path)
        return jsonify({"message": "FlaskValueError"}), 406
    req = requests.get("%s/ignition?%s" % (EC.matchbox_uri, request_raw_query))
    try:
        matchbox_ignition = json.loads(req.content)
        req.close()
    except ValueError:
        app.logger.error("%s have incorrect matchbox return" % request.path)
        return jsonify({"message": "MatchboxValueError"}), 406

    @smartdb.cockroach_transaction
    def op(caller=request.url_rule):
        with SMART.new_session() as session:
            try:
                inject = crud.InjectLifecycle(session, request_raw_query=request_r
                if json.dumps(machine_ignition, sort_keys=True) == json.dumps(matc
                    inject.refresh_lifecycle_ignition(True)
                    return jsonify({"message": "Up-to-date"}), 200
                else:
                    inject.refresh_lifecycle_ignition(False)
                    return jsonify({"message": "Outdated"}), 210
```

```
            except AttributeError:
                return jsonify({"message": "Unknown"}), 406

    return op(caller=request.url_rule)
```

Example 48

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License 5 vo

```
def record_discovery_data():
    """
    Discovery
    Report the current facts of a machine
    ---
    tags:
      - discovery
    responses:
      200:
        description: Number of machines and if the machine is new
        schema:
            type: dict
    """
    app.logger.info("%s %s" % (request.method, request.url))
    err = jsonify({u'boot-info': {}, u'lldp': {}, u'interfaces': [], u"disks": []}
    try:
        discovery_data = json.loads(request.get_data())
    except (KeyError, TypeError, ValueError):
        logger.error("fail to parse discovery data: %s" % request.get_data())
        return err

    try:
        new = repositories.discovery.upsert(discovery_data)
        repositories.machine_state.update(discovery_data["boot-info"]["mac"], Mach
        CACHE.delete(request.path)
        return jsonify({"new-discovery": new}), 200
    except TypeError as e:
        logger.error("fail to store discovery data: %s -> %s" % (request.get_data
        return err
```

Example 49

Project: *enjoliver* Author: *JulienBalestra* File: *api.py* MIT License 5 vo

```
def scheduler_post():
    """
    Scheduler
    Affect a schedule to a machine
    ---
    tags:
      - scheduler
    responses:
      406:
        description: Incorrect body content
        schema:
            type: dict
      200:
        description: The body sent
        schema:
            type: dict
    """
    try:
```

```
        req = json.loads(request.get_data())
    except ValueError:
        return jsonify(
            {
                u"roles": model.ScheduleRoles.roles,
                u'selector': {
                    u"mac": ""
                }
            }), 406

    repositories.machine_schedule.create_schedule(req)
    CACHE.delete(request.path)
    return jsonify(req)
```

Example 50

Project: *exchange-simulator* Author: *KyberNetwork* File: *fake_dev_chain_wrapper.py* MIT License 5 vo

```
def index():
    global use_delay
    timestamp = int(time.time())
    check_pending_txs(timestamp)

    req = request.get_data().decode()
    print (str(req))
    json_req = json.loads(req)
    output_is_array = False
    if (len(json_req) == 1):
        json_req = json_req[0]
        output_is_array = True
        print(str(json_req))
    method_name = json_req["method"]
    params = json_req["params"]
    rpc_version = json_req["jsonrpc"]
    id = json_req["id"]

    # some commands are not supported in delay mode
    if((method_name == "eth_sendTransaction" or
            method_name == "eth_getTransactionByHash") and use_delay):
        respone = {"id": id, "jsonrpc": rpc_version,
                   "result": "unsuppoted command in delay mode"}
    elif(method_name == "eth_sendRawTransaction" and use_delay):
        response = handle_send_raw_tx(
            method_name, params, rpc_version, id, timestamp)
    elif(method_name == "enableDelay"):
        use_delay = True
        response = {"id": id, "jsonrpc": rpc_version, "result": "Ok"}
    else:
        response = blockchain_json_call(method_name, params, rpc_version, id)
    if(output_is_array):
        response = [response]
    print(str(response))
    return json.dumps(response)
```