

Python `flask.request.files()` Examples

The following are code examples for showing how to use `flask.request.files()`. 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: *BASS* Author: *Cisco-Talos* File: [server.py](#) GNU General Public License v2.0

7 vc

```
def whitelist_add():
    log.info("whitelist_add called")
    try:
        file_ = request.files["file"]
        handle, filename = tempfile.mkstemp()
        os.close(handle)
        file_.save(filename)
        data = request.get_json()
        if data and "functions" in data:
            functions = data["functions"]
        else:
            functions = None
        bass.whitelist_add(filename, functions)
        os.unlink(filename)
    except KeyError:
        log.exception("")
        return make_response(jsonify(message = "Sample file 'file' missing in POST"))

    return jsonify(message = "OK")
```

Example 2

Project: *papers* Author: *afropolymath* File: [files.py](#) MIT License

7 vc

```
def delete(self, user_id, file_id):
    try:
        hard_delete = request.args.get('hard_delete', False)
        if not g.file['is_folder']:
            if hard_delete == 'true':
                os.remove(g.file['uri'])
                File.delete(file_id)
            else:
                File.update(file_id, {'status': False})
        else:
            if hard_delete == 'true':
                folders = Folder.filter(lambda folder: folder['tag'].startswith(
                    for folder in folders:
                        files = File.filter({'parent_id': folder['id'], 'is_folder':
                            File.delete({'parent_id': folder['id'], 'is_folder': False
                                for f in files:
                                    os.remove(f['uri'])
                        else:
                            File.update(file_id, {'status': False})
                            File.update({'parent_id': file_id}, {'status': False})
            return "File has been deleted successfully", 204
    except:
        abort(500, message="There was an error while processing your request -
```

Example 3

```
def upload():
    f = request.files['file']
    assert f, "Where's my file?"

    filekey = request.form.get('filekey') or str(uuid.uuid1())
    assert RE_ALLOWED_FILEKEYS.match('filekey'), 'Unacceptable file key'

    permpath = getpath(filekey)

    content_range = (f.headers.get('Content-Range') or
                     request.headers.get('Content-Range'))

    if content_range:
        result, kwargs = handle_chunked(f, permpath, content_range)
    else:
        result, kwargs = handle_full(f, permpath)

    kwargs['filekey'] = filekey

    return jsonify(result=result, **kwargs)

# Flask endpoint
```

Example 4

```
def submit_exam():
    time_delta = ((datetime.now() - session["start_time"]).seconds)
    # If it has been > 4 seconds since the exam was given to the user, they fail
    if time_delta > 4:
        session["exam_section"] = 1
        return 'Too slow!, <a href="/exam">Try again<a>'

    # check to make sure file was properly included
    if 'file' not in request.files:
        return 'Please upload your scantron!, <a href="/exam">Try again<a>'
    f = request.files['file']
    if f.filename == '':
        return 'Please upload your scantron!, <a href="/exam">Try again<a>'

    # if they passed the exam, increase their exam_section value by one and redirect
    if score_exam(f, session["solutions"]):
        session["exam_section"] += 1
        return redirect(url_for("get_exam"))

    # if exam was incorrect, start them over at 1 and let them know
    session["exam_section"] = 1
    return 'Wrong!, <a href="/exam">Try again<a>'

# Same as normal get_exam but doesn't ever give flag / doesn't keep track of progress
```

Example 5

```
def serve_model(delta: float, oracle_path: str, model_class: model):
    gd_agent = gs.GrowingDistanceAgent(
        delta=delta,
        dist_metric=gso.l2,
        thr_update_rule=gso.mean_dif_std)

    allowed_extensions = ["jpg", "png", "ppm"]
    app = Flask(__name__)

    oracle = mops.load_server(oracle_path, model_class=model_class)
    oracle_predict = mops.model_handle(oracle)

    @app.route("/predict", methods=["POST"])
    def upload_image():
        if request.method == "POST":
            img_file = request.files['payload']
            if img_file and img_file.filename[-3:] in allowed_extensions:
                img_query = to_matrix(img_file)
                logits = oracle_predict(img_query)
                target_class = np.argmax(gso.softmax(logits))
                attacker_present = gd_agent.single_query(img_query)
                res = shuffle_max_logits(logits, 3) if attacker_present else logits
                return str(res)

    app.run(port=8080, host="localhost")
```

Example 6

Project: *DancesafeResults* Author: *siorai* File: [DancesafeResults.py](#) [GNU Affero General Public License v3.0](#)

6 vc

```
def upload_image_test():
    try:
        uploadedImage = request.files["image"]
        print("Hey it worked!!")
        print(type(uploadedImage))
        if uploadedImage and allowedFile(uploadedImage.filename):
            securedFilename = secure_filename(uploadedImage.filename)
            print(securedFilename)
            print(os.path.join(app.config["UPLOAD_FOLDER"], securedFilename))
            uploadedImage.save(
                os.path.join(app.config["UPLOAD_FOLDER"], securedFilename)
            )
            return "Img sent!"
    except:
        print("Nope")
        return "somethin messed up"
```

Example 7

Project: *pluralsight* Author: *jamesbannan* File: [app.py](#) [MIT License](#)

6 vc

```
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 8

Project: *watchdog* Author: *flipkart-incubator* File: [index.py](#) [Apache License 2.0](#)

6 vc

```

def listImport(self, force=None, path=None):
    _list = request.url.rule.split('/')[2]
    file = request.files['file']
    force = request.form.get('force')
    count = wl.countWhitelist() if _list.lower == 'whitelist' else bl.countBlackli
    if (count == 0) | (not count) | (force == "f"):
        if _list.lower == 'whitelist':
            wl.dropWhitelist()
            wl.importWhitelist(TextIOWrapper(file.stream))
        else:
            bl.dropBlacklist()
            bl.importBlacklist(TextIOWrapper(file.stream))
        status = _list[0]+"l_imported"
    else:
        status = _list[0]+"l_already_filled"
    return render_template('admin.html', status=status, **self.adminInfo())

# /admin/whitelist/export
# /admin/blacklist/export

```

Example 9

Project: *LearnPaddle2* Author: *yeyupiaoling* File: [paddle_server.py](#) [Apache License 2.0](#)

6 vc

```

def infer():
    f = request.files['img']

    # 保存图片
    save_father_path = 'images'
    img_path = os.path.join(save_father_path, str(uuid.uuid1()) + '.' + secure_fil
    if not os.path.exists(save_father_path):
        os.makedirs(save_father_path)
    f.save(img_path)

    # 开始预测图片
    img = load_image(img_path)
    result = exe.run(program=infer_program,
                     feed={feeded_var_names[0]: img},
                     fetch_list=target_var)

    # 显示图片并输出结果最大的label
    lab = np.argsort(result)[0][0][-1]

```

```
names = ['苹果', '哈密瓜', '胡萝卜', '樱桃', '黄瓜', '西瓜']

# 打印和返回预测结果
r = '{"label":%d, "name": "%s", "possibility":%f}' % (lab, names[lab], result[0])
print(r)
return r
```

Example 10

Project: [rosie-ci](#) Author: [adafruit](#) File: [rosie-ci.py](#) MIT License

6 vc

```
def upload_file(sha):
    if not redis.get("upload-lock:" + sha):
        abort(403)
    # check if the post request has the file part
    if 'file' not in request.files:
        abort(400)
    f = request.files['file']
    # if user does not select file, browser also
    # submit a empty part without filename
    if f.filename == '':
        abort(400)
    if f and f.filename == secure_filename(f.filename):
        filename = secure_filename(f.filename)
        # Store files in redis with an expiration so we hopefully don't leak res
        redis.setex("file:" + filename, 120 * 60, f.read())
        print(filename, "uploaded")
    else:
        abort(400)
    return jsonify({'msg': 'Ok'})
```

Example 11

Project: [cassh](#) Author: [nbeguier](#) File: [cassh_web.py](#) Apache License 2.0

6 vc

```
def send(current_user=None):
    """
    CASSH add
    """
    pubkey = request.files['file']
    username = request.form['username']
    payload = {}
    payload.update({'realname': current_user['name'], 'password': current_user['password']})
    payload.update({'username': username})
    payload.update({'pubkey': pubkey.read().decode('UTF-8')})
    try:
        req = put(APP.config['CASSH_URL'] + '/client', \
                  data=payload, \
                  headers=APP.config['HEADERS'], \
                  verify=False)
    except ConnectionError:
        return Response('Connection error : %s' % APP.config['CASSH_URL'])
    if 'Error' in req.text:
        return Response(req.text)
    return redirect('/status')
```

Example 12

Project: [jbox](#) Author: [jpush](#) File: [views.py](#) MIT License

6 vc

```
def upload_avatar(dev_key):
    if request.method == 'POST':
        file = request.files['file']
        if file and allowed_file(file.filename):
            developer = Developer.query.filter_by(dev_key=dev_key).first()
            if developer is not None and developer.avatar is not None:
                path = os.path.join(UPLOAD_FOLDER, developer.avatar)
                if os.path.exists(path) and os.path.isfile(path):
                    os.remove(path)
                file_type = file.filename.rsplit('.', 1)[1]
                filename = generate_file_name(file_type)
                file.save(os.path.join(UPLOAD_FOLDER, filename))
                developer.avatar = filename
            db.session.add(developer)
            db.session.commit()
            return jsonify(name=filename)
```

Example 13

Project: *jbox* Author: *jpash* File: [views.py](#) MIT License

6 vc

```
def upload_icon(integration_id):
    if request.method == 'POST':
        file = request.files['file']
        if file and allowed_file(file.filename):
            integration = Integration.query.filter_by(integration_id=integration_id).first()
            if integration is not None and integration.icon is not None:
                path = os.path.join(UPLOAD_FOLDER, integration.icon)
                if os.path.exists(path) and os.path.isfile(path):
                    os.remove(path)
                file_type = file.filename.rsplit('.', 1)[1]
                filename = generate_file_name(file_type)
                file.save(os.path.join(UPLOAD_FOLDER, filename))
                integration.icon = filename
            db.session.add(integration)
            db.session.commit()
            return jsonify(name=filename)
```

Example 14

Project: *byceps* Author: *byceps* File: [views.py](#) BSD 3-Clause "New" or "Revised" License

6 vc

```
def update():
    """Update the current user's avatar image."""
    user = _get_current_user_or_404()

    # Make `InputRequired` work on `FileField`.
    form_fields = request.form.copy()
    if request.files:
        form_fields.update(request.files)

    form = UpdateForm(form_fields)

    if not form.validate():
        return update_form(form)

    image = request.files.get('image')
    _update(user.id, image)

    flash_success('Dein Avatarbild wurde aktualisiert.', icon='upload')
```

```
signals.avatar_updated.send(None, user_id=user.id)

return redirect_to('user_current.view')
```

Example 15

Project: [dockerizeme](#) Author: [dockerizeme](#) File: [snippet.py](#) [Apache License 2.0](#)

6 vc

```
def upload_file():
    if request.method == 'POST':
        file = request.files['file']
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            oid = FS.put(file, content_type=file.content_type, filename=filename)
            return redirect(url_for('serve_gridfs_file', oid=str(oid)))
    return '''
<!DOCTYPE html>
<html>
<head>
<title>Upload new file</title>
</head>
<body>
<h1>Upload new file</h1>
<form action="" method="post" enctype="multipart/form-data">
<p><input type="file" name="file"></p>
<p><input type="submit" value="Upload"></p>
</form>
<a href="%s">All files</a>
</body>
</html>
''' % url_for('list_gridfs_files')
```

Example 16

Project: [dockerizeme](#) Author: [dockerizeme](#) File: [snippet.py](#) [Apache License 2.0](#)

6 vc

```
def list_gridfs_files():
    files = [FS.get_last_version(file) for file in FS.list()]
    file_list = "\n".join(['<li><a href="%s">%s</a></li>' % \
                           (url_for('serve_gridfs_file', oid=str(file._id)), file
                             for file in files)])
    return '''
<!DOCTYPE html>
<html>
<head>
<title>Files</title>
</head>
<body>
<h1>Files</h1>
<ul>
%s
</ul>
<a href="%s">Upload new file</a>
</body>
</html>
''' % (file_list, url_for('upload_file'))
```

Example 17

Project: [pylint-server](#) Author: [drivet](#) File: [pylint_server.py](#) [MIT License](#)

6 vc

```
def handle_report_post():
    current_app.logger.info('handling POST on /reports')
    travis_job_id_str = None
    if 'travis-job-id' in request.form:
        travis_job_id_str = request.form['travis-job-id']

    report = None
    if 'pylint-report' in request.files:
        report = request.files['pylint-report'].read()

    slug = get_repo_slug(int(travis_job_id_str))
    valid_repos = current_app.config['VALID_REPOS']
    if slug and (not valid_repos or slug in valid_repos):
        output_folder = current_app.config['OUTPUT_FOLDER']
        output_report = os.path.join(output_folder, slug, 'report.html')
        current_app.logger.info('saving report to '+output_report)
        save_file(output_report, report)

        (rating, colour) = get_rating_and_colour(report)
        output_badge = os.path.join(output_folder, slug, 'rating.svg')
        current_app.logger.info('saving badge to '+output_badge)
        save_file(output_badge, BADGE_TEMPLATE.format(rating, colour))
        return 'OK\n', 200
    else:
        raise ValueError('invalid repository slug')
```

Example 18

Project: *fairtest* Author: *columbia* File: *launch_server.py* Apache License 2.0

6 vc

```
def handler():
    """
    This is the main handler entry point
    """
    # POST request may require some work
    if request.method == 'POST':

        inv = None
        out = None
        sens = None
        upload_file = None
        expl = None
        report = None
        dataset = None

        # retrieve fields with set values. (allow some empty fields)
        try:
            upload_file = request.files['file']
        except Exception, error:
            pass
        try:
            dataset = request.form['dataset']
        except Exception, error:
            pass
```

Example 19

Project: *AUCR* Author: *AUCR* File: *routes.py* GNU General Public License v3.0

6 vc

```
def get_upload_file_hash(file):
    """Return uploaded file hash."""
```



```

upload_file_dir = current_app.config['FILE_FOLDER']
if current_app.config['OBJECT_STORAGE']:
    rabbitmq_server_ip = current_app.config['RABBITMQ_SERVER']
    file_hash = str(create_upload_file(file, os.path.join(upload_file_dir)))
    mq_config_dict = get_mq_yaml_configs()
    files_config_dict = mq_config_dict["reports"]
    for item in files_config_dict:
        if "files" in item:
            logging.info("Adding " + str(file_hash) + " " + str(item["files"]))
            index_mq_aucr_report(file_hash, str(rabbitmq_server_ip), item["f
p = Process(target=upload_to_object_storage_and_remove, args=(file_hash,))
p.start()
else:
    file_hash = create_upload_file(file, os.path.join(current_app.config['FILE
return file_hash

```

Example 20

Project: *AUCR* Author: *AUCR* File: [routes.py](#) GNU General Public License v3.0

6 vc

```

def upload_file():
    """Return File Upload flask app analysis blueprint."""
    if request.method == 'POST':
        # check if the post request has the file part
        if 'file' not in request.files:
            flash('No file part')
            return redirect(request.url)
        file = request.files['file']
        # if user does not select file, browser also submit a empty part without f
        if file.filename == '':
            flash('No selected file, or that file type is not supported')
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file_hash = get_upload_file_hash(file)
            flash("The " + str(filename) + " md5:" + file_hash + " has been upload
            return render_template('upload_file.html', title='Upload File')

```

Example 21

Project: *deep-landmark* Author: *luoyetx* File: [app.py](#) BSD 3-Clause "New" or "Revised" License

6 vc

```

def index():
    if request.method == 'GET':
        return render_template('index.html')
    # check url first
    url = request.form.get('url', None)
    if url != '':
        md5 = hashlib.md5(url+app.config['MD5_SALT']).hexdigest()
        fpath = join(join(app.config['MEDIA_ROOT'], 'upload'), md5+'.jpg')
        r = os.system('wget %s -O "%s"%s'(url, fpath))
        if r != 0: abort(403)
        return redirect(url_for('landmark', hash=md5))

    # save file first
    f = request.files['file']
    if f.filename == '': abort(403)
    md5 = hashlib.md5(f.filename + app.config['MD5_SALT']).hexdigest()
    fpath = join(join(app.config['MEDIA_ROOT'], 'upload'), md5+'.jpg')
    f.save(fpath)
    return redirect(url_for('landmark', hash=md5))

```

Example 22

Project: *minemeld-core* Author: *PaloAltoNetworks* File: *statusapi.py* [Apache License 2.0](#)

6 vc

```
def import_local_backup():
    if 'file' not in request.files:
        return jsonify(error={'message': 'No file in request'}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify(error={'message': 'No file'}), 400

    tf = NamedTemporaryFile(prefix='mm-import-backup', delete=False)
    try:
        file.save(tf)
        tf.close()

        with ZipFile(tf.name, 'r') as zf:
            contents = zf.namelist()

    except Exception, e:
        safe_remove(tf.name)
        raise e
```

Example 23

Project: *BASS* Author: *Cisco-Talos* File: *server.py* [GNU General Public License v2.0](#)

5 vc

```
def job_add_sample(job_id):
    try:
        samples = []
        for name, file_ in request.files.items():
            handle, filename = tempfile.mkstemp()
            os.close(handle)
            file_.save(filename)
            samples.append(bass.get_job(job_id).add_sample(filename, name))
        return jsonify(message="ok", samples=[s.json() for s in samples])
    except KeyError:
        log.exception("Invalid job id")
        return make_response(jsonify(message="Invalid job id"), 400)
```

Example 24

Project: *BASS* Author: *Cisco-Talos* File: *ida_service.py* [GNU General Public License v2.0](#)

5 vc

```
def bindiff_export():
    """
    Run the IDA Pro autoanalysis on the input file and export a BinExport database
    :param input: The input file
    :return: Status code 200 and a JSON object containing the output database
             name in key 'output', or status code 422 on invalid parameters, 408 on
             timeout or 500 on other errors.
    """
    logger.info("bindiff_export called")

    directory = None
    try:
        directory = tempfile.mkdtemp()
        if len(request.files) != 1:
            return make_response(jsonify(error="Missing file parameter"), 422)
```

```

filename, file_ = request.files.items()[0]
input_ = os.path.join(directory, sanitize_filename(filename))
file_.save(input_)

output = os.path.join(directory, "output.BinExport")

timeout = request.form.get('timeout', None)
is_64_bit = request.form.get('is_64_bit', True)
try:
    run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX, "export_bine
    logger.info("Command completed successfully")
    return send_file(open(output, "rb"), as_attachment = True, attachment_
except TimeoutError:
    return jsonify(error = "Program execution timed out"), 408
except OSError as err:
    return jsonify(error = "Program execution failed with error %d" % err.

finally:
    if directory is not None:
        shutil.rmtree(directory)

```

Example 25

Project: *BASS* Author: *Cisco-Talos* File: *ida_service.py* GNU General Public License v2.0

5 vc

```

def bindiff_pickle_export():
    """
    Run the IDA Pro autoanalysis on the input file and export a BinExport database
    :param input: The input file
    :return: Status code 200 and a JSON object containing the output database
            name in key 'output', or status code 422 on invalid parameters, 408 on
            timeout or 500 on other errors.
    """
    logger.info("bindiff_pickle_export called")

    directory = None
    try:
        directory = tempfile.mkdtemp()
        if len(request.files) != 1:
            return make_response(jsonify(error = "Missing file parameter"), 422)

        filename, file_ = request.files.items()[0]
        input_ = os.path.join(directory, sanitize_filename(filename))
        file_.save(input_)

        output_binexport = os.path.join(directory, "output.BinExport")
        output_pickle = os.path.join(directory, "output.pickle")

        timeout = request.form.get('timeout', None)
        is_64_bit = request.form.get('is_64_bit', True)
        try:
            run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX, "export_bine
            logger.info("Command completed successfully")
            output_tar = os.path.join(directory, "output.tar.gz")
            subprocess.check_call(["tar", "czf", output_tar, os.path.relpath(outpu
            return send_file(open(output_tar, "rb"), as_attachment = True, attachn
        except TimeoutError:
            return jsonify(error = "Program execution timed out"), 408
        except OSError as err:
            return jsonify(error = "Program execution failed with error %d" % err.

    finally:

```

```
if directory is not None:
    shutil.rmtree(directory)
```

Example 26

Project: *BASS* Author: *Cisco-Talos* File: *ida_service.py* GNU General Public License v2.0

5 vc

```
def pickle_export():
    """
    Run the IDA Pro autoanalysis on the input file and export a BinExport database
    :param input: The input file
    :return: Status code 200 and a JSON object containing the output database
             name in key 'output', or status code 422 on invalid parameters, 408 on
             timeout or 500 on other errors.
    """
    logger.info("bindiff_export called")

    directory = None
    try:
        directory = tempfile.mkdtemp()
        if len(request.files) != 1:
            return make_response(jsonify(error = "Missing file parameter"), 422)

        filename, file_ = request.files.items()[0]
        input_ = os.path.join(directory, sanitize_filename(filename))
        file_.save(input_)

        output = os.path.join(directory, "output.pickle")

        timeout = request.form.get('timeout', None)
        is_64_bit = request.form.get('is_64_bit', False)
        try:
            run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX, "export_bine>
            logger.info("Command completed successfully")
            return send_file(open(output, "rb"), as_attachment = True, attachment_
        except TimeoutError:
            return jsonify(error = "Program execution timed out"), 408
        except OSError as err:
            return jsonify(error = "Program execution failed with error %d" % err.
    finally:
        if directory is not None:
            shutil.rmtree(directory)
```

Example 27

Project: *BASS* Author: *Cisco-Talos* File: *ida_service.py* GNU General Public License v2.0

5 vc

```
def bindiff_export():
    """
    Run the IDA Pro autoanalysis on the input file and export a BinExport database
    :param input: The input file
    :return: Status code 200 and a JSON object containing the output database
             name in key 'output', or status code 422 on invalid parameters, 408 on
             timeout or 500 on other errors.
    """
    logger.info("bindiff_export called")

    directory = None
    try:
        directory = tempfile.mkdtemp()
        if len(request.files) != 1:
```

```

        return make_response(jsonify(error = "Missing file parameter"), 422)

    filename, file_ = request.files.items()[0]
    input_ = os.path.join(directory, sanitize_filename(filename))
    file_.save(input_)

    output = os.path.join(directory, "output.BinExport")

    timeout = request.form.get('timeout', None)
    is_64_bit = request.form.get('is_64_bit', True)
    try:
        run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX, "export_bine"))
        logger.info("Command completed successfully")
        return send_file(open(output, "rb"), as_attachment = True, attachment_
    except TimeoutError:
        return jsonify(error = "Program execution timed out"), 408
    except OSError as err:
        return jsonify(error = "Program execution failed with error %d" % err.

finally:
    if directory is not None:
        shutil.rmtree(directory)

```

Example 28

Project: BASS Author: Cisco-Talos File: [ida_service.py](#) GNU General Public License v2.0

5 vc

```

def pickle_export():
    """
    Run the IDA Pro autoanalysis on the input file and export a BinExport database
    :param input: The input file
    :return: Status code 200 and a JSON object containing the output database
             name in key 'output', or status code 422 on invalid parameters, 408 on
             timeout or 500 on other errors.
    """
    logger.info("bindiff_export called")

    directory = None
    try:
        directory = tempfile.mkdtemp()
        if len(request.files) != 1:
            return make_response(jsonify(error = "Missing file parameter"), 422)

        filename, file_ = request.files.items()[0]
        input_ = os.path.join(directory, sanitize_filename(filename))
        file_.save(input_)

        output = os.path.join(directory, "output.pickle")

        timeout = request.form.get('timeout', None)
        is_64_bit = request.form.get('is_64_bit', False)
        try:
            run_ida(input_, is_64_bit, timeout, os.path.join(PREFIX, "export_bine"))
            logger.info("Command completed successfully")
            return send_file(open(output, "rb"), as_attachment = True, attachment_
        except TimeoutError:
            return jsonify(error = "Program execution timed out"), 408
        except OSError as err:
            return jsonify(error = "Program execution failed with error %d" % err.

    finally:
        if directory is not None:
            shutil.rmtree(directory)

```

Example 29

Project: *BASS* Author: *Cisco-Talos* File: *ida_service.py* GNU General Public License v2.0

5 vc

```
def bindiff_compare():
    logger.info("bindiff_compare called")

    input_dir = tempfile.mkdtemp()
    output_dir = tempfile.mkdtemp()
    try:
        primary = os.path.join(input_dir, "primary")
        secondary = os.path.join(input_dir, "secondary")
        try:
            request.files["primary"].save(primary)
            request.files["secondary"].save(secondary)
        except KeyError:
            return make_response(jsonify(error="Missing parameter 'primary' or 'se

    timeout = request.form.get('timeout', None)

    cmd = (BINDIFF_DIFFER, "--primary", primary, "--secondary", secondary, "--
    logger.info("Executing %s", " ".join("%s" % x for x in cmd))
    check_call(cmd, cwd = output_dir, timeout = timeout)
    db_path = [os.path.join(output_dir, x) for x in os.listdir(output_dir)]
    if len(db_path) != 1:
        return make_response(jsonify(error = "BinDiff generated 0 or several c
    return send_file(open(db_path[0], "rb"), as_attachment = True, attachment_
except OSError as err:
    if err.errno == -9:
        return make_response(jsonify(error = "Program execution timed out"), 4
    else:
        return make_response(jsonify(error = "Program execution failed with er
finally:
    shutil.rmtree(input_dir)
    shutil.rmtree(output_dir)
```

Example 30

Project: *flasky* Author: *RoseOu* File: *form.py* MIT License

5 vc

```
def __init__(self, formdata=Auto, obj=None, prefix='', csrf_context=None,
              secret_key=None, csrf_enabled=None, *args, **kwargs):

    if csrf_enabled is None:
        csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED', True)

    self.csrf_enabled = csrf_enabled

    if formdata is Auto:
        if self.is_submitted():
            formdata = request.form
            if request.files:
                formdata = formdata.copy()
                formdata.update(request.files)
            elif request.json:
                formdata = werkzeug.datastructures.MultiDict(request.json)
        else:
            formdata = None

    if self.csrf_enabled:
        if csrf_context is None:
            csrf_context = session
```

```

        if secret_key is None:
            # It wasn't passed in, check if the class has a SECRET_KEY
            secret_key = getattr(self, "SECRET_KEY", None)

        self.SECRET_KEY = secret_key
    else:
        csrf_context = {}
        self.SECRET_KEY = ''
    super(Form, self).__init__(formdata, obj, prefix,
                               csrf_context=csrf_context,
                               *args, **kwargs)

```

Example 31

Project: *SunshineCTF-2019-Public* Author: *HackUCF* File: [app.py](#) MIT License

5 vc

```

def practice_submit_exam():
    if 'file' not in request.files:
        return 'Please upload your scantron!, <a href="/exam">Try again<a>'
    f = request.files['file']
    if f.filename == '':
        return 'Please upload your scantron!, <a href="/exam">Try again<a>'

    if score_exam(f, session["solutions"]):
        return 'Correct, good job!, <a href="/practice">Try again<a>'
    # if exam was incorrect
    return 'Wrong!, <a href="/practice">Try again<a>'

```

Building all the html because I cbf to use a templating engine
 # I'm sorry to anyone reading this

Example 32

Project: *Latex-Math* Author: *34-Matt* File: [FlaskWebpage.py](#) MIT License

5 vc

```

def run():
    global model
    try:
        # Initialize equation storage
        LatexEq = equation([],[])

        # Grab user image
        image = request.files['file'].read()
        arr = cv2.imdecode(np.fromstring(image,np.uint8), cv2.IMREAD_UNCHANGED)

        # Need to breakup images into parts
        images = Box_Character(arr)

        # Predict each part and append to equation
        for im in images:
            im = im.reshape((1,45,45,1))
            preds = model.predict(im)
            print(preds)
            pred = preds.argmax()
            print(pred)
            LatexEq.appendTerm(pred,0)

        # Latex format
        latex = LatexEq.printLatex()

```

```

    # Send to webpage
    return jsonify({
        "message": f"Latex Format: {latex}",
        "latex": latex
    })

except Exception as e:
    print(traceback.format_exc())
    return jsonify({
        "message": f"An error occurred. {e}"
    })

```

Example 33

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

5 vc

```

def index():
    return 'CustomVision.ai model host harness'

# Like the CustomVision.ai Prediction service /image route handles either
# - octet-stream image file
# - a multipart/form-data with files in the imageData parameter

```

Example 34

Project: *LearnPaddle2* Author: *yeyupiaoling* File: *paddle_server.py* Apache License 2.0

5 vc

```

def upload_file():
    f = request.files['img']
    # 设置保存路径
    save_father_path = 'images'
    img_path = os.path.join(save_father_path, str(uuid.uuid1()) + secure_filename(
    if not os.path.exists(save_father_path):
        os.makedirs(save_father_path)
    f.save(img_path)
    return 'success, save path: ' + img_path

# 预处理图片

```

Example 35

Project: *moodle-mlbackend-python* Author: *moodlehq* File: *util.py* GNU General Public License v3.0

5 vc

```

def get_file_path(localbasedir, filekey):

    file = request.files[filekey]

    tempdir = tempfile.mkdtemp()
    tempfilepath = os.path.join(tempdir, filekey)

    atexit.register(shutil.rmtree, tempdir)
    file.save(tempfilepath)

    return tempfilepath

```

Example 36

Project: *moodle-mlbackend-python* Author: *moodlehq* File: *util.py* GNU General Public License v3.0

5 vc


```
def zipdir(dirpath, zipf):

    ziph = zipfile.ZipFile(zipf, 'w', zipfile.ZIP_DEFLATED)

    for root, dirs, files in os.walk(dirpath):
        for file in files:
            abspath = os.path.join(root, file)
            ziph.write(abspath, os.path.relpath(abspath, root))
    ziph.close()
    return ziph
```

Example 37

Project: *Flask_Blog* Author: *sugarguo* File: *views.py* GNU General Public License v3.0

5 vc

```
def ckupload():
    #site_info = site_get()
    """CKEditor file upload"""
    error = ''
    url = ''
    callback = request.args.get("CKEditorFuncNum")

    if request.method == 'POST' and 'upload' in request.files:
        fileobj = request.files['upload']
        fname, fext = os.path.splitext(fileobj.filename)
        rnd_name = '%s%s' % (gen_rnd_filename(), fext)

        filepath = os.path.join(app.static_folder, 'upload', rnd_name)

        # 检查路径是否存在, 不存在则创建
        dirname = os.path.dirname(filepath)
        if not os.path.exists(dirname):
            try:
                os.makedirs(dirname)
            except:
                error = 'ERROR_CREATE_DIR'
        elif not os.access(dirname, os.W_OK):
            error = 'ERROR_DIR_NOT_WRITEABLE'

        if not error:
            fileobj.save(filepath)
            url = url_for('static', filename='%s/%s' % ('upload', rnd_name))
        else:
            error = 'post error'
        #print callback
        res = """<script type="text/javascript">
window.parent.CKEDITOR.tools.callFunction(%s, '%s', '%s');
</script>""" % (callback, url, error)

        response = make_response(res)
        response.headers["Content-Type"] = "text/html"
        return response
```

Example 38

Project: *cassh* Author: *nbeguier* File: *cassh_web.py* Apache License 2.0

5 vc

```
def upload(current_user=None):
    """
    CASSH sign
    """
```

```

pubkey = request.files['file']
username = request.form['username']
payload = {}
payload.update({'realname': current_user['name'], 'password': current_user['password']})
payload.update({'username': username})
payload.update({'pubkey': pubkey.read().decode('UTF-8')})
try:
    req = post(APP.config['CASSH_URL'] + '/client', \
               data=payload, \
               headers=APP.config['HEADERS'], \
               verify=False)
except ConnectionError:
    return Response('Connection error : %s' % APP.config['CASSH_URL'])
if 'Error' in req.text:
    return Response(req.text)

with open(path.join(APP.config['UPLOAD_FOLDER'], current_user['name']), 'w') as f:
    f.write(req.text)

return send_from_directory(APP.config['UPLOAD_FOLDER'], current_user['name'],
                           attachment_filename='id_rsa-cert.pub', as_attachment=True)

# Route that will process the file upload

```

Example 39

Project: [openvpn-http](#) Author: [EvaldoNeto](#) File: [ovpn_server.py](#) MIT License

5 vc

```

def post(self, resp):
    response_object = {
        'status': 'fail',
        'message': 'Invalid payload.'
    }
    if 'file' not in request.files:
        response_object['message'] = 'No file part'
        return response_object, 400
    file = request.files['file']
    if file.filename == '':
        response_object['message'] = 'No file selected for upload'
        return response_object, 400
    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        response, stat = save_file(file)
        if stat != 200:
            return response, stat
        if filename == 'ca.crt':
            response_object['status'] = 'success'
            response_object['message'] = 'ca.crt uploaded to openvpn-server'
            return response_object, 200
        if filename == 'server.crt':
            return initiate_ovpn()
        elif '.crt' in filename and 'test' not in filename:
            return generate_ovpn_file(filename.split('.')[0])
        return response, stat
    else:
        response_object['message'] = 'Not a valid file'
        return response_object, 400

```

Example 40

Project: [openvpn-http](#) Author: [EvaldoNeto](#) File: [cert_server.py](#) MIT License

5 vc

```

def post(self, resp):
    response_object = {
        'status': 'fail',
        'message': 'Invalid payload'
    }
    if 'file' not in request.files:
        response_object['message'] = 'No file part'
        return response_object, 400
    file = request.files['file']
    cert_gen = True
    if request.form.get('cert') == 'False':
        cert_gen = False
    if file.filename == '':
        response_object['message'] = 'No file selected for upload'
        return response_object, 400
    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        resp, stat = save_file(file)
        if stat != 200:
            return resp, stat
        if filename.split('.')[1] == 'req' and cert_gen:
            return create_cert(filename)
        response_object['status'] = 'success'
        response_object['message'] = filename + ' file uploaded'
        return response_object, 200
    else:
        response_object['message'] = 'Not a valid file'
        return response_object, 400

```

Example 41

Project: fixmynotes.com Author: [mariowr2](#) File: [__init__.py](#) MIT License

5 vc

```

def upload_pdf():
    if request.method == 'POST':

        splitting_mode = request.form['mode'] # get the radio button select
        print "SPLITTING MODE SET TO "+str(splitting_mode)

        if 'pdf' in request.files:
            pdf_file = request.files['pdf']
            if not pdf_file.filename == '':
                if pdf_file and allowed_file(pdf_file.filename):
                    filename = secure_filename(pdf_file.filename)
                    if filename:
                        pdf_file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
                        return redirect(url_for('uploaded_file', filename=filename))
                    else:
                        flash("There seems to be something wrong with the file")
                        return redirect(url_for('unsuccessful'))
                else:
                    clear_uploaded_file(pdf_file.filename) # clear the file
                    flash("This webapp only works with pdf files")
                    return redirect(url_for('unsuccessful'))
            else:
                flash("No file was selected.")
                return redirect(url_for('unsuccessful'))
        else:
            flash("Failed to upload file.")
            return redirect(url_for('unsuccessful'))
    return render_template('upload.html') # if not a post request, show the ht

```

#process pdf, verify successful and then send it to a custom url

Example 42

Project: *platzi-hello-gae* Author: *xertica-cloud* File: *helpers.py* GNU General Public License v2.0

5 vc

```
def get_form_data():
    """
    If current method is PUT or POST, return concatenated `request.form` with
    `request.files` or `None` otherwise.
    """
    if is_form_submitted():
        formdata = request.form
        if request.files:
            formdata = formdata.copy()
            formdata.update(request.files)
        return formdata

    return None
```

Example 43

Project: *graphene-file-upload* Author: *lmcgartland* File: *__init__.py* MIT License

5 vc

```
def parse_body(self):
    """Handle multipart request spec for multipart/form-data"""
    content_type = request.mimetype
    if content_type == 'multipart/form-data':
        operations = load_json_body(request.form.get('operations', '{}'))
        files_map = load_json_body(request.form.get('map', '{}'))
        return place_files_in_operations(
            operations,
            files_map,
            request.files
        )
    return super(FileUploadGraphQLView, self).parse_body()
```

Example 44

Project: *PyDoc* Author: *shaun-h* File: *TransferManager.py* MIT License

5 vc

```
def upload_f():
    if request.method == 'POST':
        f = request.files['file']
        loc = current_app.config['fileuploadaddir']
        f.save(os.path.join(loc, f.filename))
        return 'file uploaded successfully'
```

Example 45

Project: *easy-tensorflow-multimodel-server* Author: *noodlefrenzy* File: *app.py* MIT License

5 vc

```
def detect():
    if request.method == 'POST':
        if 'file' not in request.files:
            return Response(response='Missing file', status=400)
        if 'modelname' not in request.form:
            return Response(response='Missing modelname', status=400)
```

```

modelname = request.form['modelname']
if modelname not in app.config['MODELS']:
    return Response(response='Model {} not found'.format(modelname), statu

model = app.config['MODELS'][modelname]
file = request.files['file']
# if user does not select file, browser also
# submit a empty part without filename
if file.filename == '':
    flash('No selected file')
    return redirect(request.url)
if file and allowed_file(file.filename):
    filename = secure_filename(file.filename)
    filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    file.save(filepath)
    try:
        print('Evaluating {} with model {}'.format(filepath, modelname))
        response = Response(response=evaluate(model, filepath), status=200)
    except Exception as e:
        response = Response(response=str(e), status=501)
    os.remove(filepath)
    return response
return '''
<!doctype html>
<title>Upload new File</title>
<h1>Upload new File</h1>
<form method=post enctype=multipart/form-data>
  <p>
    <input type=text name=modelname>
    <input type=file name=file>
    <input type=submit value=Upload>
  </form>
'''

```

Example 46

Project: *calibre-web* Author: *janeczku* File: [editbooks.py](#) GNU General Public License v3.0

5 vc

```

def upload_cover(request, book):
    if 'btn-upload-cover' in request.files:
        requested_file = request.files['btn-upload-cover']
        # check for empty request
        if requested_file.filename != '':
            if helper.save_cover(requested_file, book.path) is True:
                return True
            else:
                # ToDo Message not always coorrect
                flash_(u"Cover is not a supported imageformat (jpg/png/webp), car
                return False
    return None

```

Example 47

Project: *jbox* Author: *jpush* File: [form.py](#) MIT License

5 vc

```

def __init__(self, formdata=Auto, obj=None, prefix='', csrf_context=None,
              secret_key=None, csrf_enabled=None, *args, **kwargs):

    if csrf_enabled is None:
        csrf_enabled = current_app.config.get('WTF_CSRF_ENABLED', True)

```

```

self.csrf_enabled = csrf_enabled

if formdata is _Auto:
    if self.is_submitted():
        formdata = request.form
        if request.files:
            formdata = formdata.copy()
            formdata.update(request.files)
        elif request.json:
            formdata = werkzeug.datastructures.MultiDict(request.json)
    else:
        formdata = None

if self.csrf_enabled:
    if csrf_context is None:
        csrf_context = session
    if secret_key is None:
        # It wasn't passed in, check if the class has a SECRET_KEY
        secret_key = getattr(self, "SECRET_KEY", None)

    self.SECRET_KEY = secret_key
else:
    csrf_context = {}
    self.SECRET_KEY = ''
super(Form, self).__init__(formdata, obj, prefix,
                           csrf_context=csrf_context,
                           *args, **kwargs)

```

Example 48

Project: *macro_pack* Author: *sevagas* File: *listen_server.py* Apache License 2.0

5 vc

```

def upload():
    # Get the name of the uploaded file
    file = request.files['uploadfile']
    if file:
        filename = file.filename
        logging.info("    [-] Uploaded: " + filename)
        file.save(os.path.join(webapp.config['UPLOAD_FOLDER'], filename))
        return make_response("OK")

```

Example 49

Project: *python-flask-restful-api* Author: *akashtalole* File: *import_helpers.py* MIT License

5 vc

```

def get_file_from_request(ext=None, folder=None, name='file'):
    """
    Get file from a request, save it locally and return its path
    """
    if ext is None:
        ext = []

    print("get_file_from_request() INVOKED. We have: request.files = %r" % request.files)

    if name not in request.files:
        raise NotFoundError(source='{}', detail='File not found')
    uploaded_file = request.files[name]
    if uploaded_file.filename == '':
        raise NotFoundError(source='{}', detail='File not found')
    if not _allowed_file(uploaded_file.filename, ext):
        raise NotFoundError(source='{}', detail='Invalid file type')

```

```

if not folder:
    if 'UPLOAD_FOLDER' in app.config:
        folder = app.config['UPLOAD_FOLDER']
    else:
        folder = 'static/uploads/' + UPLOAD_PATHS['temp']['event'].format(uuid)
else:
    with app.app_context():
        folder = app.config['BASE_DIR'] + folder
if not os.path.isdir(folder):
    os.makedirs(folder)

filename = secure_filename(uploaded_file.filename)
uploaded_file.save(os.path.join(folder, filename))
return os.path.join(folder, filename)

```

Example 50

Project: [python-flask-restful-api](#) Author: [akashtalole](#) File: [uploads.py](#) MIT License

5 vc

```

def upload_file():
    force_local = request.args.get('force_local', 'false')
    if 'file' in request.files:
        files = request.files['file']
        file_uploaded = uploaded_file(files=files)
        if force_local == 'true':
            files_url = upload_local(
                file_uploaded,
                UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
            )
        else:
            files_url = upload(
                file_uploaded,
                UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
            )
    elif 'files[]' in request.files:
        files = request.files.getlist('files[]')
        files_uploaded = uploaded_file(files=files, multiple=True)
        files_url = []
        for file_uploaded in files_uploaded:
            if force_local == 'true':
                files_url.append(upload_local(
                    file_uploaded,
                    UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
                ))
            else:
                files_url.append(upload(
                    file_uploaded,
                    UPLOAD_PATHS['temp']['event'].format(uuid=uuid.uuid4())
                ))
    else:
        abort(
            make_response(jsonify(error="Bad Request"), 400)
        )

    return jsonify({"url": files_url})

```