

Table of Contents

1 Open ERP7 Integration.....	1
2 Overview.....	2
3 Prerequisites.....	3
4 Swivel Configuration.....	4
5 Configuring the OpenERP 7 server.....	6
5.1 Upload the custom files to OpenERP 7.....	6
5.2 Testing.....	6
6 Error Messages.....	8
6.1 On OpenERP stack trace.....	8
6.2 On Swivel Log.....	8
7 OpenERP Custom Integration.....	9
8 Overview.....	10
9 Prerequisites.....	11
10 Swivel Configuration.....	12
11 OpenERP User Authentication flow Integration.....	14
11.1 The Standard OpenERP User Authentication flow.....	14
11.2 The Swivel Integrated OpenERP User Authentication flow.....	14
12 Configuring the OpenERP Files.....	16
12.1 openerp/addons/web/static/src/xml/base.xml.....	16
12.2 openerp/addons/web/static/src/js/chrome.js.....	16
12.3 openerp/addons/web/static/src/js/coresetup.js.....	19
12.4 openerp/addons/web/controllers/main.py.....	20
13 Python 3.x changes.....	23
13.1 openerp/addons/web/controllers/main.py.....	23
13.2 Testing.....	23
14 Error Messages.....	26
14.1 On OpenERP stack trace.....	26
14.2 On the Swivel Log.....	26

1 Open ERP7 Integration

2 Overview

This document outlines the steps required to integrate the OpenERP 7.0 with Swivel using dual or single channel authentication.

The Swivel install requires configuring an agent on the Swivel server and setting up a shared secret with the code being added to OpenERP 7.0 server to allow communication for authentication.

Due to the complexity of OpenERP, several files need to be changed (or replaced) on the server, then recompiled. Integration with PINsafe is made through Agent-XML.

NOTE: This document refers to the version 7.0 of OpenERP. If using version 7.0 you can replace the files with a simple copy paste, else please refer to the KB article about [OpenERP_Custom_Integration](#).

3 Prerequisites

OpenERP 7.0 server

Swivel server

The files that need to be replaced: [Media:OpenERP7.tar.gz](#)

4 Swivel Configuration

On the Swivel server configure the agent that is permitted to request authentication. On the PINsafe Administration Console select from the server menu Agents and enter the details of the OpenERP IP address and a shared key, then click on apply. Example:

Name : OpenERP,
Hostname/IP : 10.10.10.252,
shared secret : secret

The screenshot shows the Swivel Administration Console interface. On the left is a navigation menu with the following items: Status, Log Viewer, Server (expanded), Policy, Logging, Transport, Database, Mode, Repository, RADIUS, Migration, Appliance, OATH, Synchronisation Administration, Reporting, User Administration, Save Configuration, Administration Guide, and Logout. The 'Server' menu is expanded to show: Name, Language, License, Jobs, SMTP, Agents (selected), Peers, Single Channel, Dual Channel, Third Party Authentication, and Voice Channel. The main content area is titled 'Server > Agents' and contains the following text: 'Please enter the details for any PINsafe agents below. Agents are permitted to access the au'. Below this text, there is a list of agents with a '+' icon next to 'local'. The configuration form for the selected agent is as follows:

Name:	<input type="text" value="OpenERP"/>
Can act as Repository:	<input type="text" value="Yes"/>
Hostname/IP:	<input type="text" value="10.10.10.252"/>
Shared secret:	<input type="text" value="....."/>
Group:	<input type="text" value="---ANY---"/>
Authentication Modes:	<input type="text" value="ALL"/>
Check password with Repository:	<input type="text" value="No"/>
Username attribute for repository:	<input type="text"/>
Allow alternative usernames:	<input type="text" value="No"/>
Alternative username attributes:	<input type="text"/>

At the bottom of the form, there is a '+ New Entry' button and 'Apply' and 'Re' buttons.

If Single Channel communication is to be used, select from the PINsafe Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

- [Status](#)
- [Log Viewer](#)
- [-] [Server](#)
 - ◊ [Name](#)
 - ◊ [Language](#)
 - ◊ [License](#)
 - ◊ [Jobs](#)
 - ◊ [SMTP](#)
 - ◊ [Agents](#)
 - ◊ [Peers](#)
 - ◊ [Single Channel](#)
 - ◊ [Dual Channel](#)
 - ◊ [Third Party Authentication](#)
 - ◊ [Voice Channel](#)
- [-] [Policy](#)
- [-] [Logging](#)
- [-] [Transport](#)
- [-] [Database](#)
- [-] [Mode](#)
- [-] [Repository](#)
- [-] [RADIUS](#)
- [-] [Migration](#)
- [-] [Appliance](#)
- [-] [OATH](#)
- [-] [Synchronisation Administration](#)
- [-] [Reporting](#)
- [User Administration](#)
- [Save Configuration](#)
- [Administration Guide](#)
- [Logout](#)

Server > Single Channel

Please specify how single channel security strings are delivered.

Allow session request by username:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="Yes"/> ?
Allow alternative usernames:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="No"/> ?
Alternative username attributes:	<input style="width: 100%; height: 20px; border: 1px solid #ccc;" type="text"/>
Multiple Authentications per String:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="No"/> ?
Image file:	<input style="background-color: #f0f0f0;" type="text" value="turingOrange.xml"/> ▾
Background image file:	<input style="background-color: #f0f0f0;" type="text" value="backgroundsMonochrome.xml"/> ▾
Text Alpha Value:	<input style="width: 80%; height: 20px; border: 1px solid #ccc;" type="text" value="100"/>
Only use one font per image:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="Yes"/> ?
Image Rendering:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="Static"/> ?
Jiggle characters within slot:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="No"/> ?
Add blank trailer frame to animated images:	<input style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.9em; font-weight: normal; background-color: #f0f0f0; cursor: pointer; border-radius: 3px;" type="button" value="No"/> ?
Number of complete display cycles per image:	<input style="width: 80%; height: 20px; border: 1px solid #ccc;" type="text" value="10"/>
Inter-frame delay (1/100s):	<input style="width: 80%; height: 20px; border: 1px solid #ccc;" type="text" value="25"/>
No. Characters Visible:	<input style="width: 80%; height: 20px; border: 1px solid #ccc;" type="text" value="1"/>

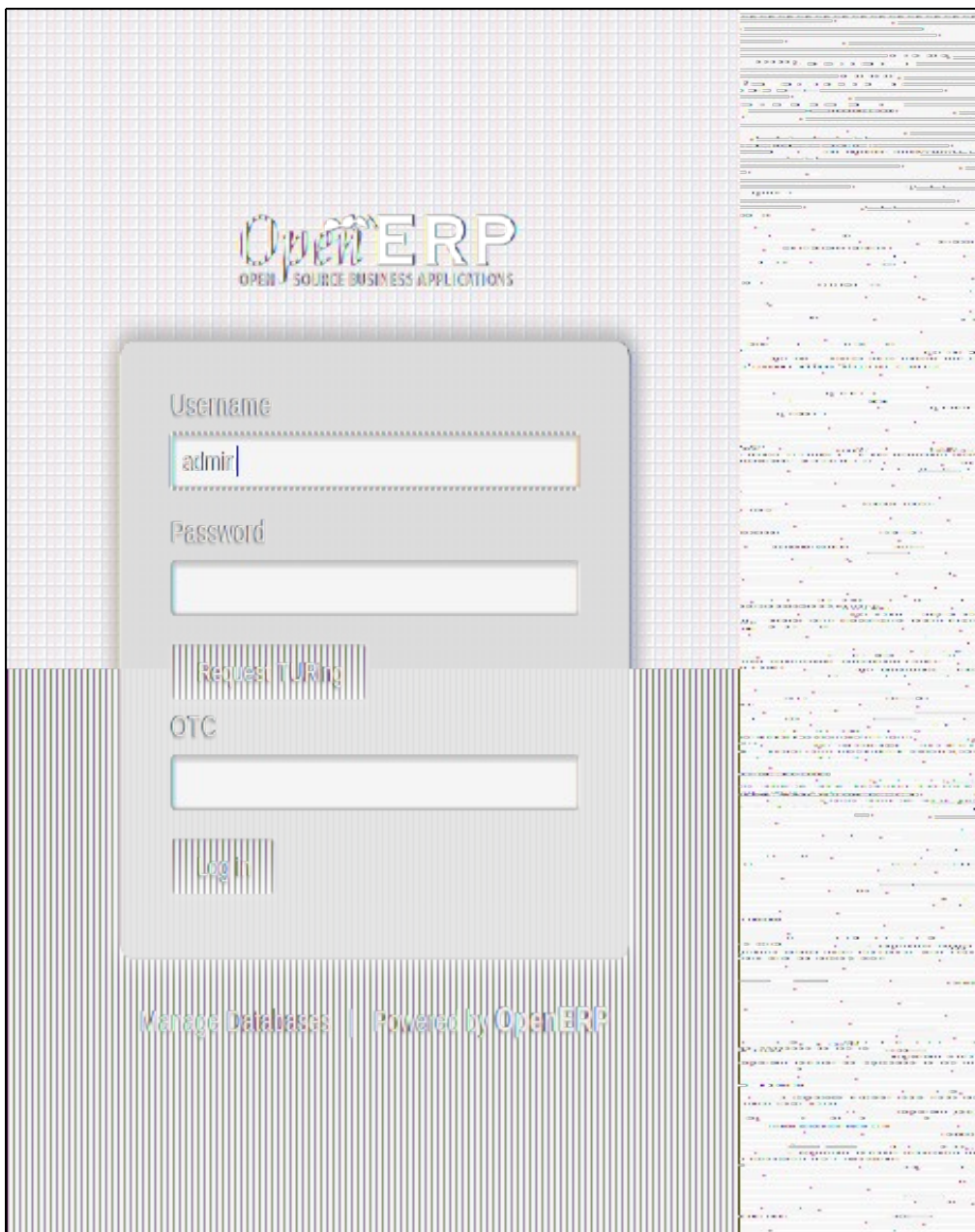
5 Configuring the OpenERP 7 server

5.1 Upload the custom files to OpenERP 7

1. Shutdown OpenERP 7 server service.
2. Unpack the "File Package" (you can find it in the Prerequisites section of this page) to a local directory.
3. Edit the file OpenERP/addons/web/controllers/main.py Search for ?10.10.10.201? (line 859) and change it with the IP address or domain name for your Swivel server
4. Edit the file OpenERP/addons/web/controllers/main.py Search for ?<Secret>secret<%2FSecret>? (line 860) and change it with ?<Secret>THE SECERT YOU CHOSE IN THE AGENT CONFIG<%2FSecret>?
5. Edit the file OpenERP/addons/web/static/source/js/chrome.js Search for ?10.10.10.201? (line 710) and change it with the IP address or domain name for your Swivel server
6. FTP ALL the files to the server running OpenERP, respecting the directory structure and overwriting
7. Start OpenERP Server with (recompile option enabled -u) : /?path to openERP?/openerp-server -u ?database to run?

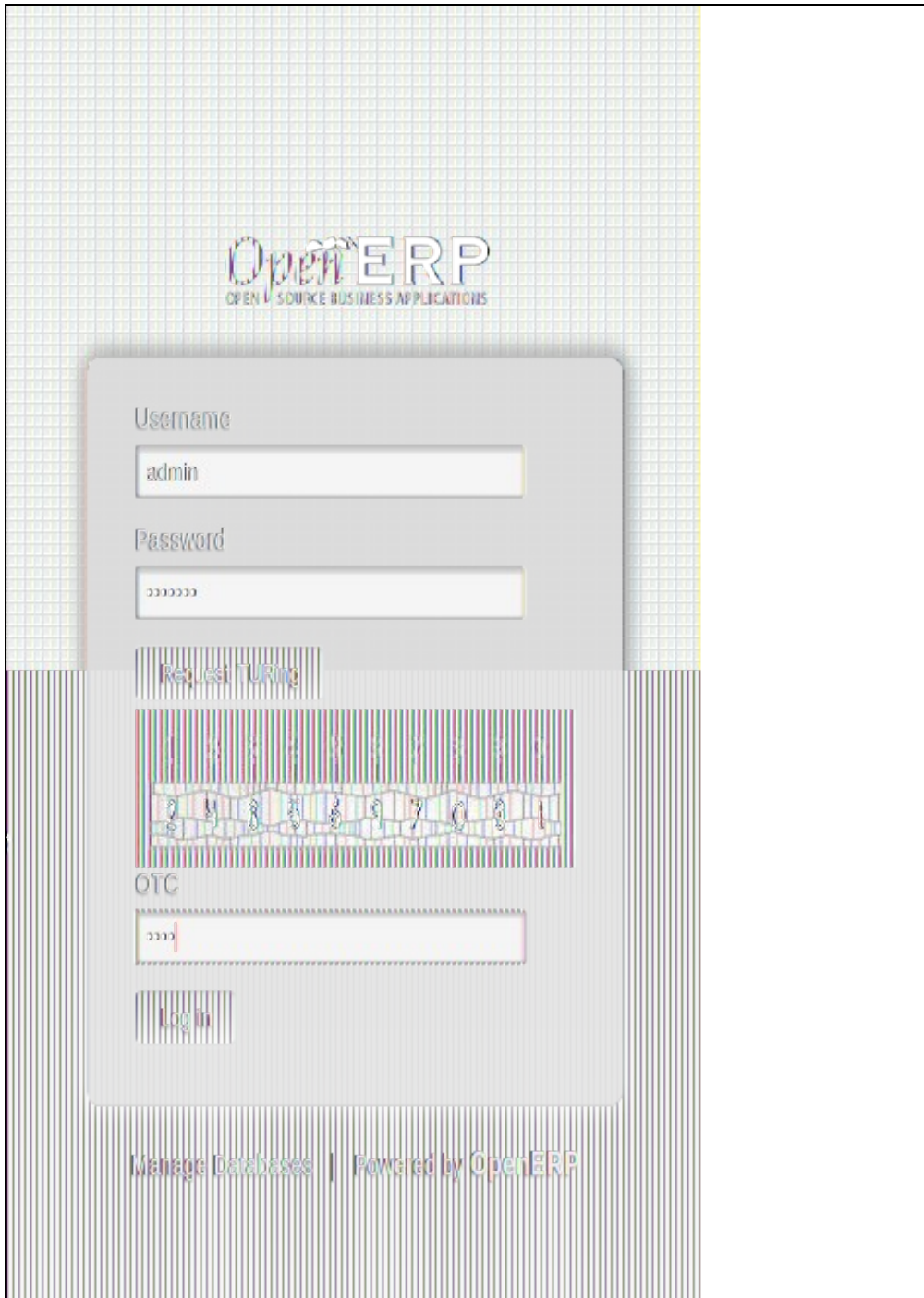
5.2 Testing

Open a browser and point to <http://yourserverip:8069>



After clicking the ?request TURing? button, the swivel appliance log should show ?127.0.0.1:Session started for user: admin.?

And this should be the result:



Then after login, the swivel appliance log should read:

?OPENERP SERVER IP? OpenERP:Login successful for user: admin.

6 Error Messages

6.1 On OpenERP stack trace

6.1.1 "ImportError: No module named urllib.urlopen"

Please refer to the [OpenERP_Custom_Integration](#) article, section Python 3.0 changes.

6.2 On Swivel Log

6.2.1 AgentXML request failed, error: The agent is not authorized to access the server

User fails to authenticate with the above error message in the Swivel log. An Agent on Swivel server has not been defined for the OpenERP server. Go to *Server/Agents* in the Swivel admin console, and add a new entry, using the IP address of the OpenERP server. Make sure the agent secret is the same as on the OpenERP configuration.

7 OpenERP Custom Integration

8 Overview

This document outlines the steps required to integrate the OpenERP with Swivel using dual or single channel authentication.

The Swivel install requires configuring an agent on the Swivel server and setting up a shared secret with the code being added to OpenERP 7.0 server to allow communication for authentication.

Due to the complexity of OpenERP, several files need to be changed (or replaced) on the server, then recompiled. Integration with Swivel is made through Agent-XML.

NOTE: This document uses the files from version 7.0 of OpenERP, however, the principle is that either customized, or from another version, the functions should be similar, making integration easier.

For an article specific for version OpenERP 7.0, please check article [Open_ERP7_Integration](#)

9 Prerequisites

OpenERP server (version 7.0 assumed in this example)

PINsafe server

Text Editor or Code editor

It is recommended to read the following Knowledgebase articles (to better understand how Swivel Agent-XM works):

<https://kb.swivelsecure.com/wiki/index.php/Agent-XML>

<https://kb.swivelsecure.com/wiki/index.php/AuthenticationAPI>

10 Swivel Configuration

On the Swivel server configure the agent that is permitted to request authentication. On the Swivel Administration Console select from the server menu Agents and enter the details of the OpenERP IP address and a shared key, then click on apply. Example:

Name : OpenERP,
Hostname/IP : 10.10.10.252,
shared secret : secret

The screenshot shows the Swivel Administration Console interface. At the top left is the Swivel logo with the tagline "the power of knowing". A navigation menu on the left lists various system settings such as Status, Log Viewer, Server, Policy, Logging, Transport, Database, Mode, Repository, RADIUS, Migration, Appliance, OATH, Synchronisation Administration, Reporting, User Administration, Save Configuration, Administration Guide, and Logout. The main content area is titled "Server > Agents" and contains a list of agents, with "local" selected. Below the list is a configuration form for the selected agent. The form fields are: Name (OpenERP), Can act as Repository (Yes), Hostname/IP (10.10.10.252), Shared secret (masked with dots), Group (---ANY---), Authentication Modes (ALL), Check password with Repository (No), Username attribute for repository (empty), Allow alternative usernames (No), and Alternative username attributes (empty). There are "Apply" and "Delete" buttons at the bottom right of the form area.

If Single Channel communication is to be used, select from the Swivel Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

- [Status](#)
- [Log Viewer](#)
- [-] [Server](#)
 - ◊ [Name](#)
 - ◊ [Language](#)
 - ◊ [License](#)
 - ◊ [Jobs](#)
 - ◊ [SMTP](#)
 - ◊ [Agents](#)
 - ◊ [Peers](#)
 - ◊ [Single Channel](#)
 - ◊ [Dual Channel](#)
 - ◊ [Third Party Authentication](#)
 - ◊ [Voice Channel](#)
- [-] [Policy](#)
- [-] [Logging](#)
- [-] [Transport](#)
- [-] [Database](#)
- [-] [Mode](#)
- [-] [Repository](#)
- [-] [RADIUS](#)
- [-] [Migration](#)
- [-] [Appliance](#)
- [-] [OATH](#)
- [-] [Synchronisation Administration](#)
- [-] [Reporting](#)
- [User Administration](#)
- [Save Configuration](#)
- [Administration Guide](#)
- [Logout](#)

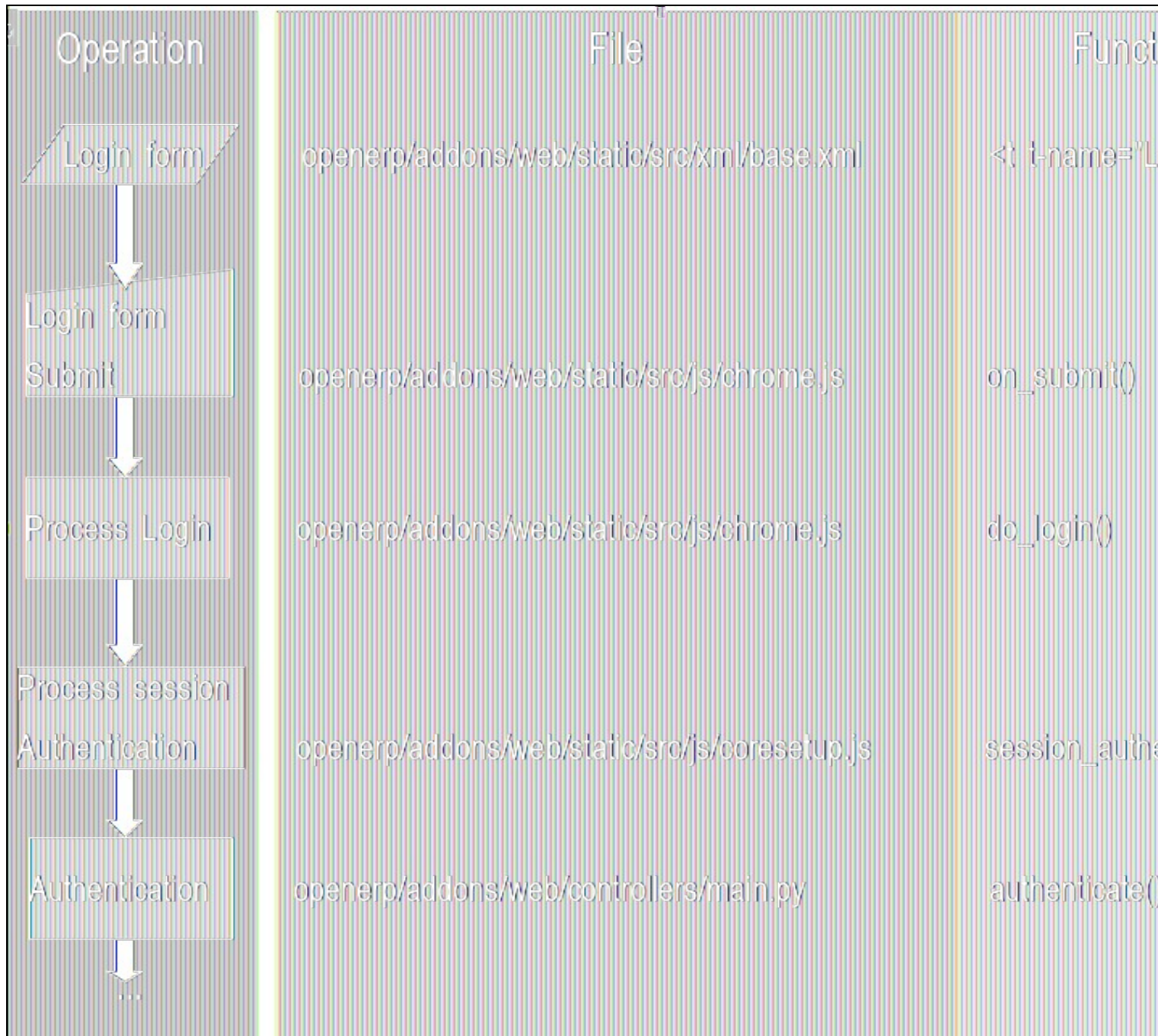
Server > Single Channel

Please specify how single channel security strings are delivered.

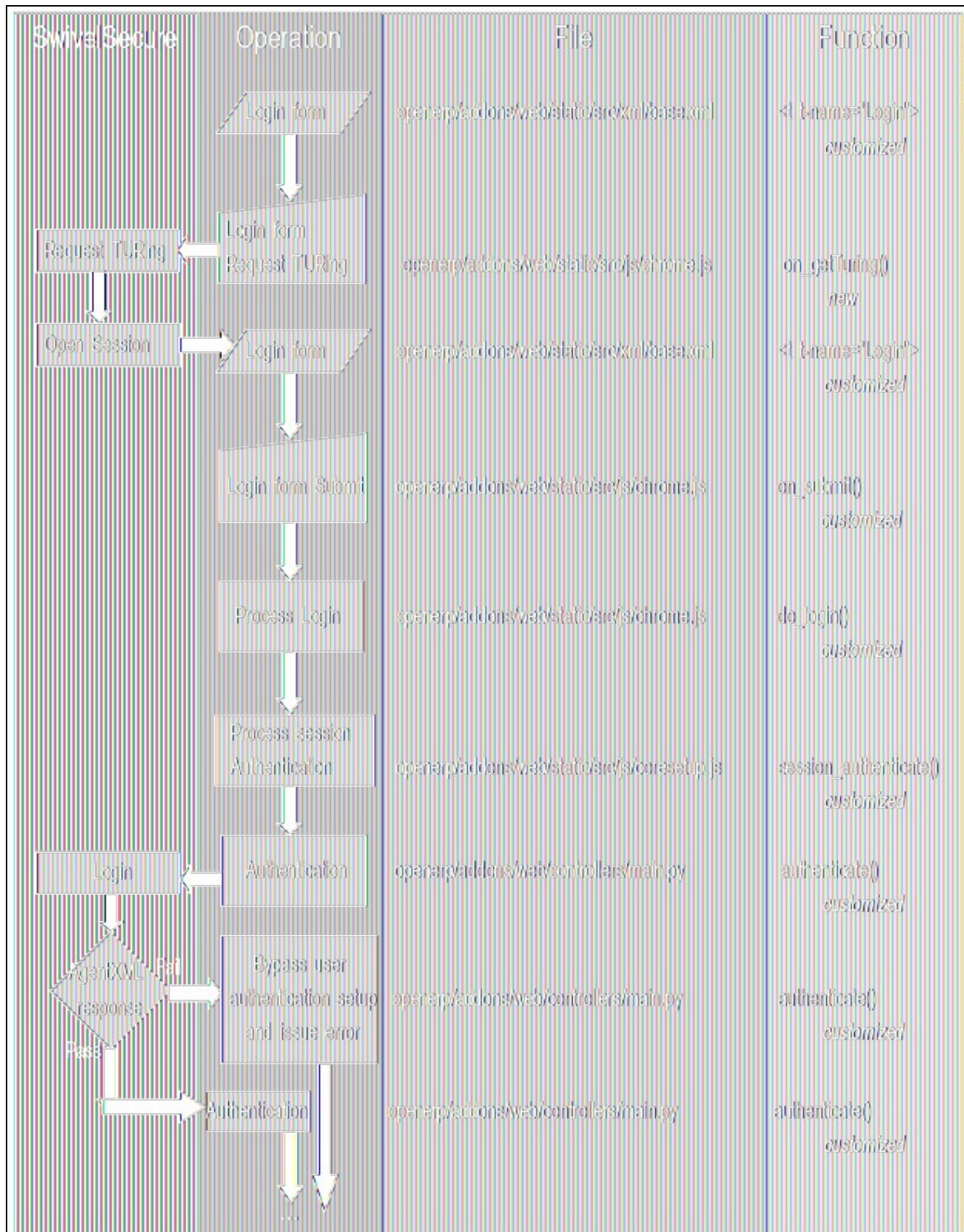
Allow session request by username:	<input style="border: 1px solid #ccc;" type="button" value="Yes"/>
Allow alternative usernames:	<input style="border: 1px solid #ccc;" type="button" value="No"/>
Alternative username attributes:	<input type="text" value=""/>
Multiple Authentications per String:	<input style="border: 1px solid #ccc;" type="button" value="No"/>
Image file:	<input style="border: 1px solid #ccc;" type="text" value="turingOrange.xml"/>
Background image file:	<input style="border: 1px solid #ccc;" type="text" value="backgroundsMonochrome.xml"/>
Text Alpha Value:	<input style="border: 1px solid #ccc;" type="text" value="100"/>
Only use one font per image:	<input style="border: 1px solid #ccc;" type="button" value="Yes"/>
Image Rendering:	<input style="border: 1px solid #ccc;" type="text" value="Static"/>
Jiggle characters within slot:	<input style="border: 1px solid #ccc;" type="button" value="No"/>
Add blank trailer frame to animated images:	<input style="border: 1px solid #ccc;" type="button" value="No"/>
Number of complete display cycles per image:	<input style="border: 1px solid #ccc;" type="text" value="10"/>
Inter-frame delay (1/100s):	<input style="border: 1px solid #ccc;" type="text" value="25"/>
No. Characters Visible:	<input style="border: 1px solid #ccc;" type="text" value="1"/>

11 OpenERP User Authentication flow Integration

11.1 The Standard OpenERP User Authentication flow



11.2 The Swivel Integrated OpenERP User Authentication flow



12 Configuring the OpenERP Files

12.1 openerp/addons/web/static/src/xml/base.xml

Unchanged code in black

Added code in green

Changed code in orange

In this file we are transforming the form to support the Get TURING button and the OTC password field.

Search for the login form code, usually called `<t t-name="Login">?` and place in line nr 61

```
Code
<t t-name="Login">
  <div class="oe_login">
    <div class="oe_login_bottom"> </div>
    <div class="oe_login_error_message"/>
    <div class="oe_login_pane">
      <div class="oe_login_logo"></div>
      <form action="" method="post">
        <div class="oe_login_dbpane" >
          Database:
          <input name="db" t-att-value="widget.selected_db || "">
        </div>
        <ul>
          <li>Username</li>
          <li><input name="login" type="text" value="" autofocus="autofocus"/></li>
          <li>Password</li>
          <li><input name="password" type="password" value=""></li>
          <li><button name="getTURING">Request TURING</button></li>
          <div class="TURING"><img name="TURINGimg" class="myTURINGimg" /></div>
          <li>OTC</li>
          <li><input name="password2" type="password" value=""></li>
          <li><button name="submit">Log in</button></li>
        </ul>
      </form>
      <div class="oe_login_footer">
        <a href="#" class="oe_login_manage_db">Manage Databases</a> |
        <a href="http://www.openerp.com" target="_blank">Powered by <span>OpenERP</span></a>
      </div>
    </div>
  </div>
</t>
```

12.2 openerp/addons/web/static/src/js/chrome.js

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be adding the functions that support automation and control over the form displaying the TURING image and Get TURING button.

First we need to hide the TURING image and register the link from the Get TURING button to the respective function.

Search for the form initiation routine, usually called ? start: function() {? and place in line nr 640

Code

```
start: function() {
    var self = this;
    self.$el.find("form").submit(self.on_submit);
    this.$(dm.TURING).hide();
    self.$el.find("form button[name=getTURING]").click(self.on_getTuring);
    self.$el.find("#oe_login_manage_db").click(function() {
        self.do_action("database_manager");
    });
    self.on("change:database_selector", this, function() {
        this.database_selected(this.get("database_selector"));
    });
    var d = $.when();
    if ($.param.fragment().token) {
        self.params.token = $.param.fragment().token;
    }
    // used by dbmanager.do_create via internal client action
    if (self.params.db && self.params.login && self.params.password) {
        d = self.do_login(self.params.db, self.params.login, self.params.password);
    } else {
        d = self.rpc("/web/database/get_list", {})
            .done(self.on_db_loaded)
            .fail(self.on_db_failed)
            .always(function() {
                if (self.selected_db && self.has_local_storage && self.remember_credentials) {
                    self.$("[name=login]").val(localStorage.getItem(self.selected_db + "last_login") || "");
                }
            });
    }
    return d;
},
```

Then we need to create the function. Find space right after the ?on_db_failed: function? on lines 703 to 707.

Change the IP address on the example with your swivel appliance IP or domain name.

Code

```
on_do_failed: function (error, event) {
    if (error.data.fault_code === 'AccessDenied') {
        event.preventDefault();
    }
},
on_getTuring: function() {
    var login = this.$("form input[name='login']").val();
    var SwivelURL = 'https://10.10.10.201:8443/proxy/SCImage?username=' + login + "&random=" + Math.floor(Math.random() * 10000);
    this.$("myTURINGimg").attr('src', SwivelURL);
    this.$("div.TURING").show();
    return false;
},
```

Finally, we need to change the functions that support the original form submit, so we can add OTC (called "password2" in this example).

Function ?on_submit: function(ev)? should be immediately after your recently added function code, on line nr 715.

You will also update function ?do_login?, placed immediately after ?on_submit? function.

Code

```
on_submit: function(ev) {
    if(ev) {
        ev.preventDefault();
    }
    var db = this.$("form [name=db]").val();
    if (!db) {
        this.do_warn(_("Login"), _("No database selected !"));
        return false;
    }
    var login = this.$("form input[name=login]").val();
    var password = this.$("form input[name=password]").val();
    var password2 = this.$("form input[name=password2]").val();
    this.do_login_db(login, password, password2);
},
/*
 * Performs actual login operation, and UI-related stuff
 *
 * @param {String} db database to log in
 * @param {String} login user login
 * @param {String} password user password
 */
do_login: function (db, login, password, password2) {
    var self = this;
    self.hide_error();
    self.$("#oe_login-pane").fadeOut("slow");

    return this.session.session_authenticate(db, login, password, password2).then(function() {
        self.remember_last_used_database(db);
        if (self.has_local_storage && self.remember_credentials) {
            localStorage.setItem(db + 'last_login', login);
        }
        self.trigger("login_successful");
    }, function () {
        self.$("#oe_login-pane").fadeIn("fast", function() {
            self.show_error(_("Invalid username or password"));
        });
    });
},
},
```

12.3 openerp/addons/web/static/src/js/coresetup.js

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be adding the parameter `?password2?` to the function that creates the bridge between the user interface (in XML, javascript and jquery) and the OpenERP server core (in python).

Search for the `?session_authenticate: function?`, usually place on line nr101

Code

```
session.authenticate: function(db, login, password, password2, _volatile) {
    var self = this;
    var base_location = document.location.protocol + '//' + document.location.host;
    var params = { t: db, login: login, password: password, password2: password2, base_location: base_location };
    return this.rpc('web/session/authenticate', params).then(function(result) {
        if (!result.uid) {
            return $.Deferred().reject();
        }
        _extend(self, result);
        if (!_volatile) {
            self.set_cookie('session_id', self.session_id);
        }
        return self.load_modules();
    });
},
```

12.4 openerp/addons/web/controllers/main.py

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be changing the OpenERP core authentication function, so that it calls the Swivel Secure server with a login function request, and then continues on the consideration of the reply given by swivel software.

First we need to add a library module to handle and parse the XML sent by the Swivel Agent-XML.

This is added in the very beginning of the code file.

Code

```
#!/usr/bin/perl -w
# coding: utf-8

import ast
import base64
import csv
import glob
import itertools
import logging
import operator
import datetime
import hashlib
import os
import re
import simplejson
import time
import urllib
import urllib2
import urlparse
import xmlrpclib
import zlib

from xml.etree import ElementTree
from cStringIO import StringIO

from xml.dom.minidom import parseString

import babel.messages.pofile
import werkzeug.utils
import werkzeug.wrappers

try:
    import xbt
except ImportError:
    xbt = None

import openerp
import openerp.modules.registry
from openerp.tools.translate import _
from openerp.tools import config

from .. import http
openerpweb = http
```

Finally, we need to change the `?def authenticate?` function to handle the the new `?password2?` parameter, call the Swivel Server, parse the response and make decision based on the response.

The `?def authenticate?` function should start in line nr 857 or 858

Don't forget to change the server IP in the code by your own Swivel server IP or domain path and the "secret".

Code

```
@ogonarpweb.jsonrequest
def authenticate(self, req, db, login, password, password2, base_location=None):

    url = 'https://10.10.201.80:80/pinsafe/AgentXML'
    params = '%3Fxml-version%3D1.0' + encoding%3D'UTF-8'%3F<SASRequest><Version>3.6<%2FVersion><Secret>secret<%2FSecret>
    <Action>login<%2FAction><Username>' + login + '<%2FUsername><Password><%2FPassword><OTC>' + password2 + '<%2FOTC><%2FSASRequest>'
    data = urllib.urlopen(url + '?' + params).read()
    dom = parseString(data)
    xmlTag = dom.getElementsByTagName('Result')[0].toxml()
    xmlData = xmlTag.replace('<Result>', '').replace('</Result>', '')
    if xmlData == 'PASS':
        wsgienv = req.hitrequest.environ
        env = dict(
            base_location=base_location,
            HTTP_HOST=wsgienv['HTTP_HOST'],
            REMOTE_ADDR=wsgienv['REMOTE_ADDR'],
        )
        req.session.authenticate(db, login, password, env)
        return self.session_info(req)
    if xmlData == 'FAIL':
        return {'error': _('Error, Invalid credentials !'), 'title': _('No access')}
```

13 Python 3.x changes

13.1 openerp/addons/web/controllers/main.py

Unchanged code in black

Added code in green

Changed code in orange

If running Python 3.0, then you have a call to the urllib module.

The urllib <<http://docs.python.org/2/library/urllib.html#module-urllib>> module has been split into parts and renamed in Python 3 to urllib.request, urllib.parse, and urllib.error.

This is shown in RED in the example from the changes to be done on file ?openerp/addons/web/controllers/main.py?

Do not forget to change the IP an "secret" in the code by your own Swivel server IP or domain path.

```
Code

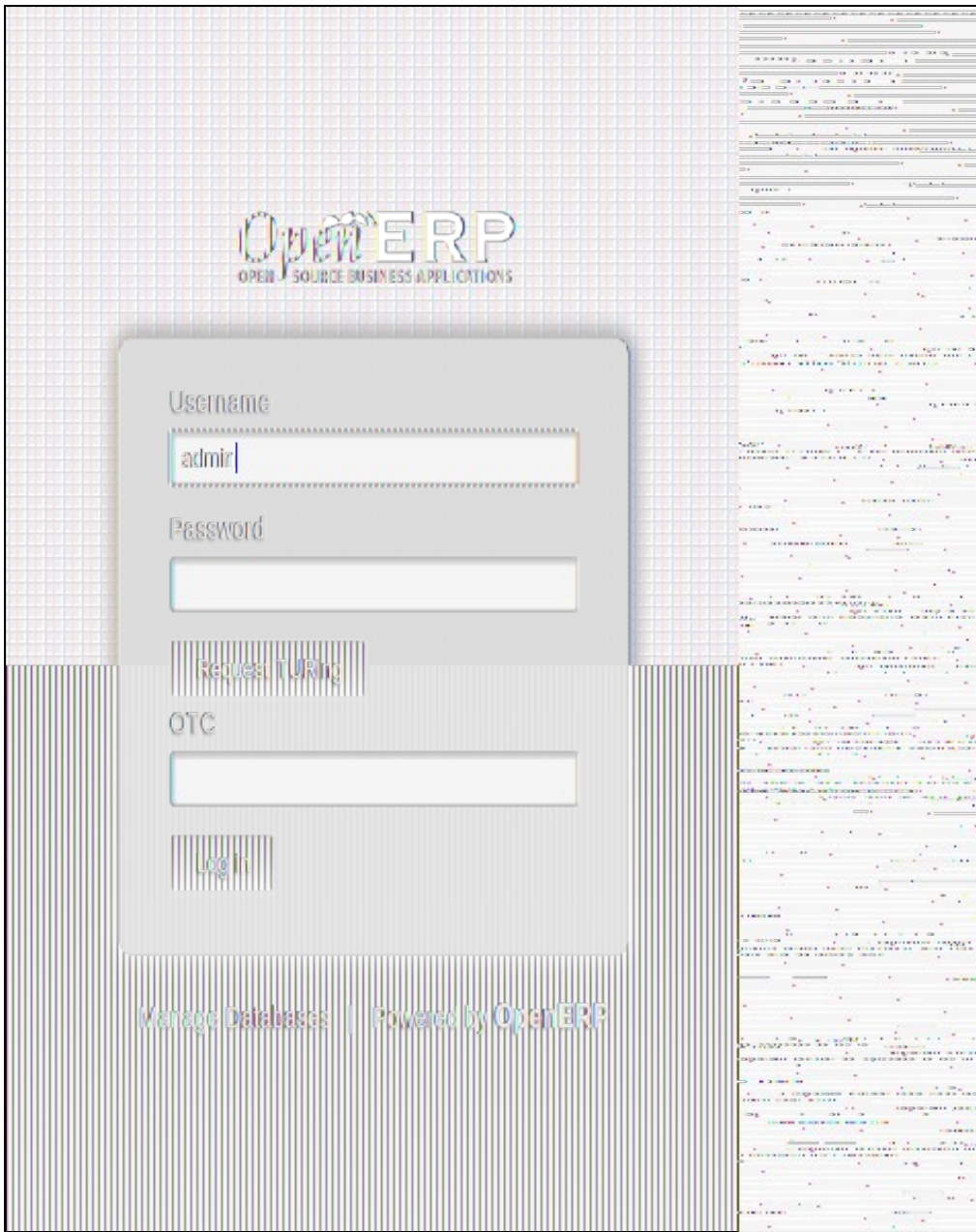
@openerpweb.jsonrequest
def authenticate(self, req, db, login, password, password2, base_location=None):

    url = 'https://10.10.10.201:8060/pinSAFE/AgentXML'
    params = '<%3Fxml+version%3D'1.0'+encoding%3D'UTF-8'%2F><SASRequest><Version>3.6<%2FVersion><Secret>secret<%2FSecret>
<Action>login<%2FAction><Username>' + login + '<%2FUsername><Password><%2FPassword><OTC>' + password2 + '<%2FOTC><%2FBASRequest>'

    data = urllib.request.urlopen(url + '?' + params).read()
    dom = parseString(data)
    xmlTag = dom.getElementsByTagName('Result')[0].toxml()
    xmlData = xmlTag.replace('<Result>', '').replace('</Result>', '')
    if xmlData == 'PASS':
        wsgienv = req.httprequest.environ
        env = dict(
            base_location=base_location,
            HTTP_HOST=wsgienv['HTTP_HOST'],
            REMOTE_ADDR=wsgienv['REMOTE_ADDR'],
        )
        req.session.authenticate(db, login, password, env)
        return self.session_info(req)
    if xmlData == 'FAIL':
        return {'error': _('Error, Invalid credentials !'), 'title': _('No access')}
```

13.2 Testing

Open a browser and point to <http://yourserverip:8069>



After clicking the ?request TURING? button, the swivel appliance log should show ?127.0.0.1:Session started for user: admin.?

And this should be the result:

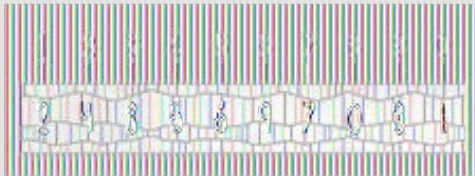


Username

admin

Password

Request Turing



OTC

Log In

Manage Databases | Powered by OpenERP

Then after login, the swivel appliance log should read:

?OPENERP SERVER IP? OpenERP:Login successful for user: admin.

14 Error Messages

14.1 On OpenERP stack trace

14.1.1 "ImportError: No module named urllib.urlopen"

Please refer to the this article, section Python 3.0 changes.

14.2 On the Swivel Log

14.2.1 AgentXML request failed, error: The agent is not authorized to access the server

User fails to authenticate with the above error message in the Swivel log.

This means that an Agent on Swivel server has not been defined for the OpenERP server.

Go to Server/Agents in the PINsafe admin console, and add a new entry, using the IP address of the OpenERP server.

Make sure the agent secret is the same as on the OpenERP configuration.