<table>
<thead>
<tr>
<th>Number</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Custom Templates</td>
<td>71</td>
</tr>
<tr>
<td>22</td>
<td>Theme: Code Snippets for Dynamic Content</td>
<td>77</td>
</tr>
<tr>
<td>23</td>
<td>Theme: Homepage Template</td>
<td>89</td>
</tr>
<tr>
<td>24</td>
<td>Theme: Mobile Templates</td>
<td>91</td>
</tr>
<tr>
<td>25</td>
<td>Theme: Structure</td>
<td>95</td>
</tr>
<tr>
<td>26</td>
<td>Tendenci Credits</td>
<td>97</td>
</tr>
</tbody>
</table>
Tendenci is a association management system (AMS) designed for organizations with powerful features for event registration, membership management, and email communication.

Tendenci allows membership management organizations to communicate, manage and conduct transactions with website visitors. Tendenci is easy to use and offers many modules as one customizable software package.

The setup process is broken into three sections. The first section covers required setup needed in all cases. The second section covers optional additional setup for development purposes. The third section covers recommended but optional setup to prepare for running a live site. Steps required for testing are documented separately.
If you are new to Tendenci, we recommend you setup a copy on your local system before attempting to install on public hosting. This will allow you to experiment with the setup process, with Tendenci itself, with your site design/layout, and with themes before building a real site. We strongly recommend installing Tendenci within a virtual environment, which will enable you to install/run Ubuntu within your existing OS, and will also make it easy to setup/test/discard multiple copies/configurations of Tendenci.

### 1.1 System Requirements

These instructions cover the installation of Tendenci on a system running Ubuntu 18.04 (recommended) and Ubuntu 16.04.

It should technically be possible to install Tendenci on nearly any Linux, Mac, or Windows system, and some partial instructions for installing on macOS are included below. However, we generally recommend installing Tendenci on Ubuntu, and this process has only been thoroughly tested and documented on Ubuntu.

For testing purposes, we recommend using a virtual machine with at least:

- 2 CPU cores
- 3GB of RAM
- 6GB of storage

If necessary, you may be able to run Tendenci on a VM with as little as 1 CPU core, 2GB of RAM, and 3GB of storage, but you will likely find that to be too slow/constrained even for testing purposes.

Please note that Tendenci 11 requires Python 3.6 or newer.

### 1.2 System Preparation

Make sure Ubuntu is up to date before beginning:
sudo apt update
sudo apt dist-upgrade

Also ensure that Ubuntu is configured to automatically install security updates:

sudo apt install unattended-upgrades update-notifier-common
sudo dpkg-reconfigure -plow unattended-upgrades

Install the required system dependencies:

sudo apt install build-essential \ libevent-dev libpq-dev \ libjpeg8 libjpeg-dev libfreetype6 libfreetype6-dev

For Ubuntu 18.04, add the universe repository:

sudo add-apt-repository universe

If on for Ubuntu 16.04 (skip if on 18.04), install Python 3.6 (Ubuntu 16.04 comes with Python 3.5. Ubuntu 18.04 has Python 3.6 installed by default.):

sudo add-apt-repository ppa:jonathonf/python-3.6
sudo apt update
sudo apt install python3.6

Install python3.6-dev and pip:

sudo apt install python3.6-dev --upgrade
# The python3-distutils is missing in Ubuntu 18.04 https://github.com/pypa/get-pip/ \ -issues/43. Install it before proceeding.
sudo apt-get install python3-distutils
curl "https://bootstrap.pypa.io/get-pip.py" | sudo python3.6

1.2.1 System Preparation on macOS

In order for images to correctly render, you will need to install the jpeg libraries:

tar -xvzf jpegsrc.v9.tar.gz
cd jpeg-9
./configure
make
sudo make install

To remove the files

cd ..
rm -r jpeg-9
rm jpegsrc.v9.tar.gz

You will also need Xcode 4.4.1 or higher (in the app store) and will need to install the Command Line tools it comes with. To install these, open Xcode, click the “Xcode” menu item in the top left of the screen near the Apple logo, then click “Preferences”, then click “Downloads”. Then click install on the line next to Command Line Tools.
1.3 Database Preparation

Tendenci is designed for use with PostgreSQL.

It may be technically possible to use Tendenci with other databases, but Tendenci has only been tested with PostgreSQL and is not expected to work with other databases without some significant development effort.

Install PostgreSQL and some required modules:

```
sudo apt install postgresql postgresql-contrib postgis
```

Create a user and database for Tendenci in PostgreSQL (Replace `DB_USER`, `DB_PASS` and `DB_NAME` with an appropriate user name, password, and database name):

```
# switch to the "postgres" user assuming installation above went correctly
sudo -u postgres -s
DB_USER=mysite
echo "DB_USER=
DB_PASS=mysite
echo "DB_PASS=
DB_NAME=mysite
echo "DB_NAME=

Verify what you typed is what you intended. Next create your database still as the 'postgres' user:

```
psql -c "CREATE USER $DB_USER WITH PASSWORD '$DB_PASS';"
psql -c "ALTER ROLE $DB_USER SET client_encoding TO 'UTF8';"
psql -c "ALTER ROLE $DB_USER SET default_transaction_isolation TO 'read committed';"
psql -c "CREATE DATABASE $DB_NAME WITH OWNER $DB_USER;"
pql -d $DB_NAME -c "CREATE EXTENSION postgis;"
pql -d $DB_NAME -c "CREATE EXTENSION postgis_topology;"
pql -d $DB_NAME -c "CREATE EXTENSION fuzzystrmatch;"
pql -d $DB_NAME -c "CREATE EXTENSION postgis_tiger_geocoder;"

# confirm your database was created correctly
psql -l
# deactivate the postgres user reverting back to "root"
exit
# verify path statement reads as your admin Ubuntu user which is by default "root"
whoami
# this should return something like:
root@tendenci:/home/ed# whoami
root
```

1.3.1 Database Preparation on macOS

To install PostgreSQL on macOS, we recommend using Postgres.app

If you would like a database GUI, we recommend PGAdmin

The commands shown above for creating the database and user on Ubuntu should also work on macOS.

If you are still in “sudo postgres” mode, meaning your command prompt reads “postgres@localhost”, you need to exit from the “sudo postgres” by typing “exit” which should return you to “root” or the admin account.
1.4 Virtualenv Preparation

Virtualenv enables a complete Python environment to be stored within a single directory, which helps keep the system clean, and makes it easy to run multiple versions of Python, Python libraries, or Tendenci within a single VM or server. While not strictly required, we recommend using virtualenv with Tendenci.

Please make sure, that you only use root permissions where advised! (Please do NOT sudo or install as root as that defeats the purpose of a virtualenv). Otherwise, the installation will fail!

Prepare a virtualenv for Tendenci:

```
# whoami - make sure you are NOT "root". If so, type "exit" to back up one level.
sudo apt-get install virtualenv
sudo mkdir -p /srv
sudo chown "$(id -u -n)" /srv/
cd /srv/
python3.6 -m virtualenv -p python3.6 mysite
sudo chown root /srv/
```

To use the virtualenv, you must “activate” it:

```
source /srv/mysite/bin/activate
# verify you are in the venv environment you created
which python
/srv/mysite/bin/python
```

You can “deactivate” the virtualenv using:

```
deactivate
```

Activating the virtualenv adjusts your $PATH so that running python, pip, or other commands will run the versions in the virtualenv instead of the versions installed in the base system.

The pip in the virtualenv automatically installs Python libraries into the virtualenv without modifying the base system. The python in the virtualenv automatically uses the Python libraries within the virtualenv instead of any system-level libraries.

Note that you must “activate” the virtualenv every time you open a new terminal or SSH session. If your command prompt does not begin with (mysite), then you have not activated the virtualenv, and any python or pip commands you run will use the system versions of those commands instead of the virtualenv versions, which may have unexpected results.

Maybe also consider virtualenvwrapper in your VM for convenience.

1.4.1 Virtualenv on macOS

To prepare a virtualenv for Tendenci:

```
virtualenv -p venv
```

If you have anaconda installed on your machine, you must specify the path to the version of Python that you would like to use:

```
virtualenv -p /Library/Frameworks/Python.framework/Versions/3.6/bin/python3 venv
```

The commands shown above for activating/deactivating the virtualenv on Ubuntu should also work on macOS.
1.5 Tendenci Installation

If you have not already activated your virtualenv, make sure you do so before continuing:


source /srv/mysite/bin/activate

Install Django:


pip install "Django>=1.11,<2.0"

Create a new Django project based on the Tendenci template:


sudo mkdir /var/www/
sudo chown "$\$(id -u -n)" /var/www/


cd /var/www/
django-admin.py startproject --template=https://github.com/tendenci/tendenci-project-template/archive/master.zip mysite


sudo chown root /var/www/

Set up some directories appropriately:


chmod -R -x+X /var/www/mysite/media/
sudo mkdir /var/log/mysite


sudo chown "$\$(id -u -n)" : /var/log/mysite/

Install Tendenci and its dependencies:


cd /var/www/mysite/
pip install --no-binary psycopg2 -r requirements/dev.txt --upgrade

1.6 Tendenci Configuration

Edit /var/www/mysite/conf/settings.py

For testing, uncomment DEBUG = True

Set SECRET_KEY and SITE_SETTINGS_KEY to two different random strings, each at least 50 characters in length. Random strings are conveniently available at <https://www.miniwebtool.com/django-secret-key-generator/>

For live sites, set ALLOWED_HOSTS appropriately. For testing, the default should be sufficient.

Configure the database settings using the user name, password, and database name you configured in the “Database Preparation” section above.

Set TIME_ZONE to the local time zone configured on the system running Tendenci.

1.7 Tendenci Preparation

Copy “tendenci2018” theme to the themes directory (adjust your path if your virtualenv path is different or you’re running a different python version rather than python 3.6):


mkdir /var/www/mysite/themes/tendenci2018
Initialize the database and static files:

```
cd /var/www/mysite/
python manage.py initial_migrate
python manage.py deploy
chmod -R -x+X /var/www/mysite/media/
python manage.py load_tendenci2018_defaults
python manage.py update_dashboard_stats
python manage.py rebuild_index --noinput
```

Set the “Site URL” setting appropriately:

```
python manage.py set_setting site global siteurl 'https://www.example.com'
```

Create an administrator login in Tendenci:

```
python manage.py createsuperuser
```

To change the theme (optional):

```
python manage.py set_theme <theme name>
```

### 1.8 Testing

Start Tendenci:

```
source /srv/mysite/bin/activate
cd /var/www/mysite/
python manage.py runserver
```

Then open [http://localhost:8000/](http://localhost:8000/) in your browser to see your Tendenci site!

Hit CTRL-C to exit from python manage.py runserver

If you only plan to use this Tendenci installation for demonstration/testing purposes, you can stop here. If you intend to do development or run a live site, continue with one of the next two sections.
Development Setup

This section covers additional recommended setup if you plan to do development work on Tendenci itself.

Install and configure git (replace the name and email with appropriate values):

```
sudo apt-get install git
git config --global user.name "John Doe"
git config --global user.email "john.doe@example.com"
```

**Note:** See this Github help article for instructions.

Clone the Tendenci repository:

```
cd /srv/
git clone https://github.com/tendenci/tendenci.git tendenci-git
```

Edit `/var/www/mysite/requirements/tendenci.txt` and comment out `tendenci>=8,<9`

Edit `/var/www/mysite/requirements/common.txt` and add:

```
-e /srv/tendenci-git/
```

Run `pip install` again:

```
source /srv/mysite/bin/activate
cd /var/www/mysite/
pip install -r requirements/dev.txt --upgrade
```

Apply any database and static media updates that are in the development version of Tendenci but are not yet in the released version of Tendenci:

```
python manage.py migrate
python manage.py deploy
```
You can now edit files under /srv/tendenci-git/ and/or run git pull to retrieve the latest changes from GitHub. python manage.py runserver should immediately pick up your changes, so you can view them simply by reloading the page in your browser.
This section covers additional recommended setup to prepare for running a live site.
You may want to check the Django documentation and PostgreSQL documentation for database configuration/tuning suggestions.

### 3.1 Additional Configuration

Review the comments and examples in `/var/www/mysite/conf/settings.py` to determine if any additional settings should be configured for your live site.

### 3.2 How far will it scale?

But first, how far will Tendenci scale?

Scalability is one of Tendenci’s strengths, with memory as your limit. t2 instances at AWS scale, Linode (or similar) offer the best economics for clients with fewer than 10k active users.

There are Tendenci installations with significantly over 100k active users. Those usually self host (for example regulatory reasons) but you should have no problem whatsoever with fewer than 100k users on a large VM with a lot of RAM.

Once you cross 100k people, deployments will almost certainly need a combination of multiple front end servers, central authentication, and a cluster database.

### 3.3 Additional Installation

Install additional system dependencies:
sudo apt install nginx memcached libmemcached-dev

Install additional Python dependencies:

source /srv/mysite/bin/activate
cd /var/www/mysite/
pip install -r requirements/prod.txt --upgrade

### 3.4 Permissions

When using the Systemd configuration below, Tendenci will run as the `www-data` user. To ensure that Tendenci has appropriate permissions when running as `www-data`:

```bash
chmod -R o+rX /srv/mysite/
sudo chgrp -Rh www-data /var/www/mysite/
chmod -R g+rwX /var/www/mysite/media/ /var/www/mysite/themes/
chmod -R g+rwX /var/www/mysite/whoosh_index/
sudo chown -Rh www-data /var/log/mysite/
chmod -R g+rwX /var/log/mysite/
```

For security, you should also ensure that there are no unnecessary permissions:

```bash
# Owner/group on /srv/mysite/ should already be your normal user account
# Tendenci needs read/search/execute but not write
chmod -R o+rX-w /srv/mysite/
# Owner on /var/www/mysite/ should already be your normal user account,
# group should be www-data
# Tendenci needs read/search but not execute, write is only needed on media, themes,...
# and
# whoosh_index, other users do not need any access
sudo chgrp -Rh www-data /var/www/mysite/
chmod -R x+X,g-w,o-rwx /var/www/mysite/
chmod -R ug-x+rwX,o-rwx /var/www/mysite/media/ /var/www/mysite/themes/
chmod -R ug-x+rwX,o-rwx /var/www/mysite/whoosh_index/
# Owner on /var/log/mysite/ should be www-data,
# group should be some group that includes your normal user account but not other,
# inappropriate
# users (Ubuntu creates a dedicated group for each user by default, so this command
# uses that)
# Tendenci and the assigned group need read/write/search but not execute, other users
# do not
# need any access (The assigned group needs write access so your normal user account
# can run
# `python manage.py ...`)
sudo chown -Rh www-data:$(id -u -n) /var/log/mysite/
sudo chmod -R x+X,g+rw,o-rwx /var/log/mysite/
```

Note that these permissions should allow your normal user account to manage/upgrade Tendenci without using sudo, which is safer than performing management/upgrades using sudo.
3.5 Systemd Setup

Systemd Unit files should be created in `/etc/systemd/system/` with a name of your choice (e.g. `mysite.service`). The contents of your unit file should look like this:

```ini
[Unit]
Description=Start Tendenci instance
#Requires=nginx.service postgresql.service
#Wants=memcached.service
#Before=nginx.service
#After=postgresql.service

[Service]
WorkingDirectory=/var/www/mysite
PIDFile=/run/mysite.pid
Type=forking
KillMode=process
Restart=on-failure
ExecStart=/srv/mysite/bin/gunicorn
   --user www-data
   --group www-data
   --workers 4
   --bind=127.0.0.1:8000
   --pid=/run/mysite.pid
   --pythonpath=/var/www/mysite
   --access-logfile=/var/log/mysite/access.log
   --error-logfile=/var/log/mysite/server.log
   --capture-output
   --daemon
   conf.wsgi

[Install]
WantedBy=multi-user.target
```

You’ll need to issue the `start` and `enable` commands to start the service and make it run automatically on boot.

To start the service manually:

```bash
sudo systemctl start mysite
```

To start the service automatically on boot:

```bash
sudo systemctl enable mysite
```

If you changed the configuration file, you can restart the service with:

```bash
sudo systemctl restart mysite
```

or issue the `reload` command:

```bash
sudo systemctl reload mysite
```

To stop the service:

```bash
sudo systemctl stop mysite
```

To disable the service from starting on boot:
To ensure that the log files remain readable/writable by your normal user account, change the second `www-data` in `create 0660 www-data www-data` to a group that includes your normal user account but not other inappropriate users (Ubuntu creates a dedicated group for each user by default with the same name as the associated user, so you can use that here).

### 3.7 NGINX Setup

First, make an NGINX configuration file for the site. This will be created at `/etc/nginx/sites-available/mysite` and should look like this:

```nginx
server {
    listen 80;
    server_name localhost;

    charset utf-8;
    keepalive_timeout 65;
    client_max_body_size 30M;
    gzip_types text/css application/javascript text/javascript text/plain text/xml;
    gzip_vary on;

    root /var/www/mysite/;

    location /static/ {
        access_log off;
        expires 30d;
    }

    location /media/ {
```

(continues on next page)
Tendenci Documentation, Release 11.4.10

(continued from previous page)

```conf
    access_log off;
    expires 30d;
}

location ^~ /media/export/ {
    return 404;
}

location ~ /themes/([a-zA-Z0-9-\-_]+)/(media|static)/ {
    access_log off;
}

location / {
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://127.0.0.1:8000/;
}
```

Be sure to update the `server_name` domain name and `root` path appropriately.

Next, create a symlink to the new file and remove the default symlink:

```bash
ln -s /etc/nginx/sites-available/mysite /etc/nginx/sites-enabled/mysite
rm /etc/nginx/sites-enabled/default
```

Finally, restart the NGINX service:

```bash
sudo service nginx restart
```

## 3.8 EMail Setup (if needed)

You can run Tendenci through AWS SES or any other SMTP relay using python directly via settings.py. As a best practice we recommend ElasticStack with alerts at scale. If you absolutely need a mail server (rkhunter for example), you can install postfix.

```bash
sudo apt-get install postfix
```

During installation, you will see 3 prompts which you must respond to in order to complete the installation. If the server is already configured with a domain name, enter that domain in the prompts. Otherwise, leave the defaults in place.

Configuration of mail servers is difficult. Be careful. Check the docs on Postfix for proper configuration.

## 3.9 memcached Setup

memcached is a service that can be used to speed up access to web pages by caching them in the systems memory for future use.

Tendenci is configured to use memcache automatically once it is installed as described in the “Additional Installation” section above.
If you wish to increase the amount of memory memcache uses, edit `/etc/memcached.conf` and change the “-m” value from 64 to your desired cache size in MB.

### 3.10 Cron Setup

Tendenci has two management commands that should be run on a regular basis.

* `run_nightly_commands` is used to handle tasks like membership notices, event reminders, and periodic cleanup.

* `process_unindexed` handles search indexing.

Run `sudo -u www-data crontab -e` and add the following lines:

```
30 2 * * * /srv/mysite/bin/python /var/www/mysite/manage.py run_nightly_commands
10 */6 * * * /srv/mysite/bin/python /var/www/mysite/manage.py process_unindexed
```

### 3.11 Multiple Sites

To install multiple sites on a single server:

- Create a unique user and database in PostgreSQL for each site.

- A unique virtualenv may optionally be created for each site, but is not required. If using multiple virtualenvs, be sure to source the correct virtualenv before running any `python` or `pip` commands.

- You can skip the `pip install "Django"` command, but otherwise follow the “Tendenci Installation”, “Tendenci Configuration”, and “Tendenci Preparation” steps above. Change the last parameter of the `django-admin.py startproject ...` command to create a different `/var/www/<project>/` directory for each site, then change `/var/www/mysite/` in all of the commands in the instructions to work with the appropriate site.

- Configure an additional Systemd Unit file and NGINX configuration file for each site, and configure additional cron jobs for each site. Each site will need to run on a different port internally, so change `8000` in the example Systemd and NGINX configurations to a unique port number for each site. Also change the names of any log files in the configuration.

- A separate memcached instance for each Tendenci site may optionally be used, but is not required.
Performing a Minor Update

These instructions will upgrade Tendenci to a new minor version.

Shut down the Tendenci server if necessary:
```
sudo service mysite stop
```

Activate your virtualenv:
```
source /srv/mysite/bin/activate
```

Upgrade Tendenci and its dependencies:
```
cd /var/www/mysite/
# For testing installations:
pip install -r requirements/dev.txt --upgrade
# For live installations:
pip install -r requirements/prod.txt --upgrade
```

Apply database and static media updates, then clear the cache:
```
python manage.py migrate
python manage.py deploy
python manage.py clear_cache
```

Verify that permissions are still set appropriately:
```
# For a testing/development installation, this may be sufficient:
chmod -Rh -x+X /var/www/mysite/media/
# For a live site, you should run:
chmod -R o+rX-w /srv/mysite/
sudo chgrp -Rh www-data /var/www/mysite/
chmod -R -x+X,g-w,o-rwx /var/www/mysite/
chmod -R ug-x+rwX,o-rwx /var/www/mysite/media/ /var/www/mysite/whoosh_index/
sudo chown -Rh www-data:"$(id -u -n)" /var/log/mysite/
sudo chmod -R -x+X,g+rw,o-rwx /var/log/mysite/
```
Restart the Tendenci server:

```
sudo service mysite restart
```
Among many other changes, Tendenci 11.0 adds support for Python 3, and allows for the use of a simplified configuration which should reduce the number of configuration changes that will be required during future upgrades.

### 5.1 Step 1: Prepare for Upgrade

Back up your site and database!

Make sure the required system dependencies are installed:

```
sudo apt-get install \
    build-essential python3-dev libevent-dev libpq-dev \
    libjpeg8 libjpeg-dev libfreetype6 libfreetype6-dev
sudo apt-get install curl wget
```

If this is a live site using memcached, also make sure libmemcached-dev is installed:

```
sudo apt-get install libmemcached-dev
```

Shut down mysite:

```
sudo service mysite stop
```

### 5.2 Step 2: Upgrade to Python 3

Upgrading to Python 3.6 or newer is required. This involves rebuilding your virtualenv.

Move your old virtualenv out of the way:

```
sudo mv /srv/mysite /srv/mysite.t7
```

Prepare a new Python 3 virtualenv:
sudo apt-get install python3-virtualenv
sudo chown "$(id -u -n)" /srv/
cd /srv/
python3 -m virtualenv -p python3 mysite
sudo chown root /srv/
source /srv/mysite/bin/activate
pip install "Django>=1.11,<2.0"

In this new virtualenv, /srv/mysite/venv/bin/ has been moved to /srv/mysite/bin/.

Update your systemd Units, Upstart jobs, cron jobs, and any other scripts which reference /srv/mysite/venv/bin/ to use /srv/mysite/bin/ instead.

If you have already upgraded to Tendenci 11.0 on Python 2 and are upgrading to Python 3 as a separate process, run the following, then skip to Step 7 (Steps 7 and 8 should be followed after both the Tendenci and Python upgrades):

cd /var/www/mysite/
source /srv/mysite/bin/activate
# For development installations:
pip install --no-binary psycopg2 -r requirements/dev.txt --upgrade
# For live sites:
pip install --no-binary psycopg2 -r requirements/prod.txt --upgrade

5.3 Step 3: Upgrade Tendenci

Make a backup of /var/www/mysite/:

cd /var/www/
cp -a mysite mysite.t7

Upgrade files in /var/www/mysite/:

sudo chown -Rh "$\{id -u -n\}:www-data" /var/www/mysite/
cd /var/www/mysite/
rm manage.py deploy.py README.md requirements.txt
rm conf/wsgi.py requirements/tendenci.txt requirements/prod.txt
rm -rf index.whoosh
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/
  manage.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/
  wsgi.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/
  README.md
cd conf/
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/
  wsgi.py
cd ../requirements/
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/
  requirements/tendenci.txt
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/
  requirements/prod.txt
cd..
mkdir whoosh_index
chgrp www-data whoosh_index
chmod g+w whoosh_index

Upgrade Tendenci and its dependencies:
5.4 Step 4: Upgrade Configuration

Tendenci 11.0 allows for the use of a simplified configuration which should reduce the number of configuration changes that will be required during future upgrades. Upgrading to this new configuration is recommended but optional.

To switch to the new configuration, continue with Step 4a. To continue using your existing configuration, skip to Step 4b.

5.4.1 Step 4a: Upgrade to New Configuration

The old configuration files came pre-configured with a number of standard settings (for example, they included standard INSTALLED_APPS, urlpatterns, CACHES, a number of *_PATH settings, etc). Tendenci now configures all of those standard settings by default, so the new settings.py and urls.py files only need to contain site-specific changes to the default configuration. Removing the old standard settings from your configuration will simplify your configuration, make your site specific changes easier to identify, understand, and track, and eliminate any need to update the standard settings during future Tendenci upgrades.

Make sure you have a backup copy of your old configuration. If you followed Step 3, there should be a copy in /var/www/tendenci.t7/conf/

Remove the old configuration and replace it with new configuration files:

```
cd /var/www/mysite/conf/
rm *
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/__init__.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/wsgi.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/settings.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/urls.py
```

Copy any site specific settings from /var/www/tendenci.t7/conf/local_settings.py to /var/www/tendenci/conf/settings.py. The new settings.py includes comments which explain how/where to configure most of the settings you are likely to need to change. Avoid copying any standard/default settings. However, be sure to copy:

- SECRET_KEY
- SITE_SETTINGS_KEY
- The settings in DATABASES
- TIME_ZONE
• Payment Gateway Settings

• EMail Settings

Note that the length recommendations for SECRET_KEY and SITE_SETTINGS_KEY have changed. Old keys will continue to work, but you should consider changing your keys to meet the new guidelines. If you change SECRET_KEY, then all currently logged in users will be logged out, and all active password reset tokens will be invalidated. If you change SITE_SETTINGS_KEY, then all settings on the “Site Settings” and “Full Settings” pages in Tendenci will be reset. Since SECRET_KEY is a critical component of the Django security model, and since changing SECRET_KEY is relatively non-disruptive, we strongly recommend changing SECRET_KEY to a new value that meets the new recommendations. However, since changing SITE_SETTINGS_KEY is relatively disruptive, and since it is a less critical component of the security model, we recommend that you do not change your SITE_SETTINGS_KEY at this time.

If you have made site specific changes to INSTALLED_APPS, Tendenci now adds the following apps to INSTALLED_APPS by default, so your INSTALLED_APPS += [...] setting no longer needs to include them:

• django.contrib.gis
• tendenci.apps.committees
• tendenci.apps.case_studies
• tendenci.apps.donations
• tendenci.apps.speakers
• tendenci.apps.staff
• tendenci.apps.studygroups
• tendenci.apps.videos
• tendenci.apps.testimonials
• tendenci.apps.social_services
• tendenci.apps.explorer_extensions
• explorer

If you wish to remove the above apps from the default INSTALLED_APPS, instructions can be found in the “Custom Application Settings” section of the new settings.py.

Next, review the comments and examples in settings.py to determine if there are any new settings that should be configured for your site. In particular, ALLOWED_HOSTS and the settings under “HTTPS and Session Settings” should be configured appropriately even if you have not previously configured them. Also note that Tendenci now enables logging by default. Review the “Logging Settings” section of the new settings.py to determine whether you need to make any site specific changes to the default logging configuration.

Copy any site specific URL patterns from /var/www/tendenci.t7/conf/local_urls.py to /var/www/mysite/conf/urls.py. Note that URL pattern in the new file must be wrapped with a url() function call.

As with INSTALLED_APPS above, Tendenci now includes URL patterns for the following apps by default, so your configuration no longer need to include them:

• explorer.urls
• tendenci.apps.explorer_extensions.urls
• tendenci.apps.committees.urls
• tendenci.apps.case_studies.urls
• tendenci.apps.donations.urls
• tendenci.apps.speakers.urls
• tendenci.apps.staff.urls
• tendenci.apps.studygroups.urls
• tendenci.apps.videos.urls
• tendenci.apps.testimonials.urls
• tendenci.apps.social_services.urls

If you wish to remove the URL patterns for the above apps from the default configuration, instructions can be found in the new urls.py file.

If you made any changes to the new urls.py file, be sure to uncomment `ROOT_URLCONF = 'conf.urls'` in the new settings.py file.

If you are using NGINX, edit `/etc/nginx/sites-available/tendenci` and change location
~ /themes/([a-zA-Z0-9\-\_]+)/media/ { to location ~ /themes/([a-zA-Z0-9\-\_]+)/ (media|static)/ {

Skip to Step 5 to continue.

5.4.2 Step 4B: Update Existing Configuration

If you are using NGINX, edit `/etc/nginx/sites-available/tendenci` and change location
~ /themes/([a-zA-Z0-9\-\_]+)/media/ { to location ~ /themes/([a-zA-Z0-9\-\_]+)/ (media|static)/ {

Edit `/var/www/tendenci/conf/local_settings.py`.

At the top of the file, you should see the following lines:

```python
def get_setting(setting):
    import settings
    return getattr(settings, setting)
```

Replace those lines with:

```python
def get_setting(setting):
    try:
        from . import settings # Python 3
    except ImportError:
        import settings # Python 2
    return getattr(settings, setting)
```

Then remove the following from INSTALLED_APPS:

• django.contrib.gis
• tendenci.apps.committees
• tendenci.apps.case_studies
• tendenci.apps.donations
• tendenci.apps.speakers
• tendenci.apps.staff
• tendenci.apps.studygroups
• tendenci.apps.videos

5.4. Step 4: Upgrade Configuration
• tendenci.apps.testimonials
• tendenci.apps.social_services
• tendenci.apps.explorer_extensions
• explorer

The above apps are now included in INSTALLED_APPS by default. If you wish to remove them from the default INSTALLED_APPS, you can add `INSTALLED_APPS.remove('tendenci.apps.name')` lines as shown in the “Custom Application Settings” section of the latest settings.py file.

Replace `django.core.cache.backends.memcached.MemcachedCache` with `django.core.cache.backends.memcached.PyLibMCCache`

If present, replace `tendenci.apps.theme.template_loaders.Loader` with `tendenci.apps.theme.template_loaders.ThemeLoader`

Tendenci now enables logging by default. Review the “Logging Settings” section of the latest settings.py file to determine whether you need to make any site specific changes to the default logging configuration.

Edit `/var/www/mysite/conf/settings.py`

Change from `local_settings` import * to from `.local_settings` import *

Delete INSTALLED_APPS += ('gunicorn',)

Change MIDDLEWARE_CLASSES to MIDDLEWARE

In HAYSTACK_CONNECTIONS, change 'PATH': os.path.join(PROJECT_ROOT, 'index.whoosh'), to 'PATH': os.path.join(PROJECT_ROOT, 'whoosh_index', 'main'),

Delete the if DEBUG_TOOLBAR_INSTALLED: block near the bottom of the file.

Replace the old urls.py file:

```
cd /var/www/mysite/conf/
rm urls.py
wget https://raw.githubusercontent.com/tendenci/tendenci-project-template/master/conf/→urls.py
```

Copy any site specific URL patterns from local_urls.py to urls.py, ignoring the following:

• explorer.urls
• tendenci.apps.explorer_extensions.urls
• tendenci.apps.committees.urls
• tendenci.apps.case_studies.urls
• tendenci.apps.donations.urls
• tendenci.apps.speakers.urls
• tendenci.apps.staff.urls
• tendenci.apps.studygroups.urls
• tendenci.apps.videos.urls
• tendenci.apps.testimonials.urls
• tendenci.apps.social_services.urls
The above URL patterns are now included by default. If you wish to remove the URL patterns for the above apps from the default configuration, instructions can be found in the new urls.py file.

Note that URL patterns in the new file must be wrapped with a `url()` function call.

When finished, you can delete your old local_urls.py file.

### 5.5 Step 5: Configure logging

If you are using the new default logging configuration, you should set up `/var/log/mysite` and configure logrotate appropriately.

**Set up `/var/log/mysite`:**

```bash
sudo mkdir /var/log/mysite
sudo chown www-data:$(id -u -n) /var/log/mysite/
chmod -R g+rwx /var/log/mysite/
```

The group configured above will enable your user account to run `python manage.py ...` without sudo, which is safer than performing management/upgrades using sudo. The configured group should include your normal user account but not other inappropriate users. Ubuntu creates a dedicated group for each user by default, so that is what is used here.

**Create `/etc/logrotate.d/mysite` containing:**

```bash
/var/log/mysite/*.log {
    daily
    minsize 100k
    missingok
    rotate 14
    compress
    create 0660 www-data www-data
    sharedscripts
    postrotate
        service tendenci restart
    endscript
}
```

To ensure that the log files remain readable/writable by your normal user account, change the second `www-data` in `create 0660 www-data www-data` to a group that includes your normal user account but not other inappropriate users (Ubuntu creates a dedicated group for each user by default with the same name as the associated user, so you can use that here).

### 5.6 Step 6: Upgrade Database and Static Files

Configure some settings for the database account used by Tendenci:

```bash
DB_USER=tendenci
sudo -u postgres psql -c "ALTER ROLE $DB_USER SET client_encoding TO 'UTF8';"
sudo -u postgres psql -c "ALTER ROLE $DB_USER SET default_transaction_isolation TO 'read committed';"
```

Work around some migration issues in djkombu:
Run:

cd /var/www/mysite/
source /srv/mysite/bin/activate
python manage.py migrate
python manage.py deploy
python manage.py clear_cache

Tendenci 11 makes significant changes to the layout of /var/www/mysite/static/ and may leave lots of abandoned files in that directory after the upgrade. You can optionally clean these up using python manage.py collectstatic --clear --link --no-input. However note that this will delete ALL of the files in that directory and then re-populate them from the installed Django apps. If you have manually added or modified any files in that directory (this is unsupported and discouraged, although there is nothing technically preventing you from doing it), then your files/changes will be lost. Make sure you have a backup copy of that directory before attempting to clean it up. If you followed Step 3, there should be a copy in /var/www/mysite.t7/static/

5.7 Step 7: Update/Fix Permissions

Ensure that filesystem permissions are set appropriately:

chmod -R o+rX-w /srv/mysite/
sudo chgrp -Rh www-data /var/www/mysite/
chmod -R -x+X,g-w,o-rwx /var/www/mysite/
chmod -R ug-x+rwX,o-rwx /var/www/mysite/media/ /var/www/mysite/whoosh_index/ /var/www/˓→mysite/themes/
sudo chown -Rh www-data:"$(id -u -n)" /var/log/mysite/
sudo chmod -R -x+X,g+rw,o-rwx /var/log/mysite/

For an explanation of these permissions, see the “Permissions” section in the latest version of the Installation instructions.

5.8 Step 8: Rebuild Search Indexes

Run:

sudo -u www-data -H -s
cd /var/www/mysite/
source /srv/mysite/venv/bin/activate # For Python 2
source /srv/mysite/bin/activate # For Python 3
python manage.py rebuild_index

5.9 Step 9: Restart Tendenci

Run:

sudo service mysite restart
If you want to migrate your site from T5 to the latest version of tendenci T7, you’d need to take two steps because of the change in the way django switched database migration methods moving to django 1.8. For Tendenci this means you must:

1) Migrate from T5 to T6
2) Migrate from T6 to T7

In this guide, we’ll show you how to migrate from T5 to T6.

In general, to migrate your site from T5 to T6, you’d need to set up a T6 site, update the theme to fit your needs, then move the database and media files from your T5 site to the new T6 site.

### 6.1 1. Set up a T6 site

The instruction [here](#) can help you to get your T6 site set up.

You might also need to look at this [guide](#) for information on server setup.

Note: Please write down your T6’s current theme name, you will need it in the last step. The current theme name can be found in the theme editor.

### 6.2 2. Dump database and zip the media files on your T5 site

```bash
mkdir /tmp/<site_name> && cd /tmp/<site_name>

sudo -u postgres pg_dump <db_name> --file=<db_name>.dump --host=localhost --username=<db_user>

tar -C <path-to-site>/media -czvf <site_name>_media.tar.gz .
```

If your T5 and T6 sites are not on the same server, you’ll need to move (with scp) the database dump and media .gz file to your T6 server.
6.3 3. Restore database on your T6 site

If your T5 and T6 sites are on the same server, skip this step.

```bash
# create and restore database
sudo -u postgres psql
CREATE USER <db_user> WITH PASSWORD '<db_password>';  
CREATE DATABASE <db_name> WITH OWNER <db_user>;
GRANT ALL PRIVILEGES ON DATABASE <db_name> TO <db_user>;
# type '\q' to quit psql
sudo -u postgres psql -d <db_name> --file=<db_name>.dump
sudo -u postgres psql -d <db_name> -c "CREATE EXTENSION postgis;"

# restore media files
 tar -C <path-to-site>/media -xzvf /tmp/<site_name>/<site_name>_media.tar.gz
```

After that, update the DATABASES setting in the conf/local_settings.py.

6.4 4. Copy over site specific settings from T5 .env and conf/local_settings.py

It can include but may not limit to the following:

- SECRET_KEY
- SITE_SETTINGS_KEY
- AWS_ACCESS_KEY_ID
- AWS_SECRET_ACCESS_KEY
- DEFAULT_FROM_EMAIL
- # payment gateway settings
- # newsletters settings

6.5 5. Migrate database

```bash
python manage.py clear_cache
python manage.py syncdb
python manage.py migrate

# We had to run the following commands. But your mileage may vary.
python manage.py migrate social_auth 0001 --fake
python manage.py migrate wp_exporter 0001 --fake
python manage.py migrate djcelery 0001 --fake
python manage.py migrate --delete-ghost-migrations
python manage.py migrate user_groups 0012 --fake
python manage.py migrate social_auth 0001 --fake
python manage.py migrate wp_exporter 0001 --fake
python manage.py migrate djcelery 0001 --fake
python manage.py migrate wp_importer 0001 --fake
python manage.py migrate wp_importer 0001 --fake
```
6.6 6. Run deploy.py

```python
deploy.py
```

6.7 7. Set theme

Use the theme name you wrote down in step 1.

```python
manage.py set_theme <your current theme>
```

6.8 8. Clear caches

```python
manage.py clear_cache
manage.py clear_theme_cache
```

Sometimes things can be bumpy depending on your database’s state. Stay cool and try to debug. Seek help if you’re not able to resolve yourself.

Finally, restart your new T6 site.

**Note:** If you want to copy over the theme from your T5 site to T6/T7:

1) Make sure the theme is bootstrap 3 compatible.

2) Update templates to replace `{% block body %}` with `{% block content %}`.

3) Update templates to add quotes to the first argument of url tags. For example, if you have url tag `{% url admin:boxes_box_change box.pk %}`, it needs to be updated to `{% url 'admin:boxes_box_change' box.pk %}`.
Upgrade 6.x to 7.1x

UPDATED: It is now for migrating from T6 to T7.1 (instead of T7.0).

7.1 Step 1: Back up your site and database

If you’re using docker, just backup your docker container:

```
docker commit <site_name> backup/<site_name>:<your tag>
```

7.2 Step 2: Update some files

These 4 files need to be updated:
- conf/settings.py
- conf/local_settings.py
- conf/local_urls.py
- deploy.py

**conf/settings.py:**

Replace:

```
TEMPLATE_DIRS += (os.path.join(PROJECT_ROOT, "themes"),)
```

With:

```
TEMPLATES[0]["DIRS"] += (os.path.join(PROJECT_ROOT, "themes"),)
```

Replace:
TEMPLATE_CONTEXT_PROCESSORS += (
    'django.core.context_processors.static',
    'tendenci.apps.base.context_processors.newrelic',
)

With:

TEMPLATES[0]['OPTIONS']['context_processors'] += (
    'django.core.context_processors.static',
    'tendenci.apps.base.context_processors.newrelic',
)

Also, check your addons to make sure they are T7 compatible. If they are not, comment out these two lines.

```
from tendenci.apps.registry.utils import update_addons
INSTALLED_APPS = update_addons(INSTALLED_APPS, SITE_ADDONS_PATH)
```

conf/local_settings.py:

Replace:

```
'OPTIONS': {'autocommit': True},
```

With:

```
'autocommit': True,
```

Remove:

```
TEMPLATE_CONTEXT_PROCESSORS = get_setting('TEMPLATE_CONTEXT_PROCESSORS')
```

Replace:

```
'committees',
'case_studies',
'donations',
'speakers',
'staff',
'studygroups',
'videos',
'testimonials',

With:

```
'tendenci.apps.committees',
'tendenci.apps.case_studies',
'tendenci.apps.donations',
'tendenci.apps.speakers',
'tendenci.apps.staff',
'tendenci.apps.studygroups',
'tendenci.apps.videos',
'tendenci.apps.testimonials',
'tendenci.apps.social_services',
```

conf/local_urls.py

Replace:
deploy.py

Replace with the content of https://github.com/tendenci/tendenci-project-template/blob/master/deploy.py

7.3 Step 3: Update files in requirements directory

Add a file `tendenci.txt` and replace all files with the content in the directory https://github.com/tendenci/tendenci-project-template/tree/master/requirements

7.4 Step 4: Remove south and common addons modules from environment

```
pip uninstall -y South
pip uninstall -y tendenci-case-studies
pip uninstall -y tendenci-committees
pip uninstall -y tendenci-donations
pip uninstall -y tendenci-speakers
pip uninstall -y tendenci-staff
pip uninstall -y tendenci-studygroups
pip uninstall -y tendenci-videos
```

Note: For some reason, we were not able to make a clean install in docker containers. We ended up rebuilding the dist-packages.

```
rm -rf /usr/local/lib/python2.7/dist-packages
mkdir /usr/local/lib/python2.7/dist-packages
chmod g+w /usr/local/lib/python2.7/dist-packages
curl https://bootstrap.pypa.io/get-pip.py | sudo python
```
7.5 Step 5: Install requirements

```
pip install -r requirements.txt --upgrade
```

**Note:** If you encountered the error “bash: /usr/bin/pip: No such file or directory” or similar, please exit and re-enter the container then try again.

7.6 Step 5: Fake initials

```
python manage.py fake_initials
```

**Note:** If you encountered the error “no module ‘testimonials’ is installed” or similar, please check if ‘testimonials’ is removed from `INSTALLED_APPS` in `conf/local_settings.py`.

7.7 Step 6: Run deploy

```bash
# If you encountered an error saying `salutation` already exists,
# you'll need to fake events 0003 `python manage.py migrate events 0003 --fake`
# then rerun the command `python manage.py migrate`.
# You'll also need to fake files 0002 `python manage.py migrate files 0002 --fake`
python manage.py migrate
python deploy.py
```

If everything goes well, restart your site.
Upgrade 7.0 to 7.1

We’ve refactored tendenci by pulling the common addons into the core. As a result, T7.1 is not backwards compatible with T7.0. This guide helps you migrate from T7.0 to T7.1.

8.1 Step 1: Update 2 files in conf/

- conf/local_settings.py
- conf/local_urls.py

conf/local_settings.py:

Replace:

```python
['committees',
 'case_studies',
 'donations',
 'speakers',
 'staff',
 'studygroups',
 'videos',
 'testimonials',
```

With:

```python
['tendenci.apps.committees',
 'tendenci.apps.case_studies',
 'tendenci.apps.donations',
 'tendenci.apps.speakers',
 'tendenci.apps.staff',
 'tendenci.apps.studygroups',
 'tendenci.apps.videos',
 'tendenci.apps.testimonials',
 'tendenci.apps.social_services',
```
conf/local_urls.py

Replace:

```python
('^', include('committees.urls')),
('^', include('case_studies.urls')),
('^', include('donations.urls')),
('^', include('speakers.urls')),
('^', include('staff.urls')),
('^', include('studygroups.urls')),
('^', include('videos.urls')),
('^', include('testimonials.urls')),
```

With:

```python
('^', include('tendenci.apps.committees.urls')),
('^', include('tendenci.apps.case_studies.urls')),
('^', include('tendenci.apps.donations.urls')),
('^', include('tendenci.apps.speakers.urls')),
('^', include('tendenci.apps.staff.urls')),
('^', include('tendenci.apps.studygroups.urls')),
('^', include('tendenci.apps.videos.urls')),
('^', include('tendenci.apps.testimonials.urls')),
('^', include('tendenci.apps.social_services.urls')),
```

8.2 Step 2: Update files in requirements/ directory

Add a file `tendenci.txt` and replace all files with the content in the directory https://github.com/tendenci/tendenci-project-template/tree/master/requirements

8.3 Step 3: Remove common addons modules from environment

```bash
pip uninstall -y tendenci-case-studies
pip uninstall -y tendenci-committees
pip uninstall -y tendenci-donations
pip uninstall -y tendenci-speakers
pip uninstall -y tendenci-staff
pip uninstall -y tendenci-studygroups
pip uninstall -y tendenci-videos
```

8.4 Step 4: Install requirements

```bash
pip install -r requirements.txt --upgrade
```

Note: If you encountered the error “no module ‘testimonials’ is installed” or similar, please check if ‘testimonials’ is removed from `INSTALLED_APPS` in `conf/local_settings.py`. If you encountered the error “bash: /usr/bin/pip: No such file or directory” or similar, please exit and re-enter the container.
8.5 Step 5: Run deploy

```
python deploy.py
```

If everything goes well, restart your site.

8.6 Troubleshooting Migrations

If you hit any errors on the migrations while running deploy.py, swbug it from the root of your python app with:

```
python manage.py showmigrations
```

Make a note of any of the migrations that are missing an “x” in the brackets. In this example for some reason the django migrations skipped the third migration in events.

```
events
[X] 0001_initial
[X] 0002_auto_20150804_1545
[ ] 0003_registrant_salutation
```

which was resolved by faking the third migration in events like this:

```
python manage.py migrate events 0003 --fake
```

Of course realize that if your database structure is off and you fake that migration it won’t update your database table and will probably cause NGINX to just deliver of 502 error without logging anything helpful.
9.1 Philosophy

This idea of the checklist is to confirm that the site has been set up correctly. This is NOT a to-do list to build a Tendenci site. If you are hitting more than one or two errors, stop, and push it back to your developers and project manager. This is for quality control only.

For developers you will note that many of these tests can be done using unit tests or selenium which is a far better method than having a human verify these things. We can look at this as a list of tests that need to be written for Tendenci while acknowledging there will always be a human element to quality control.

9.2 Basic Functionality Testing

1. Newsletter Sending has been tested
   • Instead of the settings for the regular emails, the newsletter has a separate set of the settings. In your site’s conf/settings.py, make sure you have the following settings set up.

```
NEWSLETTER_EMAIL_HOST = '<email-host>'
NEWSLETTER_EMAIL_PORT = <email-port>
NEWSLETTER_EMAIL_HOST_USER = '<email-host-user>'
NEWSLETTER_EMAIL_HOST_PASSWORD = '<email-host-password>'
```

   • Then, set up a test group. Send a newsletter. Make sure content is coming through. Has footer content been customized?

2. Email sending from site is tested and sender address is client’s email
   • Do a password reset and confirm you received the email

3. Redirects have been tested
   • If they are using /redirects - click links in the list. Try in a new browser window when you are specifically NOT logged in.
• Note the redirects module only redirects within the same site.

4. 404 page has search bar (Debug=True is off)
• Test the reflex. Make up a page that doesn’t exist example: (/blah/blah) and see what the result is. Note that example link has two slashes to see if the images break.

5. The security, 403 page is working properly
• Not logged in try to access a page that requires you to be logged in. Make sure you are prompted to login.

6. Merchant account has been tested
• Donation form is a good way to test /donations or use /py

### 9.3 Site Content Review

1. Verify homepage is not pulling in demo/blank content. An example might be an auto feed from the photos module but no photos have been added.
• twitter feed - verify it is using latest twitter embed code
• facebook feed - *make sure* it is using latest FB embed iframe method or the site will run very slow waiting on FB

2. Favicon and apple-touch-icon.png are logo

3. Check Title Tag Spelling and Unique Meta and keywords on homepage.
• Check “Site Display Name”, “Site Primary Keywords”, and “Site Secondary Keywords” first. These are located at /settings/site/global/
• Do a view source to see what is pulling in for <title></title>.
• Check for duplicate wording.
• *Danger* You can override dynamic title tags in the theme editor: /theme-editor/?file=/templates/homepage.html but this is a terrible idea. Instead check your base.html file in theme, that homepage.html is extending from base. Then check that your homepage.html is doing a block.super either at the front or back of the title tag block in your homepage.

4. Page titles are H1’s on interior pages and no duplicate H1s on homepage or interior page

### 9.4 Contact and Custom Forms

1. Verify the official “contact” form is using the “contacts” module and not the custom forms module. There are many technical reasons for this beyond the scope of a checklist.
• Default Contact form has an email address in the “Send Copies To” field.

2. Custom forms - a necessary evil. Verify all forms if used on the site are in “published” status
• Logout. Make Sure you still see them. Submit form to test.
• Test if admin receives notification as well.
• Make sure “Send Copies To” field is filled out with an email address
• Verify completion URL and/or submission text is updated to make sense for the site (no “lorem ipsum” etc)
9.5 Edit Default Data to Make Sense

To Repeat - this should already have been done because this is a QA checklist

1. Unused Applications disabled in /admin/
   - Check if default users have email address ending in @<YOUR DOMAIN NAME>. Maybe delete them but be careful for deleting relational data.
   - Company listed as <WHATEVER YOUR ORGANIZATION NAME IS>
   - Edit, don’t delete, default user groups. warning Edit/Don’t Delete.
   - Test pages edited to be relevant or possibly removed or set to draft where appropriate. (Recommended - Please check with project manager before deleting)

2. Test invoices have been voided

3. Images are using alt tags throughout the site

4. Site has been checked in an html validator (https://www.onlinewebcheck.com/check.php) and on http://www.site-analyzer.com

5. Optional: Test site speed (http://tools.pingdom.com/fpt/)

6. H2, H3 etc. subtitle tags are in place across the site

9.6 Site Settings

1. Site URL field contains an url that starts with “www”
   - The canonical name must NOT be the apex/naked url. (This is BAD -> “example.org” and so very 1990s. Do NOT do this.)
   - Do an nslookup and be sure apex “example.org” forwards to “www.example.org”
   - Do an nslookup to confirm www.example.org is using a CNAME to a host and not the IP address. Verify your site is future proof.

2. Site name is correct, spelled out and not abbreviated if possible.
   - there are exceptions but generally this is best practice.
   - Example: “California Dog Walking Association”

3. Physical address is the actual physical address.
   - Required for email delivery by law.
   - Should match registered corporate or 501c address for SSL if possible

4. Location setting is correct (and Time Zone).
   - this is important for reminders for events and merchant approvals

5. Email - CRITICAL
   - Appropriate names and emails are set for administrator, contact, webmaster in site settings.
   - Default send email is set up in local_settings.py
   - Mail relay must have MX, SPF, DKIM records in DNS or email will not go through.
   - Domain must be verified through SES, or Mailgun or whatever your email relay server is.
9.7 Search Engine Visibility Site Settings

1. HTML Sitemap is linked in the footer and is functioning properly
2. RSS feed is referenced in HTML header (view source)
3. check /robots.txt
   • if it is set to disallow all, update your site settings to make it public.
   • Make sure robots.txt links to your sitemap.xml file which is generated dynamically as well
4. check /sitemap.xml to be sure it is updating and can be crawled by Google.
5. Site Primary Keywords are set up
   • *MUST NOT* be the organization’s name
   • Consequence is spammy titles like “NAME - NAME”
   • two to three words max with no commas
   • Example: “Leadership Training”
   • Note that these show up ON PAGES like articles list view so use proper case. They are not just used in metadata.
6. Secondary Keywords are set up
   • Secondary keywords can use two to three commas max
   • Example: “dog walking, california dog walking, walking your dog in LA”
   • Don’t spam - should reflect what your site is about
   • Most modules have seo meta options at the bottom which override these on a per module or per page basis.
7. Google Analytics is set up in site settings.
   • There is a field for the UA- code
   • If using google “tags” put javascript below the <body> tag in “base.html”
   • Direct further questions on this to forums www.tendenci.com/forums/ or an SEO professional

9.8 Cross Browser Site Checking

*Note - it is best to check your site with automated testing tools like Selenium but there is always a human element for formatting*

1. Browse site in
   • Firefox
   • Chrome
   • Safari
   • IE (gah!)
   • iPhones
   • Android Devices
   • iPad
2. Themes for Tendenci are strictly Bootstrap3
• you CAN use other frameworks in conjunction with it
• use DIV IDs to avoid css conflicts if using other frameworks

3. IE - We make little to no effort to support IE more than the current and previous version.
• Example: If the current version of IE is 10, then Tendenci tries to support 10 and 9. Nothing further. (You can of course modify it to support older versions by doing a fork or submitting a pull request.)
• Safari is getting to be more like IE so same policy - current and previous versions are the only ones we check.

9.9 Accessibility

1. Check for Accessibility
   • Web Accessibility Measurement Tool http://wave.webaim.org/
   • Note: The community’s goal is to make Tendenci 100% Accessible.
2. Submit issues if you see something in core that is broken
   • https://github.com/tendenci/tendenci/issues
3. Not broken, but a pain? Submit on the forum.
   • https://www.tendenci.com/forums
4. Talk to your developer if something in your theme is broken

9.10 Security Recommendations

1. Recommended Check if web site is routing through a proxy IDS/IPS service
   • Securi is a low cost IDS/IPS that protects your site
   • Alert Logic has an AMI appliance that you can manage yourself
2. Recommended Set your server to automatically install security updates
3. Verify SSL configuration is correct
   • SSL we consider baseline although not everyone installs it unfortunately.
4. Audit your site settings monthly
   • Frequently what looks like a security issue is a site setting.
   • Audit your site settings at least monthly
5. Run the “administrators report”
   • remove security as needed

9.11 Security Awareness - YOU ARE RESPONSIBLE

1. Check that all content on your site is licensed properly
   • know where the licenses are now and in the future.
• Examples include but are not limited to images, content, video, audio, cat gifs, 3d monsters that jump out of your computer, etc.

2. Tendenci is FOSS (Free/Open Source Software)
• Read the license here https://github.com/tendenci/tendenci/blob/master/LICENSE
• Themes use a variety of licenses such as the Apache Open Source license, etc. Read them.
• Components may use other similar open source licenses which are distributed in the code.

3. YOU ARE RESPONSIBLE to confirm rights for anything uploaded to your site
• Example: a user picks the password “12345678” that is guessed by a competitor and something inappropriate gets uploaded.

4. Backups are your responsibility and strongly recommended.
• Tendenci is not your typical software. This community pushes limits and moves fast. So sometimes things break. Use common sense.
• You can download your database at /explorer/ on your site. If not enabled talk to your developer
• You can create a static backup of your site by following the instructions on this blog post http://blog.tendenci.com/tag/make-a-static-copy-of-your-website/
• You can arrange with your developer to have a scheduled backup of your site pushed to your own S3 bucket on Amazon via SSH.

5. You, or your developer, are responsible to keep the code up to date to be sure you have the latest security patches.

On a production site taking backups and restoring them for testing/recovery purposes is a critical part of site setup and maintenance.

Setting up a server to store your backups on is beyond the scope of this document, so here we cover automatically saving the data of your Tendenci site, plus its restoration.
A backup of Tendenci needs to be a copy of the website and database at a single point in time. For sites with predictable low activity periods a scheduled job to run early in the morning and perform the backup will be sufficient.

### 10.1 Website

Depending on your preferences, files can either be saved to an archive then synced to a backup location or simply copied there as a set of files.

For automated backups, drop a script fragment in `/etc/cron.daily/tendenci-daily-backup.sh` and let cron take care of scheduling.

Archiving before sending off site:

```bash
    tar -cjf /var/tmp/tendenci-nightly-backup.tar.bz2 /path/to/tendenci-website/
    rsync --quiet /var/tmp/tendenci-nightly-backup.tar.bz2 user@remoteserver:/path/to/ →backups/
```

Copying Tendenci directory directly:

```bash
    rsync --delete --quiet /path/to/tendenci-website/ user@remoteserver:/path/to/backup- →tendenci-website
```

### 10.2 Database

As with the website files, this can be compressed or not but note for a moderately large site compression will save considerable amounts of space and network bandwidth.

Backing up the database (with compression) is a single command:

```bash
    sudo -u postgres pg_dump -Fc -f /tmp/tendenci-daily.sql postgres tendenci_db_name
```

Or if compression isn’t desirable:
```bash
sudo -u postgres pg_dump tendenci > /tmp/tendenci-daily.sql
```

These files can then be copied to the backup site:

```bash
rsync --delete --quiet /path/to/tendenci-daily.sql user@remoteserver:/path/to/backup-tendenci-website
```

Take care to keep the website and backup files in sync!
11.1 Database

Restoring is as easy as taking the backup was:

```
rsync user@remoteserver:/path/to/backup-tendenci-website /path/to/tendenci-daily.sql
sudo -u postgres psql tendenci < /path/to/tendenci-daily.sql
```

This should be done AFTER the DB setup steps in the install instructions.

11.2 Website

Restoring the website replaces the instructions on using tendenci-base-template and related install/configuration steps, but does not replace external setup like installing packages from pip or setting up Nginx.

```
rsync user@remoteserver:/path/to/backup-tendenci-website /path/to/tendenci-website/
```
Boxes are specialized areas that are added to your theme. You can add HTML content to a box via a WYSIWYG. Boxes can be included based on their ID or based on tags.

To use boxes, they must be added to your theme.

### 12.1 Adding Boxes to your Theme

Boxes can be embedded into the theme with only a couple lines of code.

```latex
{% load boxes_tags %}
{% block html_body %}
  ...
  {% box 1 %}
  ...
{% endblock html_body %}
```

Boxes are referenced by their ID, which is visible in the admin area list of boxes.
Directories are listings for businesses or other services on your website. You can allow Directories to be purchased to be displayed for a preset amount of time.

Based on different settings, you can allow logged in users or members of your site to post Directories, which will be added to a pending queue for administrator approval.

13.1 Directory Pricing

Directories can have different price options for posting based on the length of time they are posted.
Setting up discount codes is a quick and painless process which can be applied to several apps within Tendenci.

14.1 Enable discounts

Discounts come enabled by default, but check at https://example.com/settings/module/discounts/#id_enabled to ensure they are current set to ‘enabled’.

14.2 Configure discountable apps

Next configure the apps you would like to have discounts apply to - some examples below.

14.2.1 Memberships

Make sure the “Discount eligible” is ticked on the relevant memberships applications (ex: https://www.example.com/admin/memberships/membershipapp/1/).

14.2.2 Events

To find the discount option tick ‘Enable Registration’ and make sure the “Discount eligible” is ticked when creating an Event (ex: https://demo.tendenci.com/events/add/).

14.3 Adding discount codes

Once the applications are configured to allow discounts they will appear in the ‘Applications’ list when issuing a discount code via https://www.example.com/discounts/add/. Choose the applications you’d like to apply discounts to when creating the discount code.
Tendenci offers robust Event management and registration capabilities. You can add events to your website to be displayed on a calendar or a list view.

Events can allow for registration. Different pricing options can be set for different levels of registration. For example, you may have

- Early Bird Rate
- Member Rate
- Member Table Rate (multiple registrants included)
- At-the-door Rate

All of these pricing options (and any others you might think up), can be configured to have a start and end time, and can be made only available members of your site or possibly only people in certain groups.

### 15.1 Adding an Event

#### 15.1.1 Basic Event details

These are things like start and end times.

#### 15.1.2 Speakers

You can add one or many speakers to be listed on your event page. Speakers include a name, photo, and bio.

#### 15.1.3 Registration options

You can configure how many people can register for the event. Start and end times for registration are controlled in the pricings.
15.1.4 Event Registration Pricing

Different pricing options can be set for different levels of registration. For example, you may have

- Early Bird Rate
- Member Rate
- Member Table Rate (multiple registrants included)
- At-the-door Rate

All of these pricing options (and any others you might think up), can be configured to have a start and end time, and can be made only available members of your site or possibly only people in certain groups.
1. The membership module handles a person’s membership. It allows people to become a member of your website and with that membership may come special privileges. Such as discounts on attending an event or when posting a job.

2. Memberships are capable of expiring and people may be allowed to renew their membership and regain their privileges.

3. Every time someone becomes a member, a new membership record is created and this set of memberships is recorded. A new membership record can be created by joining the organization for the first time or renewing your membership.

4. This allows you to view the history of a person and the length of their membership with an organization.

5. It also helps determine gaps in a person’s membership with the organization and how much they’ve contributed to the organization through membership fees.

### 16.1 Membership

1. A membership record is an agreement between a user and your website. This agreement may be offered at a price or for free. Having a membership record on your website means that this person is now considered a “member.”

2. A membership record mostly tells your website that this person is a “member.” A membership record could also hold custom information about that member.

3. All members have a membership of a specific type. In fact a person can have multiple memberships all of different types. A person cannot have multiple memberships of the same “membership type.”

### 16.2 Membership Type

1. A membership type holds meta information about a membership record.
1. The length of a membership (e.g. expiration date/time, renewal period)
2. Prices; join, renew, and admin fees
3. Used for grouping and managing of members

2. Membership types are powerful because they allow you to group members into specific sets and target these groups specifically. You can offer discounts or invite them to privileged information or events. As well as send targeted emails to these specific types of members.

### 16.3 Membership Application

1. A membership application is literally the form that someone fills out in order to become a member or to renew their membership.
2. A membership application collects basic information about a person such as who they are, where they work, and contact information. It can also collect custom information.
3. Common information is collected in the person’s profile.
4. Custom information is located within the membership record.
5. All information can be exported.
6. Detailed reports and graphs are created from profile information. Detailed reports and graphs are not created from custom information. Because custom information is custom, it’s difficult for us to know how to report on that information.
7. Your website can have as many applications as you would like.
8. Visibility of an application can be limited to any group or type of person.
9. An application can allow you to sign up for one or many membership types.
10. You decide the different ways of paying for a membership.
11. You can fill out the application for someone else and have them receive the confirmation.

### 16.4 Notices

Notices are predetermined emails sent to members or membership applicants. They can be sent out based on the following criteria:

- # if days
- before, after, or at time of
- Join, Renewal, Expiration, Approval, and Disapproval dates

The Join date is set when they apply. You may want to set up notices to go out:

- 7 days before expiration to remind members to renew
- At time of (0 days) Approval, letting them know they have been accepted
- 15 days after Approval, again touching base and thanking them for joining
- 7 days after expiration, to see if they still want to renew (usually they can based on the Membership Type grace period)
Notices cannot be sent manually. Notices are more geared towards information related to a single member (their join date, their expiration date) and not really members on a whole. If several members are expiring at once, they will all receive an individual notice with information relevant to them. There is a list of available “tokens” that can be used to in the Notice Email content that are replaced with info relevant to that person.
News

The News app can be displayed in multiple ways on Tendenci through: News Feed, News Search, and News Article Views.

To learn more about adding a News Feed to different pages, go to: http://tendenci.readthedocs.io/en/latest/templating/theme_code_snippets.html#news-feed-with-thumbnail-image

To change the look of the News Search Results page, pull down these two templates: search-result.html and search.html

The overall page layout is dictated by search.html. The look of the news results can be changed through the search-result.html template.

17.1 Social Sharing Links

You can add social sharing links to the New’s module meta.html or view.html to allow users to share the specific news articles they are viewing on your site.

Facebook

```html
<a class="fa fa-facebook fa-2x" href="https://www.facebook.com/sharer/sharer.php?u={{ SITE_GLOBAL_SITEURL }}{{ news.get_canonical_url }}">&nbsp;</a>
```

Twitter

```html
<a class="fa fa-twitter fa-2x" href="https://twitter.com/home?status={{ SITE_GLOBAL_SITEURL }}{{ news.get_canonical_url }}">&nbsp;</a>
```

Linkedin

```html
<a class="fa fa-linkedin fa-2x" href="https://www.linkedin.com/shareArticle?mini=true&url={{ SITE_GLOBAL_SITEURL }}{{ news.get_canonical_url }}&title={{ news.headline }}&summary=&source="&nbsp;</a>
```
17.2 Adding a Thumbnail

The 1st National Young Energy Professionals’ Law Conference Launches in NOLA

The 1st National Young Energy Professionals’ Law Conference will be held March 31 - April 1, 2017, in New Orleans, Louisiana, and Interface Consulting International, Inc., is excited to be a part of it. Lauren Purcell and Cody Crawford will represent Interface at this event sponsored by the Institute for Energy Law. Attendees from around the country will convene at the Hyatt French Quarter New Orleans to take advantage of networking opportunities, build professional relationships, and gain valuable continuing legal education (CLE) credits. [View Full Release]

Released: March 16th, 2017 12:00 PM

Tip: To add the thumbnail image to search-result.html, use this code:
17.3 Blog Styling: Show Image in the News Article

Showing News List


Pellentesque ultrices est vel dapibus posuere. Ut a diam eget lacus egestas commodo id eu purus. Aliquam sed leo ...

Tip: By default, the Tendenci code in search-result.html might look something like this:

{% blocktrans with s=news.body|striptags|truncatewords:20|safe %}{{ s }}{% endblocktrans %}
To achieve the look of a blog, you can remove “striptags” to reveal images, links, and headings within your news article content:

{% blocktrans with s=news.body|truncatewords:100|safe %}{{ s }}{% endblocktrans %}
Pages are the most basic type of content. The main fields are Title, Content, and URL Path.

You can add tags to pages. You can also select a template for that page. For more about these templates, see *Custom Templates*. 
CHAPTER 19

Recurring Payments

Tendenci supports recurring payments when using Stripe or Authorize.net as the payment gateway. The steps for configuring recurring payments are covered on other pages, but there are some things to note before you can proceed. Tendenci 7.4.1 or greater is required. See the upgrade documentation for more details on updating.

Note: If you want to know how to configure your tendenci.com payments see https://www.tendenci.com/help-files/recurring-payment-help/

19.1 Setting up

Setting up the payment gateway is covered in help file Setting Up Online Payment and your Merchant Provider on a Tendenci Site and auto renew with Stripe is detailed on a separate page.

Tip: If you create a recurring payment but your system is incorrectly configured you will need to delete the recurring payment and create a new entry.

19.2 Updating Payment Methods

The point of recurring payments is to allow members to opt in when performing payments, but this requires one more step be performed.

Depending on if there is already an ‘online’ Payment Method listed at https://example.com/admin/payments/paymentmethod/ (‘online’ means using your merchant account) you will either need to add a new entry, or potentially rename the existing ‘Is online’ payment entry (for example a rename might be required if you changed your merchant from PayPal to Stripe). Membership Applications will update to the new names automatically.
**Warning:** Do not delete your existing ‘Is online’ method and create a new ‘Is online’ method of payment. Deleting the existing method will remove all memberships that were paid using that method.

Help on changing your Payment Methods can be done using help for the memberships app already available on the Tendenci site, including a video for a more visual guide if you prefer.

### 19.3 Testing

Stripe and Authorize.net manage their test systems differently. For Stripe, the procedure is to use a special test only access key while Authorize.net use different API endpoints. Be sure you adjust your settings appropriately for the endpoint you have chosen!
Stories are a great way to add dynamic content around your site. Stories can be used for:

- Homepage rotators/slideshows
- Sponsors rotators
- Featured content

A story consists of the following basic fields:

- Title
- Description
- Image
- Link (internal or external)

Stories can be set to activate and expire based on a start and end date. Stories may also be set to never expire.
In the *Pages* app on your Tendenci site, you can select a custom template to use to display your page. This is useful when constructing pages for special occasions or for creating targeted landing pages for marketing efforts.

The first step is to create the custom template. Custom templates replace the `default.html` template when a page is loaded. Copy that template and rename it as `default-then your custom template name, using -'s instead of spaces or underscores. An example would be `default-blue-background.html` or `default-google-landing-page.html`. Your custom template can include template code to dynamically pull in content from other apps, as well as rearranging the elements on the page or removing certain elements. Once your template file is created, add it to the theme directory as you would with other template files.

These templates will appear in the dropdown on the page add and edit screen. Select the template there, and your page will then use it instead of the normal `default.html` template. You can find more information on tags by referring to the auto-generated django docs at: `yourdomain/admin/doc/tags/`.

### 21.1 Best Practices

If your intentions are to only change the content of the page and not the surrounding elements like the header, sidebar, or nav, there is a best practice for utilizing the `default.html` file without having to have multiple places to update when that file changes.

In your `default.html` file, locate the following code:

```html
{% block content %}{% block.super %}{% endblock content %}
```

You can wrap this code in another block that we can use for our custom page. That code would look like this:

```html
{% block custom_body %}
{% block content %}{% block.super %}{% endblock content %}
{% endblock custom_body %}
```

Then, in your new template file, `default-custom-name.html`, we will add the following code from our `pages/view.html` with a couple of changes. We will replace the beginning of the code with `{% extends "pages/base.html" %}` and replace our main block with `{% block custom_body %}.`
Now the custom template will only affect the interior area of a page, so we can create pages that use other Tendenci template tags. This is great for creating pages with rss feeds or other dynamic content like upcoming events and news.

### 21.2 Navigation Bar

#### 21.2.1 Adding a button

Tip: The Navigation Bar is set up through the Navs App.

In the Navs App, Links can be assigned their own CSS Class. This is helpful for targeting individual links to emphasize in your design.
For this example, I added the class “top-nav-button” to the “Attend Orientation” Nav Link.

### 21.2.2 Adding Font-Awesome Icons

**Tip:** Also using the Nav App Classes, you can add Font Awesome Icons to your navigation bar. Just add the Font Awesome tag into the CSS Class and leave the Label blank.

### 21.2.3 Font-Awesome After Nav Items

**Tip:** To add a dropdown caret or other icons to the Tendenci Navbar, use Font Awesome in the css. For this application, I added a css class (“t-caret”) to the nav bar items that I wanted to have a dropdown caret.
21.2.4 Simple Search Bar w/ Font-Awesome Icon

Tip:

21.2.5 Bootstrap Search Bar w/ Font-Awesome Icon

Tip: This example is from the Case Studies module, but can be used in other modules by swapping the action url. There is no added css for this snippet. You can also change the look and feel by adding Bootswatch themes.

21.3 Basic Footer

In Tendenci, the Footer is usually set up inside a box to make it easy for clients to make changes to their links.

Tip: I like to set up my Footer links in a bulleted list. The bullets can be removed later in html or in css. HTML is added inside the Footer links box.
21.4 Embed Forms in Bootstrap Modals

Embed forms in Bootstrap Modals to declutter your pages.

Tip: Don’t forget to add `{% load forms_tags %}` at the top of your template.
This code references W3schools and is for educational purposes only! Original here: https://www.w3schools.com/bootstrap/bootstrap_modal.asp
CHAPTER 22

Theme: Code Snippets for Dynamic Content

You can pull various pieces of dynamic content into your Tendenci theme template. Before adding the snippets below be sure that your template file has the “load” tag for that module at the top of your page. Example: if you were to add an event feed you would want {% load event_tags %} pasted at the very top of your html doc.

For more tag references, check out the docs built into your site at: yourdomain/admin/doc/tags/

22.1 Event Feed

Add {% load event_tags %} to the very top of your html page

Then add the following code snippet where you want to place your event feed:

```html
{% list_events as events_list limit=3 %}
<ul class="list-unstyled">
  {% for event in events_list %}
    <li><a href="{{ event.get_absolute_url }}"><strong>{{ blocktrans with title=event.title %}}{{ event.get_absolute_url }}</strong></a><br>
    <em>{{ event.start_dt|date:"F j, Y - P" }}</em>
  </li>
{% empty %}
  <em>{{ trans 'Events will be shown here. <a href="/events/add/">Start adding events now!</a> </em>
{% endif %}
</ul>

{% if event_list %}
  <a href="{{ url 'event.search' }}"/>{{ trans 'View more events' }} »</a>
{% endif %}
```

22.2 Single Event (featured event)

Add {% load event_tags %} to the very top of your html page
Then add the following code snippet where you want to place your event feed:

```html
{% list_events as events_list limit=1 %}
<section id="single-event">
<h3>Featured Event</h3>
{% if events_list %}
{% for event in events_list %}
<div class="event-info">
  {% if event.image %}
    <img alt="{{ event.title }}" src="{{ image_url event.image size=310x154|crop=true }}"/>
  {% else %}
    <img src="/themes/twenty-thirteen/media/images/box-4.jpg"/>
  {% endif %}
  <p class="event-date">{{ event.start_dt|date:"F d, Y" }}</p>
  <h4><a href="{{ event.get_absolute_url }}">{{ event.title }}</a></h4>
  <p>{{ event.description|striptags|truncatewords:30|safe }}</p>
  <p><span><a class="readmore" href="{{ event.get_absolute_url }}">Read More &raquo;</a></span></p>
</div>
{% endfor %}
{% else %}
  <div class="event-info">
    <img src="/themes/twenty-thirteen/media/images/box-4.jpg"/>
    <h4>Add an Event Today to See Something Here</h4>
    <p class="event-date">{{ now|date:"F d, Y" }}</p>
    <p>This is example of what your event description will look like. Where is this text source?. Start adding Events to start adding your own content today!</p>
    <p>Display your next upcoming event right on your homepage. Once you <a href="/events/add/">add your first event</a>, this area will be replaced by that information. Don't forget to add an event photo!</p>
    <p><span><a class="readmore" href="#">Read More &raquo;</a></span></p>
  </div>
{% endif %}
</section>

22.3 Time Display on Event Feed (hide time if all day)

{% if event.all_day %}{{ event.start_dt|date:"F j, Y" }}{% else %}{{ event.start_dt }}{% endif %}

22.4 Articles Feed

Add {% load article_tags %} to the very top of your html page

Then add the following code snippet where you want to place your articles feed:

```html
{% list_articles as articles_list limit=3 %}
<ul class="list-unstyled">
{% for article in articles_list %}
<li><a href="{{ article.get_absolute_url }}"><strong>{{ article.headline }}</strong> (continues on next page)</a></li>
{% endfor %}
</ul>

(continues on next page)
22.5 Insert Box

Add `{% load box_tags %}` to the very top of your html page

Then add the following code snippet where you want to place your box: (The number 13 should be replaced with the number associated with the box you would like to load)

```html
{% box 13 %}
```

Inside the boxes interface, adding code allows you to add images, add Font-Awesome Icons, and responsive Bootstrap Layouts.
Tip:

22.6 Bootstrap Carousel

Add `{% load story_tags %}` to the very top of your html page

Then add the following code snippet where you want to place your carousel:

```html
{% list_stories as stories_list limit=10 tags="rotator" quality="80" data-interval="5000" %}
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    {% for story in stories_list %}
    <li data-target="#myCarousel" data-slide-to="{{ forloop.counter0 }}" {% if forloop.first %} class="active" {% endif %}></li>
    {% endfor %}
  </ol>
  <!-- begin: carousel-inner -->
  <div class="carousel-inner" role="listbox">
    <!-- begin: item active -->
    {% for story in stories_list %}
    <div class="item {% if forloop.first %} active {% endif %}">
      <img class="img-responsive" src="{{ story.photo.url }}" alt="{{ story.title }}">
      <div class="container-fluid">
        <div class="carousel-caption">
          <h2><a href="{{ story.full_story_link }}">{{ story.title }}</a></h2>
          <p class="story-summary">{{ story.content|safe|truncatewords:12 }}</p>
          <a class="btn btn-lg btn-primary" href="{{ story.full_story_link }}" role="button">Read more</a>
        </div>
      </div>
    </div>
    <!-- end: item -->
    {% endfor %}
  </div>
  <!-- end: carousel-inner -->
</div>
{% endfor %}
```

(continues on next page)
22.7 Directories Carousel

Add \% load directory_tags \% to the very top of your html page

Then add the following code snippet where you want to place your directories:

```html
<div id="directories" class="clearfix">
  <div id="carousel-container">
    <h3>Platinum Corporate Sponsors</h3>
    <div class="prev" id="port-left"></div>
    <div id="carousel-wrap" class="clearfix">
      <div class="carousel-items">
        {% list_directories as directories_list limit=7 tags="featured" %}
        {% for directory in directories_list %}
          <li>
            <a href="{{ directory.get_absolute_url }}"><img src="{{ directory.get_logo_url }}" alt="{{ directory.headline }}"></a><a href="{{ directory.get_absolute_url }}">{{ directory.headline }}</a>
          </li>
        {% endfor %}
      </div><!--end: .carousel-items-->
    </div><!--end: #carousel-wrap-->
    <div class="next" id="port-right"></div>
  </div><!--end: #carousel-container-->
</div><!--end: .col -->
</div><!--end: #directories -->

22.8 Directories (use with flexslider)

Add \% load directory_tags \% to the very top of your html page

Then add the following code snippet where you want to place your directories:

{% list_directories as directories_list limit=20 tags="sponsor" %}
<ul class="slides">
  {% for directory in directories_list %}
    <li>
      <a href="{{ directory.website }}">
```
22.9 Jobs Feed

Add \% load job_tags \% to the very top of your html page

Then add the following code snippet where you want to place your jobs:

```
{% load job_tags %}
{% for job in jobs_list limit=3 %}
  <h3><a href="{{ job.get_absolute_url }}">{{ job.title }} &raquo;</a></h3>
  <p>{{ job.location }}</p>
{% empty %}
  <h4>{% if user.profile.is_superuser %}<a href="/jobs/add/">Add your first job</a>{% endif %}</h4>
  <p>Houston: A new job opening is available...</p>
{% endfor %}
```

Simple Job Feed Layout

Tip:
22.10 Photos Feed

Add {% load photo_tags %} to the very top of your html page

Then add the following code snippet where you want to place your photos:

```html
{% list_photos as photos_list limit=6 random=True %}
<ul>
    {% for photo in photos_list %}
    <li><a href="{{ photo.get_absolute_url }}"><img alt="{{ photo.title }}" src="{{ photo_image_url|photo:72x66|crop:True }}" /></a></li>
    {% endfor %}
</ul>
{% list_photo_sets as photo_sets_list tags="tendencidemo" %}
<ul>
    {% for photo_set in photo_sets_list %}
        {% for photo in photo_set.get_images %}
            <li><a href="{{ photo.get_absolute_url }}"><img alt="{{ photo.title }}" src="{{ photo_image_url|photo:72x66|crop:True }}" /></a></li>
        {% endfor %}
    {% endfor %}
</ul>
```

Drag and drop photo albums to change the order of display on /photos/. To display photos from the top of your list in your feed - as opposed to the most recently added photos - add this order selection to your code:

{% list_photos as photos_list limit=2 order='photoset__position' %}

To pull photos from one or more photo sets, tag your photo set(s) and use list_photo_sets template tag:

```html
{% list_photo_sets as photo_sets_list tags="tendencidemo" %}
<ul>
    {% for photo_set in photo_sets_list %}
        {% for photo in photo_set.get_images %}
            <li><a href="{{ photo.get_absolute_url }}"><img alt="{{ photo.title }}" src="{{ photo_image_url|photo:72x66|crop:True }}" /></a></li>
        {% endfor %}
    {% endfor %}
</ul>
```

22.11 Videos Feed

Add {% load video_tags %} to the very top of your html page Also add {% load video_filters %} to the very top of your html page

Then add the following code snippet where you want to place your videos:

```html
<ul id="video-urls">
    {% list_videos as videos_list limit=1 tags="featured" %}
    {% for video in videos_list %}
        <li> <a href="{{ video.get_absolute_url }}">{{ video|video_embed:294 }} </a></li>
    {% endfor %}
</ul>
```

22.12 News Feed (with Thumbnail Image)

Add {% load news_tags %} to the very top of your html page

Then add the following code snippet where you want to place your news feed:

```html
22.10. Photos Feed
83
```
Here’s an example of what this snippet can look like:

```html
<!-- begin: news-innerblock -->
<div class="news-innerblock">
  {% list_news as news_list limit=3 %}
  <ul class="list-unstyled">
    {% for news_item in news_list %}
    <li>
      <div class="col-md-4 col-sm-12 col-xs-12 text-left">
        <!-- news-image -->
        {% if news_item.thumbnail %}
          <a href="{{ news_item.get_absolute_url }}">
            <div class="news-item-thumbnail">
              <img class="img" src="{{ image_url news_item.thumbnail }}" alt="{{ news_item.headline|striptags|safe }}" style="width:100%;height:auto;"/>
            </div>
          </a>
        {% else %}
          <a href="{{ news_item.get_absolute_url }}">
            <div class="news-item-thumbnail-auto">
              <img src="{{ static 'quince.jpg' }}" style="width:100%;height:auto;"/>
            </div>
          </a>
        {% endif %}
        <!-- end news-image -->
        <a href="{{ news_item.get_absolute_url }}">#</a>
        <h3>{{ blocktrans with h=news_item.headline %}{{ h }}{% endblocktrans %}}</h3>
        <p>{{ blocktrans with c=news_item.summary|striptags|safe|truncatewords:"30" %}{{ c }}{% endblocktrans %}}</p>
      </div>
    </li>
    {% endfor %}
  </ul>
</div>
<!--end: news-innerblock-->
```
Tip: HTML and CSS added to the above image:

22.13 RSS Feed

Add the following code snippet where you want to place your RSS Feed, include only the elements that you can to display on your page. If the RSS feed does not contain the elements you’re calling, they won’t show on the page and may throw an error.

Job 21: Project Manager
Posted on: January 20, 2017

Job 23: Head of Product
Posted on: March 30, 2017

Job 24: Assistant Designer
Posted on: April 2, 2017

View More Jobs

<!-- begin: subscribe-block -->
<div class="col-xs-12">
<h3>Careers</h3>
<ul class="list-unstyled">
{% get_rss "https://example.com" as rss %}
{% for entry in rss.entries|slice:"0:4" %}
<li>
<p class="rss-title"><a target=_blank href="{{ entry.link }}">{{ entry.title }}</a><br /></p>
</li>
{% endfor %}
</ul>
</div>
<!-- end: subscribe-block -->

(continues on next page)
Example 2 shows more of the rss elements that could be included in your site:

```html
{% get_rss "http://rss.nytimes.com/services/xml/rss/nyt/PersonalTech.xml" as rss %}
{% if rss.feed.image %}
    <img src="{{ rss.feed.image.href }}" alt="" />
{% endif %}
{% for entry in rss.entries %}
    <div class="row entry-item">
        <div class="col-xs-4 col-md-3">
            {% if entry.media_content %}
                {% for media in entry.media_content %}
                    {% if media.medium == 'image' %}
                        <img src="{{ media.url }}" width="{{ media.width }}" height="{{ media.height }}" alt="" />
                    {% endif %}
                {% endfor %}
            {% endif %}
        </div>
        <div class="col-xs-8 col-md-9">
            {% title %}
            <h4 class="entry-title"><a href="{{ entry.link }}">{{ entry.title }}</a></h4>
            {% pubdate %}
            <div class="small">Published on: {{ entry.published }}</div>
            {% authors %}
            {% for author in entry.authors %}
                {{ author.name }}
            {% endfor %}
            {% endif %}
        </div>
    </div>
{% endfor %}
```

(continues on next page)
Tip: Other elements you can include:

```html
{{ entry.enclosure }}
{{ entry.pubdate }}
{{ entry.updateddate }}
{{ entry.author_name }}
{{ entry.author_email }}
{{ entry.author_link }}
```
The homepage.html template is rendered for the homepage. The following blocks are available.

### 23.1 Blocks

- **extra_head** Contains meta information, CSS, and JS that needs to be included in the `<head>` element of the document.
- **html_body** Required. Loads in the main `<body>` element of the document.
- **extra_body** Loads on at the end of the document just before `</body>`.
- **body_ids** This pulls in as the id attribute to the `<body>` element. While it is singular, it is recommended to only use one value. A typical value for this is ‘home’.
- **body_classes** Loads in to the class attribute of the `body` element.
- **title** Loads into the `<title>` element in the document.
- **meta_keywords** Loads into the content attribute of the `<meta name="keywords" />` element in the document.
- **meta_description** Loads into the content attribute of the `<meta name="description" />` element in the document.
Tendenci detects mobile browsers and allows you to use that information within your templates. Tendenci will assign a cookie to these visitors. This cookie can be toggled using code below to allow visitors to switch between the mobile and standard versions of a theme.

### 24.1 Mobile Template Folder

A typical Tendenci theme includes a `media` and a `templates` folder. You can add a `mobile` folder at the root of the theme. If a device is detected to be mobile, Tendenci will first look for the relevant template file in the `mobile` folder, then it will move on to the `templates` folder, then to the core software. This allows you to build a mobile template and customize other templates specifically for mobile devices.

A very basic theme with mobile and standard layouts could look like this:

- **media**
  - css
  - js
  - img
- **mobile**
  - default.html
  - homepage.html
- **templates**
  - default.html
  - homepage.html
  - footer.html
24.2 Mobile Toggling

When setting up your mobile layout, you can select to show the mobile content using the code below (a mobile stylesheet is included in the example):

```html
{% if request.mobile %}
    <link rel="stylesheet" href="{% static 'css/styles-mobile.css' %}" type="text/css">
{% endif %}
```

That same code can be used to include a separate layout, separate navigation, or anything else that you may want to display to mobile viewers but not to desktop browsers. It is **not** necessary to use these `if` tags in templates in the mobile folder.

To take advantage of the cookie opt-out for mobile viewers who want to view the full site, first load the mobile tags at the top of the template:

```html
{% load mobile_tags %}
```

To show the link to mobile theme viewers to view full site:

```html
{% if request.mobile %}
    {% toggle_mobile_link request.get_full_path "View full site" %}
{% endif %}
```

To show the view mobile site link to mobile browser viewers who previously opted out:

```html
{% if request.mobile_browser and not request.mobile %}
    {% toggle_mobile_link request.get_full_path "View mobile site" %}
{% endif %}
```

The placement of these links, as well as the text of the link is customizable.

24.3 Mobile Browsers Supported

The following mobile browsers are supported:

- iPhone
- iPod
- Android
- Opera Mini
- Blackberry
- Droid
- IEMobile
- EudoraWeb
- Fennec
- Minimo
- NetFront
- Polaris
• HTC_Dream
• HTC Hero
• HTC-ST7377
• Kindle
• LG-LX550
• LX265
• Nokia
• Palm
• MOT-V9mm
• SEC-SGHE900
• SAMSUNG-SGH-A867
• SymbianOS
• DoCoMo
• ZuneHD
• ReqwirelessWeb
• SEJ001
• SonyEricsson
Here is an overview of the folder structure for themes, along with some details of specific files within the theme. All themes should be added to the themes folder in the main root of the site.

25.1 Theme Folder Organization

- theme-name
  - theme.info
  - screenshot.png
  - settings.json
  - media
    - css
    - js
    - img
  - templates
    - homepage.html
    - default.html

25.2 Theme File Descriptions

theme.info This file contains a set of attributes associated with a theme. See the example below for some common attributes.
screenshot.png  This file is a full page screenshot of the theme after it has been loaded with the fixtures using the load_npo_defaults management command.

settings.json  This file contains settings for the Tendenci site settings. These settings are written in json and are installed whenever the update_settings, set_theme, or install_theme command is run.

templates/homepage.html  This file defines the HTML that is used when the homepage of the site is loaded. You can learn more about the homepage at Theme: Homepage Template

templates/default.html  This file defines the HTML that is used when a page other than the homepage is loaded.

25.3 Theme Media Usage

Template media in the media folder can be referenced in the template using {% static 'path/to/file' %}. 

---

name = Salonify
description = Salonify theme for Tendenci CMS.
tags = homepage rotator, spaces
screenshot = screenshot.png
author = Schipul
author uri = http://schipul.com
version = 1.0
create_dt = 2013-01-01 01:00:00
Tendenci Credits

Thank you to our current core developers adding issues, pull requests, documentation and helping the community. Without idealistic people working on this open source project and trying to change the world, Tendenci would not be possible.

- Ed Schipul @eschipul
- Jenny Qian @jennyq
- Karl Goetz @goetzk
- Paul Donohue @PaulSD
- Adam Bark @AdamBark
- Jaime Lossada @dco5
- Ben Sturmfels @BenSturmfels
- Suraj Prasad @surajprasad
- Edna Diaz @ediaz1
- Rachel Schipul @rschipul
- Leanne Luce @leanneluce

Additional thanks to the people who have contributed to the product as a whole from the beginning. The list is mostly chronological and may not directly reflect their overall contributions. This is just a huge THANK YOU list!

- Jennifer Ulmer
- Donnell Smith
- Kathryn Ciaralli
- Nse Ekpenyong
- Adil Ahmed
- Aaron Long
- Eloy Zuniga
• Glen Zangirolami
• John-Michael Oswalt
• Sam Lee
• Alex Ragsdale
• Felicia Ann Bates
• John-Eddie Ayson
• JayJay Montalbo
• Max Chua
• Sarah Worthy
• Tim Newton
• David Stagg
• Loren Lugosch
• Nabil Bani

If you know of someone who should be added to this list and deserves credit, cause we are human and it’s hard to keep up, please TELL US! For example someone active on Transifex doing translations, or adding documentation, they are just as appreciated as people submitting issues and pull requests.

PS - A particular “thank you” call out to @goetzk who posted the feedback that reminded me this file existed. Thank you Karl for posting https://github.com/goetzk/tendenci-community/blob/master/community-engagement.rst - and you are correct. I need to do better. I’m learning still, and probably always will be learning from y’all.