

TEK DB Technical Information and Maintenance Manual

Dependency Stack

- Operating System (pick 1):
 - **Ubuntu 16.04 LTS** (recommended/supported)
 - Other Linux Server distributions
 - Windows Server
- Web Server (pick 1):
 - **NGINX** (recommended/supported)
 - Apache
 - IIS (for Windows)
- Database:
 - **PostgreSQL with PostGIS extension**
- Language:
 - **Python 3.5+**
- Framework:
 - **Django 1.11**

Code Source

All code and documentation is available for free (under the [MIT license](https://github.com/ECOTRUST/TEKDB)) on Github: <https://github.com/ECOTRUST/TEKDB>

Exporting the data

If you are reinstalling the tool and need to preserve or migrate your data (not the same as migrating from the old Microsoft Access-based MTKEDB) you can create a 'fixture' (a backup file) that can be imported into new instances of the TEK DB. To do so:

- Decide on a safe place to write the file to
- Decide on a meaningful name for the file, like `tek_db_fixture_YYYYMMDD.json`
 - where YYYYMMDD is today's date

- json is the data format. If you don't name it with the .json extension this process will still work.
- Navigate to your project and activate your virtual environment:

```
cd /usr/local/apps/TEKDB
source env/bin/activate
cd ./TEKDB
```

- Then run this WHOLE command to dump the data (the order is very specific to protect data dependencies) replacing the file at the end after the > with the file you chose above

```
python manage.py dumpdata --indent=2 auth.Group Lookup.LookupPlanningUnit
Lookup.LookupTribe Lookup.LookupHabitat TEKDB.Places
Lookup.LookupResourceGroup TEKDB.Resources Lookup.LookupPartUsed
Lookup.LookupCustomaryUse Lookup.LookupSeason Lookup.LookupTiming
Relationships.PlacesResourceEvents Lookup.LookupParticipants
Lookup.LookupTechniques Lookup.LookupActivity TEKDB.ResourcesActivityEvents
Lookup.People Lookup.LookupReferenceType Lookup.LookupAuthorType
TEKDB.Citations Relationships.PlacesCitationEvents Lookup.CurrentVersion
Lookup.LookupLocalityType TEKDB.Locality Relationships.LocalityGISSelections
Relationships.LocalityPlaceResourceEvent Lookup.LookupMediaType
Lookup.LookupUserInfo TEKDB.Media Relationships.MediaCitationEvents
Relationships.PlaceAltIndigenousName Relationships.PlaceGISSelections
Relationships.PlacesMediaEvents Relationships.PlacesResourceCitationEvents
Relationships.PlacesResourceMediaEvents
Relationships.ResourceActivityCitationEvents
Relationships.ResourceActivityMediaEvents
Relationships.ResourceAltIndigenousName Relationships.ResourceResourceEvents
Relationships.ResourcesCitationEvents Relationships.ResourcesMediaEvents
Accounts.UserAccess Accounts.Users explore.PageContent >
/path/to/your/file.json
```

Updating The Code

The code is all on GitHub and available to the public. Git, a tool for sharing the code and tracking changes made to it is already installed on your server. With these two pieces in place, getting code updates fairly easy, but can contain risks:

- It is highly recommended that you take a snapshot of your server before doing this.
 - If you are not sure what new features you want out of this upgrade, there is no need to do this.
 - Major risks:
 - New code will include database migrations that did not anticipate types of data in your instance. This could result in loss of data.
 - Local changes could have been made to your code base, resulting in a 'merge conflict'. Without familiarity with 'Git' these can be difficult to resolve, and guessing can quickly break things beyond repair.
 - Restoring from backup will fix both of these issues, but will prevent you from installing new updates. If you hit any of the issues above, please contact [Ecotrust](#) and ask for the Software Developers.
1. Log in to your server (via SSH for Linux)
 2. Change directories to TEK's code: `cd /usr/local/apps/TEKDB`
 3. Retrieve all code updates from GitHub: `git pull`
 4. Activate your virtual environment: `source env/bin/activate`
 5. If 'manage.py' is not in the current folder, cd into the TEKDB folder that does have it (most likely ./TEKDB)
 6. Run database migrations: `python manage.py migrate`
 - i. If new tables are added or old tables are changed this should update your database schema and transform any necessary data appropriately, but this comes with some risk to your data, hence the backup recommendation.
 7. Collect 'static files' to be served by the application: `python manage.py collectstatic`
 8. Restart uWSGI: `sudo service uwsgi restart`

OS Security Updates

Newer versions of Ubuntu Linux can now be set up to download and install security patches by default, and a cron job can be created to restart them during convenient hours if needed.

If you would like to do this by hand on Ubuntu Linux, do the following:

```
sudo apt-get update
sudo apt-get upgrade
```

Restart the server if prompted and server traffic is anticipated to be light.

Services

A deeper dive into some of the most important and dynamic pieces of the tool stack. Please note that commands below assume you are on Ubuntu Linux, and you may need to find another command to perform the same task on other operating systems. Links to each tool's website is provided in case more information is needed.

PostgreSQL & PostGIS

The database server, if causing problems, can be restarted with `sudo service postgres restart`.

On Linux, you can find most of PostgreSQL's files here: `/etc/postgresql` By default, logs will be kept here: `/var/log/postgresql/`

For more information, check out the [PostgreSQL website](#) and the [PostGIS website](#).

NGINX

The HTTP server. Can be restarted with `sudo service nginx restart`

NGINX's files can be found in `/etc/nginx` Configuration is managed in `/etc/nginx/sites-available/tekdb`

- A symlink to this file exists in `/etc/nginx/sites-enabled/` if the site is live Both access and error logs are held in `/var/logs/nginx/` by default

For more information, check out the [NGINX website](#).

uWSGI

A service enabling Django projects (and other webapp frameworks) to be served by HTTP servers.

Can be restarted with `sudo service uwsgi restart`

For more information, check out the [uWSGI Project website](#).

Troubleshooting

Lost Django Admin Password

Only Admin users can create or edit users, and this includes passwords. If no administrator is able to login, this can be alarming, but it is easily fixed from the back end.

- Log in to the server
- Navigate to the project folder (likely `/usr/local/apps/TEKDB`)
- Activate the virtual environment `source env/bin/activate`
- Navigate to the `manage.py` file (either here or in `./TEKDB`)
- Create a new Super User with `python manage.py createsuperuser`
 - You will be prompted to provide a username and a password
- Log in with the new Super User and reset the Admin passwords to something they will remember

Site is Down

- Is the error:
 - **404 - not found:**
 - Check that NGINX is running with `sudo service nginx status`
 - replace `status` with `start` if not
 - Check the NGINX logs for clues
 - Check the PostgreSQL logs

- Check the (virtual) hardware:
 - Is the hard drive full (can happen from old logs building up or lots of media served from the tool)
 - See if you can't clear enough space by deleting old log files
 - Increase the hard drive capacity
- Check the network:
 - are you on the same network as the tool?
 - does the given URL point to the server's IP address in your DNS records?
 - is NGINX configured to listen to requests to the given url?
 - is NGINX listening on port 80 (or whichever port the request is being made to)
- If all else fails, turn it off and on again.
- **A page loads, but tool doesn't show:**
 - Check network configuration (as above)
 - Check NGINX is running
 - Check NGINX logs to confirm it is receiving the requests
 - Check NGINX config to see that it is correct
 - Check uWSGI is running and configured correctly
 - Restart services one at a time and test in this order:
 - uWSGI
 - NGINX
 - The server itself
- **A Django Error** (yellow header with gray debug information):
 - NOTE: this page should not show - be sure to set DEBUG = False in your settings.py and local_settings.py files
 - Review the page content for hints.
 - Review the NGINX docs for hints.
 - Restart uWSGI
 - Reboot the server
 - Create an issue on the GitHub project describing the error and what how it was triggered.
- **A Plain Error Page** (Bold black header, plain black text on white background):
 - Follow same steps as 404 - Not found error
- **Page content appears but is mangled:**
 - Does page still look mangled on a wider browser window?
 - No images or improper page layout:

- from the virtual environment and project folder, run `python manage.py collectstatic`
- Refresh browser cache