

Django Dev Onboarding and Run pytide

Django Developer Onboarding

Access and Order of Operations

You should receive a laptop after orientation. IT will have it set up to where you can create your password for authentication across PeopleTec sites. It may take a day or two, however, for the single sign-on system to allow access to some of the employee tools and sites.

The following dependency table will show which downloads you will need as well as the requirements for access/dependencies to run/configure each. Use the steps below to request access and set up each tool.

Item:	Dependency:
Python Download, VS Code Download, Django, pip, git	None
BossDesk, Timecard	Single sign-on access (takes 1-3 days)
Admin Rights for Local Machine	BossDesk ticket approval
AWS VPN Download	Admin Rights
PostgreSQL Download, SourceTree Download	Admin Rights
Repository Access via PAT	AWS VPN, Git Download, VS Code Download, PostgreSQL Database, Django Download
Repository access with MR rights	AWS VPN, Git Download, VS Code Download, PostgreSQL Database, Django Download, CAC and required program onboarding.

Order of Operations for Onboarding and Setup

1. Get BossDesk Access – this should occur within a few days of onboarding. https://peopletec.bosssdesk.io/users/sign_in
2. Request admin rights for your machine – Place a BossDesk ticket to acquire administrator access. Use the link in step 1 to log in to BossDesk. On the lefthand side toolbar select “Service Catalog.” Select the “Information Technology” service category to narrow down request ticket fields. Choose “Exception to Policy” request and provide necessary information. This process will take a few days to complete. The request will have to be approved by multiple people and they may require follow up information.
3. Once admin access is given, you can now download items necessary for development.
 - a. AWS Client: This is necessary to access many internal company tools such as JIRA and Confluence help sites. It’s also necessary to run the program and access the database.
 - i. To Download: <https://aws.amazon.com/vpn/client-vpn-download/>
 - ii. Setup Profile: You’ll need to ask a member of the team for the setup configuration profile. This will need to be downloaded and provided when AWS prompts for the client configuration during setup.
 - b. PostgreSQL Database: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> (see instructions for setup below)
 - c. SourceTree repository and branch management GUI (optional- a nice to have for creating, viewing, and managing branches instead of using git commands in terminal: <https://www.sourcetreeapp.com/>)
 - d. Visual Studio Code: Downloadable via the Microsoft Store
 - e. Command line installs: Python 3.12, pip, git, Django (w3schools has a good tutorial that can be used while waiting on confluence onboarding documentation access. <https://www.w3schools.com/django/index.php>) (See further instructions on Python and Django downloads and setup below)
4. Pull the Git Repo Using Personal Access Token:
 - a. A team member will need to create and provide you with a personal access token (PAT).
 - b. To run py-tide without GitLab access - In PowerShell navigate to desired project directory (i.e repos). Using the provided PAT, run the following commands:
 - \$MyPat = "PAT from gitlab" (you will have to get a PAT from a gitlab current user)
 - \$B64Pat = [Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes(".\$MyPat"))
 - git config --global --add http.<https://sync.git.mil>.extraHeader "Authorization: Basic \$B64Pat"
 - git clone <https://sync.git.mil/pti-group-n/tide/ui/py-tide.git>
 - if you receive a 500 error or a 400-error run the following commands:
 - git config --global --remove http.<https://sync.git.mil>
 - git config --global --add http.<https://sync.git.mil>.extraHeader "Authorization: Basic \$B64Pat"
 - git clone <https://sync.git.mil/pti-group-n/tide/ui/py-tide.git>
5. Connect to PostgreSQL database by adding a .env file to the py-tide project main directory. A team member can provide the correct file to add.
6. See further instructions below for running the Py-Tide Website

Python/Django/PostgreSQL Setup

Python

Install Python 3.10, 3.11 or 3.12 (supported by Django 5.0.2)

Download from here: <https://www.python.org/downloads/>

Install, make sure you select the add to path checkbox.

Test:

py --version

output: Python 3.11.9

Make sure you have the Package Manager:

pip --version

output: pip 24.0 from C:\Program Files\Python312\Lib\site-packages\pip (python 3.11)

Django

Install Django:

pip install Django

Check the version:

django-admin --version

output: 5.0.2

Test Django:

start a Project:

django-admin startproject myproject

cd into the myproject directory

cd myproject

Create virtual environment

py -3 -m venv venv

Activate it:

\venv\Scripts\activate

Install Django into that environment:

py -m pip install Django

and run it:

py manage.py runserver

point a browser to: <http://127.0.0.1:8000/>

A test web site should come up.

PostgreSQL (optional)

Install PostgreSQL 14:

Install from: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Do not use the Stack Builder

Run PyTide WebSite

Login to the AWS VPN:

Clone py-tide repo from gitlab:

```
cd Source\py-tide
```

Create Virtual Environment:

```
py -m venv venv
```

Activate the environment:

```
venv\Scripts\activate
```

Install the packages:

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

create tide database in postgresSQL (if running local Database if not ignore blue):

Login to PostgreSQL (pgAdmin4 or psql)

create the tide database

```
CREATE DATABASE pytide;
```

Create User:

```
CREATE USER "pytide-user" WITH ENCRYPTED PASSWORD 'your_password';
```

```
GRANT ALL PRIVILEGES ON DATABASE pytide TO "pytide-user";
```

```
GRANT ALL ON SCHEMA public TO "pytide-user";
```

Change .env file to point to your local database:

```
DATABASE_NAME='pytide'
```

```
DATABASE_USER='admin'
```

```
DATABASE_PASSWORD='your_password'
```

```
DATABASE_HOST='localhost'
```

```
DATABASE_PORT=5432
```

Migrate the database:

```
py manage.py makemigrations
```

```
py manage.py migrate
```

Load data into database:

Execute scripts in the "run fixtures" step in Py-tide README.md file

Create an admin user:

```
py manage.py createsuperuser
```

Run the application:

```
py manage.py runserver
```

Start up a browser:

```
http://127.0.0.1:8000
```

Type Control-C to stop the website.

Deactivate your virtual environment at the command prompt:

```
deactivate
```

Running py-tide without GitLab access - use PowerShell and run the following commands:

- \$MyPat = "PAT from gitlab" (you will have to get a PAT from a gitlab current user)
- \$B64Pat = [Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("\$MyPat"))
- git config --global --add http.<https://sync.git.mil>.extraHeader "Authorization: Basic \$B64Pat"
- git clone <https://sync.git.mil/pti-group-n/tide/ui/py-tide.git>
 - if you receive a 500 error or a 400-error run the following commands:
 - git config --global --remove http.<https://sync.git.mil>
 - git config --global --add http.<https://sync.git.mil>.extraHeader "Authorization: Basic \$B64Pat"
 - git clone <https://sync.git.mil/pti-group-n/tide/ui/py-tide.git>