Create a project

Contents

This chapter describes how you can create a web application project using Plone, with full control over development and deployment.

If instead you want to contribute to a Plone package, see Contributing to Plone.

System requirements

Plone 6 has both hardware requirements and software pre-requisites.

Hardware requirements

The hardware requirements below give a rough estimate of the minimum hardware setup needed for a Plone server.

A single Plone installation is able to run many Plone sites.

- Installation of the Plone backend and Classic UI frontend requires a minimum of 256 MB of RAM and 2GB of disk swap space.
- Installation of the Volto frontend requires a minimum of 2GB of RAM.
- After installation, running Plone requires a minimum of 256 MB RAM and 512 MB of disk swap space per Plone site. 2 GB or more RAM per Plone site is recommended.
- Minimum 512 MB hard disk space is required. 40 GB or more hard disk space is recommended.

Warning

Add-on products and caching solutions may also increase RAM and disk swap space requirements. To avoid RAM and disk swap limitations, we recommend either temporarily resizing your remote machine to accommodate the build, or build your images locally and upload them to an image store, such as Docker Hub or GitHub Packages.

See also

How much RAM is required to build a Volto front end? and Dealing with heap exhaustion while building Volto 17 on limited-RAM host.

Pre-requisites for installation

- An operating system that runs all the requirements mentioned. Most UNIX-based operating systems are supported, including many Linux distributions, macOS, or Windows Subsystem for Linux (WSL) on Windows. A UNIX-based operating system is recommended.

> **Important**
>
> Windows alone is not recommended because it does not support GNU make. If you get Plone to run on Windows alone, please feel free to document and share your process.

- Python 3.8, 3.9, 3.10, or 3.11
- pipx
- nvm
- Node.js LTS 20.x
- Yeoman
- Yarn
- GNU make
- Docker

Python

Installing Python is beyond the scope of this documentation. However, it is recommended to use a Python version manager, pyenv that allows you to install multiple versions of Python on your development environment without destroying your system's Python. Plone requires Python version 3.8, 3.9, 3.10, or 3.11.

pipx

Install pipx.

```
pip install pipx
```

nvm

The following terminal session commands use `bash` for the shell. Adapt them for your flavor of shell.

> **See also**
>
> See the **`nvm` install and update script documentation**. For the `fish` shell, see **`nvm.fish`**.

1. Create your shell profile, if it does not exist.

   ```
   touch ~/.bash_profile
   ```

2. Download and run the `nvm` install and update script, and pipe it into `bash`.

   ```
   curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.39.5/install.sh |
   bash
   ```

3. Source your profile. Alternatively close the session and open a new one.

   ```
   source ~/.bash_profile
   ```

4. Verify that the `nvm` version is that which you just installed or updated:

   ```
   nvm --version
   ```

Node.js

1. Install or update the supported LTS versions of Node.js, then activate the version
   using the latest LTS.

```
nvm install --lts
nvm use --lts
```

2. Verify that the supported version of Node.js is activated.

```
node -v
```

Yeoman

Install Yeoman.

```
npm install -g yo
```

Yarn 3

Install the latest Yarn 3 version (not the Classic 1.x one).

1. Open a terminal and type:

```
corepack enable
```

2. Verify that Yarn v3.x.x is installed and activated.

```
yarn set version 3.x
yarn -v
```

```
3.2.3
```

If you see a version that is not v3.x.x, you can try deleting the `yarn.lock` file, and installing again.

```
rm yarn.lock
corepack enable
yarn set version 3.x
yarn -v
```

```
3.2.3
```

Make

Make comes installed on most Linux distributions. On macOS, you must first install Xcode, then install its command line tools. On Windows, it is strongly recommended to Install Linux on Windows with WSL, which will include `make`.

Finally, it is a good idea to update your system's version of `make`, because some distributions, especially macOS, have an outdated version. Use your favorite search engine or trusted online resource for how to update `make`.

Install Docker

Install Docker Desktop for your operating system. Docker Desktop includes all Docker tools.

Install Plone 6

We install Plone 6 with pipx, Cookiecutter, mxdev, make, and other developer tools.

> **Note**
>
> We do not maintain documentation for installing Plone 6 or later with `buildout`. For Plone 5, `buildout` was the preferred installation method. You can read the documentation of how to install Plone 5 with **`buildout`**, and adapt it to your needs for Plone 6.

Create a new directory to hold your project, and make it your current directory.

```
mkdir my_project
cd my_project
```

Issue the following command to install or update `cookiecutter`, then run it to create a Plone project skeleton using the Cookiecutter cookiecutter-plone-starter.

```
pipx run cookiecutter gh:collective/cookiecutter-plone-starter
```

You will be presented with a series of prompts. You can accept the default values in square brackets (`[default-option]`) by hitting the `Enter` key, or enter your preferred values. For ease of documentation, we will use the default values.

> **Important**
>
> For ***Project Slug***, you must not use any of the Plone core package names listed in **`constraints.txt`**. Note that pip normalizes these names, so `plone.volto` and `plone-volto` are the same package.

```
% pipx run cookiecutter gh:collective/cookiecutter-plone-starter


Cookiecutter Plone Starter
================================================================================

Sanity checks
--------------------------------------------------------------------------------
  [1/5] Python: ✓
  [2/5] Node: ✓
  [3/5] yo: ✓
  [4/5] Docker: ✓
  [5/5] git: ✓

Project details
--------------------------------------------------------------------------------

  [1/19] Project Title (Project Title): Plone Conference Website 2070
  [2/19] Project Description (A new project using Plone 6.):
  [3/19] Project Slug (Used for repository id) (plone-conference-website-2070):
  [4/19] Project URL (without protocol) (plone-conference-website-2070.example.com):
  [5/19] Author (Plone Foundation): Elli
  [6/19] Author E-mail (collective@plone.org): elli@plone.org
  [7/19] Python Package Name (plone_conference_website_2070):
  [8/19] Volto Addon Name (volto-plone-conference-website-2070):
  [9/19] Choose a Python Test Framework
    1 - pytest
    2 - unittest
    Choose from [1/2] (1):
[10/19] Plone Version (6.0.8):
[11/19] Should we use Volto Alpha Versions? (No): yes
[12/19] Volto Version (18.0.0-alpha.1):
[13/19] Volto Generator Version (8.0.0):
[14/19] Language
    1 - English
    2 - Deutsch
    3 - Español
    4 - Português (Brasil)
    5 - Nederlands
    6 - Suomi
    Choose from [1/2/3/4/5/6] (1):
[15/19] GitHub Username or Organization (collective): ellizurigo
[16/19] Container Registry
    1 - GitHub Container Registry
    2 - Docker Hub
    Choose from [1/2] (1):
[17/19] Should we setup a caching server?
    1 - Yes
    2 - No
    Choose from [1/2] (1): 2
[18/19] Add Ansible playbooks?
    1 - Yes
    2 - No
    Choose from [1/2] (1):
[19/19] Add GitHub Action to Deploy this project?
    1 - Yes
    2 - No
    Choose from [1/2] (1):

Plone Conference Website 2070 generation
--------------------------------------------------------------------------------

Summary:
  - Plone version: 6.0.8
  - Volto version: 18.0.0-alpha.1
  - Volto Generator version: 8.0.0
  - Output folder: /Users/katjasuss/Desktop/_temp/scratch_cookiecutter_plone/plone-
conference-website-2070

Frontend codebase:
 - Installing required npm packages
 - Generate frontend application with @plone/volto 18.0.0-alpha.1

Backend codebase
 - Remove folder
src/plone_conference_website_2070/src/plone_conference_website_2070/tests not used by
pytest
 - Format generated code in the backend

================================================================================

Project "Plone Conference Website 2070" was generated
--------------------------------------------------------------------------------
Now, code it, create a git repository, push to your organization.
```

```
Sorry for the convenience,
The Plone Community.

================================================================================
```

Change to your project directory `plone-conference-website-2070`.

```
cd plone-conference-website-2070
```

Next you switch to using `make`. To see all available commands and their descriptions, enter the following command.

```
make help
```

To install both the Plone backend and frontend, use the following command.

```
make install
```

This will take a few minutes. ☕ First the backend, then the frontend will be installed.

When the process completes successfully, it will exit with no message.

> **Note**
>
> If you used a Plone core package name, then `make install` will return an error message such as the following.
>
> ```
> ERROR: Cannot install plone-volto 1.0.0a1 (from
> /home/username/projects/volto/plone-volto/backend/src/plone_volto) because
> these package versions have conflicting dependencies.
>
> The conflict is caused by:
>     The user requested plone-volto 1.0.0a1 (from
> /home/username/projects/volto/plone-volto/backend/src/plone_volto)
>     The user requested (constraint) plone-volto==4.2.0
>
> To fix this you could try to:
> 1. loosen the range of package versions you've specified
> 2. remove package versions to allow pip attempt to solve the dependency
> conflict
>
> ERROR: ResolutionImpossible: for help visit
> make[2]: *** [Makefile:112: build-dev] Error 1
> make[2]: Leaving directory '/home/username/projects/volto/plone-
> volto/backend'
> make[1]: *** [Makefile:46: install-backend] Error 2
> make[1]: Leaving directory '/home/username/projects/volto/plone-volto'
> ```
>
> You must delete your project, follow the important note, and run the cookiecutter again.

Start Plone

Plone 6 has two servers: one for the frontend, and one for the backend. As such, we need to maintain two active shell sessions, one for each server, to start your Plone site.

Start Plone backend

In the currently open session, issue the following command.

```
make start-backend
```

The Plone backend server starts up and emits messages to the console.

```
2022-09-24 01:30:17,799 WARNING [ZODB.FileStorage:411][MainThread] Ignoring index for
/<path-to-project>/my_project/project-title/backend/instance/var/filestorage/Data.fs
2022-09-24 01:30:19,639 INFO    [chameleon.config:38][MainThread] directory cache:
/<path-to-project>/my_project/project-title/backend/instance/var/cache.
2022-09-24 01:30:23,680 INFO    [plone.volto:22][MainThread] Aliasing
collective.folderish classes to plone.volto classes.
2022-09-24 01:30:24,935 INFO    [Zope:42][MainThread] Ready to handle requests
Starting server in PID 92714.
2022-09-24 01:30:24,940 INFO    [waitress:486][MainThread] Serving on
http://[::1]:8080
2022-09-24 01:30:24,940 INFO    [waitress:486][MainThread] Serving on
http://127.0.0.1:8080
```

Start Plone frontend

Create a second shell session in a new window. Change your current working directory to `project-title`. Start the Plone frontend with the following command.

```
make start-frontend
```

The Plone frontend server starts up and emits messages to the console.

```
 WAIT  Compiling...


✓ Client
  Compiled successfully in 864.83ms

✓ Server
  Compiled successfully in 9.62s

✅  Server-side HMR Enabled!
sswp> Handling Hot Module Reloading
Volto is running in SEAMLESS mode
Using internal proxy: http://localhost:3000 -> http://localhost:8080/Plone
🎭 Volto started at 0.0.0.0:3000 🚀
```

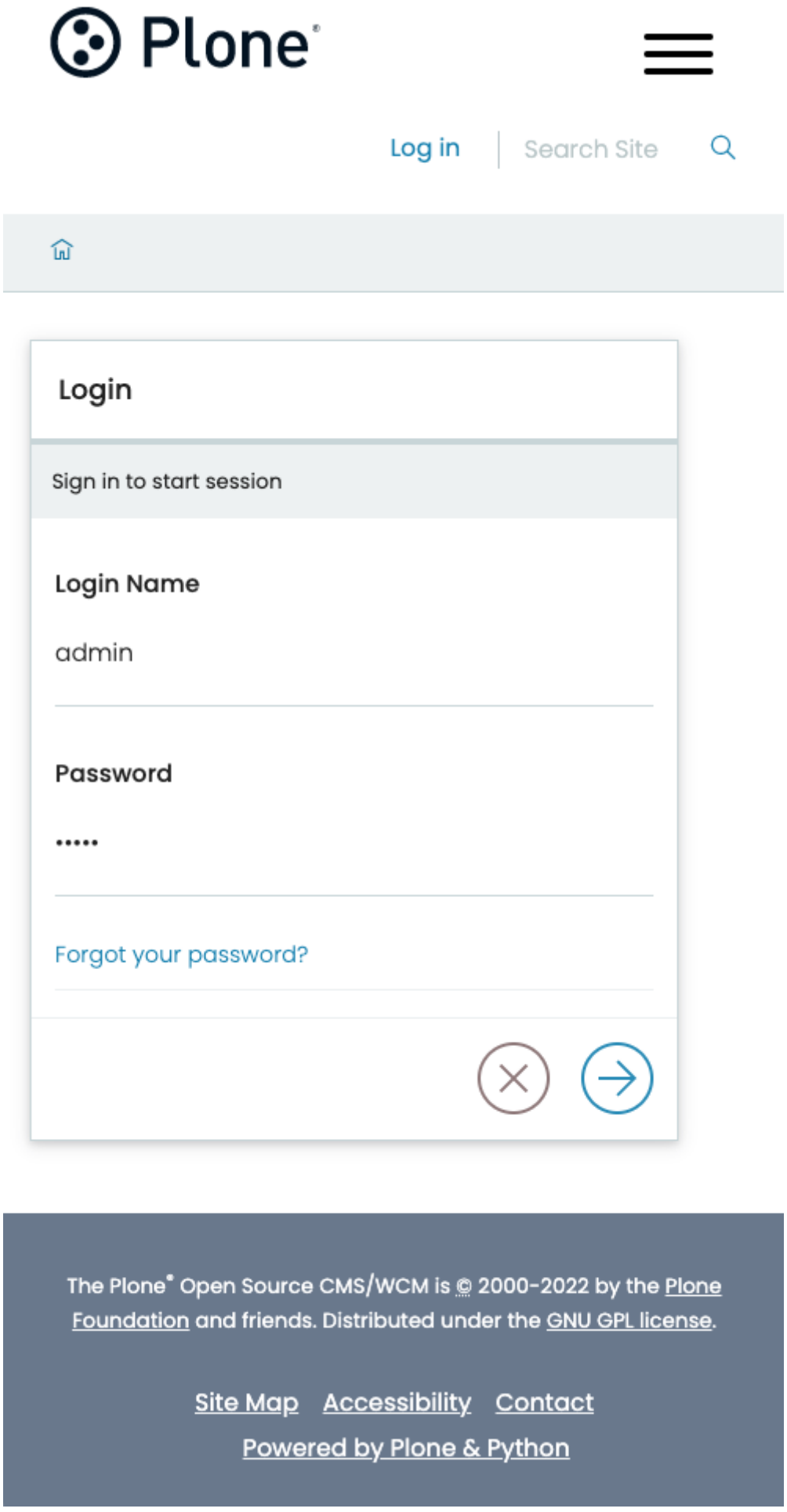
Note that the Plone frontend uses an internal proxy server to connect with the Plone backend. Open a browser at the following URL to visit your Plone site.

http://localhost:3000

You will see a page similar to the following.

Select the ***Login*** link to visit the login form, and enter the following credentials.

- ***Login name*** : `admin`
- ***Password*** : `admin`



Now you can edit content or configure your Plone site.

You can stop the site with `ctrl-c` .
