



Roadmap to Plone 6 and beyond

# Plone 6 Roadmap

---

Philip Bauer, Jens Klein, Timo Stollenwerk

December 2020

<b>Preface</b>	<b>3</b>
Previous Roadmaps	3
Status-quo	3
Challenges and Solutions	3
<b>Plone 6 Roadmap</b>	<b>4</b>
Volto Frontend / Pastanaga UI	4
Modernize Plone default theme	4
Folderish Content / Page	4
Using Dexterity for Plone Site	5
Drop Support for Python 2	5
Archetypes End of Life	5
Remove portal_quickInstaller	5
Use Zope 5 in Plone 6	5
Mockup and resource registry redone	5
Rethink the UnifiedInstaller	6
Release Schedule for Plone 6.0	6
<b>Long-term Roadmap</b>	<b>6</b>

# Preface

## Previous Roadmaps

The result of community discussions were set down in past [roadmap-documents](#).

## Status-quo

The existing Plone platform is mature, very secure, flexible, and extensible. It offers a good user experience and focuses on accessibility.

With Plone 5.2 it now supports Python 3.8, runs on WSGI, and provides a full REST API. This ensures Plone has a stable and feature-rich backend for the foreseeable future. Plone is also positioned as a powerful Headless-CMS-backend for all kinds of frontend-technologies.

It allows you to build a wide range of sites for very different audiences, from very simple sites managed by a small team or a single person with no technical background, to large sites supporting complex business processes and involving a lot of custom features.

Plone offers specific features (security, complex workflows, sophisticated permissions, high pluggability) which make it a better solution compared to its competitors in many cases.

## Challenges and Solutions

In the past years the Plone community has identified a set of challenges and tasks. Put simply the two main tasks were a) modernizing the existing frontend for long term support and b) providing a new, modern frontend with improved usability.

The planned Plone 6 release will ship with a modernized server-side rendered frontend - as we know it, but tidied up and ready for future long term maintenance.

The rising importance of mobile devices and the pursuit to improve user experience and loading times of modern websites and applications led to significant innovation in web technology in recent years. JavaScript technology became one of the main drivers of innovation in web technology. Open source frameworks like React offer a superior user experience and speed together with rich features, well maintained code, and a large developer community.

To stay at the front of these web-development trends Plone 6 will ship with a new React-based frontend - called Volto - that provides a superior user experience and speed on top of a modern and well maintained JavaScript-framework together with the maturity, reliability, and security of the Python-based Plone backend.

# Plone 6 Roadmap

There will be three major changes in Plone 6:

- Plone 6 will ship with Volto, a new ReactJS-based frontend for Plone implemented on top of the plone.restapi. This combines the stability, maturity, and security of the Plone backend with a modern, mature, user-friendly and well maintained frontend.
- The “core” Barceloneta-based frontend will be modernized using Bootstrap 5, with cleaned up markup and CSS.
- Plone 6 will run on Python 3 only, fading out compatibility code to Python 2..

## Volto Frontend / Pastanaga UI

Plone 6 will ship with Volto as default frontend. It implements the new Pastanaga UI in React.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/2703>

Status: PLIP has been accepted by the Plone Framework Team. Volto has frequent releases and is used in production.

## Modernize Plone default theme

The server-side rendered frontend (barceloneta) is rewritten based on bootstrap 4 and cleaned up a lot. Based on this you can implement custom themes.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/3061>

Related: <https://github.com/plone/Products.CMFPlone/issues/2967>

Status: PLIP has been accepted by the Plone Framework Team, implementation is active and progressed far.

## Folderish Content / Page

We plan to simplify the Plone User Experience by getting rid of the default\_page in Plone 6 for Volto.

PLIP: not yet (follow-up the old PLIP #20144)

Status: Needs discussion. No PLIP written yet.

## Using Dexterity for Plone Site

In order to be able to move to folderish content types, we need the Plone site root to implement the Dexterity base class. Otherwise we would not be able to display a front page in Plone.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/2775>

Status: PLIP has been accepted. Implementation is finished and needs to be reviewed.

## Drop Support for Python 2

Python 2 has been declared as deprecated with Plone 5.2. Python 3 is the default Python version. We do not support Python 2 in Plone 6.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/2812>

Status: PLIP has been accepted by the Plone Framework Team

## Archetypes End of Life

Archetypes has been declared as deprecated with Plone 5.2 and will not be supported in Plone 6. Migrations from Archetypes to Dexterity can still be done in 5.2 (in-place) or leveraging the RestAPI.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/2775>

Status: Done.

## Remove portal\_quickInstaller

PLIP: <https://github.com/plone/Products.CMFPlone/issues/1775>

Status: PLIP is implemented and reviewed

## Use Zope 5 in Plone 6

Zope 5 is mostly feature-equivalent to Zope 4 and no longer supports Python 3.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/3058>

Status: PLIP has been accepted by the Plone Framework Team

## Mockup and resource registry redone

Mockup and dependencies goes ES6

Drop through-the-web compilation via RequireJS and LESS.

Resource Registry: Bundle registration is kept but resources will be removed as there is no need for them anymore.

PLIP: <https://github.com/plone/Products.CMFPlone/issues/3211>

Status: PLIP has been accepted by the Plone Framework Team, work in progress.

## Rethink the UnifiedInstaller

Combine Plone Core and Plone Volto setup in one installer.

Status: Installer Team is discussing different concepts.

## Release Schedule for Plone 6.0

The old schedule was to have a final Plone release before the Plone Conference. This schedule was upended by the corona-virus. At the Plone-conference 2020 (online) the community needs to decide on a new schedule.

The main blocker is that the two “big” PLIPs (“Volto Frontend” and “Modernize Plone default theme”) need to be merged and production-ready.

## Long-term Roadmap

A long-term roadmap has not been planned yet. Here are a few aspects that have been discussed so far:

- We need to focus on training and on-boarding the community for Volto. Every Plone-developer needs to be able to create, customize and reuse react components. It would be great for the acceptance of these technologies if there were good examples of how to include react-components in the current frontend.
- Simplify the stack for new developers. The documentation has improved a lot in the last few years but there is still a need for more condensed and accessible documentation for junior developers that needs to focus on the core-aspects of modern Plone-Development.
- As a very long-term future switch we can think about removing the Barceloneta frontend and all browser views that render page-templates from the code. This will only be possible after a large part of the Plone community moved to Volto and Volto can provide a comparable ecosystem of add-ons.