

September Update

September included a Windows patch to the SP and some significant fixes for (uncommon, so far) issues with the HTTP stack in the IdP. The latter are not expected to ship unless they cause more trouble until more critical bug fixes warrant a new patch.

Over the last month's work on IdP 4.1, much of the supporting work to make the new plugin framework useful has been completed and we have been able to address most of the manual configuration challenges of adding new flows to the system. This work has led to revisiting portions of the configuration, and much of the high level authentication behavior (what was previously in *conf/authn/general-authn.xml*) will now be configurable with properties instead of XML.

In the process of working on the problem of populating default configuration files after install instead of all up front, a companion notion to the plugin framework patterned after Jetty modules has been developed this month called IdP modules. Along with a new command line tool to manage them, the module system is a way of bundling rules for installing, updating, and removing configuration files, views, and any other necessary files in a deployed system. The IdP need not be running (and usually shouldn't be) to alter module state.

As an example, all of the individual login flow implementations will be exposed as IdP modules, and the default configurations hidden (and generally unusable) until the module is enabled to make use of the feature, causing the files to be added to the tree. In addition, the module system understands how to manage the files it owns, and supports RPM-like semantics to preserve configurations during upgrades or on removal/disablement. So people should expect to run into *.idpsave* and *.idpnew* file extensions on occasion when defaults are changed.

We plan to extend this module design to most of the "supplemental" features in the software that are managed with dedicated configuration files today, reducing the number of files in the way out of the box.

The module system supports the plugin framework by allowing plugins to expose their own modules as needed, and taking care of much of the configuration file placement that would otherwise be left to the plugin installer. Plugins can obviously enable their modules on install and disable them on removal automatically.

We also hope that for many older deployers, reducing system clutter by simply disabling modules not being used should be easy.

As this work has taken shape, it's becoming clearer that we are targeting 4.1 as the next "big" release of the IdP rather than deferring certain work to 5.0. We have started internal discussions about exactly what work might be moved in to this version (and to the plugins that would accompany it), and will adjust the expected release date accordingly. We're already looking at no earlier than the very end of 2020 and may push the release into 2021. At this point, 5.0 probably is more like 4.0: a release done out of necessity at some future point to make changes to the Java and Spring platform, rather than a big feature release. Since we don't support older minor versions anyway, extending the useful life of 4.x is ultimately better for everybody.