Pydio Cells 2.0 Release Candidate

A – Presentation

What's new

This version provides a desktop sync client that replaces (at last) PydioSync for Cells. **CellsSync** provides offline synchronization on your Windows, Linux and Mac computers.

This is a major step in fulfilling the promise of the initial rewrite of Pydio in GO language, as CellsSync is written in the same language and shares most of its libraries with the server. For this reason, many server components where also extended under the hood.

Below are the most important ones :

- **The "Sync" library**, that was used inside DataSources to mirror the *object service* modifications to the *index service* of each datasource was fully rewritten. This same library is used for server-to-desktop synchronization, and can even be used in other interesting scenariis : server-side folder-to-folder sync (directly through datasources), server-to-s3 storage sync (for realtime or scheduled backups), etc.
- **The Authentication brick**, originaly based on Open ID Connect, has been greatly rethought to allow external applications to use the server as a standard OIDC Identity Provider. Typically, adding an account inside CellsSync opens a common OAuth2 workflow by opening the server login page and getting a token back to the application. CellsSync does not know anything about users credentials locally.

Readme first : new gRPC gateway requirements

For best performances and real-time events, **CellsSync** communicates with the server using a gRPC connection. gRPC is an **HTTP/2** protocol, which implies that **HTTP/2 must be enabled on the bind address facing the outside world**.

If you are behind a proxy or inside a private network, you may have to check your proxy settings:

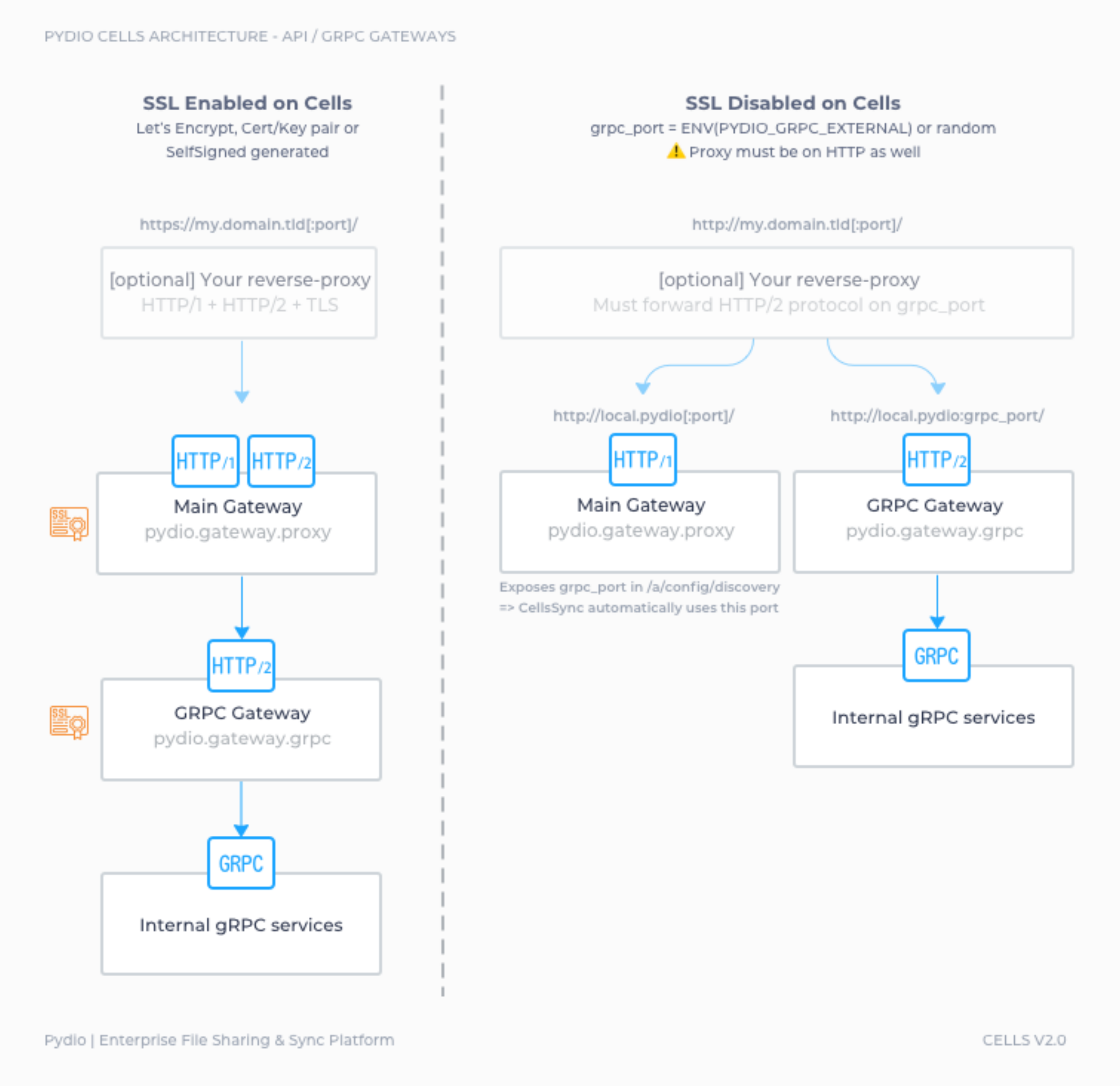
- **[SSL Enabled]** Cert/Key couple, Let's Encrypt, Self-signed config (see note below)

  Cells will serve HTTP/1.1 and HTTP/2 on the same port, the one you define for external url (e.g; 443, or 8080 or anything you choose). You don't have to open any other port in your firewall.

  - **[No Proxy]** Your Cells is directly facing the outside world with a proper SSL configuration, everything should be working out of the box
  - **[Proxy]** Just make sure your proxy is HTTP/2 enabled. If Cells using self-signed configuration (see note below), you can either install the generated rootCA.pem on the proxy machine, or configure the proxy to SkipVerify (for example *insecure_skip_verify* on Caddy).
- **[No SSL]** Cells will serve HTTP/1.1 and HTTP/2 on two different ports. By default, gRPC will pick a randomly available port and advertise it in the /a/config/discovery API. The CellsSync client will automagically query this API to connect.

  - **[No firewall, No Proxy]** If you are on a local machine with all ports open, this should work out of the box.
  - **[Firewall and/or Proxy]** You will have to make proper configuration to open and forward the HTTP/2 on this port. To avoid using a random port at each restart, you can fix this port by using the **PYDIO_GRPC_EXTERNAL** environnement variable at startup. Your proxy will probably not be able to serve HTTPS but HTTP only.

The various cases are summarized in the figure below.


PYDIO CELLS ARCHITECTURE - API / GRPC GATEWAYS
SSL Enabled on Cells
Let's Encrypt, Cert/Key pair or SelfSigned generated
https://my.domain.tld[:port]/
[optional] Your reverse-proxy
HTTP/1 + HTTP/2 + TLS
HTTP/1
HTTP/2
Main Gateway
pydio.gateway.proxy
HTTP/2
GRPC Gateway
pydio.gateway.grpc
GRPC
Internal gRPC services
SSL Disabled on Cells
grpc_port = ENV(PYDIO_GRPC_EXTERNAL) or random
⚠ Proxy must be on HTTP as well
http://my.domain.tld[:port]/
[optional] Your reverse-proxy
Must forward HTTP/2 protocol on grpc_port
http://local.pydio[:port]/
http://local.pydio:grpc_port/
HTTP/1
Main Gateway
pydio.gateway.proxy
HTTP/2
GRPC Gateway
pydio.gateway.grpc
Exposes grpc_port in /a/config/discovery
=> CellsSync automatically uses this port
GRPC
Internal gRPC services
Pydio | Enterprise File Sharing & Sync Platform
CELLS V2.0

[Note about self-signed] : in Cells v1, we were using the embedded self-signed feature of Caddy (our main gateway), that was managed in-memory. Cells v2 has to share the self-signed certificate between the main gateway and the gRPC gateway, so we now generate a self-signed certificate and pass this one to both services. As such, **if you are upgrading from v1 and using self-signed mode**, please re-run the `cells config ssl mode` step to regenerate a working configuration.

B – Install / Run instructions

Cells Server

Cells 2.0 does not have any new parameters for installation. Please refer to the official Cells documentation for installation instructions.

In-App Update or binary replacement

Update can be performed from a running Cells 1.X by using the in-app update tool. Just switch the update config to the `dev` channel and click refresh to see the new version appear, perform the update and restart. You can also download the new binary manually and replace it and restart.

Except for the Self-Signed case (see Note above), you should be good to go.

Fresh Install

Binaries are available for all OS on our download server :

Linux: https://download.pydio.com/pub/cells/release/2.0.0-rc2/linux-amd64/cells

MacOSX: https://download.pydio.com/pub/cells/release/2.0.0-rc2/darwin-amd64/cells

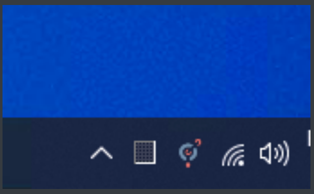
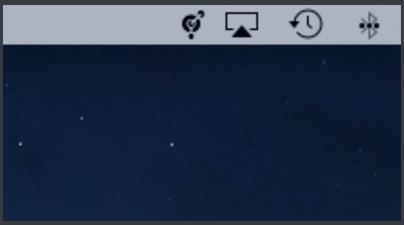
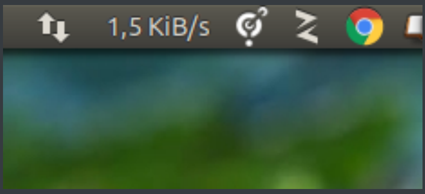
Windows : https://download.pydio.com/pub/cells/release/2.0.0-rc2/windows-amd64/cells.exe

CellsSync

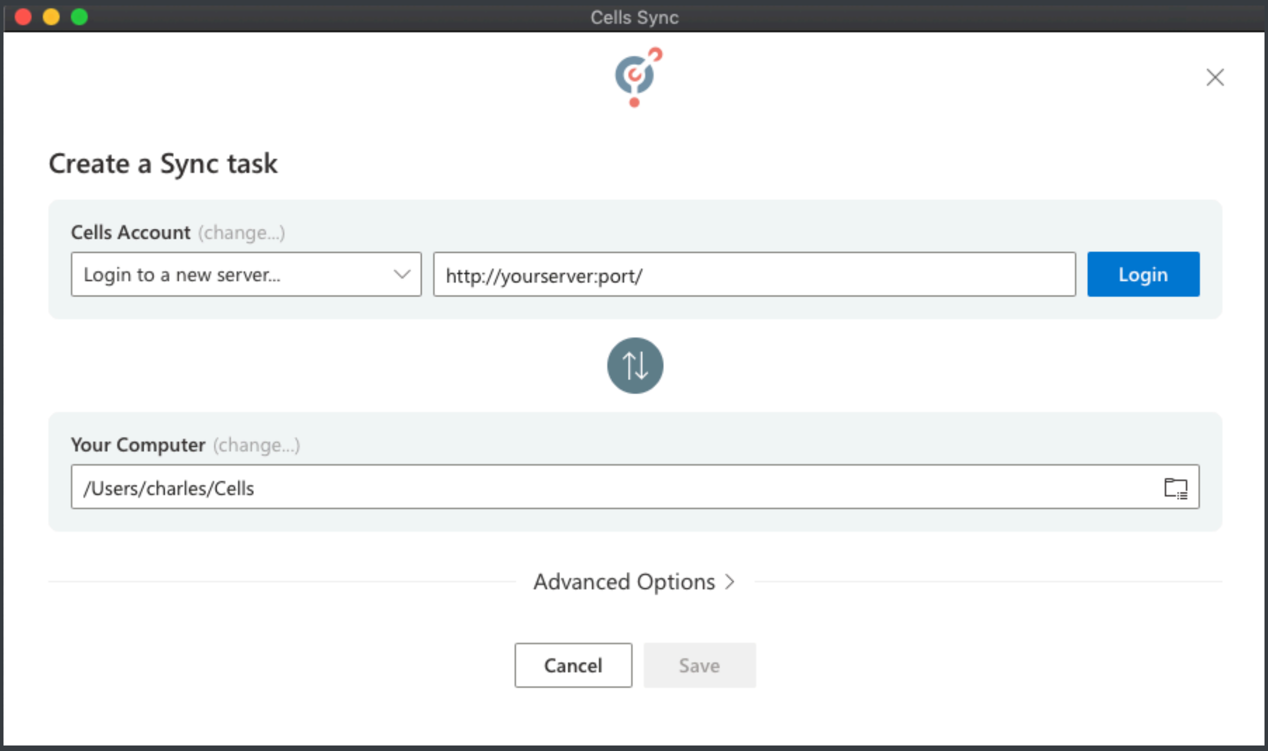
CellsSync is available for Windows, MacOS and Linux with installers in the following formats

- **Windows signed MSI** : will install cells-sync.exe in your local Program Files. Switching on the "Run At Startup" configuration should create a ".*lnk*" shortcut inside the Windows Start Up menu. Download URL : https://download.pydio.com/pub/cells-sync/release/0.9.0/windows-amd64/CellsSync-0.8.0.msi
- **MacOS signed DMG** provides a drag'n'drop Finder window for installing cells-sync in your Applications folder. Switching on the "Run At Startup" should install CellsSync as a *launchd service* in the user scope (under `~/Library/LaunchAgents`) Download URL : https://download.pydio.com/pub/cells-sync/release/0.9.0/darwin-amd64/CellsSync-0.8.0.dmg
- **Linux Binary** : The "Run At Startup" option is ubuntu-specific and installs ".*desktop*" shortcuts in the required locations. We are working on providing a SNAP for installing more easily no desktop-enable linux distributions. Download URL : https://download.pydio.com/pub/cells-sync/release/0.9.0/linux-amd64/cells-sync

Once installed, you should find a new icon in your Systrem Tray.

Windows	MacOSX	Linux
		

At first run, the application interface (webview) will open automatically on the "Create Task" screen. You should be able to create a new server with your Cells 2.0 server URL and be redirected back and from the server for identifying your user.



Cells Sync

Create a Sync task

Cells Account (change...)

Login to a new server...

http://yourserver:port/

Your Computer (change...)

/Users/charles/Cells

After authentication completes, you should see the new server in the "Cells Account" list, and then click in the "Path" to select a workspace or a subfolder to sync (make sure to make one workspace sync-able as explained above.)

Select the target Path for the local folder (a default location should be created automatically) and save the task. Sync should be on its way!

C – CellsSync Known Issues

- On MacOS, once enabled, you cannot disable the "Run At Startup" option from the app. You have to remove the service by running `/Applications/CellsSync/Contents/MacOS/cells-sync service uninstall` command.
- The "Stop" button is sometimes not working as it should
- MacOS ".keynote" files may be buggy, especially if you are migrating them from an old version of Keynote (stored as a folder) to a newer version (stored as file). Saving the file to a new name should make the process easier.

D – Testing / Reporting Bugs

The Sync library has been already thoroughly tested, especially with a big amount of files and folders. But of course there might still be some glitches. Please make sure to backup everything before starting syncing! We don't want you to loose any data...

Please use Github issues under https://github.com/pydio/cells-sync/issues. Make sure to always provide as much info as possible about your setup:

- CellsSync version
- Desktop OS and version
- Cells Server version
- Scenario of what went wrong, sample files that create problem, etc..
- CellsSync && CellsServer logs if possible (see below)

Cells Server Logs

As before, the logs are located under %APPLICATION_DATA%/logs.

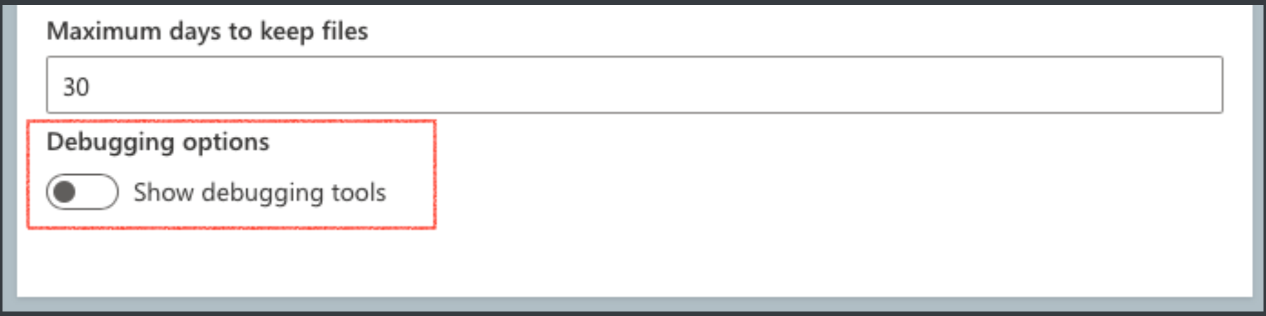
- On windows, %APPLICATION_DATA% = `C:\Users\%username%\AppData\Roaming\Pydio\cells`
- On Linux %APPLICATION_DATA% = `/home/%username%/.config/pydio/cells`
- On Mac %APPLICATION_DATA% = `/Users/%username%/Library/Application Support/Pydio/cell s`

CellsSync Logs

Same as Cells server, the logs are located under %APPLICATION_DATA%/logs.

- On windows, %APPLICATION_DATA% = `C:\Users\%username%\AppData\Roaming\Pydio\cells-sync`
- On Linux %APPLICATION_DATA% = `/home/%username%/.config/pydio/cells-sync`
- On Mac %APPLICATION_DATA% = `/Users/%username%/Library/Application Support/Pydio/cells-sync`

For testing, it can be helpfull to enable the debugging tools inside the CellsSync settings.



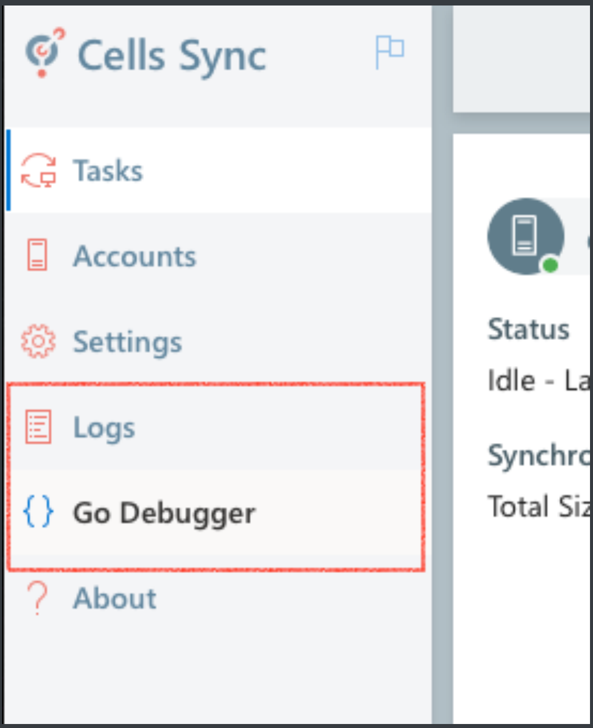
Maximum days to keep files

30

Debugging options

○ Show debugging tools

This will add two more entries in the menu : a log viewer for directly show application logs, and a GO debugger for showing various states of the program. The latter is particularly usefull if you feel that the application is "stuck" in a state where nothing more happens : open the Go debugger and please send us the "goroutines" list that you see in there.


Cells Sync
Tasks
Accounts
Settings
Logs
Go Debugger
About
Status
Idle - La
Synchro
Total Siz