

Digging deep

A short overview on debugging Nextcloud and apps

Julius Härtl
julius@nextcloud.com

Debugging

- Observe a bug → Try to fix → Test → repeat
- `console.log()`
- `\OC::$server→getLogger()→error($myDebugOutput);`
- Time consuming approach

Debugging Basics

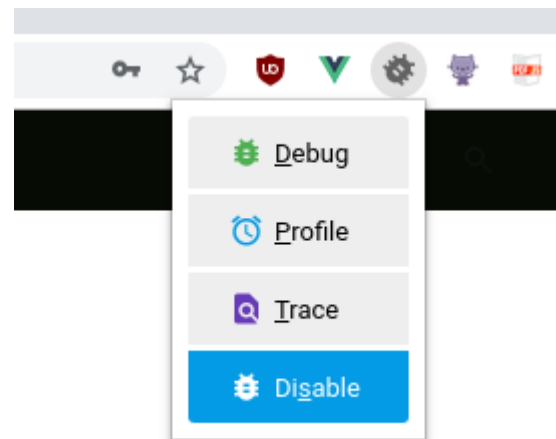
- Breakpoints
- Step over, step into, step out
- Watchers

PHP

- Xdebug keeps you covered

```
xdebug.remote_enable=1  
xdebug.remote_connect_back=1  
xdebug.idekey=PHPSTORM
```

- Browser extension for triggering



PHP Debugger

The screenshot displays a PHP development environment with a code editor at the top and a debugger interface at the bottom. The code editor shows a file named `Jail.php` with the following PHP code:

```
58 if ($path === '') {
59     return $this->rootPath;
60 } else {
61     return Filesystem::normalizePath( $path: $this->rootPath . '/' . $path);
62 }
63
64
65 public function getJailedPath($path) {
66     $root = rtrim($this->rootPath, charlist: '/') . '/';
67
68     if ($path !== $this->rootPath && strpos($path, $root) !== 0) {
69         return null;
70     } else {
71         $path = substr($path, strlen($this->rootPath));
72         return trim($path, charlist: '/');
73     }
74 }
```

The debugger interface at the bottom shows the following components:

- Debug:** A tab labeled "Markdown though editor.ul" and a "docker-debug-sqlite" icon.
- Debugger:** A section with icons for console, variables, and other debugging tools.
- Frames:** A list of call stack frames. The current frame is `Jail.php:68, OC\Files\Storage\Wrapper\Jail->getJailedPath()`. Other frames include `Wrapper.php:506, call_user_func_array()`, `Wrapper.php:506, OC\GroupFolders\Mount\GroupFolderStorage->__call()`, `Wrapper.php:506, OC\GroupFolders\Mount\GroupFolderStorage->getJailedPath()`, `Wrapper.php:506, call_user_func_array()`, `Wrapper.php:506, OC\Files\Storage\Wrapper\PermissionsMask->__call()`, `Wrapper.php:506, OC\Files\Storage\Wrapper\PermissionsMask->getJailedPath()`, `Wrapper.php:506, call_user_func_array()`, `Wrapper.php:506, OC\Files_Trashbin\Storage->__call()`, `Wrapper.php:506, OC\Files_Trashbin\Storage->getJailedPath()`, `Wrapper.php:506, call_user_func_array()`, `Wrapper.php:506, OC\FilesAccessControl\StorageWrapper->__call()`, `Manager.php:77, OC\FilesAccessControl\StorageWrapper->getJailedPath()`, `Manager.php:77, OC\WorkflowEngine\Manager->setFileInfo()`, `Operation.php:66, OC\FilesAccessControl\Operation->checkFileAccess()`, and `CacheWrapper.php:63, OC\FilesAccessControl\CacheWrapper->formatCacheEntry()`.
- Variables:** A window showing the current state of variables. The `$this` variable is highlighted, showing its value as `(OC\Files\Storage\Wrapper\Jail) [7]`. Other variables include `$path = "__groupfolders/1"`, `$root = "__groupfolders/1/"`, `$storage = (OC\GroupFolders\ACL\ACLStorageWrapper) [8]`, `$cache = null`, `$scanner = null`, `$watcher = null`, `$propagator = null`, `$updater = null`, `$_COOKIE = (array) [10]`, `$_ENV = (array) [35]`, `$_REQUEST = (array) [10]`, `$_SERVER = (array) [47]`, `$_SESSION = (array) [1]`, and `$GLOBALS = (array) [22]`.

PHP Debugger

- IDE integration available
 - PHPStorm
 - Visual Studio Code
- `curl http://nextcloud -b XDEBUG_SESSION=PHPSTORM`

PHP Profiling

- Xdebug keeps you covered

```
xdebug.profiler_enable_trigger = 1
```

```
# Add request URI and timestamp to the filename
```

```
xdebug.profiler_output_name = profile.%R.%u
```

PHP Profiling

profile_index_php_apps_files_texteditor_ajax_savefile.1564584592_658634

Server: <no server> Time: ms Refresh

Execution Statistics Call Tree

Callable	Time	Calls
OC\FilesView->changeLock	28 0.0%	1 0.0%
OC\FilesView->writeUpdate	4 0.0%	1 0.0%
OCA\Files_Trashbin\Storage->file_put_contents	1 0.0%	1 0.0%
OC\FilesView->changeLock	0 0.0%	1 0.0%
OC\Files\FileSystem->resolvePath	0 0.0%	1 0.0%
OC\FilesView->runHooks	69,081 98.6%	1 0.0%
OC_Hook->emit	69,081 98.6%	1 0.0%
OC\FilesNode\HookConnector->write	3 0.0%	1 0.0%
OCA\Files_Versions\Hooks->write_hook	69,077 98.6%	1 0.0%
OCA\Files_Versions\Storage->store	69,076 98.6%	1 0.0%
OCA\Files_Versions\VersionManager->createVersion	83 0.1%	1 0.0%
OC\User\Manager->get	0 0.0%	1 0.0%
OC\Server->getUserManager	0 0.0%	1 0.0%
OC\Server->query	4 0.0%	1 0.0%
OC\FilesFileInfo->getSize	0 0.0%	1 0.0%
Symfony\Component\EventDispatcher\EventDispatcher->dispatch	0 0.0%	1 0.0%
Symfony\Component\EventDispatcher\EventDispatcher->dispatch	0 0.0%	1 0.0%
php:spl_autoload_call	0 0.0%	1 0.0%
OC\FilesNode\LazyRoot->getById	68,971 98.5%	1 0.0%
OC\Server->getRootFolder	0 0.0%	1 0.0%
OC\FilesFileInfo->getId	0 0.0%	1 0.0%
OC\FilesView->getFileInfo	1 0.0%	1 0.0%
OC\Server->getEventDispatcher	0 0.0%	1 0.0%
OC\FilesView->__construct	0 0.0%	1 0.0%
OCA\Files_Versions\Storage->getUidAndFilename	10 0.0%	1 0.0%
OC\Files\FileSystem->is_dir	1 0.0%	1 0.0%
OC\Files\FileSystem->file_exists	1 0.0%	1 0.0%
php:pathinfo	0 0.0%	1 0.0%
OCA\Files_Versions\Events\CreateVersionEvent->shouldCreateVersion	0 0.0%	2 0.0%
OCA\Files_Versions\Events\CreateVersionEvent->__construct	0 0.0%	2 0.0%
php:spl_autoload_call	0 0.0%	1 0.0%
OC\Files\FileSystem->isBlacklisted	0 0.0%	1 0.0%
php:array_key_exists	0 0.0%	2 0.0%
OC_Hook->emit	0 0.0%	1 0.0%
OC\FilesView->shouldEmitHooks	0 0.0%	1 0.0%
OC\FilesView->getHookPath	0 0.0%	1 0.0%
OC\FilesView->lockFile	9 0.0%	1 0.0%
OC\FilesView->getRelativePath	0 0.0%	1 0.0%

Callers

Callable	Time	Calls
OC\FilesNode\LazyRoot->getById	68,971 98.5%	1 0.0%
OC\FilesNode\LazyRoot->__call	68,999 98.5%	7 0.0%
php:call_user_func({var/www/html/nextcloud/lib/private/Files/Node/LazyRoot.php:61})	3 0.0%	1 0.0%
php:call_user_func_array({var/www/html/nextcloud/lib/private/Files/Node/LazyRoot.php:64})	68,995 98.5%	7 0.0%

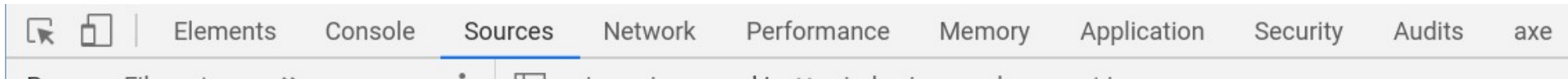


PHP Profiling

▼ m OC_Hook::emit	69,081	98.6%	1 0.0%
▶ m OC\Files\Node\HookConnector->write	3	0.0%	1 0.0%
▼ m OCA\Files_Versions\Hooks::write_hook	69,077	98.6%	1 0.0%
▼ m OCA\Files_Versions\Storage::store	69,076	98.6%	1 0.0%
▶ m OCA\Files_Versions\Versions\VersionManager->createVersion	83	0.1%	1 0.0%
▶ m OC\User\Manager->get	0	0.0%	1 0.0%
▶ m OC\Server->getUserManager	0	0.0%	1 0.0%
▶ m OC\Server->query	4	0.0%	1 0.0%
▶ m OC\Files\FileInfo->getSize	0	0.0%	1 0.0%
▶ m Symfony\Component\EventDispatcher\EventDispatcher->dispatch	0	0.0%	1 0.0%
▶ m Symfony\Component\EventDispatcher\EventDispatcher->dispatch	0	0.0%	1 0.0%
▶ fx php::spl_autoload_call	0	0.0%	1 0.0%
▶ m OC\Files\Node\LazyRoot->getId	68,971	98.5%	1 0.0%
▶ m OC\Server->getRootFolder	0	0.0%	1 0.0%
▶ m OC\Files\FileInfo->getId	0	0.0%	1 0.0%

JavaScript debugger

- Browser have a powerful toolset
- There is more than a console and element inspector



JavaScript debugger

The screenshot shows a web browser's developer tools interface. The 'Sources' panel is active, displaying the code of viewer.js. A breakpoint is set at line 46, and the debugger is paused. The 'Call Stack' and 'Scope' panels are visible on the right.

Call Stack:

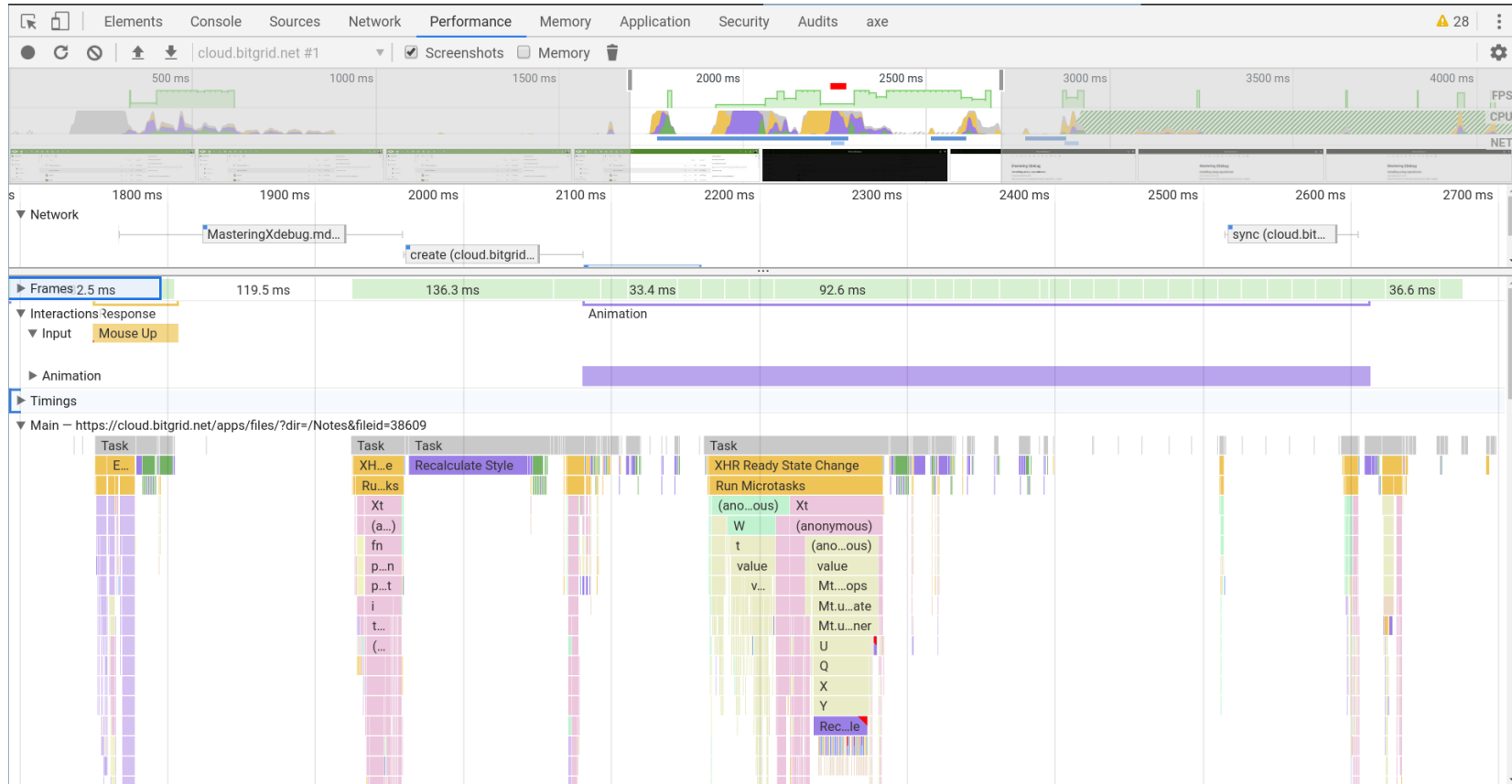
Function	File	Line
s	url.js	46
showEditor	document.js	160
(anonymous)	document.js	146
setTimeout (async)		
showEditor	document.js	146
loadDocument	document.js	396
(anonymous)	document.js	391
c	jquery.js	3099
add	jquery.js	3145
initSession	document.js	390
(anonymous)	document.js	462
c	jquery.js	3099
fireWith	jquery.js	3211
ready	jquery.js	3417
P	jquery.js	3433

Scope:

Variable	Value
c	"https://cloud.bitgrid.net/index.php/a..."
u	"https%3A%2F%2Fcloud.bitgrid.net%2Ffind..."
a	true
e	"811184_oc4e6adde34d"
n	false
s	true
t	"New Document.odt"
this	undefined



Profiling



Vue.js

- Vue.js devtools
- Inspect components
- Observe vuex state changes
- Watch events
- Performance measurements

Now

- Use a debugger
- Profile all the things