



Un ami qui vous veut du bien...

24/02/2017 @b_viguier

Echo Debug
À l'ancienne

I ❤ Echo

XDebug

Beaucoup plus de possibilités



Qu'est ce que c'est ?

- « *The Xdebug extension helps you debugging your script by providing a lot of valuable debug information* »
- Derick Rethans [@derickr](#)
- Existe depuis 2002 (Php 4)

```
43
44
45     * NOTE: For standard formats, Symfony will also automatically choose the best
46     * Content-Type header for the response.
47     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
48     */
49     public function indexAction($page, $_format) $page: "1" $_format: "html"
50 {
51     $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
52
53     // Every template name also has two extensions that specify the format and
54     // engine for that template.
55     // See https://symfony.com/doc/current/templating.html#template-suffix
56     return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
57 }
58
59 /**
60 * @Route("/posts/{slug}", name="blog_post")
61 * @Method("GET")
62 *
63 * NOTE: The $post controller argument is automatically injected by Symfony
64 * after performing a database query looking for a Post with the $slug.
```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "1"
- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

#SPOILER



Installation

C'est le premier pas qui coûte...

Il existe un package pour ça...

- Unix: `sudo apt-get install php7-xdebug`
- OsX: `brew install php7-xdebug`
- Windows: <http://www.ubuntu.com/download>

Pour être sûr...

- `zend_extension= "/wherever/you/put/it/xdebug.so"`
- `php -v` => with Xdebug v2.4.1, Copyright (c) 2002-2016,
by Derick Rethans
- `phpinfo()` =>

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
with Xdebug v2.4.1, Copyright (c) 2002-2016, by Derick Rethans

Avant

Fatal error: Uncaught Error: Call to a member function hello() on null in
/Users/bviguier/workspace/symfony-demo/web/app_dev.php:4 Stack trace: #0
/Users/bviguier/workspace/symfony-
demo/vendor/symfony/symfony/src/Symfony/Bundle/FrameworkBundle/Resources/config/router_dev.php(40):
require() #1 {main} thrown in **/Users/bviguier/workspace/symfony-demo/web/app_dev.php** on line **4**

Après

(!)	Fatal error: Uncaught Error: Call to a member function hello() on null in /Users/bviguier/workspace/symfony-demo/web/app_dev.php on line 4			
(!)	Error: Call to a member function hello() on null in /Users/bviguier/workspace/symfony-demo/web/app_dev.php on line 4			
Call Stack				
#	Time	Memory	Function	Location
1	0.0005	364384	{main}()	.../router_dev.php:0
2	0.0009	372904	require('/Users/bviguier/workspace/symfony-demo/web/app_dev.php')	.../router_dev.php:40

Mais bon...



[Need support?](#)

“

Call to a member function hello() on null

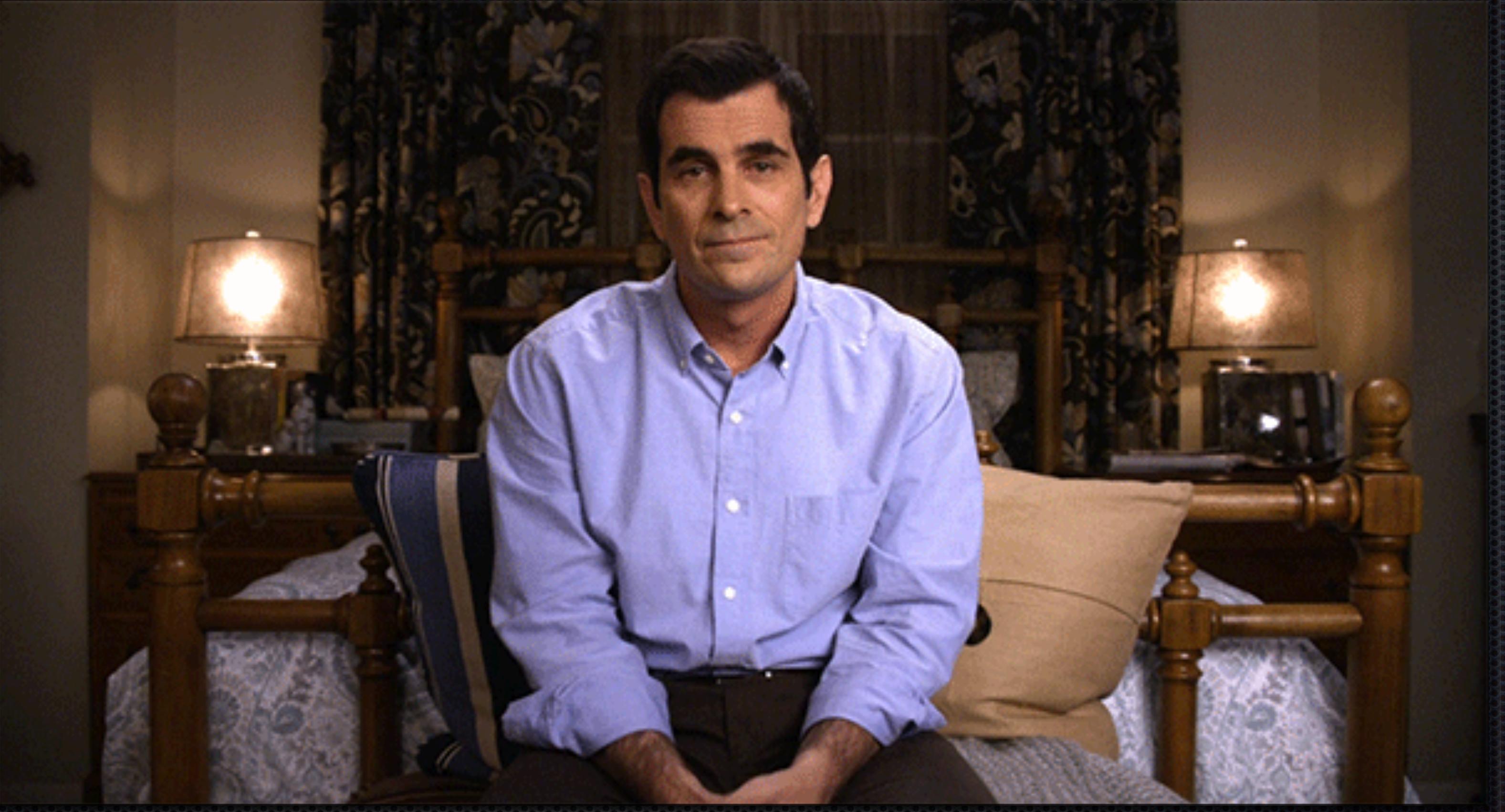
500 Internal Server Error - FatalThrowableError

”

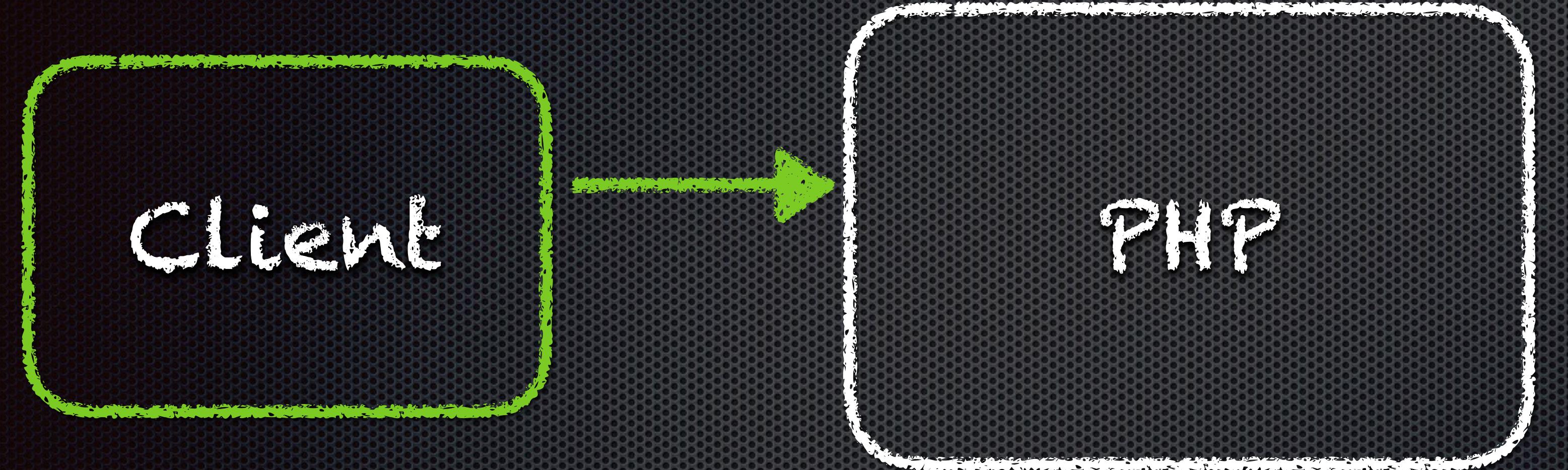
Stack Trace

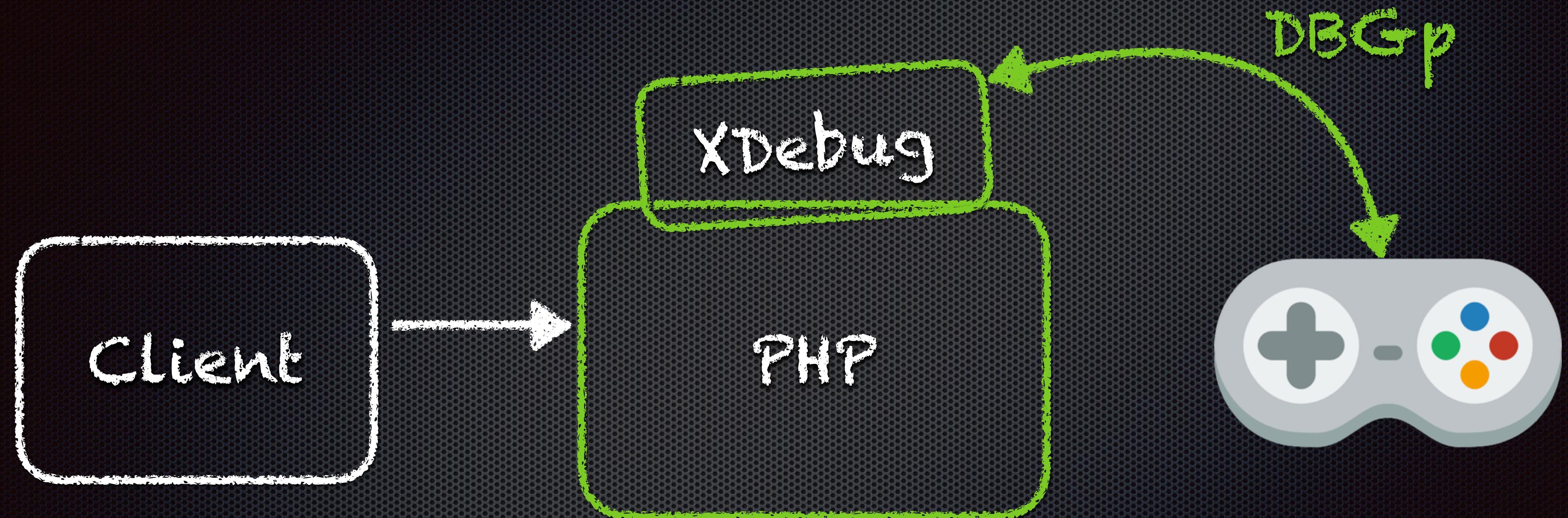
1. in src/AppBundle/Controller/BlogController.php at line 52 -

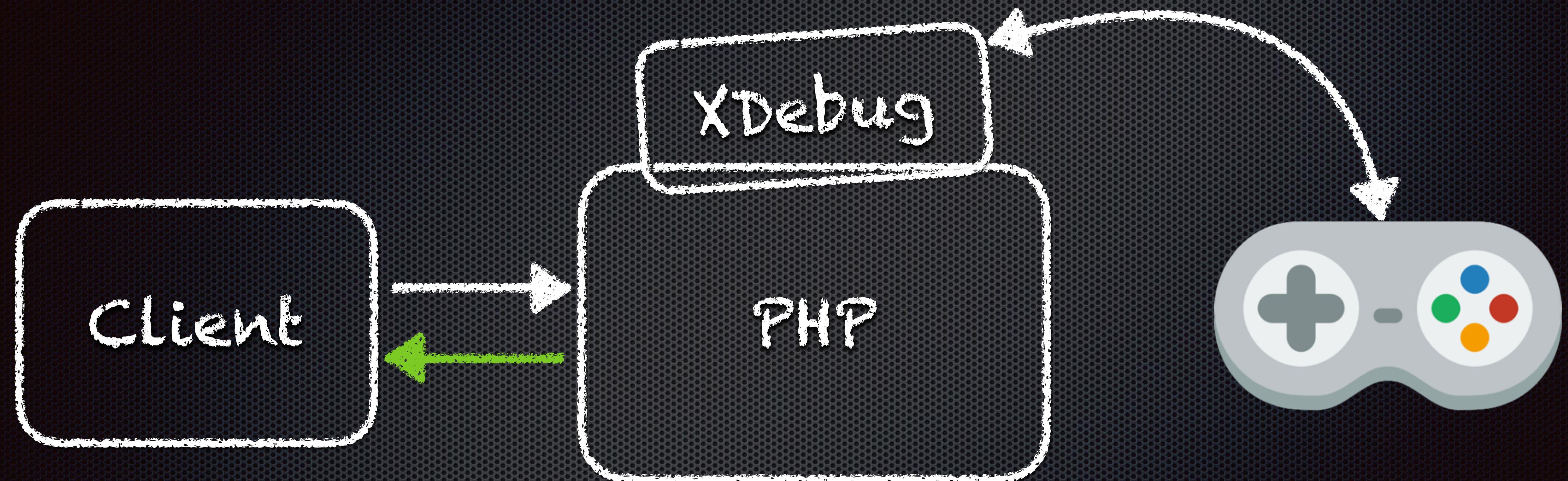
```
49.     public function indexAction($page, $_format)
50. {
51.     $a = null;
52.     $a->hello();
53.     $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
54.
55.     // Every template name also has two extensions that specify the format and
```

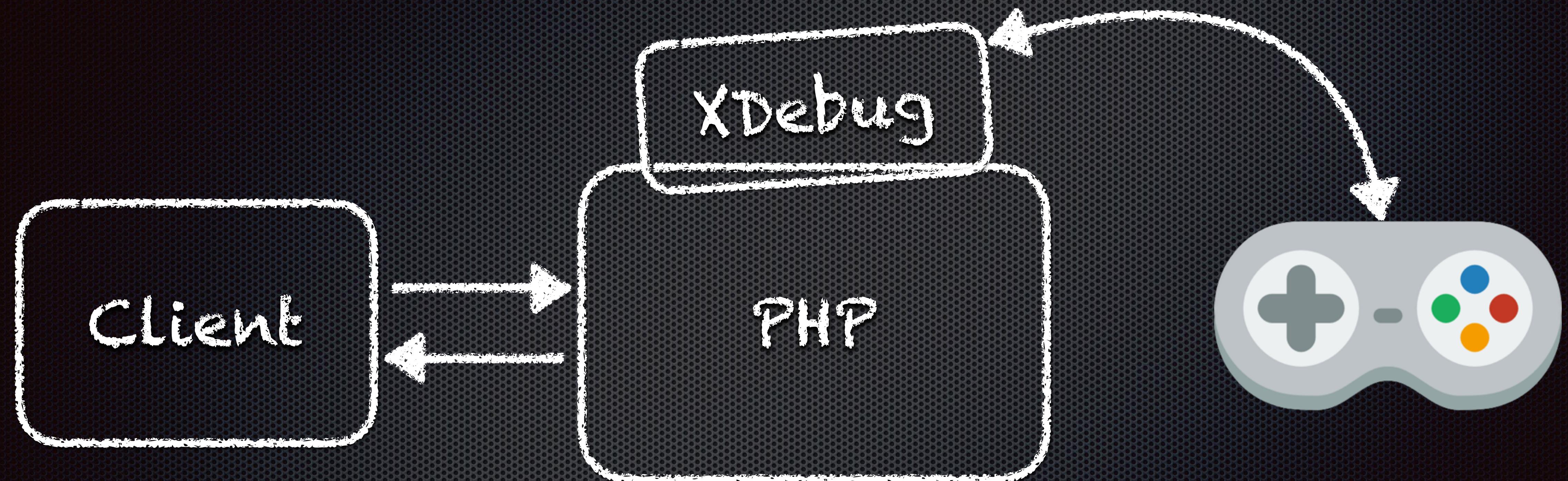


Comment ça marche
Ce qu'il y a sous le capot...







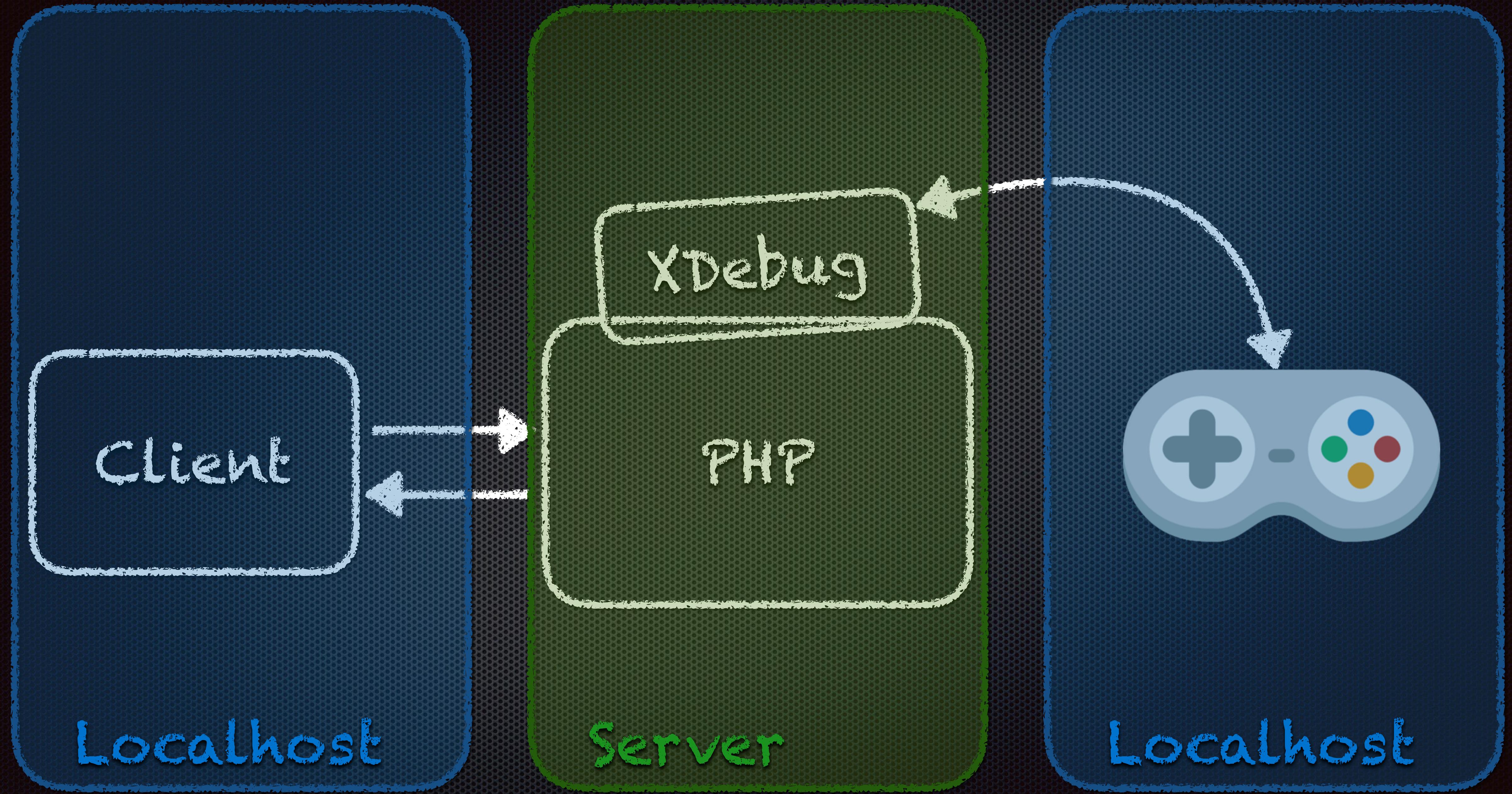


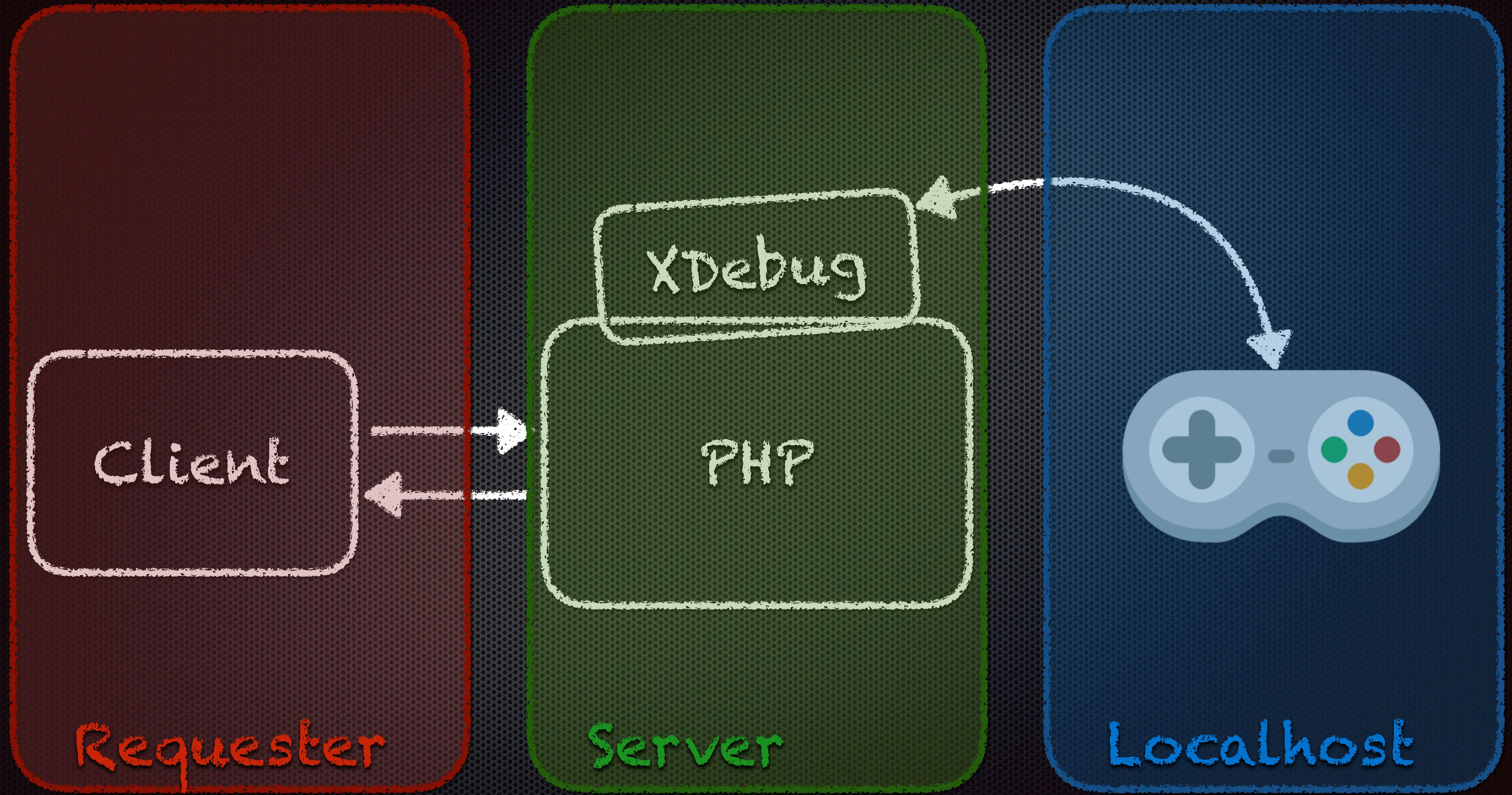
Client

XDebug
PHP



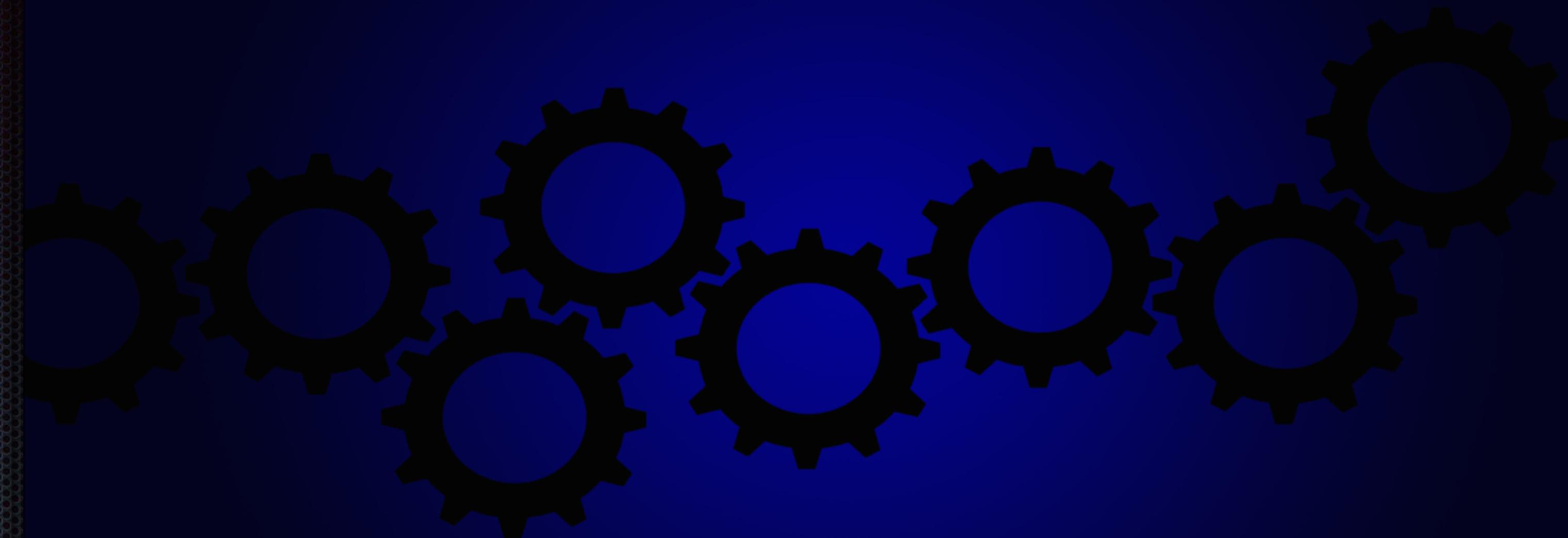
localhost





Configuration

Les paramètres indispensables...



Documentation

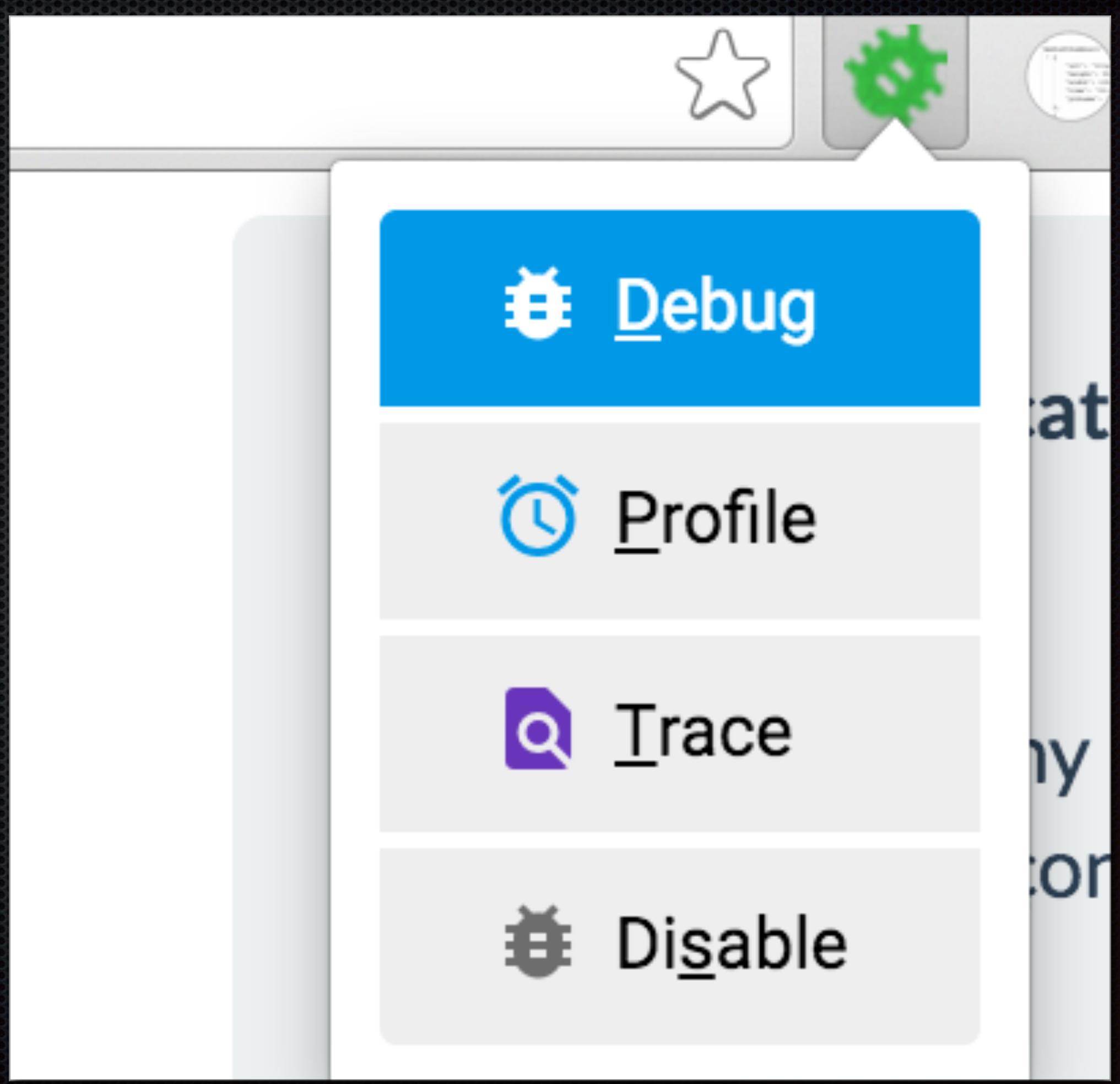
Bien mais pas top

Tenu x
mieux x
faire A.B.

Demandé
être placé de
l'attribution
des cours
au lieu de
le pavillon A

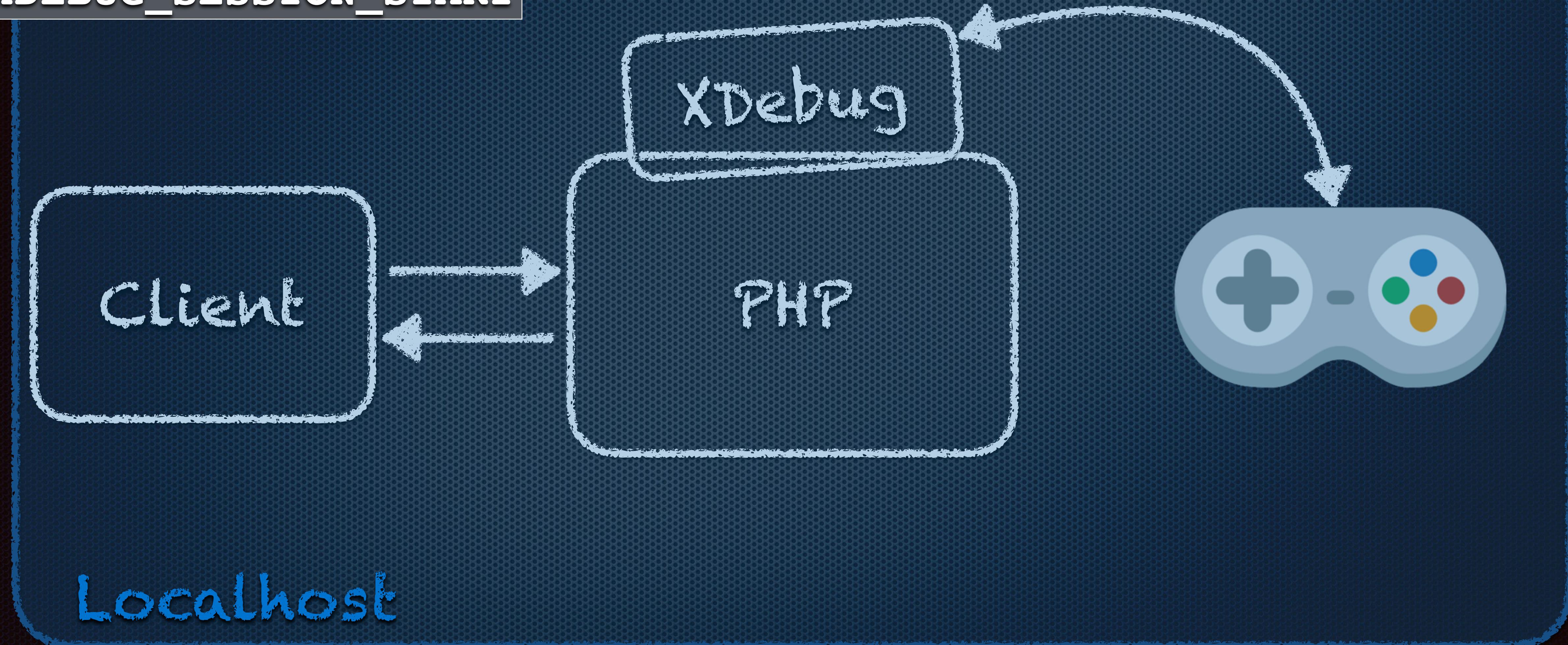
Activer XDebug

- xdebug.remote_autostart
- xdebug.remote_enable
 - browser extension
 - XDEBUG_SESSION_START
- ...



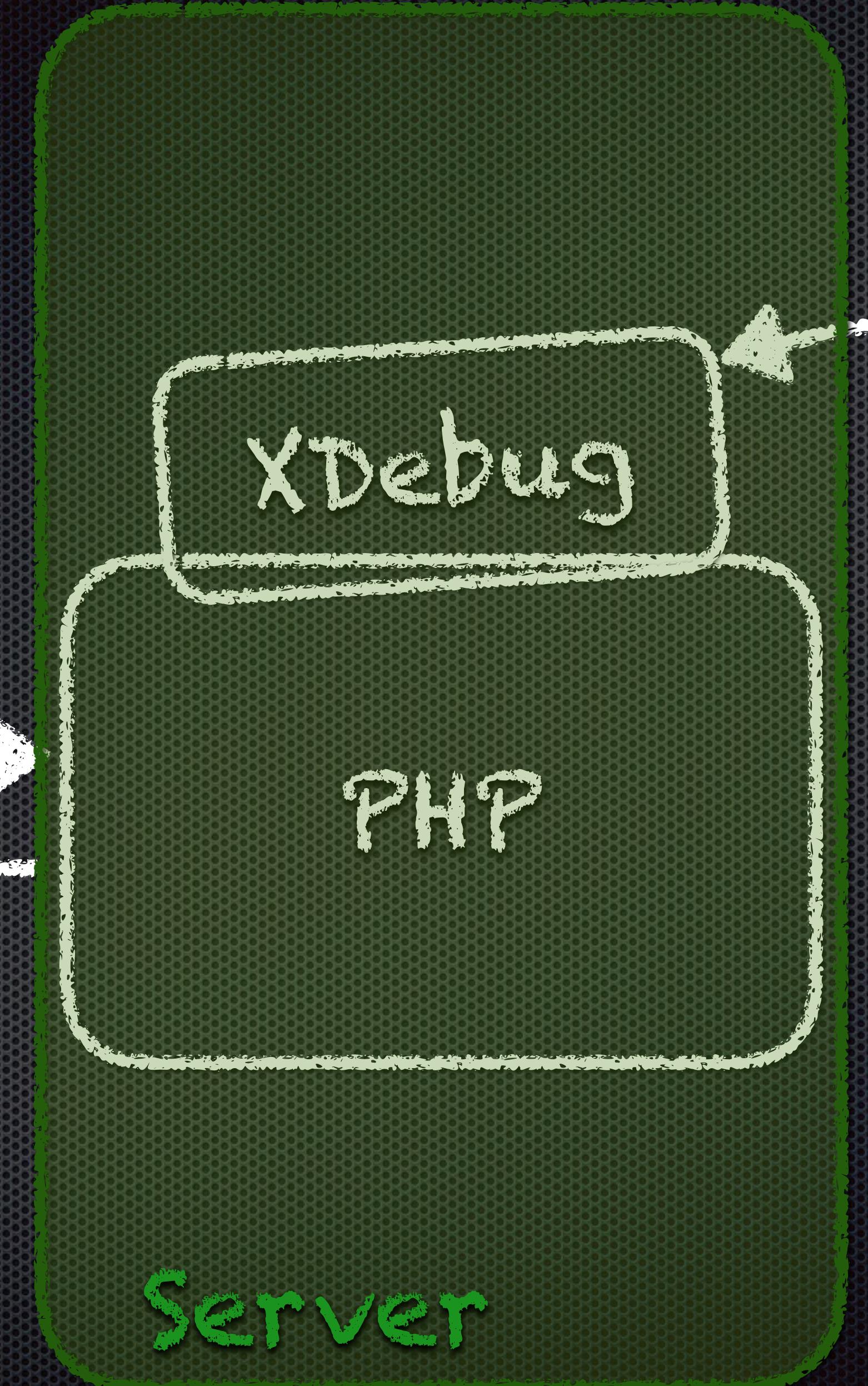
`xdebug.
remote_enable`

`XDEBUG_SESSION_START`



`xdebug.
remote_enable`

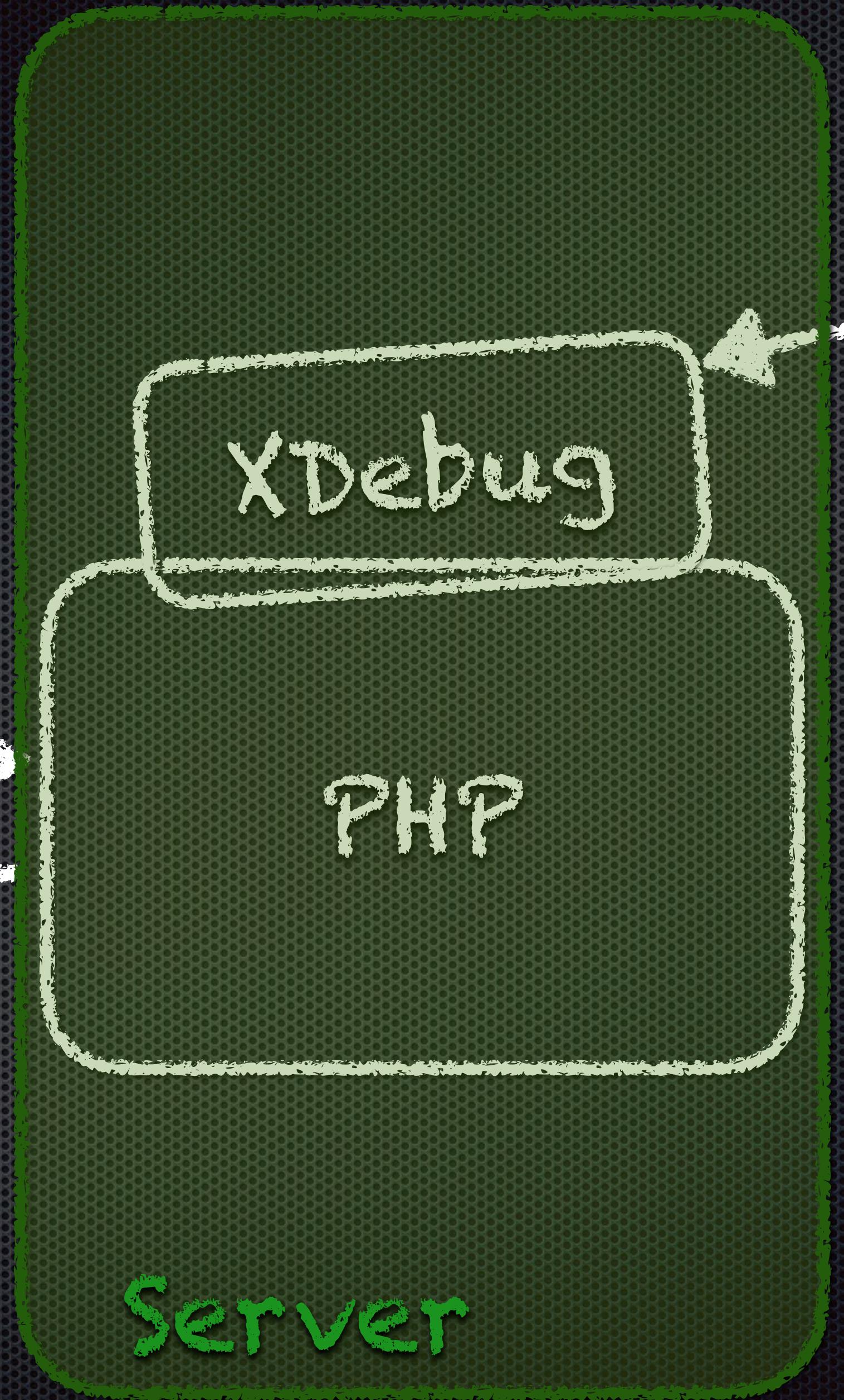
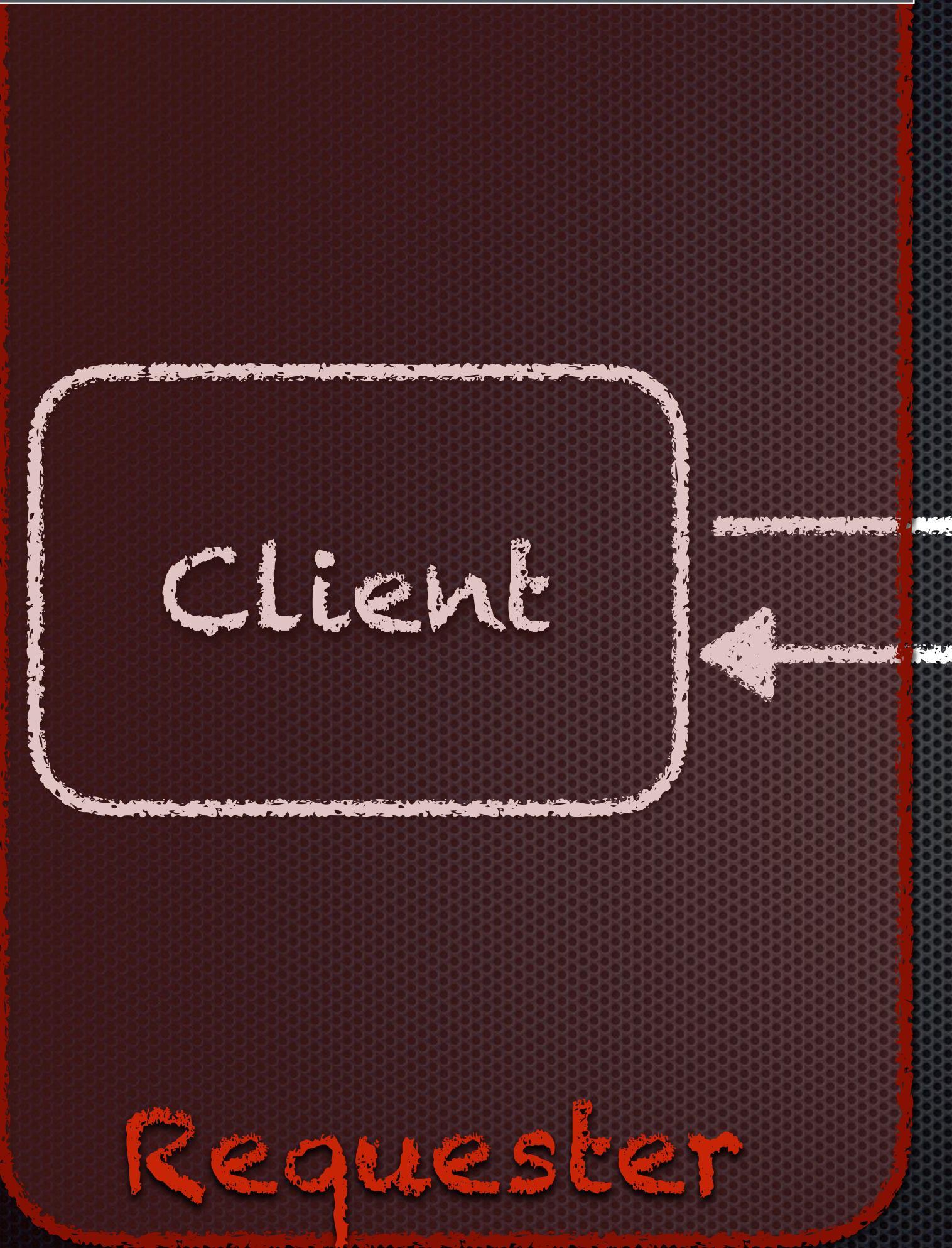
`XDEBUG_SESSION_START`



`xdebug.
remote_connect_back`



`xdebug.
autostart`



`xdebug.
remote_host
remote_port`





Utilisation
DBGp inside...

Client CLI

Xdebug Simple DBGp client (0.10.0)

Copyright 2002-2007 by Derick Rethans.

- libedit support: enabled

Waiting for debug server to connect.

Connect

```
<?xml version="1.0" encoding="iso-8859-1"?>
<init xmlns="urn:debugger_protocol_v1" xmlns:xdebug="http://
xdebug.org/dbgp/xdebug" fileuri="file:///Users/bviguier/workspace/
symfony-demo/vendor/symfony/symfony/src/Symfony/Bundle/
FrameworkBundle/Resources/config/router_dev.php" language="PHP"
xdebug:language_version="7.0.9" protocol_version="1.0"
appid="24676" idekey="PHPSTORM"><engine version="2.4.1"><!
[CDATA[Xdebug]]></engine><author><! [CDATA[Derick Rethans]]></
author><url><! [CDATA[http://xdebug.org]]></url><copyright><!
[CDATA[Copyright (c) 2002-2016 by Derick Rethans]]></copyright></
init>
(cmd)
```

```
(cmd) status -i 1
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<response xmlns="urn:debugger_protocol_v1"
xmlns:xdebug="http://xdebug.org/dbgp/xdebug"
command="status" transaction_id="1"
status="starting" reason="ok"></response>
```

```
(cmd) run -i 1
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<response xmlns="urn:debugger_protocol_v1"
xmlns:xdebug="http://xdebug.org/dbgp/xdebug"
command="run" transaction_id="1" status="break"
reason="ok"><xdebug:message filename="file:///Users/
bviquier/workspace/symfony-demo/web/app_dev.php"
lineno="30"></xdebug:message></response>
```

```
(cmd) stack_get -i 1
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<response xmlns="urn:debugger_protocol_v1"
xmlns:xdebug="http://xdebug.org/dbgp/xdebug"
command="stack_get" transaction_id="1"><stack
where="require" level="0" type="file" filename="file:///Users/bviguier/workspace/symfony-demo/web/app_dev.php"
lineno="30"></stack><stack where="{main}" level="1"
type="file" filename="file:///Users/bviguier/workspace/
symfony-demo/vendor/symfony/symfony/src/Symfony/Bundle/
FrameworkBundle/Resources/config/router_dev.php"
lineno="40"></stack></response>
```

```
(cmd) run -i 1
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<response xmlns="urn:debugger_protocol_v1"
xmlns:xdebug="http://xdebug.org/dbgp/xdebug"
command="run" transaction_id="1" status="stopping"
reason="ok"></response>
```

```
(cmd) run -i 1
Disconnect
```





PhpStorm
Quand les outils s'accordent...

Server name: 127.0.0.1
Server port: 8000
Request uri: /fr/blog/
File path on server: bvguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Bundle/FrameworkBundle/Resources/config/router_dev.php

Configure local file path

Import mappings from deployment Manually choose local file or project

Select a project or a file to debug

▼  symfony-demo
 /Users/bvguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Bundle/FrameworkBundle/Resources/config/router_dev.php



Ignore

Accept

Preferences

Languages & Frameworks > PHP > Servers > For current project

Name: 127.0.0.1 Shared

Host: 127.0.0.1 Port: 80 Debugger: Xdebug

Use path mappings (select if the server is remote or symlinks are used)

File/Directory	Absolute path on the server
Project files	
/Users/bviguier/workspace/symfony-demo	/other/server/path
Include path	

Servers

- Behat
- Code Sniffer
- Composer
- Mess Detector
- Phing
- PHPUnit
- Blade
- Drupal
- Google App Engine for PHP
- Joomla! Support
- Smarty
- WordPress

Cancel Apply OK

Navigation

-  Step Over
-  Force Step Over
-  Step Into
-  Force Step Into
-  Smart Step Into
-  Step Out
-  Run to Cursor
-  Force Run to Cursor
-  Resume Program

```
43
44
45     *
46     * NOTE: For standard formats, Symfony will also automatically choose the best
47     * Content-Type header for the response.
48     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
49     */
50
51     public function indexAction($page, $_format)
52     {
53         $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
54
55         // Every template name also has two extensions that specify the format and
56         // engine for that template.
57         // See https://symfony.com/doc/current/templating.html#template-suffix
58         return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
59     }
60
61     /**
62      * @Route("/posts/{slug}", name="blog_post")
63      * @Method("GET")
64      *
65      * NOTE: The $post controller argument is automatically injected by Symfony
66      * after performing a database query looking for a Post with the $slug
67  
```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "1"
- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [13]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
145  
146  
147     $event = new FilterControllerArgumentsEvent($this, $controller, $arguments, $request, $type);  
148     $this->dispatcher->dispatch(KernelEvents::CONTROLLER_ARGUMENTS, $event);  
149     $controller = $event->getController();  
150     $arguments = $event-> getArguments();  
151  
152     // call controller  
153     $response = call_user_func_array($controller, $arguments);  
154  
155     // view  
156     if (!$response instanceof Response) {  
157         $event = new GetResponseForControllerResultEvent($this, $request, $type, $response);  
158         $this-> dispatcher-> dispatch(KernelEvents::VIEW, $event);  
159  
160         if ($event-> hasResponse()) {  
161             $response = $event->getResponse();  
162         }  
163  
164         if (!$response instanceof Response) {  
165             $msg = sprintf('The controller must return a response (%s given).', $this-> varToString($response));  
166     }
```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
27
28
29     }
30
31     if (is_file($_SERVER['DOCUMENT_ROOT'].$_SERVER['SCRIPT_NAME'])) {
32         return false;
33     }
34
35     $_SERVER = array_merge($_SERVER, $_ENV);  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
36     $_SERVER['SCRIPT_FILENAME'] = $_SERVER['DOCUMENT_ROOT'].$_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
37
38     // Since we are rewriting to app_dev.php, adjust SCRIPT_NAME and PHP_SELF accordingly
39     $_SERVER['SCRIPT_NAME'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
40     $_SERVER['PHP_SELF'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
41
42     require 'app_dev.php';
43
44     error_log(sprintf('%s:%d [%d]: %s', $_SERVER['REMOTE_ADDR'], $_SERVER['REMOTE_PORT'], http_response_code(), $_SERVER['REQUEST_URI']), 4);
45
```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$_composer_autoload_files = {array} [10]
- \$kernel = {AppKernel} [10]
- \$loader = {Composer\Autoload\ClassLoader} [10]
- \$request = {Symfony\Component\HttpFoundation\Request} [28]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```

27
28
29     }
30
31     if (is_file($_SERVER['DOCUMENT_ROOT'].$_SERVER['SCRIPT_NAME'])) {
32         return false;
33     }
34
35     $_SERVER = array_merge($_SERVER, $_ENV);  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
36     $_SERVER['SCRIPT_FILENAME'] = $_SERVER['DOCUMENT_ROOT'].$_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
37
38     // Since we are rewriting to app_dev.php, adjust SCRIPT_NAME and PHP_SELF accordingly
39     $_SERVER['SCRIPT_NAME'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
40     $_SERVER['PHP_SELF'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
41     require 'app_dev.php';
42
43     error_log(sprintf('%s:%d [%d]: %s', $_SERVER['REMOTE_ADDR'], $_SERVER['REMOTE_PORT'], http_response_code(), $_SERVER['REQUEST_URI']), 4);

```

Debug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()**

Variables

- \$_composer_autoload_files = {array} [10]
- \$kernel = {AppKernel} [10]
- \$loader = {Composer\Autoload\ClassLoader} [10]
- \$request = {Symfony\Component\HttpFoundation\Request} [28]
 - acceptableContentTypes = null
 - attributes = {Symfony\Component\HttpFoundation\ParameterBag} [1]
 - basePath = null
 - baseUrl = ""
 - charsets = null
 - content = null
 - cookies = {Symfony\Component\HttpFoundation\ParameterBag} [1]
 - defaultLocale = "en"
 - encodings = null
 - files = {Symfony\Component\HttpFoundation\FileBag} [2]
 - format = null
 - formats = null
 - headers = {Symfony\Component\HttpFoundation\HeaderBag} [2]
 - httpMethodParameterOverride = true

```

27
28
29     }
30
31     if (is_file($_SERVER['DOCUMENT_ROOT'].$_SERVER['SCRIPT_NAME'])) {
32         return false;
33     }
34
35     $_SERVER = array_merge($_SERVER, $_ENV);  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
36     $_SERVER['SCRIPT_FILENAME'] = $_SERVER['DOCUMENT_ROOT'].$_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
37
38     // Since we are rewriting to app_dev.php, adjust SCRIPT_NAME and PHP_SELF accordingly
39     $_SERVER['SCRIPT_NAME'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';
40     $_SERVER['PHP_SELF'] = $_SERVER['DIRECTORY_SEPARATOR'].'app_dev.php';  $_SERVER: {DOCUMENT_ROOT => "/Users/bviguier/workspace/symfony-demo/web", REMOTE_ADDR => "127.0.0.1", REMOTE_
41     require 'app_dev.php';
42
43     error_log(sprintf('%s:%d [%d]: %s', $_SERVER['REMOTE_ADDR'], $_SERVER['REMOTE_PORT'], http_response_code(), $_SERVER['REQUEST_URI']), 4);

```

Debug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$_composer_autoload_files = {array} [10]
- \$kernel = {AppKernel} [10]
- \$loader = {Composer\Autoload\ClassLoader} [10]
- \$request = {Symfony\Component\HttpFoundation\Request} [28]
 - acceptableContentTypes = null
 - attributes = {Symfony\Component\HttpFoundation\ParameterBag} [1]
 - can not get property
 - basePath = null
 - baseUrl = ""
 - charsets = null
 - content = null
- cookies = {Symfony\Component\HttpFoundation\ParameterBag} [1]
 - defaultLocale = "en"
 - encodings = null
- files = {Symfony\Component\HttpFoundation\FileBag} [2]
 - format = null
 - formats = null
- headers = {Symfony\Component\HttpFoundation\HeaderBag} [2]

```
43
44
45     *
46     * NOTE: For standard formats, Symfony will also automatically choose the best
47     * Content-Type header for the response.
48     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
49     */
50
51     public function indexAction($page, $_format) $page: "1" $_format: "html"
52     {
53         $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
54
55         // Every template name also has two extensions that specify the format and
56         // engine for that template.
57         // See https://symfony.com/doc/current/templating.html#template-suffix
58         return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
59     }
60
61     /**
62      * @Route("/posts/{slug}", name="blog_post")
63      * @Method("GET")
64      *
65      * NOTE: The $post controller argument is automatically injected by Symfony
66      * after performing a database query looking for a Post with the $slug.
67  
```

Debug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "1"
- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
43
44
45     *
46     * NOTE: For standard formats, Symfony will also automatically choose the best
47     * Content-Type header for the response.
48     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
49     */
50
51     public function indexAction($page, $_format) $page: "1" $_format: "html"
52     {
53         $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
54
55         // Every template name also has two extensions that specify the format and
56         // engine for that template.
57         // See https://symfony.com/doc/current/templating.html#template-suffix
58         return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
59     }
60
61     /**
62      * @Route("/posts/{slug}", name="blog_post")
63      * @Method("GET")
64      *
65      * NOTE: The $post controller argument is automatically injected by Symfony
66      * after performing a database query looking for a Post with the $slug.
67  
```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "1"
- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
43
44
45     *
46     * NOTE: For standard formats, Symfony will also automatically choose the best
47     * Content-Type header for the response.
48     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
49     */
50
51     public function indexAction($page, $_format) $page: "3" $_format: "html"
52     {
53         $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);
54
55         // Every template name also has two extensions that specify the format and
56         // engine for that template.
57         // See https://symfony.com/doc/current/templating.html#template-suffix
58         return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
59     }
60
61     /**
62      * @Route("/posts/{slug}", name="blog_post")
63      * @Method("GET")
64      *
65      * NOTE: The $post controller argument is automatically injected by Symfony
66      * after performing a database query looking for a Post with the $slug
67  
```

Debug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "3"
- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
349  
350  
351       *  
352       * @return Registry  
353       *  
354       * @throws \LogicException If DoctrineBundle is not available  
355       */  
356       protected function getDoctrine()  
357       {  
358           if (!$this->container->has('doctrine')) { container: appDevDebugProjectContainer  
359               throw new \LogicException('The DoctrineBundle is not registered in your application.');//  
360           }  
361           return $this->container->get('doctrine');  
362       }  
363  
364       /**  
365        * Get a user from the Security Token Storage.  
366        *  
367        * @return mixed  
368        *  
369        * @throws \LogicException If SecurityBundle is not available  
370        */
```

Debug router_dev.php

Debugger Console

Frames

- Controller.php:357, Symfony\Bundle\FrameworkBundle\Controller\Controller->getDoctrine()
- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library/HTMLPurifier/PureParser.php:153
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
349  
350  
351       *  
352       * @return Registry  
353       *  
354       * @throws \LogicException If DoctrineBundle is not available  
355       */  
356       protected function getDoctrine()  
357      {  
358          if (!$this->container->has('doctrine')) {  
359              throw new \LogicException('The DoctrineBundle is not registered in your application.');//  
360          }  
361          return $this->container->get('doctrine'); // container: appDevDebugProjectContainer  
362      }  
363  
364     /**  
365      * Get a user from the Security Token Storage.  
366      *  
367      * @return mixed  
368      *  
369      * @throws \LogicException If SecurityBundle is not available  
370     */
```

Debug *router_dev.php

Debugger Console

Frames

- Controller.php:361, Symfony\Bundle\FrameworkBundle\Controller\Controller->getDoctrine()
- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library/HTMLPurifier/PureData.php:153
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$this = {AppBundle\Controller\BlogController} [1]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$_GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
234
235     }
236
237     /**
238     * {@inheritDoc}
239     */
240     public function getRepository($persistentObjectName, $persistentManagerName = null) $persistentObjectName: "AppBundle\Entity\Post" $persistentMan
241     {
242         return $this->getManager($persistentManagerName)->getRepository($persistentObjectName);
243     }
244
245     /**
246     * {@inheritDoc}
247     */
248     public function resetManager($name = null)
249     {
250         if (null === $name) {
251             $name = $this->defaultManager;
252         }
253
254         if (!isset($this->managers[$name])) {
255             throw new \Doctrine\ORM\InvalidDocumentException(sprintf('Doctrine: No Manager named "%s" does not exist. (%s)', $name));
256         }
257     }
258 }
```

Debug router_dev.php

Debugger Console

Frames

- AbstractManagerRegistry.php:242, Doctrine\Common\Persistence\AbstractManagerRegistry
- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpFoundation\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpFoundation\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$PersistentManagerName = null
- \$PersistentObjectName = "AppBundle\Entity\Post"
- \$this = {Doctrine\Bundle\DoctrineBundle\Registry} [7]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
32
33     /**
34      * @param int $page  $page: "3"
35      *
36      * @return Pagerfanta
37      */
38     public function findLatest($page = 1)  $page: "3"
39     {
40         $query = $this->getEntityManager()
41             ->createQuery('
42                 SELECT p, t
43                 FROM AppBundle:Post p
44                 LEFT JOIN p.tags t
45                 WHERE p.publishedAt <= :now
46                 ORDER BY p.publishedAt DESC
47             ')
48             ->setParameter('now', new \DateTime())
49             ;
50
51         return $this->createPaginator($query, $page);
52     }

```

Debug *router_dev.php

Debugger Console

Frames

- PostRepository.php:39, AppBundle\Repository\PostRepository->findLatest()
- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:/Users/bviguier/workspace/symfony-demo/vendor
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$page = "3"
- \$this = {AppBundle\Repository\PostRepository} [3]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [14]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
43
44
45     * NOTE: For standard formats, Symfony will also automatically choose the best
46     * Content-Type header for the response.
47     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
48     */
49     public function indexAction($page, $_format) $page: "3" $_format: "html"
50 {
51     $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page); $page: "3" $posts: {adapter => Pagerfanta\Adapter\DoctrineORMAd
52
53         // Every template name also has two extensions that specify the format and
54         // engine for that template.
55         // See https://symfony.com/doc/current/templating.html#template-suffix
56         return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
57
58
59     /**
60     * @Route("/posts/{slug}", name="blog_post")
61     * @Method("GET")
62     *
63     * NOTE: The $post controller argument is automatically injected by Symfony
64     * after performing a database query looking for a Post with the $slug

```

Debug router_dev.php

Debugger Console

Frames

- BlogController.php:56, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array{/Users/bviguier/workspace/symfony-demo/vendor/symfony/http-kernel/HttpKernel.php:153, call_user_func_array}
- HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main()}

Variables

- \$_format = "html"
- \$page = "3"
- \$posts = {Pagerfanta\Pagerfanta} [7]
 - adapter = {Pagerfanta\Adapter\DoctrineORMAdapter} [1]
 - allowOutOfRangePages = false
 - currentPage = 3
 - currentPageResults = null
 - maxPerPage = 10
 - nbResults = 30
 - normalizeOutOfRangePages = false
 - this = {AppBundle\Controller\BlogController} [1]
 - \$_COOKIE = {array} [3]
 - \$_SERVER = {array} [25]
 - \$_GLOBALS = {array} [14]
 - HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

13:23 Debug session was finished without being paused

It may be caused by path mappings misconfiguration or not synchronized local and remote projects.

To figure out the problem check path mappings configuration for '127.0.0.1' server at [PHP|Servers](#) or enable [Break at first line in PHP scripts](#) option (from Run menu).

[Do not show again](#)

13:23 Debug session was finished without being paused

It may be caused by path mappings misconfiguration or not synchronized local and remote projects.

To figure out the problem check path mappings configuration for '127.0.0.1' server at [PHP|Servers](#) or enable [Break at first line in PHP scripts](#) option (from Run menu).

[Do not show again](#)

13:23 Debug session was finished without being paused

It may be caused by path mappings misconfiguration or not synchronized local and remote projects.

To figure out the problem check path mappings configuration for '127.0.0.1' server at [PHP|Servers](#) or enable [Break at first line in PHP scripts](#) option (from Run menu).

[Do not show again](#)

13:23 Debug session was finished without being paused

It may be caused by path mappings misconfiguration or not synchronized local and remote projects.

To figure out the problem check path mappings configuration for '127.0.0.1' server at [PHP|Servers](#) or enable [Break at first line in PHP scripts](#) option (from Run menu).

[Do not show again](#)



Symfony
Subtilités...

```
\AppBundle\Controller\BlogController indexAction
{
    /**
     * @Route("/", defaults={"page": "1", "_format": "html"}, name="blog_index")
     * @Route("/rss.xml", defaults={"page": "1", "_format": "xml"}, name="blog_rss")
     * @Route("/page/{page}", defaults={"_format": "html"}, requirements={"page": "[1-9]\d*"}, name="blog_index_paginated")
     * @Method("GET")
     * @Cache(smaxage="10")
     *
     * NOTE: For standard formats, Symfony will also automatically choose the best
     * Content-Type header for the response.
     * See http://symfony.com/doc/current/quick\_tour/the\_controller.html#using-formats
     */
    public function indexAction($page, $_format)
    {
        $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest($page);

        // Every template name also has two extensions that specify the format and
        // engine for that template.
        // See https://symfony.com/doc/current/templating.html#template-suffix
        return $this->render('blog/index.' . $_format . '.twig', ['posts' => $posts]);
    }
}
```

The screenshot shows the PhpStorm debugger interface with the title "Debug router_dev.php". The "Debugger" tab is selected in the top navigation bar. The left sidebar displays the call stack, with the current frame "BlogController.php:51, AppBundle\Controller\BlogController->indexAction()" highlighted in blue. Below it are frames from "classes.php", "Symfony\Component\HttpKernel\HttpKernel", and "Kernel.php", followed by "app_dev.php" and "router_dev.php". The right sidebar shows the "Variables" panel, which lists several PHP superglobals and environment variables. The variables listed are: \$_format = "html", \$page = "1", \$this = {AppBundle\Controller\BlogController} [1], \$_COOKIE = {array} [3], \$_SERVER = {array} [25], \$GLOBALS = {array} [13], and HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library".

```
4208 \Symfony\Component\HttpKernel\HttpKernel handleRaw
4209     $event = new FilterControllerEvent($this, $controller, $request, $type);
4210     $this->dispatcher->dispatch(KernelEvents::CONTROLLER, $event);
4211     $controller = $event->getController();
4212     $arguments = $this->argumentResolver-> getArguments($request, $controller); argumentResolver: Symfony\Component\HttpKernel\Controller\TraceableArgument
4213     $event = new FilterControllerArgumentsEvent($this, $controller, $arguments, $request, $type);
4214     $this->dispatcher->dispatch(KernelEvents::CONTROLLER_ARGUMENTS, $event); dispatcher: Symfony\Component\HttpKernel\Debug\TraceableEventDispatcher
4215     $controller = $event->getController();
4216     $arguments = $event-> getArguments();
4217     $response = call_user_func_array($controller, $arguments);
4218     if (!$response instanceof Response) {
4219         $event = new GetResponseForControllerResultEvent($this, $request, $type, $response);
4220         $this->dispatcher->dispatch(KernelEvents::VIEW, $event);
4221         if ($event->hasResponse()) {
4222             $response = $event->getResponse();
4223         }
4224         if (!$response instanceof Response) {
4225             $msg = sprintf('The controller must return a response (%s given).', $this->varToString($response));
4226             if (null === $response) {
4227                 $msg .= ' Did you forget to add a return statement somewhere in your controller?';
4228             }
4229             throw new \LogicException($msg);
4230         }
4231     }
4232 }
```

Debug *router_dev.php

Debugger Console ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌍ ⌎

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- classes.php:4216, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/var/cache/dev/appDevDebugContainer.php:10}
- classes.php:4216, Symfony\Component\HttpKernel\HttpKernel->handleRaw()
- classes.php:4171, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$this = {Symfony\Component\HttpKernel\HttpKernel} [4]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [13]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
\Symfony\Component\HttpKernel\HttpKernel handleRaw
  $arguments = $this->argumentResolver->getArguments($request, $controller); argumentResolver: Symfony\Component\HttpKernel\Controller\TraceableEventDispatcher
  $event = new FilterControllerArgumentsEvent($this, $controller, $arguments, $request, $type);
  $this->dispatcher->dispatch(KernelEvents::CONTROLLER_ARGUMENTS, $event); dispatcher: Symfony\Component\HttpKernel\Debug\TraceableEventDispatcher
  $controller = $event->getController();
  $arguments = $event-> getArguments();

  // call controller
  $response = call_user_func_array($controller, $arguments);

  // view
  if (!$response instanceof Response) {
    $event = new GetResponseForControllerResultEvent($this, $request, $type, $response);
    $this->dispatcher->dispatch(KernelEvents::VIEW, $event);

    if ($event->hasResponse()) {
      $response = $event->getResponse();
    }

    if (!$response instanceof Response) {
      $msg = sprintf('The controller must return a response (%s given).', $this->varToString($response));
    }
  }
}
```

bug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel.php:153, Symfony\Component\HttpKernel\HttpKernel->handleRaw()}
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$this = {Symfony\Component\HttpKernel\HttpKernel} [4]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [13]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"

```
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```

```
    ) {
        header('HTTP/1.0 403 Forbidden');
        exit('You are not allowed to access this file. Check '.basename(__FILE__).' for more information.');
    }

/** @var Composer\Autoload\ClassLoader $loader */
$loader = require __DIR__.'/../app/autoload.php';
Debug::enable();

$kernel = new AppKernel('dev', true);
if (PHP_VERSION_ID < 70000) {
    $kernel->loadClassCache();
}

$request = Request::createFromGlobals();
$response = $kernel->handle($request);
$response->send();
$kernel->terminate($request, $response);
```

Debug *router_dev.php

Debugger Console

Frames

- BlogController.php:51, AppBundle\Controller\BlogController->indexAction()
- HttpKernel.php:153, call_user_func_array:{/Users/bviguier/workspace/symfony-demo/vendor/symfony/symfony/src/Symfony/Component/HttpKernel/HttpKernel->handleRaw()}
- HttpKernel.php:68, Symfony\Component\HttpKernel\HttpKernel->handle()
- Kernel.php:168, Symfony\Component\HttpKernel\Kernel->handle()
- app_dev.php:36, require()
- router_dev.php:40, {main}()

Variables

- \$_composer_autoload_files = {array} [10]
- \$kernel = {AppKernel} [10]
- \$loader = {Composer\Autoload\ClassLoader} [10]
- \$request = {Symfony\Component\HttpFoundation\Request} [28]
- \$_COOKIE = {array} [3]
- \$_SERVER = {array} [25]
- \$GLOBALS = {array} [13]
- HTMLPURIFIER_PREFIX = "/Users/bviguier/workspace/symfony-demo/vendor/ezyang/htmlpurifier/library"



Atoum
Ça marche aussi !

Configuration

- ✖ **-mcn 1**



Profiling

C'est possible!

Configuration

- **xdebug.profiler_enable_trigger=1**
- **xdebug.profiler_output_name="cachegrind.out.%R"**



Meow!

(That's cat
for “Thanks!”)

San Diego
Humane
SOCIETY