

Techniques de débogage

-- Débogage avec PHP/Eclipse

Il est préférable d'utiliser un véritable débogueur, plutôt que de se satisfaire des **echo** et **var_dump** insérés dans le code pour évaluer une variable

-- XDebug

XDebug est l'un des outils de débogage existant pour PHP. Il est généralement livré par défaut avec les solutions Wampp, Xampp ou easyPhp.

-- Vérifications

Vérifier son activation en lançant **phpinfo** :

xdebug

xdebug support	enabled
Version	2.2.1
IDE Key	ECLIPSE_DBGP_192.168.1.13

Supported protocols	Revision
DBGp - Common DeBuGger Protocol	\$Revision: 1.145 \$

Directive	Local Value	Master Value
xdebug.auto_trace	Off	Off
xdebug.cli_color	0	0
xdebug.collect_assignments	Off	Off
xdebug.collect_includes	On	On
xdebug.collect_params	0	0
xdebug.collect_return	Off	Off
xdebug.collect_vars	Off	Off
xdebug.coverage_enable	On	On
xdebug.default_enable	On	On

Dans certains cas, il faut l'activer en éditant **php.ini** :

```
[XDebug]
zend_extension = "C:\xampp\php\ext\php_xdebug.dll"
xdebug.profiler_append = 0
xdebug.profiler_enable = 1
xdebug.profiler_enable_trigger = 0
xdebug.profiler_output_dir = "C:\xampp\tmp"
```

```

xdebug.profiler_output_name = "cachegrind.out.%t-%s"
xdebug.remote_enable = 1
xdebug.remote_handler = "dbgp"
xdebug.remote_host = "127.0.0.1"
xdebug.remote_port = "9000"
xdebug.trace_output_dir = "C:\xampp\tmp"
xdebug.idekey = "ECLIPSE_DBGP"

xdebug.auto_trace=0      ;to always profile, set to 1 & comment out
xdebug.trace_enable_trigger
xdebug.trace_enable_trigger=1
xdebug.collect_params=4   ;displays full parameter variable names and values
xdebug.collect_return=1   ;display function return values
xdebug.trace_format=2

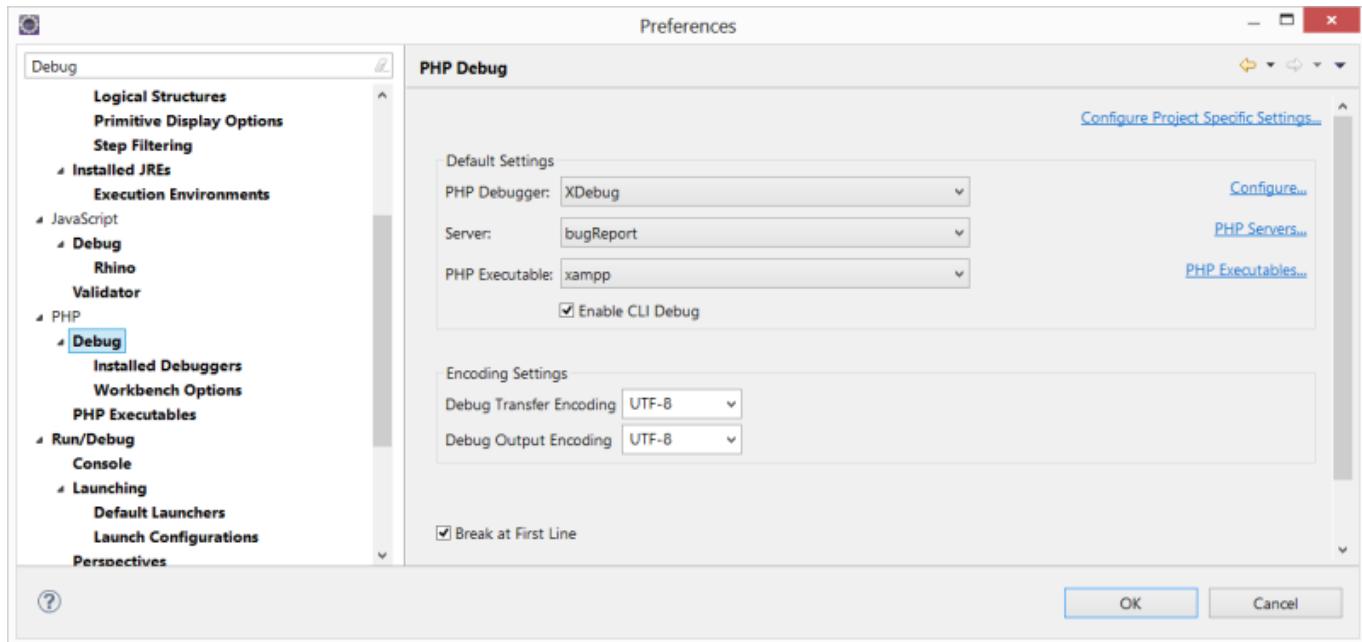
```

- Vérifier la présence de la dll sous Windows et son emplacement
- Noter la valeur de **xdebug.idekey** qui servira dans la communication avec Eclipse et Chrome
- Vérifier l'adresse (remote_host) et le port de communication (9000 par défaut)

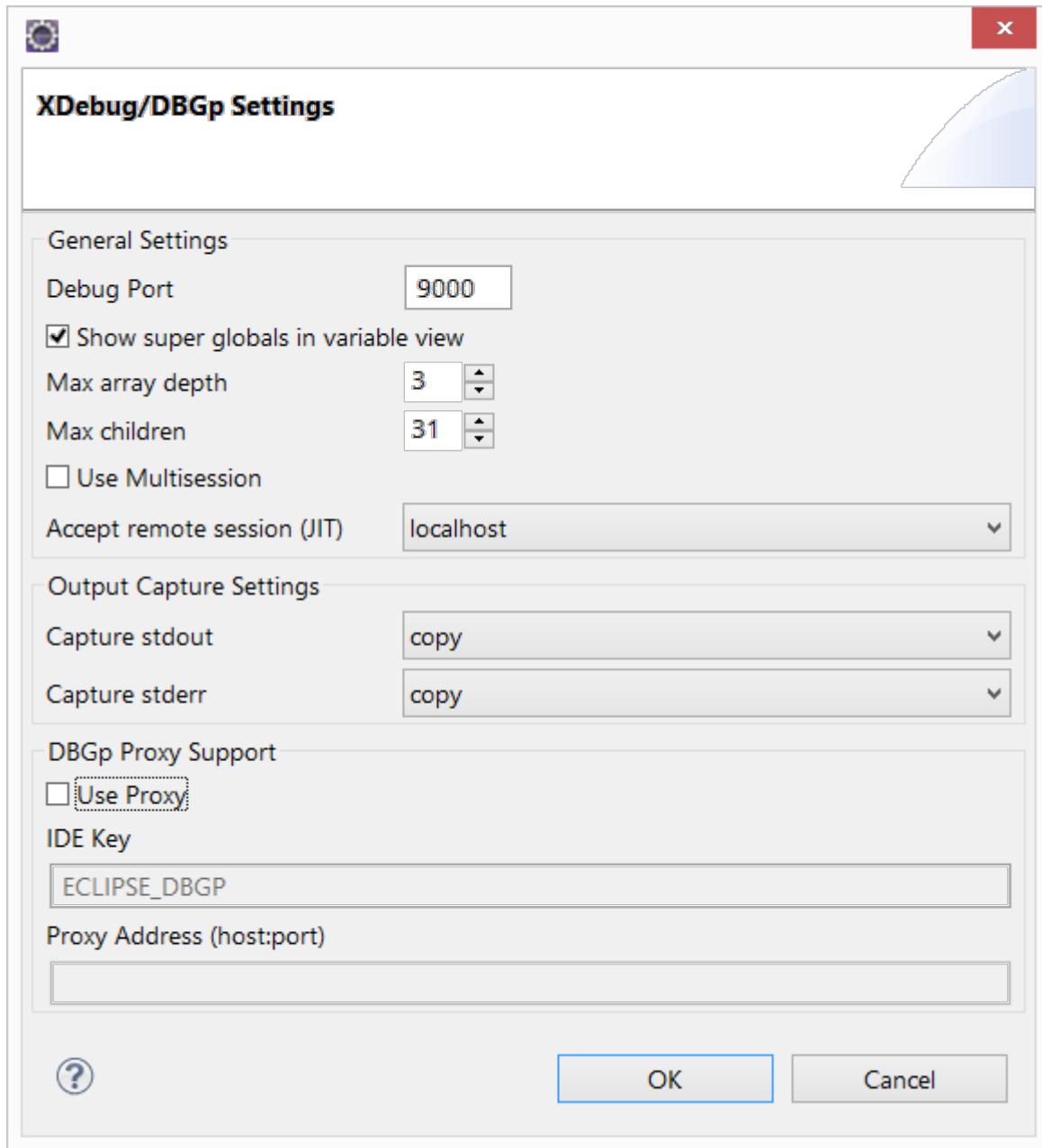
-- Configuration d'Eclipse

L'extension pdt doit être installée.

- Aller dans le menu **Window/preferences**, puis frapper **debug** dans la zone de recherche.
- Activer l'élément **PHP/debug**



Choisir ensuite Configurer XDebug :



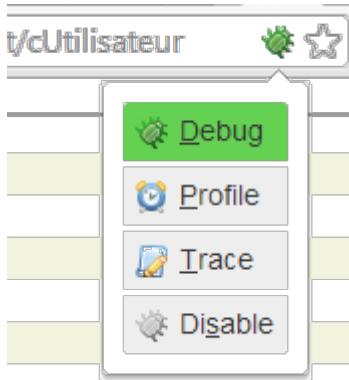
Vérifier au passage les paramètres **php servers** et **php executables**.

-- Installation du Helper de Chrome

Installer l'extension XDebug Helper de chrome : [XDebug helper](#)

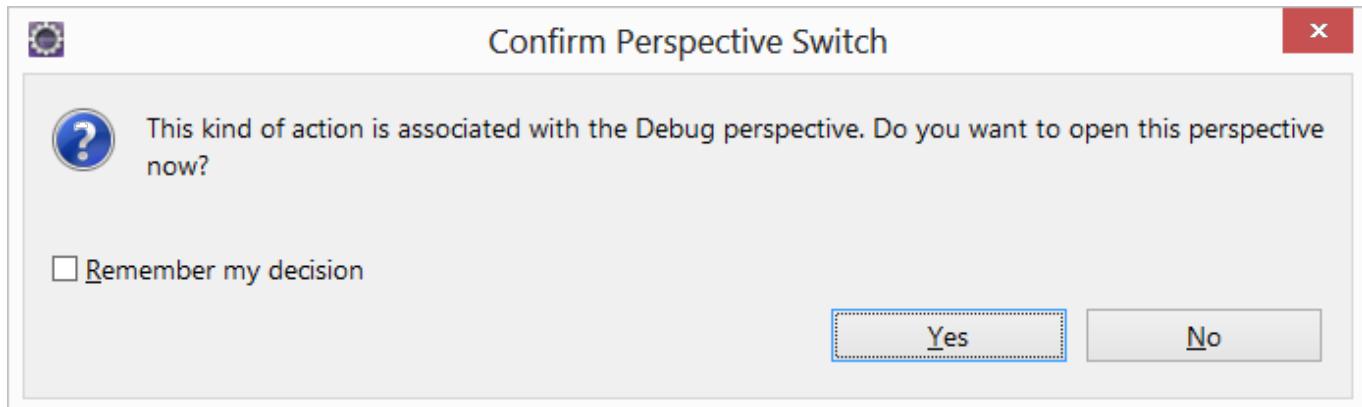
-- Débogage

- Démarrer Eclipse
- Dans chrome, aller à l'adresse de la page à déboguer et activer le débogage sur la droite de la barre d'adresse :



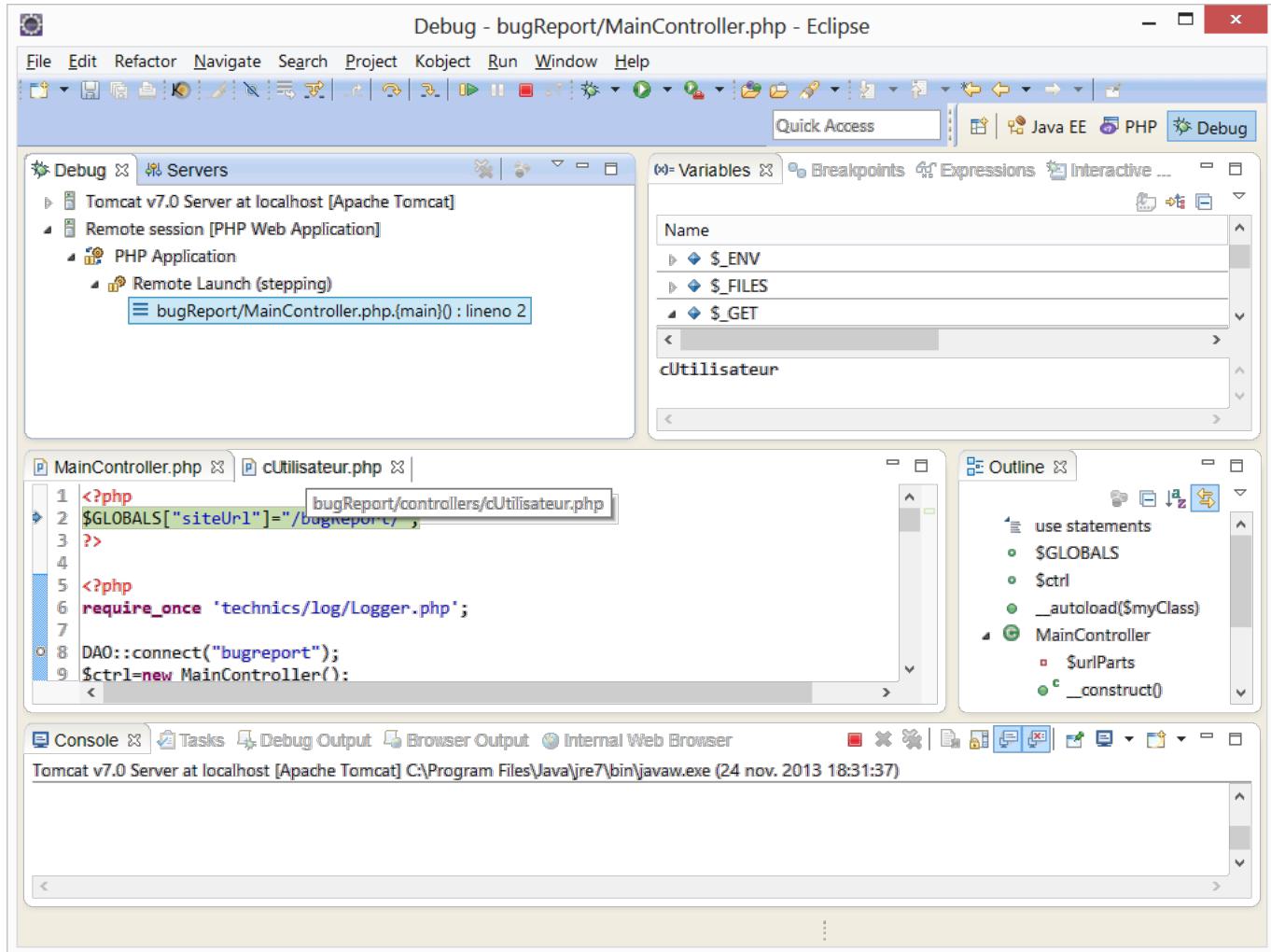
Rafraîchir la page pour provoquer une nouvelle exécution du script, puis retourner à Eclipse :

Eclipse propose de passer en perspective Debug :



En cas de non fonctionnement (pas d'arrêt dans Eclipse lors de l'exécution), changez le port xdebug dans php.ini (passez le à 9101 par exemple); et modifiez le également dans Eclipse.

L'exécution s'est arrêtée sur la première ligne.



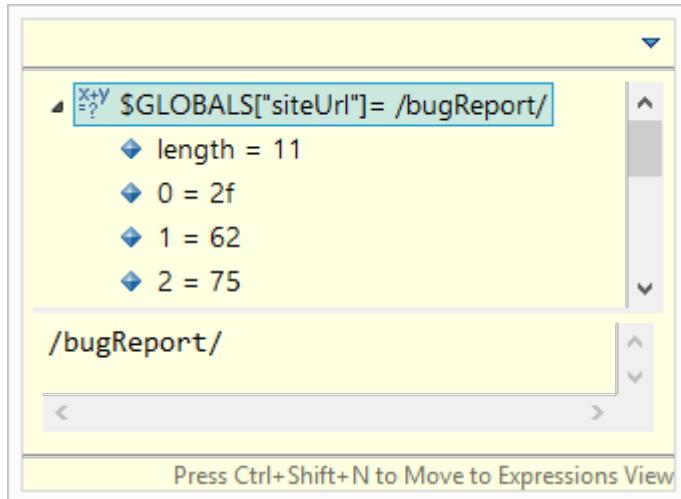
Il est alors possible de reprendre l'exécution :

	en pas à pas, d'instruction en instruction sans parcourir les fonctions appelées (step over)
	en pas à pas, d'instruction en instruction en parcourant les fonctions appelées (step into)
	de reprendre normalement l'exécution du script (jusqu'à sa fin ou jusqu'au prochain point d'arrêt (resume)
	d'arrêter la session de débogage (stop)

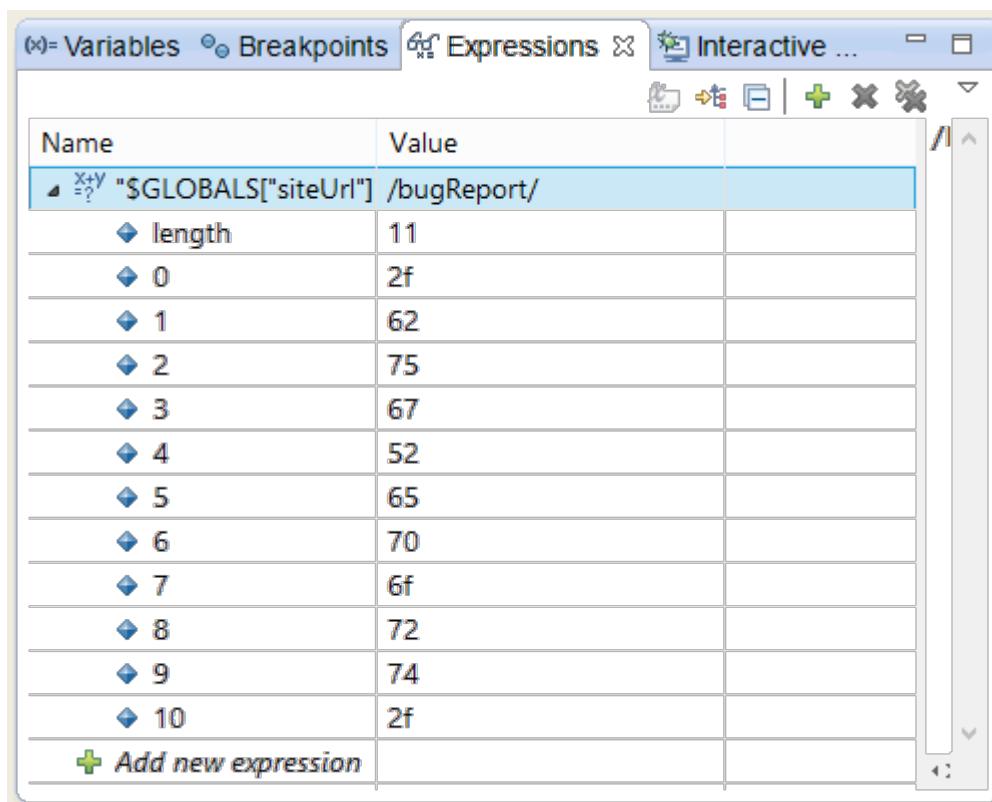
Inspection de variables

Il est alors possible :

- d'inspecter ponctuellement la valeur prise par une expression (Bouton droit de la souris sur l'expression à évaluer puis **Inspect** dans le menu) :

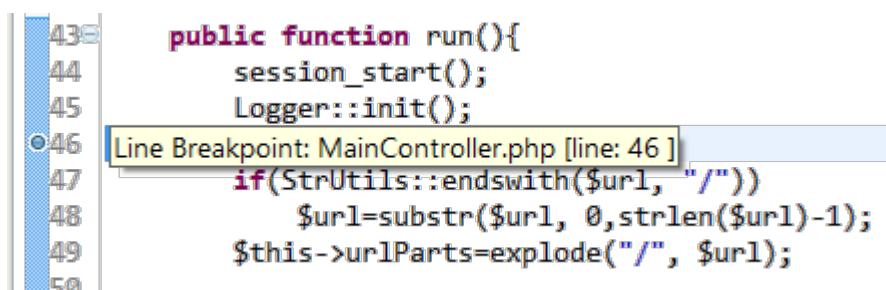


De maintenir la valeur de l'expression affichée (Bouton droit de la souris sur l'expression à évaluer puis **Watch** dans le menu :



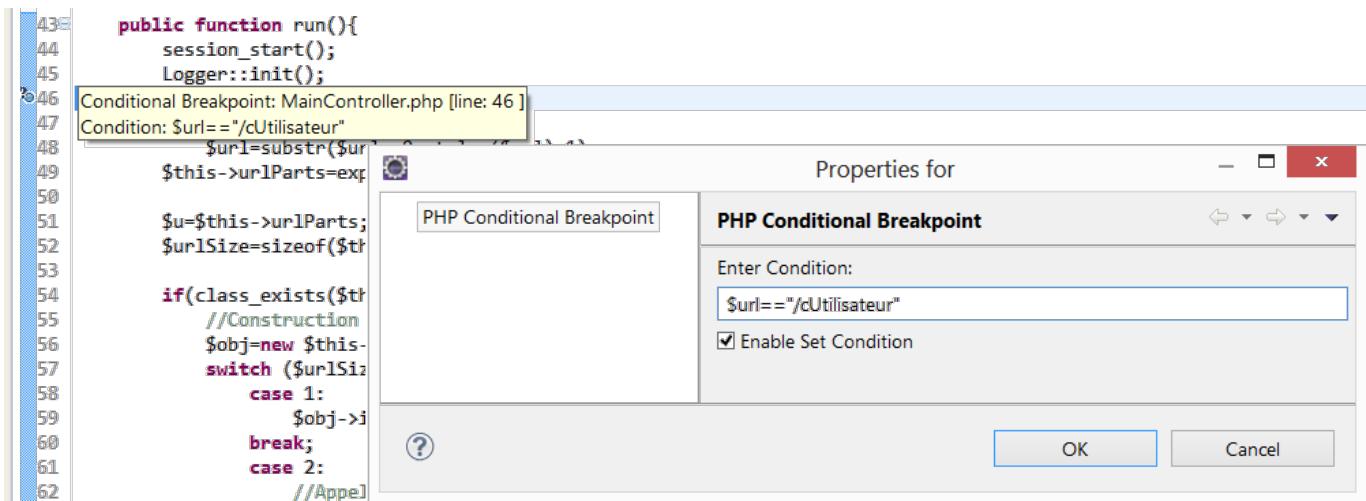
Poser un point d'arrêt

Double cliquer dans la barre verticale des numéros de ligne, au niveau de la ligne sur laquelle l'exécution doit s'interrompre :



Relancer ensuite l'exécution (F5 dans le navigateur par exemple).

Il est également possible de poser un point d'arrêt conditionnel. L'exécution ne sera interrompue dans ce cas que si la condition est vraie :



Poser le point d'arrêt normalement (double-clic), puis aller dans **breakpoint properties** en utilisant le menu contextuel.

-- Débogage côté client

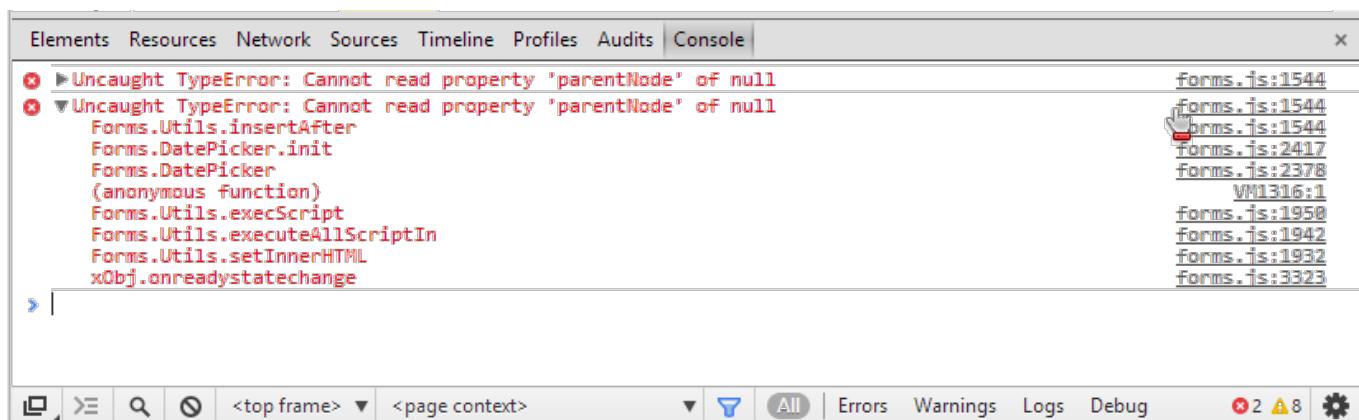
Difficile parfois de trouver les erreurs dans les scripts côté client et les requêtes ajax, il convient donc d'utiliser les outils de développement fournis avec les navigateurs, ou d'en installer en supplément.

-- Chrome

-- Débogage des scripts

Chrome (Version 31.0.1650.57 m) est fourni avec des outils de développement efficaces, accessibles par le menu : **Outils/Outils de développement**

La console d'erreur affichée par défaut montre les erreurs, warnings et/ou informations de log.



Il est ensuite possible d'ouvrir le fichier concerné (dans le cas présent **forms.js**), puis de poser un point d'arrêt sur une ligne en cliquant sur la barre de numérotation des lignes.

Elements Resources Network Sources Timeline Profiles Audits Console

log.js forms.js

```

1538     var sm=new Forms.MessageDialog("sm");
1539     sm.addButton("Okay",13);
1540     sm.show();
1541     return sm;
1542   },
1543   insertAfter:function(newElement,target
1544     var parent = targetElement.parentNode
1545     if(parent.lastchild == targetEleme
1546       parent.appendChild(newElement)
1547     } else {
1548       parent.insertBefore(newElement
1549     }
1550   },
1551   insertBefore:function(newElement,targe
1552   var parent = targetElement.parentNode
1553   if(parent.lastchild == targetEleme
1554     parent.appendChild(newElement)
1555   } else {
1556     parent.insertBefore(newElement
1557   }
1558

```

Line 1544, Column 31

Watch Expressions

- s: <not available>
- target: <not available>
- element: input#id-line-d4f6a235ff7fe8ff1142d56d3024949a
- xObj.responseText.isJSON(): <not available>

Call Stack

Scope Variables

Breakpoints

forms.js:1544

DOM Breakpoints

XHR Breakpoints

Event Listener Breakpoints

Workers

La nouvelle exécution du script permettra de s'arrêter au point d'arrêt, puis de reprendre l'exécution :

- | | |
|--|---|
| | en pas à pas d'instruction en instruction sans parcourir les fonctions appelées |
| | en pas à pas en parcourant les fonctions appelées |

Le point d'arrêt peut être conditionnel (à poser avec le bouton droit de la souris) :

2469 if(dtpType.indexOf("time")!=-1){

The breakpoint on line 2408 will stop only if this expression is true:

dtpType!=undefined

Il est également possible d'inspecter des variables, soit en les survolant à la souris, soit en les ajoutant aux addWatch :

Watch Expressions

- s: <not available>
- target: div#ajxid1bKCatég
- element: input#idCategorie.fieldValid
- xObj.responseText.isJSON(): <not available>
- \$(this.inputId): ""

Call Stack

Scope Variables

Breakpoints

No Breakpoints

DOM Breakpoints

XHR Breakpoints

Event Listener Breakpoints

Workers

-- Inspection des requêtes

Afficher le volet réseau (**network**), cliquer sur la requête ajax à inspecter :

Elements Resources Network Sources Timeline Profiles Audits Console

Name	Method	Status	Type	Initiator	Size	Time	Timeline
Path		Text			Content	Latency	
/qstart	POST	OK	text/html	Script	6.3 KB	3 ms	11574.1 days
changeForm.do?className=.../qstart	GET	200 OK	text/html	forms.js:3361	3.29 B	42 ms	
KSValidator.frm	POST	200 OK	text/html	forms.js:3361	3.29 B	42 ms	
KSValidator.frm	POST	200 OK	text/html	forms.js:3361	3.29 B	42 ms	
changeForm.do?className=.../qstart	GET	200 OK	text/html	forms.js:3361	3.29 B	42 ms	
KSValidator.frm	POST	200 OK	text/html	forms.js:3361	3.29 B	42 ms	
KSValidator.frm	POST	200 OK	text/html	forms.js:3361	6.9 KB	16030 ms	

Icons: Refresh, Stop, All, Documents, Stylesheets, Images, Scripts, XHR, Fonts, WebSockets, Other. Status: 8 errors.

Chrome affiche le contenu de la réponse :

Name Path

Name	Path
testChangeUser.do?_ajx	/qstart
refresh.do	/qstart
changeForm.do?className=KUtilisateur&keyValues=1&excludedFields=&...	/qstart
KSValidator.frm	/qstart
changeForm.do?className=KUtilisateur&keyValues=2&excludedFields=&...	/qstart
KSValidator.frm	/qstart

Headers Preview Response Cookies Timing

```

37 </div>
38 </fieldset>
39 <div class="boxButtons"><input class="btn" type="button" value="Annuler">
40 </form>
41 <input type='hidden' id='koKeyValue' name='koKeyValue' value='1'>
42 Forms.DOM.onReady(function(){$post('Field-idUtilisateur', {
43   'koKeyValue': koKeyValue
44 }, 'new Forms.Elements('btSubmitUser',document).addEvent('click', function(e) {
45   e.preventDefault();
46   new Forms.Elements('btAddUser',document).addEvent('click', function(e) {
47     e.preventDefault();
48     $infoBulle('body', 'body.onLoad', {"request": "/qstart/testChangeUser.do?_ajx", "method": "post", "data": "koKeyValue=1", "type": "text/javascript"});
49     Forms.DOM.onReady(function() {
      ...
    
```

Icons: Refresh, Stop, All, Documents, Stylesheets, Images, Scripts, XHR, Fonts, WebSockets, Other. Status: 8 errors.

Il est possible d'inspecter également les en-têtes et paramètres de la requête.

Headers Preview Response Cookies Timing

Request URL: http://127.0.0.1:8080/qstart/testChangeUser.do?_ajx

Request Method: GET

Status Code: 200 OK

Request Headers [view source](#)

```
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
Connection: keep-alive
Cookie: JSESSIONID=329DADB4E0E3561E40500E0435552CA1; PHPSESSID=tuj4k623d6i9i2v6sf9i6qtvu7; __utma=96992031.1760393890.1383006835.1384903547.1385169990.57; __utmc=96992031; __utmz=96992031.1383006835.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)
Host: 127.0.0.1:8080
Referer: http://127.0.0.1:8080/qstart/
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.57 Safari/537.36
```

Query String Parameters [view source](#) [view URL encoded](#)

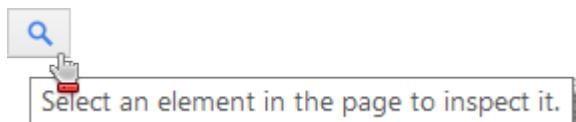
```
_ajx:
```

Response Headers [view source](#)

```
Content-Type: text/html; charset=UTF-8
Date: Sun, 24 Nov 2013 19:18:06 GMT
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked
```

-- Inspection DOM

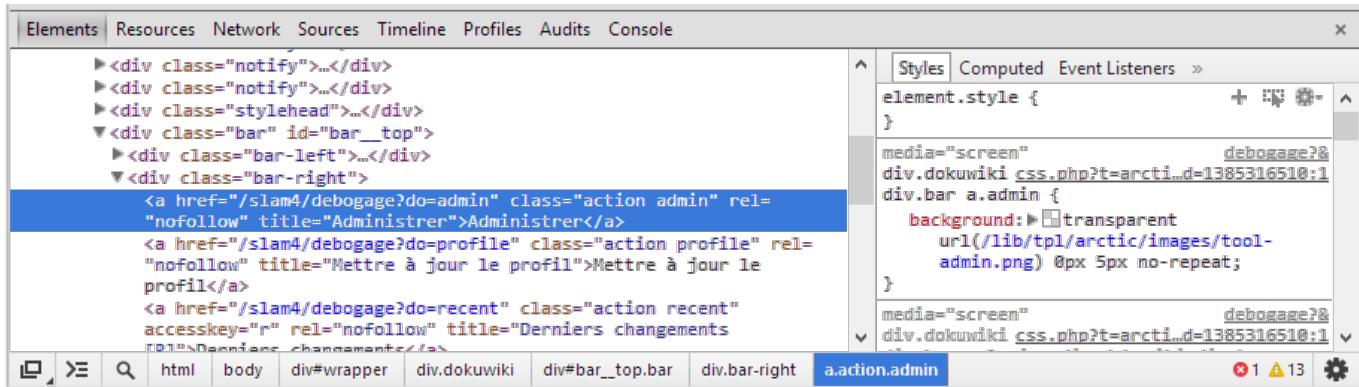
L'outil d'inspection DOM permet d'obtenir des informations sur l'arborescence DOM de la page :



Il suffit de cliquer ensuite sur l'élément DOM à inspecter



Pour obtenir la structure à laquelle il appartient, ou les informations de style :



The screenshot shows the Chrome DevTools interface. The Elements tab is active, displaying the HTML structure of a page. The Styles tab is also active, showing the CSS rules applied to the selected element. The selected element is an anchor tag with the class 'action admin'. The CSS rules for this element are as follows:

```
element.style {  
}  
media="screen" debogage?&  
div.dokuwiki_css.php?t=arctic_d=1385316510:1  
div.bar a.admin {  
    background: transparent  
    url(/lib/tpl/arctic/images/tool-  
    admin.png) 0px 5px no-repeat;  
}  
media="screen" debogage?&  
div.dokuwiki_css.php?t=arctic_d=1385316510:1
```

The bottom status bar shows the count of errors (1) and warnings (13).

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/slam4/debogage>

Last update: **2019/08/31 14:21**

