
Diabolo Documentation

1.0.0

Michael Luthor

12 05, 2017

Contents:

1		1
1.1	1
1.2	1
1.3	1
1.4	2
1.5	2
2		3
2.1	3
2.2	4
3		5
3.1	5
3.2	5
4		9
4.1	9
4.2	10
4.3	10
5		11
5.1	11
5.2	12
5.3	<code>start()</code> <code>stop()</code>	12
6		13
6.1	13
6.2	13
6.3	14
7		15
7.1	15
7.2	16
7.3	17
7.4	<code>Web</code>	17
8	Indices and tables	19

CHAPTER 1

1.1

Diabolo PHP

1.2

-
-
-

1.3

- Mysql/Sqlite/ActiveRecord
- Redis/
- /
- ///S3
- Web/ /
- OAuth2.0 //
- /
-
- /
-

1.4

- //
- /

1.5

-
-

2.1

-
- Demo:

```
$ mkdir Module          #
$ mkdir Module/Demo     # Demo
$ vi Module/Demo/Module.php #
```

:

```
<?php
namespace X\Module\Demo; #
use X\Core\Module\XModule; #
class Module extends XModule {
    #
    public function run($parameters = array()) {
        echo "This is a demo module.";
    }
}
```

- index.php :

```
<?php
define("DIABOLO_PATH", __DIR__); #
require DIABOLO_PATH.'/Core/X.php'; #
$config = array( #
    'document_root' => __DIR__, #
    'module_path' => array(), #
    'service_path' => array(), #
    'library_path' => array(), #
    'modules' => array( #
        'Demo' => array( #
            'enable' => true, #
            'default' => true, #
        )
    )
);
```

```
        'params' => array(),      #
    ),
    'services' => array(),        #
    'params' => array(),          #
);
X\Core\X::start($config)->run();
```

• :

```
project
| -- Core          # Diabolo
`-- Module         #
    `-- Demo       # Demo
        `-- Module.php #
```

2.2

http://127.0.0.1/:

This **is** a demo module.

3.1

```
array(  
    'document_root' => __DIR__,      #  
    'module_path' => array(),        #  
    'service_path' => array(),       #  
    'library_path' => array(),       #  
    'modules' => array(),            #  
    'services' => array(),           #  
    'params' => array(),             #  
);
```

3.2

- document_root

```
X::system()->getPath() document_root /myproject/demo/
```

```
echo X\Core\X::system()->getPath("View/Index.php");
```

```
/myproject/demo/View/Index.php
```

- module_path

```
:
```

```
'module_path' => array(  
    '/myproject/demo/Module',  
    '/usr/local/diabolo/Module',  
)
```

- service_path

```
:
```

```
'service_path' => array(
    '/myproject/demo/Service',
    '/usr/local/diabolo/Service',
)
```

- library_path

lib :

```
'library_path' => array(
    '/myproject/demo/Library',
    '/usr/local/diabolo/Library',
)
```

- modules

:

```
'modules' => array(
    'Demo' => array(
        'enable' => true,
        'default' => true,
        'params' => array(
            'param001' => 'value of param001',
        ),
    ),
)
```

- enable : truefalse

- default : truefalse

- params : , \X\Module\Demo::getModule()->getConfiguration()->get("param001")
 “value of param001”

- services

:

```
'services' => array(
    'Demo' => array(
        'class' => '\\X\\Service\\Demo\\Service',
        'enable' => true,
        'delay' => true,
        'params' => array(
            'param001' => 'value of param001',
        ),
    ),
)
```

- class

- enable truefalse

- delay truefalse false

- params : , \X\Service\Demo::getService()->getConfiguration()->get("param001")
 “value of param001”

- params

```
, \X\Core\X::system()->getConfiguration()->get("params")->get("param001")  
“value of param001”:
```

```
'params' => array(  
    'param001' => 'value of param001',  
) ,
```


4.1

- ```
project
 `-- Module
 |-- ModuleName
 | `-- Module.php
 `-- Example
 `-- Module.php
```

- `module_path`

- 

- `Module.php,:`

```
<?php
namespace X\Module\Example;
use X\Core\Module\XModule;

class Module extends XModule {
 /**
 * {@inheritDoc}
 * @see \X\Core\Module\XModule::run()
 */
 public function run($parameters = array()) {
 echo "This is an example module.";
 }
}
```

```
run()
```

- `modules`

```
'modules' => array(
 'Example' => array(
 'enable' => true,
```

```
'default' => true,
'params' => array(
 'pretty_name' => '',
),
),
),
```

params

## 4.2

module

```
web : http://example.com/index.php?module=example
cli : php index.php --module=example
```

This is an example module..

## 4.3

- ```
#
echo $this->getPath("Resource/Template.txt");

#
\X\Module\Example\Module::getModule()->getPath("Resource/Template.txt");
```

/home/project/example,

```
/home/project/example/Module/Resource/Template.txt
```

- ```
#
echo $this->getConfiguration()->get("pretty_name");

#
\X\Module\Example\Module::getModule()->getConfiguration()->get("pretty_name");
```

## 5.1

- ```
project
  `-- Service
      |-- ServiceName
      |   `-- Service.php
      `-- Example
          `-- Service.php
```

- service_path
-
- Service.php

```
<?php
namespace X\Service\Example;

class Service extends \X\Core\Service\XService {
    /** */
    protected static $serviceName = 'Example';

    /** */
    public function doSomething() {
        return "something";
    }
}
```

- services

```
'services' => array(
    'Example' => array(
        'class' => '\\X\\Service\\Example\\Service',
        'enable' => true,
        'delay' => true,
        'params' => array(
```

```
        'param001' => 'value of param001',  
    ),  
),  
)
```

params

5.2

getService()

```
echo X\Service\Example\Service::getService()->doSomething();
```

something

5.3 start() stop()

/ start() stop()

```
class Service extends \X\Core\Service\XService {  
    /** */  
    private $startTime = null;  
  
    /** */  
    public function start() {  
        parent::start();  
  
        $this->startTime = time();  
    }  
  
    /** */  
    public function stop() {  
        echo time() - $this->startTime();  
  
        parent::stop();  
    }  
}
```

delay false

6.1

, event001

```
$EventManager = \X\Core\X::system()->getEventManager();  
  
#  
$EventManager->registerHandler('event001', function( $param1 ) {  
    return "THIS IS CALLBACK RESPONSE";  
});  
  
#  
$EventManager->registerHandler('event001', "THIS IS A TEST RESULT");
```

6.2

event001

```
$EventManager = \X\Core\X::system()->getEventManager();  
$result = $EventManager->trigger('event001', 'par001');
```

\$result

```
array(  
    "THIS IS CALLBACK RESPONSE",  
    "THIS IS A TEST RESULT",  
);
```

"par001", \$param1 "par001"

6.3

`null`

7.1

:

```
'services' => array(
    'XAction' => array(
        'class' => '\\X\\Service\\XAction\\Service',
        'enable' => true,
        'delay' => true,
        'params' => array(
            'ActionParamName' => 'action',
            'Error404Handler' => array(Error404::class, 'handle404Error'),
            'Error500Handler' => array(Error500::class, 'handle500Error'),
            'CommonViewPath' => 'View',
        ),
    ),
),
```

- ActionParamName action, **web** action --action

```
GET http://example.com/index.php?module=example&action=show/hello
```

```
php index.php --module=example --action=show/hello
```

```
show/hello
```

- Error404Handler 404 404; 404 404

```
\X\Service\XAction\Service::getService()->triggerErrorHandler('404');
```

```
$this->throw404IfTrue(true);
```

- Error500Handler 500, null null
- CommonViewPath View,

```
project
|- Module
|   \- Example
|       \- View
|           |- Particle
|           |   \- Index.php
|           \- Layout
|
\-- View
    |- Particle
    |   \- Header.php
    \- Layout
        \- Default.php
```

7.2

```
<?php
namespace X\Module\Example;
use X\Core\Module\XModule;
use X\Service\Action\Service as XActionService;

/**
 * ,
 * <li> Action </li>
 * <li> View    View/Layout,  View/Particle </li>
 */
class Module extends XModule {
    /**
     * XAction
     */
    public function run($parameters = array()) {
        $actionService = XActionService::getService();

        #
        $group = 'example';
        #
        $actionService->addGroup($group, '\\X\\Module\\Example\\Action');
        #
        $actionService->setGroupOption($group, 'defaultAction', 'index');
        #
        $actionService->setGroupOption($group, 'viewPath', $this->getPath('View/'));
        #
        $actionService->getParameterManager()->merge($parameters);
        #
        return $actionService->runGroup($group);
    }
}
```

-
- user/login \X\Module\Example\Action\User\Login
-
-
-

7.3

```
<?php
namespace X\Module\Example\Action\User;
use X\Service\Action\Handler\WebPageAction;

class Index extends WebPageAction {
    /** */
    protected $layout = '/SingleColumn';
    /** */
    protected $title = '';

    #
    protected function runAction( $userName ) {
        $this->addParticle('/header');

        $this->addParticle('login', array(
            'default' => $userName, #
        ));

        $this->addParticle('/footer');
        $this->display();
    }
}
```

- WebPageAction WebPageAction,CommandAction AjaxAction
- /SingleColumn /header,login / CommonViewPath
- runAction() , \$userName

```
GET http://example.com/index.php?module=example&action=user/login&userName=example
```

7.4 Web

webphp web html body head

```
$this X\Service\Action\Component\WebView\Html $this
X\Service\Action\Component\WebView\ParticleView
```

```
<div>
    <?php echo date('Y-m-d H:i:s'); ?>
</div>
```

```
<?php
/* @var $this X\Service\Action\Component\WebView\Html */

/* @var $link X\Service\Action\Component\WebView\LinkManager */
$link = $this->getLinkManager();

# CSS
$link->addCSS('bootstrap', '/lib/bootstrap/dist/css/bootstrap.css');

/* @var $script X\Service\Action\Component\WebView\ScriptManager */
$script = $this->getScriptManager();
```

```
# Js
$script->add('jquery', '/lib/jquery/dist/jquery.min.js');
```

```
<?php
/* @var $this \X\Service\Action\Component\Webview\ParticleView */
$html = $this->getManager()->getParent();

/* @var $link \X\Service\Action\Component\Webview\LinkManager */
$link = $html->getLinkManager();

# CSS
$link->addCSS('bootstrap', '/lib/bootstrap/dist/css/bootstrap.css');

/* @var $script \X\Service\Action\Component\Webview\ScriptManager */
$script = $html->getScriptManager();
# Js
$script->add('jquery', '/lib/jquery/dist/jquery.min.js');
```

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`