LNXMCP CHEAT SHEET (v3.5.0)

Site Url https://github.com/lnxhcm/lnxmcp **Doc Url** http://lnxmcp.uk/ **Phar Url** https://github.com/lnxhcm/lnxmcp/blob/v3.5.0/dist/lnxmcp.phar

Cfg

file	cfg/mcp.settings.json
	/app.php
(app.php)	$lnxmcp_phar
resource	app.def
	$app_path
	app.path
	app.lang
	app.menu.InitCommon
	app.debug and app.level

Sys Cfg Settings

lnxmcp()→	getCfg($name)
	setCfg($name,$value)
	getResource($name)
	getCommon($name)
	setCommon($name,$value)

Sys Tools

lnxmcp()→	header($string,$end,$retcode,
(legacy move)	move($file,$filedef,$ext,$path,$end)
	Rem($var,[$var])
	supportmail($message)
	mail($page,$scope,module,$vendor)
	escapeClear($string)
	ConvertToAscii($string)
	Object2Array($object)
function	lnxMcpExit([$message])
	lnxMcpTag($tagname, $scopeIn)
	lnxMcpCmd($scopeCtl, $scopeIn)
	lnxMcpExtLoad($fle,$pth,$ext,$scp,$cvt)

Sys Admin

resource	mcp.web.api
	mcp.web.admin
function	lnxmcpadm($cmd)
shell	lnxmcp-adm $cmd
web	/lnxmcpadm
	/lnxmcpapi
web function	home , form ,mail
shell function	http, checksintax,

Sys Command

lnxmcp()→	controller($proc,$ispre,$scope,$mod,$subc,$ven)
	api($srv, $ispre, $scope, $mod, $subc, $ven)
	service($srv,$ispre,$scope,$mod,$subc,$ven)
	page($page,$scope,$mod,$ven,$pathtpl,$hasreturn)
	block($page,$scope,$mod,$ven,$pathtpl,$hasreturn)
	render($page,$scope,$mod,$ven,$pathtpl,$hasreturn)
	driver($name,$isp,$scpe,$mod,$subc,$ven,$path)
	shell($ctrlproc,$scope,$mod,$subc,$ven)
	remote($proc,$scopeIn,$mod,$subcall,$ven)
	legacyClass($name,$scpe,$mod,$subc,$ven,$path)
	runSequence($actionseq, $scopeIn)
	runMenu($action, $scopeIn)
	runTag($action, $scopeIn)
	runCommand($scopeCtl, $scopeIn)

Sys Debug

lnxmcp()→	debug($message)	
	info($message)	
	warning($message)	
	error($message	
	rem($msg [,$msg2])	(F:app.web.rem)
	dump($msg)	(F:app.web.dump)

Database

resource	(“Driver.[db label])	
lnxmcp()→	queryJsonR($name,$scope,$module)	
	query($db,$ispreload,$scope,$mod,$subc,$ven)	
	queryR($db,$ispreload,$scope,$mod,$subc,$ven)	
	queryCommonR($db,$ispreload,$scope,$mod,$subc)	
	queryArrayR($scopeIn)	

exTemplate tags

	[scope-<vars>]	[scope-dump]
	[common-<vars>	[server-<vars>]
	[lnxmcp-<tags>]	[<label>-<tags>]
<lnxmcp>	name	module
</lnxmcp>	vendor	type
	disable-rem	Block-type
(block-type)	config	json
	common	scope
	translate	javascript
	print_r	

mcpBaseModelClass

$this→	getRes($resname)	getArgCtl($name)
	getCfg($cfgname)	getArgOut($name)
	getDriver($drvlabel)	setArgOut($name, $value)
	getCommon($name)	setReturn($return)
	callCmd(array $scopeCtl, array $scopeIn)	
	callTag($action,$scopeIn, $buffer)	
	debug($messge)	info($messge)
	warning($messge)	error($messge)

pdoDriver

ScopeIn	“E” as Driver envs	“Q” as Query
	“V” value	“T” type
Type	“e” execute	“q” simpleQuery
	“f” firstRow	“c” simpleCount
	“er” executeWithRollback	's' return sql

lnxmcp()→runCommand() / lnxmcpCmd

cmd list	extTemplate	extFile
	page	showPage
	showFullCommonBlock	
	blockShell	blockRemote
	block	blockCommon
	showBlock	showCommonBlock
	render	renderCommon
	service	serviceCommon
	apiArray	apiArrayCommon
	api	apiReturn
	apiController	ApiService
	shell	remote
	driver	mail
	run	load
	controllerReturn	tag
	controller	controllerCommon
	queryArray	queryJson
	query	queryCommon
	headerHttp	headerClose
	header	legacy
	javascript	javascriptCommon
	shell	remote
	print	clear
	exit	dumpexit
	move	goto
	virtFs	

Example Controller:

```
<?php
/**
 ** <DESCRIPTION>
 **/
namespace App\<Module>\Controller;
use LinHUniX\Mcp\Model\mcpBaseModelClass;`
class <Name>Controller extends mcpBaseModelClass {
  /**
   *  Ideally this method shuld be used to first esecution
   */
  protected function moduleInit(){
        $this->spacename=__NAMESPACE__;
        $this->classname=__CLASS__;
    }
  /**
   *  Ideally this method shuld be used to insert
   *  the model code and the other are to be used only as normal
   */
  protected function moduleCore() {
     $mydataarray=$this->argIn;
     /// is empty waith to be implemented
     $this->argOut=$arrayout;
  }
  /**
   *  Ideally this method shuld be used to insert
   *  the model code and the other are to be used only as normal
   */
  protected function moduleSingleTon() {
     /// is empty waith to be implemented
   }
}`
```

Example Service:

```
<?php
/**
 ** <DESCRIPTION>
 **/
namespace App\<Module>\Service;
use LinHUniX\Mcp\Model\mcpServiceModelClass;
Class <Name> extends mcpServiceModelClass {
  /**
   *  Ideally this method shuld be used to first esecution
   */
  Protected function moduleInit(){
        $this->spacename=__NAMESPACE__;
        $this->classname=__CLASS__;
    }
   /**
    * standard 1 shot user
    */
   protected function moduleSingleTon() {}
   /***
    * function <t>_<e>(){
    * [T]=xxx
    * [E]=yyy
    */
   public function xxx_yyy(){}
}
```

Json Query:

```
{
    "E": "db label",
    "T": "query type",
    "Q": "query sql [field] [R:raw field] [S:stripslashed field] "
}
```

Channge Path module

```
lnxmcp()→setCfg('app.mod.path.<vendro>.<module>','fullpaht');
```

cfg/PathRewrite.json

```
{
"/<path>":"Location: /<newpath>",
"*/<path>":"Content-type: <mimevalue>"
}
```

cfg/PathConvert.json

```
{
  "<path>/*":{
        "vendor":"<vendor>","module":"<modle>","name":"<name>", "type":"<type>"
    },
  "<path>":"<tag>"
}
```

LNXMCP CHEAT SHEET (v3.5.0)					
Site Url https://github.com/lnhunix/lnxmcp	**Doc Url**	http://lnxmcp.uk/	**Phar Url**	https://github.com/lnhunix/lnxmcp/blob/3...0/dist/lnxmcp.phar	
DATABASE		**MAIL**			
lnxmcp()→	getResource($name) ex “Driver.xxx.db1”	lnxstmail($to,$sbj,$msg,$head,$parm,$from,$docs,tpl)			
		lnxsimplemail($to,$sbj,$msg,$from,$docs)			
$db1/db1()→	simpleQuery($sql,$arravars)				
	indexQuery($sql, $idfield = 'id', $sort = true)				
	queryLabel($sql,$arrayvar)				
	queryNoLabel($sql,$arrayvar)				
	firstRow($sql,$arrvars)				
	firstRowfield($sql,$row0field)				
	getLastRun()				
	getLastError()				
	intexec($sql)				
	rawexec($sql)				
	execute($sql)				
	executeWithRollback($sql,arrayvar)				
	dataWalk($sql, $callback, $vararray, $funarr,$err)				
	getRows($fields,$table,$orde,$from,$size,$where,$id)				