Cheat Sheet

Site Url	https://github.com/linhunix/lnxmcp	Doc Url	http://lnxmcp.uk/	Phar Url	https://github.com	/linhunix/lnxmcp/blob/3 5.0/d	ist/Inxmcp.phar
Cfg		Sys Comma			mcpBaseModelC		
file	cfg/mcp.settings.json	Inxmcp() →	controller(\$proc,\$ispre,\$scope	,\$mod,\$subc,\$ven)	\$this →	getRes(\$resname)	getArgCtl(\$name)
-		1	api(\$srv, \$ispre, \$scope, \$mod		1	getCfg(\$cfgname)	
	/app.php	1			-		getArgOut(\$name)
app.php)	\$Inxmcp_phar	4	service(\$srv,\$ispre,\$scope,\$m			getDriver(\$drvlabel)	setArgOut(\$name, \$value
resource	app.def		page(\$page,\$scope,\$mod,\$ve	n,\$pathtpl,\$hasreturn)		getCommon(\$name)	setReturn(\$return)
	\$app_path		block(\$page,\$scope,\$mod,\$ve	en,\$pathtpl,\$hasreturn)		callCmd(array \$scopeCtl, array	\$scopeIn)
	app.path	1	render(\$page,\$scope,\$mod,\$v	en.\$nathtnl.\$hasreturn)	1	callTag(\$action,\$scopeIn, \$buffer)	
		1	driver(\$name,\$isp,\$scpe,\$mod				:-4-(0)
	app.lang	-				debug(\$messge)	info(\$messge)
	app.menu.InitCommon		shell(\$ctrlproc,\$scope,\$mod,\$	subc,\$ven)		warning(\$messge)	error(\$messge)
	app.debug and app.level		remote(\$proc,\$scopeIn,\$mod,	\$subcall,\$ven)	pdoDriver		
Sys Cfg Settir	าตร		legacyClass(\$name,\$scpe,\$	mod,\$subc,\$ven,\$path)	Scopeln	"E" as Driver envs	"O" as Query
nxmcp() →	getCfg(\$name)	1	runSequence(\$actionseq, \$s	conein)	1	"V" value	"T" type
шхтер() ->		1			Tuno		
	setCfg(\$name,\$value)		runMenu(\$action, \$scopeIn)		Туре	"e" execute	"q" simpleQuery
	getResource(\$name)		runTag(\$action, \$scopeIn)			"f" firstRow	"c" simpleCount
	getCommon(\$name)		runCommand(\$scopeCtl, \$se	copeln)		"er" executeWithRollback	's' return sql
	setCommon(\$name,\$value)	Sys Debug	•		Inxmc	p() → runCommand() /	InxmcpCmd
	Seconimon(¢name,¢valae)	Inxmcp() →	d-b(b)		cmd list		· · · · · · · · · · · · · · · · · · ·
TI-		linxinch() →	debug(\$message)		Citiu iist	extTemplate	extFile
ys Tools			info(\$message)			page	showPage
nxmcp() →	header(\$string,\$end,\$retcode,		warning(\$message)			showFullCommonBlock	
egacy move)	move(\$file,\$filedef,\$ext,\$path,\$end)	1	error(\$message			blockShell	blockRemote
egacy move,		1		(Franciscale ram)	1		
	Rem(\$var,[\$var])	1	rem(\$msg [,\$msg2])	(F:app.web.rem)	-	block	blockCommon
	supportmail(\$message)	4	dump(\$msg)	(F:app.web.dump)		showBlock	showCommonBlock
	mail(\$page,\$scope,module,\$vendor)					render	renderCommon
	escapeClear(\$string)	Database				service	serviceCommon
	ConvertToAscii(\$string)	resource	("Driver (db Jaham)		1		apiArrayCommon
		-	("Driver.[db label])	L	-	apiArray	
	Object2Array(\$object)	Inxmcp() →	queryJsonR(\$name,\$scope,\$r			арі	apiReturn
unction	InxMcpExit([\$message])	_	query(\$db,\$ispreload,\$scope,\$	smod,\$subc,\$ven)]	apiController	ApiService
	InxMcpTag(\$tagname, \$scopeIn)		queryR(\$db,\$ispreload,\$scope	,\$mod,\$subc,\$ven)		shell	remote
	InxMcpCmd(\$scopeCtl, \$scopeIn)	1	queryCommonR(\$db,\$ispreloa		1	driver	mail
		1			1		
	InxMcpExtLoad(\$fle,\$pth,\$ext,\$scp,\$cvt)	4	queryArrayR(\$scopeIn)			run	load
ys Admin]	controllerReturn	tag
esource	mcp.web.api	exTemplate	tags			controller	controllerCommon
	mcp.web.admin		[scope- <vars>]</vars>	[scope-dump]	1	queryArray	queryJson
		1			1		
		-	[common- <vars></vars>	[server- <vars>]</vars>		query	queryCommon
unction	Inxmcpadm(\$cmd)		[Inxmcp- <tags>]</tags>	[<label>-<tags>]</tags></label>		headerHttp	headerClose
hell	Inxmcp-adm \$cmd	<inxmcp></inxmcp>	name	module		header	legacy
veb	/Inxmcpadm		vendor	type	İ	javascript	javascriptCommon
WCD	·	7.1.baop	disable-rem				
	/Inxmcpapi			Block-type	1	shell	remote
		Ha			1		
		(block-type)	config	json		print	clear
web function	home , form ,mail	(block-type)					
web function		(block-type)	config common	json scope		print exit	clear dumpexit
	home , form ,mail	(block-type)	config common translate	json		print exit move	clear
		(block-type)	config common	json scope		print exit	clear dumpexit
	home , form ,mail http, checksintax,		config common translate	json scope		print exit move virtFs	clear dumpexit
	home , form ,mail		config common translate	json scope	Ex	print exit move	clear dumpexit
shell function	home , form ,mail http, checksintax,		config common translate	json scope	Ex	print exit move virtFs	clear dumpexit
shell function	home , form ,mail http, checksintax, Example Contro		config common translate	json scope	Ex	print exit move virtFs	clear dumpexit
<pre>shell function </pre> <pre></pre> <pre>** ** < CDESCRIPTIO</pre>	home , form ,mail http, checksintax, Example Contro		config common translate	json scope	Ex	print exit move virtFs	clear dumpexit
shell function <pre> </pre> <pre> ** ** ** ** *DESCRIPTIO **/ ** ** ** ** ** ** ** ** ** ** ** **</pre>	home , form ,mail http, checksintax, Example Contro		config common translate	json scope javascript php</td <td>Ex</td> <td>print exit move virtFs</td> <td>clear dumpexit</td>	Ex	print exit move virtFs	clear dumpexit
shell function php *** < DESCRIPTIO ***/ namespace Appl <</td <td>home , form ,mail http, checksintax, Example Contro N> Module>\Controller;</td> <td></td> <td>config common translate</td> <td>json scope javascript</td> <td>Ex</td> <td>print exit move virtFs</td> <td>clear dumpexit</td>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller;		config common translate	json scope javascript	Ex	print exit move virtFs	clear dumpexit
shell function php *** ** < DESCRIPTIO **/ namespace Appl< sse LinHUniXIMcp</td <td>home , form ,mail http, checksintax, Example Contro</td> <td></td> <td>config common translate</td> <td> son scope </td> <td></td> <td>print exit move virtFs</td> <td>clear dumpexit</td>	home , form ,mail http, checksintax, Example Contro		config common translate	son scope		print exit move virtFs	clear dumpexit
shell function **?php ** ** < DESCRIPTIO **/ namespace App\< sse LinHUniX\Mcp lass < Name>Con /** /**	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; N\ode\ncpBaseModelClass; troller extends mcpBaseModelClass {		config common translate	son scope	dule>\Service:	print exit move virtFs ample Service:	clear dumpexit
<pre><?php ** ** <DESCRIPTIO **) namespace Appl< lass <\number <\n</td><td>home , form ,mail http, checksintax, Example Contro N> Module>\Controller; ModelmcpBaseModelClass;</td><td></td><td>config common translate</td><td> son scope </td><td>dule>\Service:</td><td>print exit move virtFs ample Service:</td><td>clear dumpexit</td></pre>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; ModelmcpBaseModelClass;		config common translate	son scope	dule>\Service:	print exit move virtFs ample Service:	clear dumpexit
shell function	home , form ,mail http, checksintax, Example Control N> Module>\Controller; ModelmcpBaseModelClass; thod shuld be used to first esecution		config common translate	son scope	dule>\Service; odel\mcpServiceM ds mcpServiceMod	print exit move virtFs ample Service:	clear dumpexit
c?php *** **CPESCRIPTIO *** **Inamespace Appl lass <name>Con *** *I leally this me */ protected function</name>	home , form ,mail http, checksintax, Example Control N> Module>\Controller: b\model\mcpBaseModelClass; titroller extends mcpBaseModelClass { thod shuld be used to first esecution		config common translate	son scope javascript	dule>\Service; odel\mcpServiceM ds mcpServiceMod	print exit move virtFs ample Service:	clear dumpexit
shell function **?php *** ** <pescriptio \$this:="" *="" **="" appl<="" deally="" function="" i="" linhunixmcp="" me="" namespace="" protected="" sse="" this="">space*</pescriptio>	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \Model\mcpBaseModelClass; \text{intoller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=_NAMESPACE_;}		config common translate	son scope javascript	dule>\Service; odel\mcpServiceM ds mcpServiceMor d shuld be used to	print exit move virtFs ample Service:	clear dumpexit
*** *** *** *** *** ** ** ** ** ** ** *	home , form ,mail http, checksintax, Example Control N> Module>\Controller: b\model\mcpBaseModelClass; titroller extends mcpBaseModelClass { thod shuld be used to first esecution		config common translate	son scope	dule>\Service; ode\\mcpServiceM ds mcpServiceMod d shuld be used to oduleInit(){	print exit move virtFs ample Service: oodelClass; delClass {	clear dumpexit
chell function	home , form ,mail http, checksintax, Example Contro N> Module>\Controller: b\Model\mcpBaseModelClass; \troller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_;		config common translate	son scope javascript	dule>\Service; ode\mcpServiceMo ds mcpServiceMo d shuld be used to noduleInit(){ ne=NAMESPAC	print exit move virtFs ample Service: oodelClass; delClass {	clear dumpexit
*** *** *** *** *** ** ** ** *	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \Model\mcpBaseModelClass; intoller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert	ler:	config common translate	son scope javascript	dule>\Service; ode\mcpServiceMo ds mcpServiceMo d shuld be used to noduleInit(){ ne=NAMESPAC	print exit move virtFs ample Service: oodelClass; delClass {	clear dumpexit
shell function *** ** CPESCRIPTIO *** ** CPESCRIPTIO ** isse LinHUniXMcp lass <name>Con f** * Ideally this me * this>-spacer * lideally this me * lideally this me * ledeally this me * the model code</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller: b\Model\mcpBaseModelClass; \troller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_;	ler:	config common translate	son scope javascript	dule>\Service; ode\mcpServiceMo ds mcpServiceMo d shuld be used to noduleInit(){ ne=NAMESPAC	print exit move virtFs ample Service: oodelClass; delClass {	clear dumpexit
Shell function "" "" "" "" "" "" "" "" ""	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; N\odelincpBaseModelClass; troller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=NAMESPACE_; ame=CLASS_; thod shuld be used to insert e and the other are to be used only as normal	ler:	config common translate	son scope javascript	dule>\Service; odel\mcpServiceMor ds mcpServiceMor d shuld be used to oduleInit(){ ne= NAMESPAC e=_CLASS_;	print exit move virtFs ample Service: oodelClass; delClass {	clear dumpexit
** c>php ** ** ** ** CPESCRIPTIO ** lamespace App\ lass <name>Con ** lass <name>Con ** ** ** ** ** ** ** ** **</name></name>	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \Model\mcpBaseModelClass; \introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; ame=CLASS_; thod shuld be used to insert e and the other are to be used only as norman n moduleCore() {	ler:	config common translate	son scope javascript	dule>\Service; odel\mcpServiceMor ds mcpServiceMor d shuld be used to odule\mit(){ ne=_NAMESPAC e=_CLASS_; user	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
*** *** *** *** ** ** *DESCRIPTIO *** ** ** ** ** ** ** ** **	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=NAMESPACE; ame=CLASS; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented	ler:	config common translate	son scope javascript	dule>\Service; odel\mcpServiceMor ds mcpServiceMor d shuld be used to odule\mit(){ ne=_NAMESPAC e=_CLASS_; user	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function *** ** CPSCRIPTIO ** ** CPSCRIPTIO ** lamespace App\< lass < Name>Con /* * Ideally this me */ * Ideally this me * Idea	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=NAMESPACE; ame=CLASS; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented	ler:	config common translate	son scope javascript	dule>\Service; ode\mcpServiceMor ds mcpServiceMor d shuld be used to nodule\nit(){ ne=_NAMESPAC e=_CLASS; user moduleSingleTon(print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
c?php *** ** CPBCRIPTIO ** ** ** ** ** ** ** ** ** *	home , form ,mail http, checksintax, Example Control N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=NAMESPACE; ame=CLASS; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented	ler:	config common translate	son scope javascript	dule>\Service; ode\mcpServiceMor ds mcpServiceMor d shuld be used to nodule\nit(){ ne=_NAMESPAC e=_CLASS; user moduleSingleTon(print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function	home , form ,mail http, checksintax, Example Control N> Module>\Controller; Model\mcpBaseModelClass; Itroller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis->argln; n to be implemented arrayout;	ler:	config common translate	son scope javascript	dule>\Service; ode\mcpServiceMor ds mcpServiceMor d shuld be used to nodule\nit(){ ne=_NAMESPAC e=_CLASS; user moduleSingleTon(print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function """ "" < DESCRIPTIO "" < DESCRIPTIO ""	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \Model\mcpBaseModelClass; \text{ithod shuld be used to first esecution} \text{in moduleInit()}{\text{name} =NAMESPACE;} \text{ame} =CLASS; \text{thod shuld be used to insert} \text{e and the other are to be used only as normal n moduleCore() {\text{5this}>argin;} \text{it to be implemented be used to insert} \text{thod shuld be used to insert} \text{thod shuld be used to insert} \text{thod shuld be used to insert}	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
c?php *** ** CPBCRIPTIO ** ** ** ** ** ** ** ** ** *	home , form ,mail http, checksintax, Example Control N> Module>\Controller; Model\mcpBaseModelClass; Itroller extends mcpBaseModelClass { thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis->argln; n to be implemented arrayout;	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function *** ** ** ** ** ** ** ** **	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \Model\mcpBaseModelClass; \text{ithod shuld be used to first esecution} \text{in moduleInit()}{\text{name} =NAMESPACE;} \text{ame} =CLASS; \text{thod shuld be used to insert} \text{e and the other are to be used only as normal n moduleCore() {\text{5this}>argin;} \text{it to be implemented be used to insert} \text{thod shuld be used to insert} \text{thod shuld be used to insert} \text{thod shuld be used to insert}	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function **?php *** ** <pescriptio **="" <name="" lass="">Con **/ shell function **/ size LinHUniXIMcp lass <name>Con **/ ** Ideally this me ** Ideally this me ** Ideally this me ** Ideally this me ** Ideally distance ** Ideally distance ** Ideally this me ** Ideally this me</name></pescriptio>	home , form ,mail http, checksintax, Example Control N	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; name=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { this>argln; h to be implemented sarrayout;	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
?php ** *CESCRIPTIO ** *CESCRIPTIO ** *SE LinHUniXIMcp lass <name>Con ** * Ideally this me * Ideally t</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; name=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { this>argln; h to be implemented sarrayout;	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
?php ** *CESCRIPTIO ** *CESCRIPTIO ** *SE LinHUniXIMcp lass <name>Con ** * Ideally this me * Ideally t</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; name=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { this>argln; h to be implemented sarrayout;	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; name=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { this>argln; h to be implemented sarrayout;	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; ame=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>arglin; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>arglin; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { h to be implemented	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function "php" ** CPSCRIPTIO ** CPSCRIPTIO ** CPSCRIPTIO ** * Ideally this me * Ideally this	home , form ,mail http, checksintax, Example Control No	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function *** ** ** ** ** ** ** ** **	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; ame=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>arglin; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>arglin; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { h to be implemented	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
shell function *** *** ** CPESCRIPTIO **/ lamespace Appl< lass <name>Con **/ ** * Ideally this me */ * Ideally this me * Indeally this</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; ModelmcpBaseModelClass; Introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=_NAMESPACE_; name=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { n to be implemented in the other are to be used field in the limit of limit of the limit of the limit of	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; ModelmcpBaseModelClass; Introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=_NAMESPACE_; name=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { n to be implemented in the other are to be used field in the limit of limit of the limit of the limit of	ler:	config common translate	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
php ** ** *DESCRIPTIO ** ** *DESCRIPTIO ** ** * DESCRIPTIO ** ** * Ideally this me * the model cod */ /// is empty wait * Ideally this me * the model cod */ * Ideally this me * Ideall	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; ModelmcpBaseModelClass; Introller extends mcpBaseModelClass { thod shuld be used to first esecution n modulenit() { name=_NAMESPACE_; name=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthis>argin; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { n to be implemented in the other are to be used field in the limit of limit of the limit of the limit of	Jer:	config common translate print_r	son scope javascript	dule>\Service; odelmcpServiceMods mcpServiceMod ds mcpServiceMod d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
hell function ?php ** <>DESCRIPTIO */ se LinHUniXIMcp lass <name>Con ** to leadly this me * the model cod */ */ ** Ideally this me * the model cod */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** "E": "db label", ** ** ** ** ** ** ** ** ** ** ** ** **</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \Model\mcpBaseModelClass; \text{\text{\text{intoller extended}}} thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthits>argin; h to be implemented be used to insert e and the other are to be used only as normal n moduleCore() { the object of	Jer:	config common translate print_r	son scope javascript	dule>\Service; odefimcpServiceMor ds mcpServiceMor d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{yyy(){}	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
hell function ?php ** <>DESCRIPTIO */ se LinHUniXIMcp lass <name>Con ** to leadly this me * the model cod */ */ ** Ideally this me * the model cod */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** Ideally this me * the model cod */ */ */ ** "E": "db label", ** ** ** ** ** ** ** ** ** ** ** ** **</name>	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \Model\mcpBaseModelClass; \text{\text{\text{intoller extended}}} thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthits>argin; h to be implemented be used to insert e and the other are to be used only as normal n moduleCore() { the object of	Jer:	config common translate print_r	son scope javascript	dule>\Service; odefimcpServiceMor ds mcpServiceMor d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{yyy(){}	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function Phip	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \Model\mcpBaseModelClass; \text{\text{\text{intoller extended}}} thod shuld be used to first esecution n moduleInit(){ name=_NAMESPACE_; ame=_CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { sthits>argin; h to be implemented be used to insert e and the other are to be used only as normal n moduleCore() { the object of	Jer:	config common translate print_r	son scope javascript	dule>\Service; odefimcpServiceMor ds mcpServiceMor d shuld be used to noduleInit(){ ne=_NAMESPAC e=_CLASS_; user moduleSingleTon() 0{yyy(){}	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit
chell function **Phyp ** ** ** ** ** ** ** ** ** *	home , form ,mail http, checksintax, Example Contro N> Module>\Controller; \ModelmcpBaseModelClass; \introller extends mcpBaseModelClass; \thod shuld be used to first esecution n modulenit(){ name=NAMESPACE_; name=CLASS_; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { bthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleCore() { bthis>argln; h to be implemented sarrayout; thod shuld be used to insert e and the other are to be used only as normal n moduleSingleTon() { h to be implemented iffield] [R:raw field] [S:stripslashed field] " n module etCfg('app.mod.path. <vendro>.<mc <newpath="" ite.json="" n:="">",</mc></vendro>	Jer:	config common translate print_r	son scope javascript	dule>\Service; odel\mcpServiceMord ds mcpServiceMord d shuld be used to oduleInit(){ ne=NAMESPAC e=CLASS; user moduleSingleTon() 0{yyy(){}}	print exit move virtFs ample Service: odelClass; telClass { of irst esecution EE_;	clear dumpexit goto
chell function **Phyp ** ** ** ** ** ** ** ** ** *	home , form ,mail http, checksintax, Example Control N	Jer:	config common translate print_r	son scope javascript	dule>\Service; odel\mcpServiceMord ds mcpServiceMord d shuld be used to oduleInit(){ ne=NAMESPAC e=CLASS; user moduleSingleTon() 0{yyy(){}}	print exit move virtFs ample Service: lodelClass; letClass {	clear dumpexit goto

Cheat Sheet

	LNXMCP CHEAT SHEET (v3.5.0)								
Site Url	https://github.com/linhunix/Inxmep Doc Url	http://lnxmcp.uk/	Phar Url https://github.com	n/linhunix/lnxmcp/bl	.0/dist/lnxmcp.phar				
	DATABASE		MAIL						
Inxmcp() →	getResource(\$name) ex "Driver.xxx.db1"	Inxstmail(\$to,\$sbj,	\$msg,\$head,\$parm,\$from,\$docs,tpl)						
		Inxsimplema	ail(\$to,\$sbj,\$msg,\$from,\$docs)] [
				l					
\$db1/db1() →	simpleQuery(\$sql,\$arravars)								
	indexQuery(\$sql, \$idfield = 'id', \$sort = true)								
	queryLabel(\$sql,\$arrayvar)] [
	queryNoLabel(\$sql,\$arrayvar)] [
	firstRow(\$sql,\$arrvars)			J l					
	firstRowfield(\$sql,\$row0field)			J					
	getLastRun()								
	getLastError()								
	intexec(\$sql)] [
	rawexec(\$sql)] [
	execute(\$sql)			J l					
	executeWithRollback(\$sql,arrayvar)] [
	dataWalk(\$sql, \$callback, \$vararray, \$funarr,\$err)] [
	getRows(\$fields,\$table,\$orde,\$from,\$size,\$where,\$id)] [
]					
]					
] [
] [
] [
] [
] [