Cheat Sheet

Site Url Cfg	https://github.com/linhunix/lnxmcp	Doc Url	http://lnxmcp.uk/	Phar Url	https://github.com	/linhunix/lnxmcp/blob/3.5.0/d	list/Inxmcp.phar
_ •	The post grand block in the fact that the part of the	Sys Comma			mcpBaseModelC		остинор:рна
ile	cfg/mcp.settings.json	Inxmcp() →	controller(\$proc,\$ispre,\$scope	\$mod \$suhc \$ven\	\$this →	getRes(\$resname)	antAraCtl(\$come)
iiC		- Invitrich() →			Ψα 113 →		getArgCtl(\$name)
	/app.php	4	api(\$srv, \$ispre, \$scope, \$mo	1, \$subc, \$ven)	_	getCfg(\$cfgname)	getArgOut(\$name)
app.php)	\$Inxmcp_phar		service(\$srv,\$ispre,\$scope,\$n	nod,\$subc,\$ven)		getDriver(\$drvlabel)	setArgOut(\$name, \$valu
resource	app.def		page(\$page,\$scope,\$mod,\$ve	en,\$pathtpl,\$hasreturn)		getCommon(\$name)	setReturn(\$return)
	\$app_path		block(\$page,\$scope,\$mod,\$ve	en.\$nathtol.\$hasreturn)	1	callCmd(array \$scopeCtl, array	
		+	render(\$page,\$scope,\$mod,\$		1		*****
	app.path	-			_	callTag(\$action,\$scopeIn, \$buffer)	
	app.lang		driver(\$name,\$isp,\$scpe,\$mo	d,\$subc,\$ven,\$path)		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 Settin					-		
, , , , , , , , , , , , , , , , , , , 	Ť	-	runSequence(\$actionseq, \$		Scopeln	"E" as Driver envs	"Q" as Query
nxmcp() →	getCfg(\$name)		runMenu(\$action, \$scopeIn)		"V" value	"T" type
	setCfg(\$name,\$value)	Sys Debug			Туре	"e" execute	"q" simpleQuery
	getResource(\$name)	Inxmcp() →	debug(\$message)		1	"f" firstRow	"c" simpleCount
					1		·
	getCommon(\$name)		info(\$message)			"er" executeWithRollback	's' return sql
	setCommon(\$name,\$value)		warning(\$message)		Inxmc	$p() \rightarrow runCommand() /$	InxmcpCmd
			error(\$message		cmd list	extTemplate	extFile
ve Toole				(F	1		
ys Tools		4	rem(\$msg [,\$msg2])	(F:app.web.rem)	4	page	showPage
nxmcp() →	header(\$string,\$end,\$retcode,	1	dump(\$msg)	(F:app.web.dump)	1	showFullCommonBlock	
egacy move)	move(\$file,\$filedef,\$ext,\$path,\$end)					blockShell	blockRemote
(regacy move)	Rem(\$var,[\$var])	Database	·	·	1	block	blockCommon
			T	T T	4		
	supportmail(\$message)	resource	("Driver.[db label])		1	showBlock	showCommonBlock
	mail(\$page,\$scope,module,\$vendor)	Inxmcp() →	queryJsonR(\$name,\$scope,\$i	module)		render	renderCommon
	escapeClear(\$string)	7	query(\$db,\$ispreload,\$scope,		1	service	serviceCommon
		+			1		
	ConvertToAscii(\$string)	4	queryR(\$db,\$ispreload,\$scop		4	apiArray	apiArrayCommon
	Object2Array(\$object)	_	queryCommonR(\$db,\$ispreloa	ad,\$scope,\$mod,\$subc)]	арі	apiReturn
function	InxMcpExit([\$message])		queryArrayR(\$scopeIn)			apiController	ApiService
	InxMcpTag(\$tagname, \$scopeIn)	1			1	shell	remote
		ovTone lei	togo		1		
	InxMcpCmd(\$scopeCtl, \$scopeIn)	exTemplate	tags		4	driver	mail
	InxMcpExtLoad(\$fle,\$pth,\$ext,\$scp,\$cvt)		[scope- <vars>]</vars>	[scope-dump]		run	load
ys Admin			[common- <vars></vars>	[server- <vars>]</vars>		controllerReturn	tag
•		1			1		-
resource	mcp.web.api	4 .	[Inxmcp- <tags>]</tags>	[<label>-<tags>]</tags></label>	4	controller	controllerCommon
	mcp.web.admin	<inxmcp></inxmcp>	name	module		queryArray	
			vendor	type		query	queryCommon
unction	Inxmcpadm(\$cmd)	•	disable-rem	Block-type		headerHttp	headerClose
		(1-11-4)			-		neauer Close
shell	Inxmcp-adm \$cmd	(block-type)	config	json		header	
web	/Inxmcpadm		common	scope		javascript	javascriptCommon
	/Inxmcpapi	1	translate	javascript	1	shell	remote
	пипорарі	+		javaconpt	1		
		-	print_r		-	print	clear
web function	home , form ,mail					exit	dumpexit
		-			1		
shell function	http. checksintay				1		
shell function	http, checksintax,				4		
shell function	http, checksintax,						
hell function	http, checksintax, Example Control	ler:			Ex	ample Service:	
		ler:			Ex	ample Service:	
:?php	Example Control	ler:			Ex	ample Service:	
php ** ** <DESCRIPTIO</td <td>Example Control</td> <td>ler:</td> <td></td> <td><2nhn</td> <td>Ex</td> <td>ample Service:</td> <td></td>	Example Control	ler:		<2nhn	Ex	ample Service:	
:?php ** ** <descriptio **/</descriptio 	Example Control	ler:		php</td <td>- Ex</td> <td>ample Service:</td> <td></td>	- Ex	ample Service:	
<pre><?php ** ** <DESCRIPTIO **/ namespace App\</pre></pre>	Example Control ON> :Module>\Controller;	ler:		php /** ** <DESCRIPTION		ample Service:	
php ** ** <DESCRIPTIO **/ namespace App\< use LinHUniX\Mcp</td <td>Example Control DN> *Module>\Controller; D\Model\mcpBaseModelClass;`</td> <td>ler:</td> <td></td> <td>/** ** <description> **/</description></td> <td></td> <td>ample Service:</td> <td></td>	Example Control DN> *Module>\Controller; D\Model\mcpBaseModelClass;`	ler:		/** ** <description> **/</description>		ample Service:	
*** ** < DESCRIPTIO ***/ hamespace Appl< slass < Name>Cor	Example Control ON> :Module>\Controller;	ler:		/** ** <description> **/ namespace App\<mc< td=""><td>odule>\Service;</td><td></td><td></td></mc<></description>	odule>\Service;		
??php ** ** <descriptio **="" <name="" appl<="" lamespace="" lass="" linhuniximcp.="" sse="">Cor /**</descriptio>	Example Control DN> SModule>Controller: p\Model\mcpBaseModelClass; htroller extends mcpBaseModelClass {	ler:		/** ** <description> **/ namespace App\<mc linhunix\mcp\m<="" td="" use=""><td>odule>\Service; lodeNmcpServiceN</td><td>odelClass;</td><td></td></mc></description>	odule>\Service; lodeNmcpServiceN	odelClass;	
<pre><?php ** *> <descriptio *="" **="" <name="" appl<="" lass="" linhuniximc="" namespace="" use="">Cor /**</descriptio></pre>	Example Control DN> *Module>\Controller; D\Model\mcpBaseModelClass;`	ler:		/** ** <description> **/ namespace App\<mc< td=""><td>odule>\Service; lodeNmcpServiceN</td><td>odelClass;</td><td></td></mc<></description>	odule>\Service; lodeNmcpServiceN	odelClass;	
c?php ** ** *CDESCRIPTIO **/ namespace App\< lass <\name>Cor /** * Ideally this me */ protected function	Example Control DN> *Module>Controller; p)ModelmcpBaseModelClass; rtroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){	ler:		/** ** < DESCRIPTION> **/ namespace App\ <mc <="" class="" linhunix\mcp\m="" name="" use=""> exten /**</mc>	odule>\Service; ode\mcpServiceMo ds mcpServiceMo	iodelClass; delClass {	
use LinHUniXiMcc class <name>Cor /** * Ideally this me */ protected function \$this->space</name>	Example Control DN> Module>\Controller; piMode\mcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit(){ name=_NAMESPACE_;	ler:		/** ** <description> **/ namespace App\<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this metho</mc></description>	odule>\Service; ode\mcpServiceMo ds mcpServiceMo	iodelClass; delClass {	
<pre><?php ** ** <DESCRIPTIO ** ** <DESCRIPTIO ** ** ** ** ** ** ** ** ** ** ** ** **</td><td>Example Control DN> *Module>Controller; p)ModelmcpBaseModelClass; rtroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){</td><td>ler:</td><td></td><td>/** *** <description> **/ namespace App\<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this metho */</mc></description></td><td>ndule>\Service; lode\mcpServiceMo ds mcpServiceMo od shuld be used to</td><td>iodelClass; delClass {</td><td></td></pre>	Example Control DN> *Module>Controller; p)ModelmcpBaseModelClass; rtroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){	ler:		/** *** <description> **/ namespace App\<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this metho */</mc></description>	ndule>\Service; lode\mcpServiceMo ds mcpServiceMo od shuld be used to	iodelClass; delClass {	
*** *** ** ** ** ** ** ** **	Example Control DN> Module>\Controller; piMode\mcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit(){ name=_NAMESPACE_;	ler:		/** <description> *** <description> **/ namespace App\<mc <name="" class="" linhunix\mcpim="" use=""> exten /** * Ideally this metho */ Protected function in</mc></description></description>	odule>\Service; lodel\mcpServiceMods mcpServiceMods shuld be used to	odelClass; delClass { definition of the control of	
:?php ** ** ** ** ** ** ** ** **	Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_;	ler:		/** <description> *** <description> **/ namespace App\<mc <name="" class="" linhunix\mcpim="" use=""> exten /** * Ideally this metho */ Protected function in</mc></description></description>	odule>\Service; ode\mcpServiceMods mcpServiceMod ds mcpServiceMod ds shuld be used to noduleInit(){ me=NAMESPAC	odelClass; delClass { definition of the control of	
?php ** ** ** *>DESCRIPTIO ** ** ** * * * * * * * * * * * * * *	Example Control DN> Module>Controller; plModel\mcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit() { ename=NAMESPACE; name=CLASS; ethod shuld be used to insert			/ ***OESCRIPTION> ***/ namespace App\ <m <name="" class="" linhuniximcpim="" use=""> exten /** * Ideally this meth */ Protected function n \$this->spacenar \$this->classnam }</m>	odule>\Service; ode\mcpServiceMods mcpServiceMod ds mcpServiceMod ds shuld be used to noduleInit(){ me=NAMESPAC	odelClass; delClass { definition of the control of	
:?php *** ** <descriptio **="" **<="" td=""><td>Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_;</td><td></td><td></td><td>/** **<oescription> ** namespace Appl<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this meth */ Protected function n \$this>spacenar \$this>spacenar } /**</mc></oescription></td><td>odule>\Service; lode\mcpServiceX ds mcpServiceMo od shuld be used to nodule\nit(){ ne=NAMESPAC le=CLASS;</td><td>odelClass; delClass { definition of the control of</td><td></td></descriptio>	Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_;			/** ** <oescription> ** namespace Appl<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this meth */ Protected function n \$this>spacenar \$this>spacenar } /**</mc></oescription>	odule>\Service; lode\mcpServiceX ds mcpServiceMo od shuld be used to nodule\nit(){ ne=NAMESPAC le=CLASS;	odelClass; delClass { definition of the control of	
??php ** ** <descriptio **="" <\name="" amespace="" appl-="" lass="">Cor ** Ideally this me */ protected function \$this->space \$this->class } ** Ideally this me ** ** ** ** ** ** ** ** ** *</descriptio>	Example Control DN> Module>Controller: plModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit(){ name=NAMESPACE; name=CLASS; ethod shuld be used to insert le and the other are to be used only as normal			/** ***OESCRIPTION> ***/ namespace App\ <m <name="" class="" linhuniximcpim="" use=""> exten /** * Ideally this meth */ Protected function n \$this->spacenar \$this->classnam }</m>	odule>\Service; lode\mcpServiceX ds mcpServiceMo od shuld be used to nodule\nit(){ ne=NAMESPAC le=CLASS;	odelClass; delClass { definition of the control of	
:?php *** ** ** ** ** ** ** *DESCRIPTIO ** ** ** ** *Ideally this me */ ** *Ideally this me */ ** *Ideally this me */ ** *Ideally this me * *Ideally this me * *This->class * ** *Ideally this me * * the model cod * * *This-* * ** ** ** ** ** ** ** ** ** ** ** **	Example Control DN> Module>Controller: p)ModelmcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name= _NAMESPACE_; name= _CLASS_; ethod shuld be used to insert le and the other are to be used only as normal on moduleCore() { sthis->argin;			/** ** <obscription> ***/ namespace Appl<mc <name="" class="" linhuniximcpm="" use=""> exter /** * Ideally this meth */ Protected function n Sthis->spacenas Sthis->classnam } /** * standard 1 shot */</mc></obscription>	odule>\Service; lode\mcpServiceX ds mcpServiceMo dd shuld be used to noduleInit(){ ne=CLASS; user	iodelClass; delClass { first esecution :E;	
:?php ** ** ** *CPSCRIPTIO ** ** ** *CPSCRIPTIO ** *Ideally this me */ *Ideally this me * the model cod */ */ *Ideally this me * the model cod */ */ */ */ */ */ */ ** ** ** ** ** **	Example Control DN> Module>\Controller; p\Model\mcpBase\ModelClass; p\model\mcpBase\ModelClass; ethod shuld be used to first esecution on moduleInit(){ name=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { sthis>-argin; th to be implemented			/** ** <oescription> ** namespace Appl<mc <name="" class="" linhunix\mcp\m="" use=""> exten /** * Ideally this meth */ Protected function n \$this>spacenar \$this>spacenar } /**</mc></oescription>	odule>\Service; lode\mcpServiceX ds mcpServiceMo dd shuld be used to noduleInit(){ ne=CLASS; user	iodelClass; delClass { first esecution :E;	
:?php *** ** <descriptio *="" **="" cod="" descriptio="" ideally="" me="" model="" shis-="" the="" this="">argout=3 */ */ */ */ */ */ */ */ */ */ */ */ */</descriptio>	Example Control DN> Module>\Controller; p\Model\mcpBase\ModelClass; p\model\mcpBase\ModelClass; ethod shuld be used to first esecution on moduleInit(){ name=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { sthis>-argin; th to be implemented			/** **CDESCRIPTION> **/ namespace Appl <mu <name="" class="" linhuniximcpm="" use=""> exter /** * Ideally this meth */ Protected function n \$this->spacenan \$this->classnam } * standard 1 shot */ protected function /* /**</mu>	odule>\Service; lode\mcpServicew ds mcpServiceMo ad shuld be used to moduleInit(){ me=_NAMESPAC le=_CLASS_; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php * *** *** *** ** ** ** **	Example Control DN> Module>\Controller; p\Model\mcpBase\ModelClass; p\model\mcpBase\ModelClass; ethod shuld be used to first esecution on moduleInit(){ name=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { sthis>-argin; th to be implemented			/** ** <description> ** namespace Appl<mu <name="" class="" linhuniximcpim="" use=""> exten /* * Ideally this metho */ Protected function n \$his->spacenan \$his->classnam } /* * standard 1 shot */ protected function /** * tunction <>>= exten /* * function <>= exten /* * function <= exte</mu></description>	odule>\Service; lode\mcpServicew ds mcpServiceMo ad shuld be used to moduleInit(){ me=_NAMESPAC le=_CLASS_; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
:?php *** ** <descriptio *="" **="" <="" cod="" ideally="" me="" model="" odescriptio="" td="" the="" this=""><td>Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { \$\text{</td><td></td><td></td><td>/** **<\DESCRIPTION> ** namespace Appl<mc <name="" class="" linhunix\mcplm="" use=""> exten /** * Ideally this meth */ Protected function n \$this>>spacena \$this>>classnam } /** * standard 1 shot */ protected function /** * function <1>_ee * [T]=xxx * [E]=yyy</mc></td><td>odule>\Service; lode\mcpServicew ds mcpServiceMo ad shuld be used to moduleInit(){ me=_NAMESPAC le=_CLASS_; user moduleSingleTon(</td><td>iodelClass; delClass { first esecution :E;</td><td></td></descriptio>	Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { \$\text{			/** **<\DESCRIPTION> ** namespace Appl <mc <name="" class="" linhunix\mcplm="" use=""> exten /** * Ideally this meth */ Protected function n \$this>>spacena \$this>>classnam } /** * standard 1 shot */ protected function /** * function <1>_ee * [T]=xxx * [E]=yyy</mc>	odule>\Service; lode\mcpServicew ds mcpServiceMo ad shuld be used to moduleInit(){ me=_NAMESPAC le=_CLASS_; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php * *** **> **DESCRIPTIO *** ** ** ** ** ** ** ** **	Example Control DN> Amodule>Controller: plModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution an moduleInit() { name=NAMESPACE_; name=CLASS_; ethod shuld be used to insert de and the other are to be used only as normal an moduleCoreo { sthis->argin; th to be implemented sarrayout; ethod shuld be used to insert	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
*** *** *** ** ** ** ** ** **	Example Control DN> *Module>\Controller; p\Model\mcpBaseModelClass; ntroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as normal on moduleCore() { \$\text{	ı		/** **<\DESCRIPTION> ** namespace Appl <mc <name="" class="" linhunix\mcplm="" use=""> exten /** * Ideally this meth */ Protected function n \$this>>spacena \$this>>classnam } /** * standard 1 shot */ protected function /** * function <1>_ee * [T]=xxx * [E]=yyy</mc>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
*** *** *** *** ** ** ** ** **	Example Control DN> Amodule>Controller: plModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution an moduleInit() { name=NAMESPACE_; name=CLASS_; ethod shuld be used to insert de and the other are to be used only as normal an moduleCoreo { sthis->argin; th to be implemented sarrayout; ethod shuld be used to insert	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
*** *** *** ** ** ** ** ** **	Example Control DN> Module>\Controller; pilvlodeI/mcpBaseModelClass; pilvlodeI/mcpBaseModelClass; ethod shuld be used to first esecution In moduleInit(){	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php * ** <descriptio **="" **<="" td=""><td>Example Control DN> Amodule>Controller: piModelmcpBaseModelClass; piModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution an moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert le and the other are to be used only as normal an moduleCore() { \$\$this>argln; th to be implemented \$arrayout; ethod shuld be used to insert le and the other are to be used only as normal</td><td>ı</td><td></td><td>/** ** **/ namespace App\<mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu></td><td>odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(</td><td>iodelClass; delClass { first esecution :E;</td><td></td></descriptio>	Example Control DN> Amodule>Controller: piModelmcpBaseModelClass; piModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution an moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert le and the other are to be used only as normal an moduleCore() { \$\$this>argln; th to be implemented \$arrayout; ethod shuld be used to insert le and the other are to be used only as normal	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
??php ** ** *CDESCRIPTIO ** ** ** *CPESCRIPTIO ** *Ideally this me */ *Ideally this me */ ** *Ideally this me */ ** *Ideally this me */ ** *Ideally this me * the model cod */ */ ** ** ** ** ** ** ** ** ** ** **	Example Control DN> Module>\Controller; pilvlodeI/mcpBaseModelClass; pilvlodeI/mcpBaseModelClass; ethod shuld be used to first esecution In moduleInit(){	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php ** *CESCRIPTIO ** ** *CESCRIPTIO ** *Image and the search an	Example Control DN> Module>\Controller; pilvlodeI/mcpBaseModelClass; pilvlodeI/mcpBaseModelClass; ethod shuld be used to first esecution In moduleInit(){	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
??php ** ** *CDESCRIPTIO ** ** ** *CPESCRIPTIO ** *Ideally this me */ *Ideally this me */ ** *Ideally this me ** *Ideally this me ** *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 */ */ *Ideally this me * the model cod */ */ */ *Ideally this me * the model cod */ */ */ *Ideally this me * the model cod */ */ */ */ */ */ */ */ */ */ */ */ */	Example Control DN> Module>\Controller; pilvlodeI/mcpBaseModelClass; pilvlodeI/mcpBaseModelClass; ethod shuld be used to first esecution In moduleInit(){	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php ** ** *DESCRIPTIO ** ** *Jeasupe Apple * *Ideally this me */ *Ideally this me *Ideally t	Example Control DN> Module>\Controller; pilvlodeI/mcpBaseModelClass; pilvlodeI/mcpBaseModelClass; ethod shuld be used to first esecution In moduleInit(){	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
?php * * <descriptio *="" *<="" *cpescriptio="" td=""><td>Example Control DN> Module>Controller; pNModelnropBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit() { iname=_NAMESPACE_; iame=_CLASS_; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { Sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleSingleTon() { th to be implemented</td><td>ı</td><td></td><td>/** ** **/ namespace App\<mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu></td><td>odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(</td><td>iodelClass; delClass { first esecution :E;</td><td></td></descriptio>	Example Control DN> Module>Controller; pNModelnropBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit() { iname=_NAMESPACE_; iame=_CLASS_; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { Sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleSingleTon() { th to be implemented	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
Pphp * DESCRIPTIO // mespace App\ see LinHUniXMcgreass <\Name>Cor ** * Ideally this me // * Ideally this me * the model cod * the model cod * mydataarray= /// * Ideally this me * the model cod // * the model cod /// * Time model cod	Example Control DN> Module>Controller: piModelmcpBaseModelClass; piModelmcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert le and the other are to be used only as normal on moduleCore() { \$this>argln; th to be implemented \$arrayout; ethod shuld be used to insert le and the other are to be used only as normal on moduleSingleTon() { th to be implemented	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
Phphp Company Compa	Example Control DN> Module>Controller; pNModelnropBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution in moduleInit() { iname=_NAMESPACE_; iame=_CLASS_; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { Sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleCore() { sthis>argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma in moduleSingleTon() { th to be implemented	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	
Phph * DESCRIPTIO * DESCRIPTIO * amespace App\< see LinHUniXIMc; ass <\Name>Cor * Ideally this me * Ideally this me * the model cod * ideally this me * the model cod * mydataarray= *// is empty wait * the model cod *// is empty wait * the model cod * mydataarray= * I// is empty wait * the model cod * mydataarray= * ideally this me * the model cod * mydataarray= * ideally this me * the model cod * mydataarray= * ideally this me * the model cod * if is empty wait * if is empty	Example Control ON> **Module>Controller; p)ModelmcpBaseModelClass; introller extends mcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ name=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as norma on moduleCore() { \$this>-argin; th to be implemented \$arrayout; ethod shuld be used to insert de and the other are to be used only as norma on moduleCore() { this on insert de and the other are to be used only as norma on moduleSingleTon() { the tobe implemented **In the other are to be used only as norma on moduleSingleTon() { the tobe implemented **In the implemented **In th	ı		/** ** **/ namespace App\ <mu <name="" class="" linhunixmcplm="" use=""> exten /* * Ideally this methe */ Protected function n</mu>	odule>\Service; lode\mcpServiceN ds mcpServiceMor od shuld be used to noduleInit(){ ne=NAMESPAC ne=CLASS; user moduleSingleTon(iodelClass; delClass { first esecution :E;	