## Cheat Sheet

Site Url	https://github.com/linkusiu/fauses	Doc Url	LNXMCP CHEAT S http://lnxmcp.uk/	Phar Url	https://gith.ub.com	/linhuniy/lnymen/blob/2 = 0/d	ict/Inymon phos
Cfa	https://github.com/linhunix/lnxmcp	Sys Comma		Pilai Uli		/linhunix/lnxmcp/blob/3.5.0/d	ist/inxmcp.pnar
Cfg	T	-			mcpBaseModel0		I
file	cfg/mcp.settings.json	_Inxmcp() →	controller(\$proc,\$ispre,\$scope		\$this →	getRes(\$resname)	getArgCtl(\$name)
	/app.php	_	api(\$srv, \$ispre, \$scope, \$mod			getCfg(\$cfgname)	getArgOut(\$name)
(app.php)	\$Inxmcp_phar		service(\$srv,\$ispre,\$scope,\$m	nod,\$subc,\$ven)		getDriver(\$drvlabel)	setArgOut(\$name, \$value
resource	app.def		page(\$page,\$scope,\$mod,\$ven,\$pathtpl,\$hasreturn)			getCommon(\$name)	setReturn(\$return)
	\$app_path		block(\$page,\$scope,\$mod,\$ve	en,\$pathtpl,\$hasreturn)		callCmd(array \$scopeCtl, array	\$scopeIn)
	app.path		render(\$page,\$scope,\$mod,\$v	ven,\$pathtpl,\$hasreturn)		callTag(\$action,\$scopeIn, \$buffer)	
	app.lang	=	driver(\$name,\$isp,\$scpe,\$moo			debug(\$messge)	info(\$messge)
	app.menu.InitCommon	-	shell(\$ctrlproc,\$scope,\$mod,\$		1		error(\$messge)
		$\dashv$		· · · · · · · · · · · · · · · · · · ·		warning(\$messge)	enor(aniessge)
Of O - 44	app.debug and app.level	0	remote(\$proc,\$scopeIn,\$mod,	ΦSubcaii,Φveri)	pdoDriver	I	I
Sys Cfg Setti	Ť	Sys Debug	T		Scopeln	"E" as Driver envs	"Q" as Query
nxmcp() →	getCfg(\$name)	Inxmcp() →	debug(\$message)			"V" value	"T" type
	setCfg(\$name,\$value)		info(\$message)		Type	"e" execute	"q" simpleQuery
	getResource(\$name)		warning(\$message)			"f" firstRow	"c" simpleCount
	getCommon(\$name)		error(\$message			"er" executeWithRollback	's' return sql
	setCommon(\$name,\$value)		rem(\$msg [,\$msg2])	(F:app.web.rem)	Inxmo	p() → runCommand() /	
	Setcommon(whame, walac)	$\dashv$			cmd list	i ·	· ·
		4	dump( \$msg )	(F:app.web.dump)	Citiu iist	extTemplate	extFile
			<u> </u>		4	page	showPage
		Database				showFullCommonBlock	
Sys Tools		resource	("Driver.[db label])			blockShell	blockRemote
nxmcp() →	header(\$string,\$end,\$retcode,	Inxmcp() →	queryJsonR(\$name,\$scope,\$r	module)		block	blockCommon
legacy move)	move(\$file,\$filedef,\$ext,\$path,\$end)	7 . ,	query(\$db,\$ispreload,\$scope,\$		1	showBlock	showCommonBlock
(legacy move)	Rem(\$var,[\$var])	┪	queryR(\$db,\$ispreload,\$scope		1	render	renderCommon
		+	queryCommonR(\$db,\$ispreloa		†		
	supportmail(\$message)	$\dashv$		ια,ψουυμε,φιπου,ΦSuBC)	1	service	serviceCommon
	mail(\$page,\$scope,module,\$vendor)	4	queryArrayR(\$scopeIn)		4	apiArray	apiArrayCommon
	escapeClear(\$string)	_			1	арі	apiReturn
	ConvertToAscii(\$string)	┙				apiController	ApiService
function	InxMcpExit([\$message])					shell	remote
		exTemplate	tags			driver	mail
			[scope- <vars>]</vars>	[scope-dump]	1	run	load
Suo Admin		-			-		
Sys Admin		-	[common- <vars></vars>	[server- <vars>]</vars>		controllerReturn	tag
resource	mcp.web.api	4.	[Inxmcp- <tags>]</tags>	[ <label>-<tags>]</tags></label>		controller	controllerCommon
	mcp.web.admin	_ <lnxmcp></lnxmcp>	name	module		queryArray	
			vendor	type		query	queryCommon
unction	Inxmcpadm(\$cmd)		disable-rem	Block-type		headerHttp	headerClose
shell	Inxmcp-adm \$cmd	(block-type)	config	ison		header	
web	•	(block type)		<u> </u>	-		ious a assint Common
web	/Inxmcpadm	4	common	scope	-	javascript	javascriptCommon
	/Inxmcpapi	_	translate	javascript	4	shell	remote
			print_r			print	clear
web function	home , form ,mail					exit	dumpexit
shell function	Ihttp. checksintax.						
shell function	http, checksintax,	7				ample Service:	
shell function	·	oller:			Ev		
	http, checksintax,  Example Conti	oller:			Ex	ample Service.	
	·	oller:			Ex	ample Service.	
php ** ** <DESCRIPTIC</td <td>Example Conti</td> <td>roller:</td> <td></td> <td>c2nhn</td> <td>Ex</td> <td>ample Service.</td> <td></td>	Example Conti	roller:		c2nhn	Ex	ample Service.	
php ** ** <DESCRIPTIC **/</td <td>Example Contr</td> <td>roller:</td> <td></td> <td><?php /**</td> <td>Ex</td> <td>апріє Зегисе.</td> <td></td>	Example Contr	roller:		php<br /**	Ex	апріє Зегисе.	
php  **  *DESCRIPTIC  **/ namespace App\</td <td>Example Control  ON&gt;  <module>\Controller;</module></td> <td>oller:</td> <td></td> <td>/** ** <description></description></td> <td></td> <td>ample Service.</td> <td></td>	Example Control  ON> <module>\Controller;</module>	oller:		/** ** <description></description>		ample Service.	
php  **  *DESCRIPTIC  **/ namespace Appl«</td <td>Example Controller; pp\Module&gt;\Controller;</td> <td>oller:</td> <td></td> <td>/** ** <description> **/</description></td> <td></td> <td>ample Service.</td> <td></td>	Example Controller; pp\Module>\Controller;	oller:		/** ** <description> **/</description>		ample Service.	
<pre><?php ** ** <DESCRIPTIC **/ namespace App\s use LinHUniXIMc class <Name>Co /**</pre>	Example Controller;  SModule>\Controller;  p\Model\mcpBaseModelClass;  introller extends mcpBaseModelClass {	roller:		/**  ** <description>  **/ namespace App\<mo< td=""><td>odule&gt;\Service;</td><td></td><td></td></mo<></description>	odule>\Service;		
<pre><?php ** *< CDESCRIPTIC **/ namespace Appl use LinHUniXIMc class <Name>Co /** * Ideally this me</pre>	Example Controller; pp\Module>\Controller;	coller:		/** ** <description> **/</description>	, odule>\Service; lode\lmcpServiceW	lodelClass;	
<pre><?php ** ** < DESCRIPTIC **/ namespace Apply use LinHUniX\Mc class < Name > Co /** * Ideally this me */</pre>	Example Control  N>  Module>\Controller; pl\Model\mcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution	coller:		/**  ** < DESCRIPTION>  **/ namespace App\ <m <="" class="" linhunix\mcp\\="" name="" use=""> exter /**</m>	odule>\Service; lode\mcpServiceN ds mcpServiceMo	todelClass; delClass {	
php ** ** *CDESCRIPTIC **/ namespace App\- sse LinHUniXIMc class <Name Co /* * Ideally this me */ protected function */	Example Contri  ON> <module>\Controller;  thodule&gt;\Controller;  thodule&gt;\Controller;  thodule&gt;\Controller;  thodule&gt;\Controller;  thodule&gt;\Controller extends mcpBaseModelClass;  ethod shuld be used to first esecution  on moduleInit(){</module>	oller:		/**  ** <description>  **/ namespace App\<muse <name="" class="" linhunix\mcp\\m=""> exter /**  * Ideally this methe</muse></description>	odule>\Service; lode\mcpServiceN ds mcpServiceMo	todelClass; delClass {	
<pre><?php ** ** < DESCRIPTIC **/ namespace App\* set LinHUniXIMc class < Name>Co /** * I deally this m */ protected functic \$this&gt;-space</pre>	Example Control  N>  Module>\Controller; pl\Model\mcpBaseModelClass; introller extends mcpBaseModelClass { ethod shuld be used to first esecution	oller:		/**  ** <description>  **/ namespace App\<muse <name="" class="" linhunix\mcp\m=""> exter /**  * Ideally this methor */</muse></description>	bdule>\Service; lode\mcpServiceMo ds mcpServiceMo bd shuld be used to	todelClass; delClass {	
***  ***  ** ODESCRIPTIC  ***  ***  ***  **  **  **  **  **  *	Example Controller; pModeleNcontroller; pModeleNcopBaseModelClass; entroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_;	oller:		/**  *** <description>  **/ namespace App\<mo <name="" class="" linhunix\mcpt\="" use=""> exter  /**  * Ideally this meth  */ Protected function in</mo></description>	odule>\Service; lodel\mcpServiceN ds mcpServiceMo od shuld be used to	todelClass; delClass { 0 first esecution	
*** *** *** *** *** ** ** ** ** ** ** *	Example Control  ON> <module>\Controller; pp\Mode\mopBaseMode\Class; introller extends mcpBaseMode\class { ethod shuld be used to first esecution on moduleInit(){ ename= _NAMESPACE_; name= _CLASS_;</module>	oller:		/**  ** <description>  **/ namespace App\<mu <name="" class="" linhunix\mcp\m="" use=""> exter /**  * Ideally this meth */ Protected function r \$this-&gt;spacena</mu></description>	bdule>\Service; lode\mcpServiceMo ds mcpServiceMo bd shuld be used to	todelClass; delClass { 0 first esecution	
:?php ** ** ** *> CDESCRIPTIC ** */ */ */ */ * * * * * * * * * * * *	Example Controller; pModeleNcontroller; pModeleNcopBaseModelClass; entroller extends mcpBaseModelClass { ethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_;			/**  **OESCRIPTION>  **/ namespace App\ <m <name="" class="" linhunixmcplm="" use=""> exter /**  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan }</m>	odule>\Service; lode\mcpServiceM ds mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC	todelClass; delClass { 0 first esecution	
??php ** ** ** ** ** ** ** * * * * * * * *	Example Control  NN>  *Module>\Controller;  p\\Model\mcpBaseModelClass;  introller extends mcpBaseModelClass {  ethod shuld be used to first esecution  on moduleInit() {  ename =NAMESPACE;  name =CLASS;  ethod shuld be used to insert  de and the other are to be used only as no			/**  ** <description: ***="" <="" app\<muse="" linhunix\mcpl\mcpl\mclass="" name="" namespace=""> exter /**  * Ideally this meth */ Protected function n \$this-&gt;spacena \$this-&gt;classnan } /**</description:>	odule>\Service; lode\mcpServiceM ids mcpServiceMo od shuld be used to noduleInit(){ ne=CLASS;	todelClass; delClass { 0 first esecution	
:?php ** ** ** *DESCRIPTIC ** *Ideally this me */ *I deally this me *I deally thi	Example Controller; phModelNcopBaseModelClass; phModelmcpBaseModelClass; tethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_; name=_CLASS_;  ethod shuld be used to insert de and the other are to be used only as no on moduleCore() {			/**  ** <description: ***="" <name="" appl<muse="" class="" linhuniximcplm="" namespace=""> exter /**  * Ideally this meth */ Protected function n \$this-&gt;spacena \$this-&gt;classnan } /**  * standard 1 shot */</description:>	odule>\Service; lode\mcpService\; lode\mcpServiceMo dd shuld be used to nodulelnit(){ me=CLASS; user	iodelClass; delClass { first esecution SE;	
??php ** ** ** *>DESCRIPTIC ** ** ** * * * * * * * * * * * * * *	Example Controller; phModelNcopBaseModelClass; phModelmcpBaseModelClass; tethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_; name=_CLASS_;  ethod shuld be used to insert de and the other are to be used only as no on moduleCore() {			/**  ** <description: ***="" <="" app\<muse="" linhunix\mcpl\mcpl\mclass="" name="" namespace=""> exter /**  * Ideally this meth */ Protected function n \$this-&gt;spacena \$this-&gt;classnan } /**</description:>	odule>\Service; lode\mcpService\; lode\mcpServiceMo dd shuld be used to nodulelnit(){ me=CLASS; user	iodelClass; delClass { first esecution SE;	
??php ** ** ** *>DESCRIPTIC ** ** ** * * * * * * * * * * * * * *	Example Controller; pModelNcontroller; pModelNcopBaseModelClass; pModelNmcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>=argin; th to be implemented			/**  **CDESCRIPTION: **/ namespace Appl <mu <name="" class="" linhunixmcplm="" use=""> exter /**  * Ideally this meth */ Protected function r \$this-&gt;spacenau \$this-&gt;classnan }  * standard 1 shot */ protected function /*/ protected function /*/ /**</mu>	odule>\Service; lodel\mcpServicew ds mcpServiceMo ad shuld be used to modulelnit(){ me=_NAMESPAC me=_CLASS_; user moduleSingleTon(	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  ***  **  **  **  *	Example Controller; pModelNcontroller; pModelNcopBaseModelClass; pModelNmcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>=argin; th to be implemented			**  **CDESCRIPTION: **I namespace Appl <mu <name="" class="" linhunx="" mcplm="" use=""> exter  **  * Ideally this meth */ Protected function   \$this-&gt;classnan } } * standard 1 shot */ protected function  **  * standard 5 shot */ protected function  **  * function &lt;&gt; &gt;=e: *      *  </mu>	odule>\Service; lodel\mcpServicew ds mcpServiceMo ad shuld be used to modulelnit(){ me=_NAMESPAC me=_CLASS_; user moduleSingleTon(	iodelClass; delClass { first esecution SE;	
***  **  **  **  **  **  **  **  **  *	Example Controller; ppModelmcpBaseModelClass; pmModelmcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ pname=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>argin; the beinplemented sarrayout;			/**  **<\DESCRIPTION: *** namespace App\ <mu linhunix\mcpl\mcdl\mcpl\mcdl\mcdl\mcpl\mcdl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcp<="" td="" use=""><td>odule&gt;\Service; lodel\mcpServicew ds mcpServiceMo ad shuld be used to modulelnit(){ me=_NAMESPAC me=_CLASS_; user moduleSingleTon(</td><td>iodelClass; delClass { first esecution SE;</td><td></td></mu>	odule>\Service; lodel\mcpServicew ds mcpServiceMo ad shuld be used to modulelnit(){ me=_NAMESPAC me=_CLASS_; user moduleSingleTon(	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Controller; pModelNcontroller; pModelNcopBaseModelClass; pModelNmcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=NAMESPACE; name=CLASS; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>=argin; th to be implemented	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
<pre> ??php ** ** ** *DESCRIPTIC ** ** ** * * * * * * * * * * * * * *</pre>	Example Control  NN>  *Module>\Controller; pp\Mode\mopBaseMode\Class; introller extends mcpBaseMode\class; introller extends mcpBaseMode\class { ethod shuld be used to first esecution on module\nit(){ ename=_CLASS_;  ethod shuld be used to insert de and the other are to be used only as no on module\core() { sthis>arg(n; ith to be implemented sarrayout;  ethod shuld be used to insert de and the other are to be used only as no on module\core() { sthis>arg(n; ith to be implemented sarrayout;  ethod shuld be used to insert de and the other are to be used only as no	ormal		/**  **<\DESCRIPTION: *** namespace App\ <mu linhunix\mcpl\mcdl\mcpl\mcdl\mcdl\mcpl\mcdl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcp<="" td="" use=""><td>odule&gt;\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{</td><td>iodelClass; delClass { first esecution SE;</td><td></td></mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Control  N>  Module>\Controller; p\Model\model	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Control  NN>  *Module>\Controller; pp\Mode\mopBaseMode\Class; introller extends mcpBaseMode\class; introller extends mcpBaseMode\class { ethod shuld be used to first esecution on module\nit(){ ename=_CLASS_;  ethod shuld be used to insert de and the other are to be used only as no on module\core() { sthis>arg(n; ith to be implemented sarrayout;  ethod shuld be used to insert de and the other are to be used only as no on module\core() { sthis>arg(n; ith to be implemented sarrayout;  ethod shuld be used to insert de and the other are to be used only as no	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Control  N>  Module>\Controller; p\Model\model	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  ***  **  **  **  *	Example Control  N>  Module>\Controller; p\Model\model	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Control  N>  Module>\Controller; p\Model\model	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
***  ***  ***  ***  **  **  **  **  **	Example Control  N>  Module>\Controller; p\Model\model	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
:?php *** ** ** ** ** ** ** *DESCRIPTIC ** *Ideally this me */ ** *I deally this me * the model cod */ ** *I deally this me * the model cod */ ** *I deally this me * the model cod */ ** *I deally this me * the model cod */ */ ** *I deally this me * the model cod */ */ */ ** *I deally this me * the model cod */ */ */ ** *I deally this me * the model cod */ */ */ ** *I deally this me * the model cod */ */ */ ** *I deally this me * the model cod */ */ */ ** ** ** ** ** ** ** ** ** **	Example Controller; pModel\mopBaseModelClass; pModel\mopBaseModelClass; ptmodel\mopBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>argin; th to be implemented sarrayout; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>argin; the tobe implemented strayout; ethod shuld be used to insert de and the other are to be used only as no on moduleSingleTon() { tith to be implemented	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
?php ** *>DESCRIPTIO* ** *  amespace Apply se LinHUniXMc !ss Name>Co ** * Ideally this me *  protected functic \$this->space \$this->class! } ** * Ideally this me * the model co *  ** * Ideally this me * the model co *  ** * Ideally this me * the model co *  *  * Ideally this me * the model co *  *  * Ideally this me * the model co *  *  * Ideally this me * the model co *  *  * Ideally this me * the model co *  *  * Ideally this me * the model co *  *  * Son Query: *  ** ** ** ** ** ** ** ** ** ** ** **	Example Controller; pModel\mopBaseModelClass; pModel\mopBaseModelClass; ptmodel\mopBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=_NAMESPACE_; name=_CLASS_; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>argin; th to be implemented sarrayout; ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis>argin; the tobe implemented strayout; ethod shuld be used to insert de and the other are to be used only as no on moduleSingleTon() { tith to be implemented	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution SE;	
?php ** ** <descriptic* **="" *<="" td=""><td>Example Control  ON&gt;  *Module&gt;\Controller; pp\Mode\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ethod shuld be used to first esecution on module\nit(){ ename=_NAMESPACE_; name=_CLASS; ethod shuld be used to insert de and the other are to be used only as no on module\tcore() { *\ships-ships-\arguning{sh</td><td>ormal</td><td></td><td>/**  ** **/ namespace App\<mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu></td><td>odule&gt;\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{</td><td>iodelClass; delClass { first esecution E;</td><td></td></descriptic*>	Example Control  ON>  *Module>\Controller; pp\Mode\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ethod shuld be used to first esecution on module\nit(){ ename=_NAMESPACE_; name=_CLASS; ethod shuld be used to insert de and the other are to be used only as no on module\tcore() { *\ships-ships-\arguning{sh	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution E;	
??php ** ** *DESCRIPTIC* ** *Jeasly Name>Co ** *Ideally this me */ *Ideally this me *Ideally	Example Control  ON>  *Module>\Controller; pp\Mode\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ph\model\mcpBaseMode\(Class;\) ethod shuld be used to first esecution on module\nit(){ ename=_NAMESPACE_; name=_CLASS; ethod shuld be used to insert de and the other are to be used only as no on module\tcore() { *\ships-ships-\arguning{sh	ormal		/**  ** **/ namespace App\ <mu <="" linhunix\mcp\mcdlass="" name="" wes=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * []==xxx * [E]=yyy */</mu>	odule>\Service; lodel\mcpServiceN ids mcpServiceMo od shuld be used to moduleInit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon( e/0{	iodelClass; delClass { first esecution E;	