## Cheat Sheet

| Site Url  | https://github.com/linkumin/lenger   | Doc Url           | LNXMCP CHEAT S<br>http://lnxmcp.uk/ | Phar Url   | https://github.com  | llinhuniy/lnymen/blab/0.F.0/d                       | ict/Inymon phas           |
|---|--|-------------------|-------------------------------------|--|---|---|---------------------------|
| Cfa   | https://github.com/linhunix/lnxmcp   | Sys Comma         |                                     | Priar Uri  |   | /linhunix/lnxmcp/blob/3.5.0/d                       | ist/inxmcp.pnar           |
| Cfg   | T  | -                 |                                     |  | mcpBaseModel  |   | 1                         |
| 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   | iod,\$subc,\$ven)  |   | getDriver(\$drvlabel)                               | setArgOut(\$name, \$value |
| esource   | app.def  |                   | page(\$page,\$scope,\$mod,\$ve      | en,\$pathtpl,\$hasreturn)  |   | getCommon(\$name)                                   | setReturn(\$return)       |
|   | \$app_path   |                   | block(\$page,\$scope,\$mod,\$ve     | en,\$pathtpl,\$hasreturn)  |   | callCmd(array \$scopeCtl, array                     | \$scopeIn)                |
|   | app.path   | 7                 | render(\$page,\$scope,\$mod,\$v     | en,\$pathtpl,\$hasreturn)  | 1   | callTag(\$action,\$scopeIn, \$buffer)               |                           |
|   | app.lang   | =                 | driver(\$name,\$isp,\$scpe,\$moo    |  | 1   | debug(\$messge)                                     | info(\$messge)            |
|   | app.menu.InitCommon  | +                 | shell(\$ctrlproc,\$scope,\$mod,\$   |  | 1   |   | error(\$messge)           |
|   |  | $\dashv$          |                                     | · · · · · · · · · · · · · · · · · · ·  | d. Duitere  | warning(\$messge)                                   | enor(aniessge)            |
|   | app.debug and app.level  |                   | remote(\$proc,\$scopeIn,\$mod,      | \$Subcaii,\$ven)   | pdoDriver   |   | T                         |
| Sys Cfg Setti   | ings   | Sys Debug         |                                     |  | 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)  | 1                 | error(\$message                     |  | 1   | "er" executeWithRollback                            | 's' return sql            |
|   | setCommon(\$name,\$value)  | _                 | rem(\$msg [,\$msg2])                | (F:app.web.rem)  | Invmc   | p() → runCommand() /                                |                           |
|   | setCommon(sname,svalue)  | -                 |                                     |  |   | i ·   | T .                       |
|   |  | 4                 | dump( \$msg )                       | (F:app.web.dump)   | cmd list  | extTemplate   | extFile                   |
|   |  |                   |                                     |  | 4   | page  | showPage                  |
|   |  | Database          |                                     |  |   | showFullCommonBlock                                 |                           |
| Sys Tools   |  | resource          | ("Driver.[db label])                |  |   | blockShell  | blockRemote               |
| nxmcp() →   | header(\$string,\$end,\$retcode,   | Inxmcp() →        | queryJsonR(\$name,\$scope,\$r       | nodule)  |   | block   | blockCommon               |
| legacy move)  | move(\$file,\$filedef,\$ext,\$path,\$end)  | 7                 | query(\$db,\$ispreload,\$scope,\$   | <del></del>  | 1   | showBlock   | showCommonBlock           |
| (legacy move)   |  | -                 | queryR(\$db,\$ispreload,\$scope     |  | 1   |   |                           |
|   | Rem(\$var,[\$var])   | -                 |                                     |  | +   | render  | renderCommon              |
|   | supportmail(\$message)   | 4                 | queryCommonR(\$db,\$ispreloa        | u,\$scope,\$mod,\$subc)  | 4   | service   | serviceCommon             |
|   | mail(\$page,\$scope,module,\$vendor)   | _                 | queryArrayR(\$scopeIn)              | 1  | 1   | apiArray  | apiArrayCommon            |
|   | escapeClear(\$string)  |                   |                                     |  |   | арі   | apiReturn                 |
|   | ConvertToAscii(\$string)   |                   |                                     |  |   | shell   | remote                    |
| unction   | InxMcpExit([\$message])  |                   |                                     |  | 1   | driver  | mail                      |
|   | op=.m([4mossage])  | exTemplate        | tans                                |  |   | run   | load                      |
|   |  | extemplate        |                                     | L  | -   |   |                           |
|   |  | _                 | [scope- <vars>]</vars>              | [scope-dump]   |   | controllerReturn                                    | tag                       |
| Sys Admin   | _ <del>_</del>   |                   | [common- <vars></vars>              | [server- <vars>]</vars>  |   | controller  | controllerCommon          |
| resource  | mcp.web.api  |                   | [Inxmcp- <tags>]</tags>             | [ <label>-<tags>]</tags></label>   |   | queryArray  |                           |
|   | mcp.web.admin  | <inxmcp></inxmcp> | name                                | module   |   | query   | queryCommon               |
|   | ·  |                   | vendor                              | type   | 1   | headerHttp  | headerClose               |
| function  | Investor a des (Camel)   | - T               |                                     |  | 1   | header  | nedder Globe              |
|   | Inxmcpadm(\$cmd)   | (hla al. h.ma)    | disable-rem                         | Block-type   | -   |   |                           |
| shell   | Inxmcp-adm \$cmd   | _(block-type)     | config                              | json   |   | javascript  | javascriptCommon          |
| web<br>web function   | /Inxmcpadm   |                   | common                              | scope  |   | shell   | remote                    |
|   | /Inxmcpapi   |                   | translate                           | javascript   |   | print   | clear                     |
|   |  |                   | print r                             |  |   | exit  | dumpexit                  |
|   | home , form ,mail  |                   | _                                   |  |   |   |                           |
| chall function  |  | _                 |                                     |  |   |   |                           |
|   | http://do.dicintory  | _                 |                                     |  | -   |   |                           |
| shell function  |  |                   |                                     |  | +   |   |                           |
| shell function  | http, checksintax,   |                   |                                     |  |   |   |                           |
| shell function  |  |                   |                                     |  |   |   |                           |
|   | Example Contr  | oller:            |                                     |  | Ex  | ample Service:                                      |                           |
|   |  | oller:            |                                     |  | Ex  | ampie dei vice.                                     |                           |
| php</td <td>Example Conti</td> <td>roller:</td> <td></td> <td></td> <td>Ex</td> <td>ample dervice.</td> <td></td>   | Example Conti  | roller:           |                                     |  | Ex  | ample dervice.                                      |                           |
| php ** ** <DESCRIPTIC **/</td <td>Example Conti</td> <td>roller:</td> <td></td> <td><?php</td><td>Ex</td><td>ample Service.</td><td></td></td>  | Example Conti  | roller:           |                                     | php</td <td>Ex</td> <td>ample Service.</td> <td></td>  | Ex  | ample Service.                                      |                           |
| php  **  *DESCRIPTIC  **/ namespace App\</td <td>Example Control ON&gt;</td> <td>oller:</td> <td></td> <td><?php<br>/**<br/>** <description></description></td> <td></td> <td>ample Service.</td> <td></td>   | Example Control ON>  | oller:            |                                     | php<br /**<br>** <description></description>   |   | ample Service.                                      |                           |
| php  ***  *DESCRIPTIC  **/ namespace Appl« use LinHUniX\Mc</td <td>Example Control  ON&gt;  <module>\Controller; pp\Model\mcpBaseModelClass;`</module></td> <td>oller:</td> <td></td> <td>/**<br/>** <description><br/>**/</description></td> <td>,</td> <td>ample dervice.</td> <td></td>  | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass;`</module>  | oller:            |                                     | /**<br>** <description><br/>**/</description>  | ,   | ample dervice.                                      |                           |
| php *** ** <DESCRIPTIC *** namespace App\suse LinHUniX\Mc class <Name Co /**  | Example Control  N> <module>\Controller;  pp\Model\mcpBaseModelClass;  ntroller extends mcpBaseModelClass {</module>   | coller:           |                                     | /**  ** <description>  **/ namespace App\<mo< td=""><td>odule&gt;\Service;</td><td></td><td></td></mo<></description>  | odule>\Service;   |   |                           |
| php  **  **  DESCRIPTIO  **/ namespace Appl use LinHUniXIMc class <Name Co  /**  ** Ideally this me   | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass;`</module>  | coller:           |                                     | /**  ** <description>  **/ namespace App\<mo linhunix\mcp\m<="" td="" use=""><td>odule&gt;\Service;</td><td>lodelClass;</td><td></td></mo></description>   | odule>\Service;   | lodelClass;   |                           |
| php ht*  ** <DESCRIPTIC **/ namespace Apply ase LinHUNIWM class <Name Co /**  * Ideally this me */  | Example Controller; <a href="Module">Module</a> <a href="Module">Controller;</a> <a href="Module">Module</a> <a h<="" td=""><td>coller:</td><td></td><td>/**  ** <description>  **/ namespace App\<mo< td=""><td>odule&gt;\Service;</td><td>lodelClass;</td><td></td></mo<></description></td></a> | coller:           |                                     | /**  ** <description>  **/ namespace App\<mo< td=""><td>odule&gt;\Service;</td><td>lodelClass;</td><td></td></mo<></description>   | odule>\Service;   | lodelClass;   |                           |
| php ***  *CDESCRIPTIC **/ namespace App\- use LinHUniXIMc class <Name Co /**  * Ideally this me */ protected function ** **  **  **  **  **  **  **  **  **   | Example Contri  ON> <module>\Controller; pp\Model\mcpBaseModelClass; introller extends mcpBaseModelClass {     ethod shuld be used to first esecution     on moduleInit(){</module>  | oller:            |                                     | /**  ** < DESCRIPTION>  **/ namespace App\ <muse <="" class="" linhunix\mcp\\="" name=""> exter /**  * Ideally this methi</muse>   | odule>\Service;<br>lodel\mcpServiceN<br>ids mcpServiceMo  | todelClass;<br>delClass {                           |                           |
| php ** ** < DESCRIPTIC **/ namespace App\subseteq LinHUniXIMc class < Name Co /** * I deally this m */ protected functic \$this->space  | Example Controller; p)Module>\Controller; p)Mode\mcpBaseMode\Class; introller extends mcpBaseMode\Class { tethod shuld be used to first esecution on module\nit(){ ename=_NAMESPACE_;  | oller:            |                                     | /** *** <description> ***/ namespace App\<mu <name="" class="" linhunix\mcp\m="" use=""> exter /** * Ideally this methe */</mu></description>  | odule>\Service;<br>fodelmcpServiceM<br>ids mcpServiceMo<br>od shuld be used to  | todelClass;<br>delClass {                           |                           |
| <pre><?php *** ** <DESCRIPTIC *** namespace Apply se LinHUniXIMc lass <Name>Co /** * Ideally this m */ protected functic \$this-&gt;space \$this-&gt;classr \$this-&gt;classr </pre>  | Example Contri  ON> <module>\Controller; pp\Model\mcpBaseModelClass; introller extends mcpBaseModelClass {     ethod shuld be used to first esecution     on moduleInit(){</module>  | oller:            |                                     | /** * DESCRIPTION> *** (DESCRIPTION) **/ namespace App\\Mcptlw use LinHUniX\Mcptlw Class < Name> exter /*  * Ideally this meth */ Protected function in  | odule>\Service;<br>lodel\mcpServiceN<br>ids mcpServiceMo<br>od shuld be used to<br>modulelnit(){                                    | todelClass;<br>delClass {<br>0 first esecution      |                           |
| <pre><?php *** ** <DESCRIPTIC **/ namespace Appl\se LinHUniXMc class <Name>co /** * Ideally this mr */ protected functic \$this&gt;&gt;space \$this&gt;&gt;class } /**</pre>  | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass; ontroller extends mcpBaseModelClass {  ethod shuld be used to first esecution on moduleInit() { ename=_NAMESPACE_; name=_CLASS_;</module>  | oller:            |                                     | /**  ** < DESCRIPTION>  **/ namespace App\< M use LinHUniX\Mcp\M Class < Name> exter /**  * Ideally this meth */ Protected function r \$this->spacena  | odule>\Service;<br>lode\mcpServiceM<br>ids mcpServiceMo<br>and shuld be used to<br>moduleInit(){<br>me=NAMESPAC                     | todelClass;<br>delClass {<br>0 first esecution      |                           |
| <pre> ??php ** ** <descriptic **="" <name="" appl="" lass="" linhunixmc="" namespace="" sse="">Co /** * I deally this m */ protected functic \$this-&gt;space \$this-&gt;classr } /* * I deally this m */ */ */ */ */ */ */ */ */ */ */ */ */</descriptic></pre>  | Example Control  ON>  *Module>\Controller;  top\Model\mcpBaseModelClass;  introller extends mcpBaseModelClass {  tethod shuld be used to first esecution  on moduleInit() {  ename=NAMESPACE;  name=CLASS;  tethod shuld be used to insert   |                   |                                     | /**  *** CDESCRIPTION>  **/ namespace App\ <m <name="" class="" linhunixmcplm="" use=""> exter /**  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan }</m>  | odule>\Service;<br>lodel\mcpServiceN<br>ids mcpServiceMo<br>od shuld be used to<br>modulelnit(){                                    | todelClass;<br>delClass {<br>0 first esecution      |                           |
| ??php ** ** ** *>DESCRIPTIC ** ** * * * * * * * * * * * * * * * *   | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass; ontroller extends mcpBaseModelClass {  ethod shuld be used to first esecution on moduleInit() { ename=_NAMESPACE_; name=_CLASS_;</module>  |                   |                                     | /**  ** <description: ***="" <name="" appl<mu="" class="" linhuniximcplm="" namespace="" use=""> exter /**  * Ideally this meth */ Protected function r \$this-&gt;spacena \$this-&gt;classnan } /**</description:>  | odule>\Service;<br>fode\mcpServiceM<br>ids mcpServiceMo<br>od shuld be used to<br>moduleInit(){<br>me=NAMESPAC<br>ne=CLASS;         | todelClass;<br>delClass {<br>0 first esecution      |                           |
| :?php ** ** ** *DESCRIPTIC ** *Ideally this me */ *I total system */ *I deally this me *I deally this   | Example Controller; phModelmcpBaseModelClass; phModelmcpBaseModelClass; tethod shuld be used to first esecution on moduleInit(){ ename= _NAMESPACE_; name=_CLASS_; tethod shuld be used to insert de and the other are to be used only as no on moduleCore() {   |                   |                                     | /**  *** CDESCRIPTION>  **/ namespace App\ <m <name="" class="" linhunixmcplm="" use=""> exter /**  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan }</m>  | odule>\Service;<br>fode\mcpServiceM<br>ids mcpServiceMo<br>od shuld be used to<br>moduleInit(){<br>me=NAMESPAC<br>ne=CLASS;         | todelClass;<br>delClass {<br>0 first esecution      |                           |
| <pre> ??php *** ** <descriptic ***="" <name="" alass="" amespace="" apply="">Co /** * Ideally this me */ protected functic \$this-&gt;class } /* * Ideally this me * the model cot */ protected functic \$this-&gt;class } /* * Ideally this me * the model cot */ * The mo</descriptic></pre>  | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass; ontroller 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     on moduleCore() {     sthis-&gt;argin;</module>  |                   |                                     | /**  **CDESCRIPTION: *** namespace Appl <me <name="" class="" linhuniximcplm="" use=""> exter /**  * Ideally this meth */ Protected function r \$this-&gt;spacena \$this-&gt;classnan } /**  * standard 1 shot */</me>   | odule>\Service;<br>fode\mcpServiceM<br>ids mcpServiceMo<br>od shuld be used to<br>moduleInit(){<br>me=NAMESPAC<br>ne=CLASS;         | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  **  **  **  **  **  **  **  | Example Controller;  pModule>\Controller;  pModel\mcpBaseModelClass;  tethod shuld be used to first esecution  on moduleInit(){  ename=_NAMESPACE_;  name=_CLASS_;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *sthis>-argin; this to be implemented  |                   |                                     | ***  **CDESCRIPTION: ***  namespace Appl <mu linhunix\mcpl\mcdl\mcpl\mcdl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcpl\mcp<="" td="" use=""><td>odule&gt;\Service; flode\\mcpService\\ ids mcpServiceMod shuld be used to moduleInit(){ me=_NAMESPAC ne=_CLASS_; user moduleSingleTon(</td><td>iodelClass;<br/>delClass {<br/>first esecution<br/>SE;</td><td></td></mu>   | odule>\Service; flode\\mcpService\\ ids mcpServiceMod shuld be used to moduleInit(){ me=_NAMESPAC ne=_CLASS_; user moduleSingleTon( | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  ***  ***  **  **  **  **  **  | Example Controller;  pModule>\Controller;  pModel\mcpBaseModelClass;  tethod shuld be used to first esecution  on moduleInit(){  ename=_NAMESPACE_;  name=_CLASS_;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *sthis>-argin; this to be implemented  |                   |                                     | /**  ** <description: ***="" <name="" appl<muse="" class="" linhuliximcplm="" namespace=""> exter /*  * Ideally this methe */ Protected function n \$this-&gt;spacenan \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * standard 1 shot */ * tunction &lt;&gt;&gt; /* * function &lt;&gt;&gt;  &gt;* function &lt;&gt;&gt; </description:>   | odule>\Service; flode\\mcpService\\ ids mcpServiceMod shuld be used to moduleInit(){ me=_NAMESPAC ne=_CLASS_; user moduleSingleTon( | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** ** <DESCRIPTIC *** namespace Apply sse LinHUniXMc lass <Name>Co /** * Ideally this m */ protected functic \$this-&gt;classr } /** * Ideally this m * the model co */ protected functic \$mydataarray= /// if is empty wai \$this-&gt;argOut= } /** </pre>  | Example Controller; pplModelmcpBaseModelClass; piModelmcpBaseModelClass; 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 on moduleCore() { sthis->argin; ithis->argin; ithis->argin; ithis->arrayout;   |                   |                                     | **  **CDESCRIPTION: ***  namespace Appl <mu <="" linhunix\mcpl\mcpl\mclass="" name="" use=""> exter  **  * Ideally this meth */ Protected function in \$this-&gt;spacenal \$this-&gt;classnan }  **  * standard 1 shot */ protected function  **  * function &lt; &gt; &lt;=: * fill=xxx.</mu>   | odule>\Service; flode\\mcpService\\ ids mcpServiceMod shuld be used to moduleInit(){ me=_NAMESPAC ne=_CLASS_; user moduleSingleTon( | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** ** <DESCRIPTIC *** ** ** ** ** ** ** ** ** ** ** ** *</td><td>Example Control  Notation of the control of the con</td><td>ormal</td><td></td><td>/**  ** **/ namespace App\<mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu></td><td>odule&gt;\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(</td><td>iodelClass;<br/>delClass {<br/>first esecution<br/>SE;</td><td></td></pre>  | Example Control  Notation of the control of the con   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** ** <DESCRIPTIC *** ** ** ** ** ** ** ** ** ** ** ** *</td><td>Example Controller; pplModelmcpBaseModelClass; piModelmcpBaseModelClass; 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 on moduleCore() { sthis-&gt;argin; ithis-&gt;argin; ithis-&gt;argin; ithis-&gt;arrayout;</td><td>ormal</td><td></td><td>/**  **CESCRIPTION: ***/ namespace App\<mu <name="" class="" linhunix\mcp\mcp\m="" use=""> exter /**  * Ideally this meth */ Protected function n \$this-&gt;spacena \$this-&gt;classnan } /**  * standard 1 shot */ protected function /**  * function <t>&lt;=: * IT=xxx * [E=yyy * Testandard   Section   Section</t></mu></td><td>odule&gt;\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(</td><td>iodelClass;<br/>delClass {<br/>first esecution<br/>SE;</td><td></td></pre> | Example Controller; pplModelmcpBaseModelClass; piModelmcpBaseModelClass; 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 on moduleCore() { sthis->argin; ithis->argin; ithis->argin; ithis->arrayout;   | ormal             |                                     | /**  **CESCRIPTION: ***/ namespace App\ <mu <name="" class="" linhunix\mcp\mcp\m="" use=""> exter /**  * Ideally this meth */ Protected function n \$this-&gt;spacena \$this-&gt;classnan } /**  * standard 1 shot */ protected function /**  * function <t>&lt;=: * IT=xxx * [E=yyy * Testandard   Section   Section</t></mu> | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** <> Color</pre>   | Example Controller;  phModel>Controller;  phModel\text{Controller};  phModel\text{modelClass},  terthod shuld be used to first esecution  on moduleInit(){  ename=NAMESPACE;  name=CLASS;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleSingleTon() {   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** <> Color</pre>   | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass; introller extends mcpBaseModelClass {     ethod shuld be used to first esecution     on moduleInit() {     ename=_CLASS_;     ethod shuld be used to insert     de and the other are to be used only as no     on moduleCore() {     sthis&gt;arg(n;     ith to be implemented     sarrayout;     ethod shuld be used to insert     de and the other are to be used only as no     moduleCore() {     sthis&gt;arg(n;     ith to be implemented     sarrayout;     ethod shuld be used to insert     de and the other are to be used only as no     ethod shuld be used to insert     de and the other are to be used only as no</module>  | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  ***  **  **  **  **  **  **   | Example Controller;  phModel>Controller;  phModel\text{Controller};  phModel\text{modelClass},  terthod shuld be used to first esecution  on moduleInit(){  ename=NAMESPACE;  name=CLASS;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleSingleTon() {   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  ***  **  **  **  **  **  **   | Example Controller;  phModel>Controller;  phModel\text{Controller};  phModel\text{modelClass},  terthod shuld be used to first esecution  on moduleInit(){  ename=NAMESPACE;  name=CLASS;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleSingleTon() {   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  ***  **  **  **  **  **  **   | Example Controller;  phModel>Controller;  phModel\text{Controller};  phModel\text{modelClass},  terthod shuld be used to first esecution  on moduleInit(){  ename=NAMESPACE;  name=CLASS;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleCore() {  *shris>-argin; thit to be implemented  *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no  on moduleSingleTon() {   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| <pre><?php *** ** <DESCRIPTIC *** ** ** ** ** ** ** ** ** ** ** ** *</td><td>Example Control  N&gt;  Module&gt;\Controller; pytMode\mcpBaseMode\class; introller extends mcpBaseMode\class; introller extends 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 moduleCore() {     \$\frac{1}{2}\text{shis} &gt; \arg g n;     iit to be implemented     \frac{1}{2}\text{sarrayout;}      ethod shuld be used to insert     de and the other are to be used only as no     on moduleSingleTon() {     iith to be implemented</td><td>ormal</td><td></td><td>/**  ** **/ namespace App\<mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu></td><td>odule&gt;\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(</td><td>iodelClass;<br/>delClass {<br/>first esecution<br/>SE;</td><td></td></pre>   | Example Control  N>  Module>\Controller; pytMode\mcpBaseMode\class; introller extends mcpBaseMode\class; introller extends 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 moduleCore() {     \$\frac{1}{2}\text{shis} > \arg g n;     iit to be implemented     \frac{1}{2}\text{sarrayout;}      ethod shuld be used to insert     de and the other are to be used only as no     on moduleSingleTon() {     iith to be implemented   | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ***  ***  ***  ***  **  **  **  **  **  | Example Controller; controller; controller extends mcpBaseModelClass; controller 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     on moduleCore() {         sthis>argin;         this 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 to be implemented         sarrayout;         ethod shuld be used to insert         de and the other are to be used only as no         on moduleSingleTon() {         ith to be implemented  | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>SE; |                           |
| ?php ** ** *>DESCRIPTIC* ** ** *  ** *  ** *  *  *  ** *  *  *  | Example Control  ON> <module>\Controller; pp\Model\mcpBaseModelClass; pntroller extends mcpBaseModelClass; ethod shuld be used to first esecution on moduleInit(){ ename=_CLASS_;  ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { sthis&gt;argin; iith to be implemented sarrayout;  ethod shuld be used to insert de and the other are to be used only as no on moduleCore() { iith to be implemented starrayout;  ethod shuld be used to insert de and the other are to be used only as no on moduleSingleTon() { iith to be implemented  ethod shuld be used to insert de and the other are to be used only as no on moduleSingleTon() { iith to be implemented</module>  | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>E;  |                           |
| ?php ** ** *>DESCRIPTIC* ** ** *  ** *  ** *  *  *  ** *  *  *  | Example Controller; controller; controller extends mcpBaseModelClass; controller 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     on moduleCore() {         sthis>argin;         this 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 to be implemented         sarrayout;         ethod shuld be used to insert         de and the other are to be used only as no         on moduleSingleTon() {         ith to be implemented  | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>E;  |                           |
| ?php ** ** *>DESCRIPTIC* ** ** *  ** *  ** *  *  *  ** *  *  *  | Example Control  CN>  *Module>\Controller; pp\Mode\mcpBaseMode\(Class;\) phtroller extends mcpBaseMode\(Class;\) phtroller extends mcpBaseMode\(Class;\) pethod shuld be used to first esecution on module\(lait\) pethod shuld be used to insert de and the other are to be used only as no on module\(Core\) { *\$this>argin; th to be implemented *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no on module\(Core\) { *sthis>argin; th to be implemented *sarrayout;  tethod shuld be used to insert de and the other are to be used only as no on module\(Core\) f th to be implemented  the core module\(Core\) f the c  | ormal             |                                     | /**  ** **/ namespace App\ <mu <name="" class="" linhuniximcplm="" use=""> exter /*  * Ideally this meth */ Protected function or \$this-&gt;spacena \$this-&gt;classnan } /*  * standard 1 shot */ protected function /**  * function &lt;&gt; &lt;=: * fin=xxx * [E]=yyy */ */</mu>  | odule>\Service; todel\mcpService\ and shuld be used to modulelnit(){ me=NAMESPAC ne=CLASS; user moduleSingleTon(                    | iodelClass;<br>delClass {<br>first esecution<br>E;  |                           |