# REQEST LIFE CYCLE

http Request start

Define constant var

[Require config.inc.php](#config_ini_php)

[file](#config_ini_php)

[Context Class construct and initialization](#context_class_constructs_and_initial)

Is SSO

Authentication?

(is Single sign-on)

No

yes

Yes

[ModuleHandler](#ModuleHandl_class_constructs_and_initial)

[Initialization](#ModuleHandl_class_constructs_and_initial)

[Successful](#ModuleHandl_class_constructs_and_initial)

End for Request：

1. close session
2. close DB resource

Execute module action procedure

No

Display module contents

# Config.ini.php:

Error Report level setting

If \_\_ZBXE\_\_ defined

No

Exit script

Yes

Define version /path … constant(include the constants defined in package.inc.php)

Define debug/CDN/QUERY LOG/TEST/PROXY… Parameters

If \_\_XE\_LOADED\_CLASS\_\_ defined

Loading

func.inc.php

No

No

If \_\_DEBUG\_\_ is true

Yes

Start Record Time of \_\_elapsed\_class\_load\_\_

Loading

func.inc.php

Record Time of \_\_elapsed\_class\_load\_\_

# Context Class constructs and Initial:

* Construct the Context Class:

Construct a new FrontEndFileHandler

set FrontEndFileHandler variable of context

* Initialization the Context Class:
* Cookie setting($\_COOKIE);
* Language setting();
* Setting Request Method (GET/POST/XMLRPC/JSON);
* setRequestArgument(REQUEST/XmlRpc/JSON/UPLOAD);
* loadDBInfo : Load from the db.config.php and setDBInfo;
* set lang\_type to the cookie

Setting the context variables

No

If XE is installed

Set authentication information in Context and session

Yes

Initial session handler

No

If use db session

Loading language file of modules/module/lang

Set the language selected to Cookie and session

Yes

get virtual site information

* Session handler:
  + Open/Close/Read/Write/Destroy/Gc function defination

These functions defined in SessionController/

sessionModel

If is GET request

No

Yes

Set allow rewrite option

Set url and uri path

# ModuleHandler Class constructs and Initial:

* Construct the ModuleHandler Class:
* Var setting:
  + Module/act/mid/document\_srl/module\_srl/entry

Setting the moduleHandler variables

Yes

Validate variable above is XSS leak

echo error message

No

Execute addon(before module initialization)

Exit script

* Initialization the ModuleHandler Class:

get the variables

* Var getting:
  + oModuleModel
  + site\_module\_info

If document\_srl in request var

Yes

No

Get alian document\_srl by Alias

Get Module info by document\_srl/mid succeed

No

Set error message

Yes

Initial the module information

* Var setting:
  + setBrowserTitle
  + addHtmlHeader
  + getModulePartConfig
  + module info
* Redirect to the site default page

or

* 404 error message setting (return false)

Return true

* the ModuleHandler Class “procModule”:

If occur error in the initialization procedure

Yes

No

Check if has admin menu and the cache for the menu

Yes

Read module configuration in module.xml file

If find out the Module’s action

* Other prams:
  + Action type
  + Action ruleset
  + Kind(is admin action)
  + Is mobile action

Get other prams defined in xml file

Return error message Obj

* Find action:
  + Using action in xml
  + Using default action in xml

Return error message Obj

No

Return error message Obj

No

Yes

If is admin manage page request and denied\_admin in session

start

Yes

Create the admin menu and cache

No

Request for the View and case for using mobile phone

Create the module object

Yes

Create the mobile module object

If has rule set and the rule set file

No

If module object create succeed

No

Yes

Return error message Obj

No

Validate the rule set

Yes

Yes

If action in module object

Set session validate error message

Yes

No

Find out the action info in forward table

t

Return error message Obj

Yes

Reset the module object with forward info

Set module info:

Action/module info from xml

No

If default action exist

No

If from view and use mobile

No

Return error message Obj

Yes

Set action as default action

Yes

Add a mobile html Footer

If from view and not a admin type action

No

Yes

Add a module html Footer

If has failed message in session

No

Yes

Set a error message to context

[Execute the module object proc procedure](#ModuleObjec_class_proc)

If not a XMLRPC/JSON request

No

Yes

If proc procedure has error

No

Yes

Set successful message to session

Set error message to session

Return module object

* the ModuleHandler Class displayContent:

If param module not set or not a object

No

If not a XMLRPC/JSON call

No

Yes

Set error message

Yes

If the error message had occurred

If module not install and DB not connect

Yes

No

No

Yes

Redirect to the error return url

If other error occurred

Set error message

No

If module after trigger call fail

Yes

No

Set error message

Yes

Set object as message object to show error message

If layout\_srl exist

No

Yes

Print content

Set layout info

start

# Module Object Class:

* “proc” procedure in Module Object Class

If set stop\_proc

Yes

Proc procedure fails Return false

No

If module before triggers call succeed

No

If module after triggers call succeed

Proc procedure fails Return false

Set error message

No

Set error message

Yes

execute an addon(call called\_position as before\_module\_proc)

Proc procedure fails Return false

Yes

execute an addon(call called\_position as after\_module\_proc)

If action function exist in module

No

Yes

If module Running return error object

Proc procedure fails Return false

Yes

execute an addon(call called\_position as before\_module\_proc)

Proc procedure fails Return false

Set error message

No

execute api methods of the module if view action is and result is XMLRPC or JSON

Combine skin information with module information

Return true

Run the module action

# DisplayHandler object class:

* printContent procedure:

Set if gzip encoding supported

* + The request method type:
    - Xml request
    - XMLRPC request
    - JSON request
    - general HTTP request

[Extract contents to display by the request method](#handler_class_proc)

[Produce compliant content according to the request type](#handler_class_proc)(toDoc procedure)

No

If gzip encoding supported

Call display before Tringger

Yes

Gzip encode Header output

Call addon as position “before\_display\_content”

If http status is 200(successful request)

No

If prepareToPrint procedure exist in handler

Yes

No

Print http status code header

Yes

Print http header according to request type

[Call prepareToPrint procedure](#handler_class_prepareToPrint_proc)

Print debug message

No

If gzip encoding supported

Print result directly

Yes

Print gzip output

Call trigger display after

# \*\*\*\*\*\*\*\*Handler Class object:

* HTMLDisplayHandler toDoc procedure
* DisplayHandler toDoc procedure:
* JSONDisplayHandler toDoc procedure:
* VirtualXMLDisplayHandler toDoc procedure:
* XMLDisplayHandler toDoc procedure: