CommonGLPIInterface static _construct() #\$othertabs #\$showTitleInNavigation Header +\$rightname CommonGLPI + \$showdebug + \$taborientation + \$get_item_to_display_tab # \$type # \$displaylist + __construct() + can(\$ID, \$right, array &\$input=null) + defineTabs(\$options=[]) + addStandardTab(\$itemtype, array &\$ong, array \$options) + addDefaultFormTab (array &\$ong) + getTabNameForItem (CommonGLPI \$item, #\$foreign_key_fields \$withtemplate=0) + isNewItem() #\$forward_entity + showTabsContent(\$options=[]) to #\$notable + getHeaderName() #\$plugins_forward + display(\$options=[]) entity #\$tables_of + getErrorMessage(\$error, +\$undisclosedFields \$object=") + static getTypeName (\$nb=0)+ static getFieldLabel() + static getType() + static canCreate() + static canView() + static canUpdate() + static canDelete() + static canPurge() + static registerStandardTab (\$typeform, \$typetab) + static getOtherTabs (\$typeform) et 17 de plus... # getFormHeaderToolbar() CommonDBTM + \$fields + \$input + \$updates + \$oldvalues + \$dohistory + \$history_blacklist + \$auto_message_on_action + \$no_form_page + \$additional_fields _tor_dictionnary + \$taborientation + \$get_item_to_display_tab + \$deduplicate_queued _notifications + \$right # \$fkfield # \$searchopt # \$usenotepad + getFromResultSet(\$rs) + getID() + post_getFromDB() + showForm(\$ID, array \$options=[]) + getSNMPCredential() + getLockedFields() + post_getEmpty() + getLogTypeID() + post_deleteFromDB() + cleanDBonPurge() et 87 de plus... + static getTable(\$classname =null) + static forceTable(\$table) + static getForeignKeyField() + static getTableField (\$field, \$classname=null) + static getFromIter (DBmysqlIterator \$iter)

saveInput()

et 26 de plus...

+ static getIndexName() + static isNewID(\$ID) + static getNameField()

(&\$fields)

clearSavedInput()

+ static unsetUndisclosedFields

+ static getCompleteNameField()

restoreInput(array \$default=[])

restoreSavedValues

(array \$saved=[])

cleanLockedsOnAdd() # cleanLockeds()

computeFriendlyName()

OlaLevel_Ticket

+ static getTypeName (\$nb=0)

+ static cronInfo(\$name) + static doLevelForTicket (array \$data, \$olaType)