CommonGLPIInterface _construct() CommonGLPI + \$showdebug + \$taborientation + \$get_item_to_display_tab + static \$rightname # \$type # \$displaylist # static \$showTitleInNavigation Header # static \$othertabs + __construct() + can(\$ID, \$right, array &\$input=null) + defineTabs(\$options=[]) + addStandardTab(\$itemtype, array &\$ong, array \$options) + addDefaultFormTab (array &\$ong) + getTabNameForItem (CommonGLPI \$item, \$withtemplate=0) + isNewItem() + showTabsContent(\$options=[]) + getHeaderName() + display(\$options=[]) + getErrorMessage(\$error, \$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 _for_dictionnary + \$taborientation + \$get_item_to_display_tab + \$deduplicate_queued _notifications + \$right + static \$undisclosedFields # \$fkfield # \$searchopt # \$usenotepad # static \$notable # static \$forward_entity_to # static \$plugins_forward _entity # static \$tables_of # static \$foreign key _fields_of + 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) + static unsetUndisclosedFields (&\$fields) + static getIndexName() + static isNewID(\$ID) + static getNameField() + static getCompleteNameField() et 26 de plus... # saveInput() # clearSavedInput() # restoreInput(array \$default=[]) # restoreSavedValues (array \$saved=[]) # cleanLockedsOnAdd() # cleanLockeds() # computeFriendlyName() CommonDBConnexity + const DONT_CHECK_ITEM _RIGHTS + const HAVE_VIEW_RIGHT _ON_ITEM + const HAVE_SAME_RIGHT ON_ITEM + static \$canDeleteOnItemClean + static \$disableAutoEntity Forwarding + getCloneRelations() + getConnexityItem(\$itemtype, \$items_id, \$getFromDB=true, \$getEmpty =true, \$getFromDBOrEmpty=false) + checkAttachedItemChanges Allowed(array \$input, array \$fields) + tryEntityForwarding() + canConnexityItem(\$methodItem, \$methodNotItem, \$item_right, \$itemtype, \$items_id, CommonDBTM &\$item=null) + getHistoryChangeWhenUpdate Field(\$field) + getItemsForLog(\$itemtype, \$items_id) + static getSQLCriteriaToSearch ForItem(\$itemtype, \$items_id) + static getItemsAssociated To (\$itemtype, \$items_id) + static getItemField (\$itemtype) + static getItemFromArray (\$itemtype, \$items_id, array \$array, \$getFromDB =true, \$getEmpty=true, \$getFromDBOr Empty=false) + static canConnexity (\$method, \$item_right, \$itemtype, \$items_id) static getConnexityMassive ActionsSpecificities() + static getMassiveActions ForItemtype(array &\$actions, \$itemtype, \$is_deleted=false, CommonDBTM \$checkitem=null) + static showMassiveActions SubForm(MassiveAction \$ma) + static getConnexityInput ForProcessingOfMassiveActions (\$action, CommonDBTM \$item, array \$ids, array \$input) + static processMassiveActions ForOneItemtype(MassiveAction \$ma, CommonDBTM \$item, array \$ids) CommonDBChild + static \$itemtype + static \$items_id + static \$checkParentRights + static \$mustBeAttached + static \$logs_for_parent + static \$log_history_add + static \$log_history _update + static \$log_history _delete + static \$log_history_lock + static \$log_history _unlock + canCreateItem() + canViewItem() canUpdateItem() + canDeleteItem() + canChildItem(\$methodItem, \$methodNotItem) + getItem(\$getFromDB =true, \$getEmpty=true) + recursivelyGetItems() + getEntityID() + isEntityAssign() + maybeRecursive() et 12 de plus... + static getSQLCriteriaToSearch ForItem(\$itemtype, \$items_id) + static canCreate() + static canView() + static canUpdate() + static canDelete() + static canPurge() + static canChild(\$method) + static displayRecursiveItems (array \$recursiveItems, \$elementToDisplay, bool \$display=true) + static getJSCodeToAddFor ItemChild(\$field_name, \$child_count_js_var) + static showAddChildButton ForItemForm(CommonDBTM \$item, \$field_name, \$canedit =null, \$display=true) + static getItemField (\$itemtype) # autoinventoryInformation() NotificationTarget + \$prefix + \$table + \$notification_targets + \$notification_targets labels + \$notificationoptions + \$data + \$tag_descriptions + \$dohistory + \$target + \$entity et 12 de plus... + static \$itemtype + static \$items_id __construct(\$entity =", \$event=", \$object =null, \$options=[]) + getFromDBForTarget (\$notifications_id, \$type, \$ID) + validateSendTo(\$event, array \$infos, \$notify _me=false, \$emitter=null) + getSubjectPrefix(\$event=") + getContentHeader() + getContentFooter() + getMessageID() + showForNotification (Notification \$notification) + addAdditionnalInfosForTarget() + addAdditionnalUserInfo (array \$data) et 35 de plus... + static getTable(\$classname + static getMessageIdForEvent (?string \$itemtype, ?int \$items_id, ?string \$event) + static getTypeName (\$nb=0)+ static getInstance (\$item, \$event=", \$options=[]) + static getInstanceClass (string \$itemtype) + static getInstanceByType (\$itemtype, \$event=", \$options=[]) + static updateTargets (\$input) + static displayTabContent ForItem(CommonGLPI \$item, \$tabnum=1, \$withtemplate=0) # computeFriendlyName() # isMailMode() # isTargetItemRecursive() NotificationTargetCommon **ITILObject** + \$private_profiles + \$central_profiles + __construct(\$entity =", \$event=", \$object =null, \$options=[]) + validateSendTo(\$event, array \$infos, \$notify _me=false, \$emitter=null) + getSubjectPrefix(\$event=") + getEvents() + addItemAuthor() + addRecipientAddress() + addAdditionnalUserInfo (array \$data) + getProfileJoinCriteria() + isPrivate() + addAdditionalTargets (\$event=") + addSpecificTargets (\$data, \$options) + addDataForTemplate (\$event, \$options=[]) + getTags() # addMentionnedUser (array \$options) getActorData(CommonDBTM \$actor, int \$actortype, string \$key_prefix) NotificationTargetTicket + const HEADERTAG + const FOOTERTAG + validateSendTo(\$event, array \$infos, \$notify NotificationTargetProblem NotificationTargetChange me=false, \$emitter=null) + \$private_profiles + \$private_profiles + getSubjectPrefix(+ getEvents() + getEvents() \$event=") + getDataForObject(Common + getDataForObject(Common + getContentHeader() DBTM \$item, array \$options, DBTM \$item, array \$options, + getContentFooter() \$simple=false) \$simple=false) + getObjectItem(\$event=") + getTags() + getTags() + getEvents() + getDataForObject(Common DBTM \$item, array \$options, \$simple=false) + getTags()