```
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

       _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)
          CommonDBRelation
 + $no_form_page
 + static $itemtype_1
 + static $items_id_1
 + static $take_entity_1
 + static $checkItem_1
   _Rights
 + static $mustBeAttached_1
 + static $logs_for_item_1
 + static $log_history
    _1_add
 + static $log_history
    _1_update
 + static $log_history
    _1_delete
 + static $log_history
    _1_lock
   et 14 de plus...
 # $_force_log_option
 + getOnePeer($number)
 + getFromDBForItems
   (CommonDBTM $item1
    CommonDBTM $item2)
 + rawSearchOptions()
 + isAttach2Valid(array
    &$input)
 + isAttach1Valid(array
    &$input)
 + canRelationItem($method,
    $methodNotItem, $check
    entity=true, $forceCheckBoth=false)
 + canCreateItem()
 + canViewItem()
 + canUpdateItem()
 + canDeleteItem()
   et 13 de plus...
 + static getOpposite
   (CommonDBTM $item, &
   $relations_id=null)
 + static canRelation
   ($method, $forceCheckBoth
   =false)
 + static canCreate()
 + static canView()
 + static canUpdate()
 + static canDelete()
 + static canPurge()
 + static getHTMLTableHeader
   ($itemtype, HTMLTableBase
    $base, HTMLTableSuperHeader
$super=null, HTMLTableHeader
$father=null, array $options=[])

    static getRelationMassive

   ActionsSpecificities()
 + static showRelationMassive
   ActionsSubForm(MassiveAction
    $ma, $peer_number)
   et 6 de plus...
 # static getDistinctTypesParams
   ($items_id, $extra_where=[])
          PendingReason_Item
 + static $itemtype_1
 + static $items_id_1
 + static $take_entity_1
 + static $itemtype_2
 + static $items_id_2
 + static $take_entity_2
 + getNextFollowupDate()
 + getAutoResolvedate()
 + canCreateItem()
 + canViewItem()
 + canUpdateItem()
 + canDeleteItem()
 + canPurgeItem()
 + static getTypeName
   ($nb=0)
 + static getForItem(Common
   DBTM $item, bool $get
   _empty=false)
 + static createForItem
   (CommonDBTM $item, array
    $fields)

    static updateForItem

   (CommonDBTM $item, array
    $fields)
 + static deleteForItem
   (CommonDBTM $item)

    static isLastPendingForItem

   (CommonITILObject $item,
    CommonDBTM $timeline_item)
 + static canDisplayPending
   ReasonForItem(CommonDBTM
    $item)
 + static handlePendingReason
```

**UpdateFromNewTimelineItem** 

(CommonDBTM \$timeline\_item)

+ static handleTimelineEdits

+ static canCreate()+ static canView()+ static canUpdate()+ static canDelete()+ static canPurge()

(CommonDBTM \$new\_timeline\_item)