Api Documentation

SaltOS 4.0 r2006

April 2025

Contents

1 Actions

1.1 Add log action

`api/php/action/add.php`

This file implements the addlog action, requires a POST JSON request with an element in the json that contains the message to be added

- @msg => message that you want to add to the log file

Add error action

This file implements the adderror action, requires a POST JSON request with the follow elements: jserror, details and backtrace, this action is called from window.onerror in order to store the details of the js error

- @jserror => text used as title in the error report
- @details => text used as details in the error report
- @backtrace => array with the backtrace used in the error report

1.2 Application action

`api/php/action/app.php`

This file implements the app action, requires a GET REST request and the order of the elements are:

- @1 => the app that you want to execute
- @2 => the action that tou want to use, if the app only contains one action, this parameter is not necesary
- @3 => the id used in some actions, for example, to get the data of specific customer using the id

1.3 Authentication helper module

`api/php/action/auth.php`

This file contains all needed code to do authentications with all features suck as the main authentication using a user and password pair, the checktoken and the deauthtoken to control it.

1.4 Garbage Collector action

`api/php/action/cron.php`

This action executes the gc_exec function in the gc.php library, the execution of this accion only is allowed from the command line

1.5 Garbage Collector action

`api/php/action/gc.php`

This action executes the gc_exec function in the gc.php library, the execution of this accion only is allowed from the command line

1.6 BarCode action

`api/php/action/image.php`

This action allow to generate a barcode, you can pass the desired message that you want to convert in barcode

- @msg => the msg that you want to codify in the qrcode
- @format => the format used to the result, only can be png or json
- @w => width of each unit's bar of the barcode
- @h => height of the barcode (without margins and text footer)
- @m => margin of the barcode (white area that surround the barcode)
- @s => size of the footer text, not used if zero
- @t => type of the barcode, C128 is the most common type used

QRCode action

This action allow to generate a qrcode with the SaltOS logo embedded in the center of the image, you can pass the desired message that you want to convert in qrcode.

- @msg => the msg that you want to codify in the qrcode
- @format => the format used to the result, only can be png or json
- @s => size of each pixel used in the qrcode
- @m => margin of the qrcode (white area that that surround the qrcode)
- @l => level error correction: L (low), M (medium), Q (better), H (best)

Captcha action

This action allo to retrieve the captcha of a randomly number or math operation, used to prevent massive requests, can perform the action of create the captcha image and return the result as a simple image or as a json image

- @type => the type used to the result, only can be number or math
- @format => the format used to the result, only can be png or json
- @width => the width of the generated image
- @height => the height of the generated image
- @letter => the size of the letters of the generated image
- @number => the size of the numbers of the generated image
- @angle => the angle allowed to rotate the letters and numbers
- @color => the color user to paint the code
- @bgcolor => the background color of the image
- @fgcolor => the color used to paint the letters of the background of the image

- @period => parameter for the wave transformation
- @amplitude => parameter for the wave transformation
- @blur => true or false to enable or disable the blur effect

Score action

This action allo to retrieve the score of a password, intended to be used as helper previously to the authupdate call, can perform the action of compute the score and return the result as a simple image or as a json image

- @pass => the password that you want to compute the score
- @format => the format used to the result, only can be png or json
- @width => the width of the generated image
- @height => the height of the generated image
- @size => the size of the font of the generated image

1.7   Make indexing action

`api/php/action/indexing.php`

This fie contains useful functions related to the indexing action that internally uses the mroonga engine to search in the fulltext string generated by this action

1.8   Make indexing action

`api/php/action/integrity.php`

This fie contains useful functions related to the indexing action that internally uses the mroonga engine to search in the fulltext string generated by this action

1.9   Garbage Collector action

`api/php/action/push.php`

This action executes the gc_exec function in the gc.php library, the execution of this accion only is allowed from the command line

1.10   DB Schema action

`api/php/action/setup.php`

This action executes the db_schema and db_static functions in the dbschema.php library, the execution of this accion only is allowed from the command line

1.11   Add files action

`api/php/action/upload.php`

This file implements the delete files action, requires a POST JSON request with an array of files, and each array must contain the follow entries: id, name, size, type, data, error, file, hash

This action checks that not error is found, get the data and clear the data element of the array, check the prefix of the data using the type, check the size of the data, and then, set the file and hash to the array and store the file in the upload directory

This action checks that not error is found, checks the file element, the size of the file, the hash of the file, and then, remove the file and clear the file and hash element of the array

- @file => array that contains the follow elements:
- @id => unique id that is used by the client to identify the response
- @app => the hash, used to know the app from where the file is uploaded
- @name => the name of the file
- @size => the size of the file
- @type => the type of the file
- @data => the contents of the file encoded as inline base64
- @error => the error in case of errors
- @file => this field is used here to put the local filename used in the file
- @hash => this field contains the hash of the contents of the file

2 Autoload

2.1 Apps helper module

`api/php/autoload/apps.php`

This file contains functions intended to be used as hepers of other functions, allowing to convert between formats as the name of the app to and app id, or viceversa

2.1.1 Apps helper function

`function __apps($fn, $arg)`

This function is used by the XXX2YYY functions as helper, it stores the dictionary of all conversions and resolves the data using it

- @fn => the caller function
- @arg => the argument passed to the function

2.1.2 Subtables helper

`function __apps_subtables_helper($subtables)`

This function helps the follow apps to returns the subtables information into a structure instead of string, this is very helpfull by the code that consumes this information, allow to use it quickly without parsing the original string.

- @subtables => the string that contains the subtables specification

2.1.3 Id to App

```
function id2app($id)
```

This function resolves the code of the app from the app id

- @id => the app id used to resolve the code

2.1.4 App to Id

```
function app2id($app)
```

This function resolves the id of the app from the app code

- @app => the code used to resolve the id

2.1.5 Id to Table

```
function id2table($id)
```

This function resolves the table of the app from the app id

- @id => the app id used to resolve the table

2.1.6 App to Table

```
function app2table($app)
```

This function resolves the table of the app from the app code

- @app => the app code used to resolve the table

2.1.7 Table to Id

```
function table2id($table)
```

This function resolves the id of the app from the app table

- @table => the app table used to resolve the id

2.1.8 Table to App

```
function table2app($table)
```

This function resolves the code of the app from the app table

- @table => the app table used to resolve the app code

2.1.9   Id to Subtables

```
function id2subtables($id)
```

This function resolves the subtables of the app from the app id

- @id => the app id used to resolve the subtables

2.1.10   App to Subtables

```
function app2subtables($app)
```

This function resolves the subtables of the app from the app code

- @app => the app code used to resolve the subtables

2.1.11   Table to Subtables

```
function table2subtables($table)
```

This function resolves the subtables of the app from the app table

- @table => the app table used to resolve the subtables

2.1.12   App Exists

```
function app_exists($app)
```

This function detect if an app exists

- @app => the app that you want to check if exists

2.1.13   App to Index

```
function app2index($app)
```

This function returns the has_index of the app

- @app => the app code used to resolve the index

2.1.14   App to Control

```
function app2control($app)
```

This function returns the has_control of the app

- @app => the app code used to resolve the control

2.1.15   App to Version

```
function app2version($app)
```

This function returns the has_version of the app

- @app => the app code used to resolve the version

2.1.16   App to Files

```
function app2files($app)
```

This function returns the has_files of the app

- @app => the app code used to resolve the files

2.1.17   App to Notes

```
function app2notes($app)
```

This function returns the has_notes of the app

- @app => the app code used to resolve the notes

2.1.18   App to Log

```
function app2log($app)
```

This function returns the has_log of the app

- @app => the app code used to resolve the log

2.1.19   Subtable to Id

```
function subtable2id($subtable)
```

This function resolves the id of the app from the subtable

- @subtable => the app subtable used to resolve the id

2.1.20   Subtable to App

```
function subtable2app($subtable)
```

This function resolves the code of the app from the app subtable

- @subtable => the app subtable used to resolve the app code

2.1.21 Subtable to table

```
function subtable2table($subtable)
```

This function resolves the table of the app from the app subtable

- @subtable => the app subtable used to resolve the table

2.1.22 Table Exists

```
function table_exists($table)
```

This function detect if a table exists

- @table => the table that you want to check if exists

2.1.23 Subtable Exists

```
function subtable_exists($subtable)
```

This function detect if a subtable exists

- @subtable => the subtable that you want to check if exists

2.1.24 Id to Field

```
function id2field($id)
```

This function resolves the field of the app from the app id

- @id => the app id used to resolve the field

2.1.25 App to Field

```
function app2field($app)
```

This function resolves the field of the app from the app code

- @app => the app code used to resolve the field

2.1.26 Table to Field

```
function table2field($table)
```

This function resolves the field of the app from the app table

- @table => the app table used to resolve the field

2.1.27 Id to Name

```
function id2name($id)
```

This function resolves the name of the app from the app id

- @id => the app id used to resolve the name

2.1.28 App to Name

```
function app2name($app)
```

This function resolves the name of the app from the app code

- @app => the app code used to resolve the name

2.1.29 Table to Name

```
function table2name($table)
```

This function resolves the name of the app from the app table

- @table => the app table used to resolve the name

2.1.30 Detect apps files

```
function detect_apps_files($file)
```

This function returns the files found in the main path and in the apps path

- @file => the pattern used to search files

2.1.31 Current app

```
function current_app()
```

This function returns the current app as string

2.1.32 Detect app file

```
function detect_app_file($app)
```

This function returns the path found for the requested app

- @app => the app that wants to retrieve the file path

Notes:

If the app uses the yaml specification instead of the xml, this function call the make_app_file_helper function with the yaml file to get the xml file that must to be used in the app layer

2.1.33 Detect app folder

```
function detect_app_folder($app)
```

This function returns the path found for the requested app

- @app => the app that wants to retrieve the folder path

2.1.34 Make Apps File helper

```
function make_app_file_helper($yamlfile)
```

function call the make_app_file function defined in the required field of the yaml file to generate the equivalent xml file used as app file and returns the path to the cached file

- @yamlfile => the yaml file desired to be converted to xml

2.2 Array helper module

```
api/php/autoload/array.php
```

This file contain useful array functions, too we have added some features that historically appear in the import module, they allow to modify arrays using xpath notation, apply filters and patchs or simply, manipulate the data stored in an array of the form row/rows.

2.2.1 Null to array converter

```
function array_protected($x)
```

This function convert all nulls into an array, is intended to be used as helper for example in the glob output, to force to have an array in all cases

- @arr => the input, generally must to be an array, if a null is passed, then a void array will be returned

2.2.2 Join attr value

```
function join_attr_value($array)
```

This function allow to join the #attr and value to get only an associative array, it is intended to be used to simplify the specification of some elements and to simplify the usage of this elements in the client side

- @array => the input that can contains an array with #attr and value

2.2.3 Get Node helper

```
function __array_getnode($path, $array)
```

This function is a helper used to get a node in a xml structure

- @path => a path of the desired node
- @array => the array with nodes of the xml structure

Returns the contents of the node of the specified path

2.2.4   Get Value helper

```
function __array_getvalue($array)
```

This function is a helper used to get a value if exists of a node structure

- @array => an array

Retusn the value if exists, otherwise the same input

2.2.5   Get Attr helper

```
function __array_getattr($elem, $array)
```

This function is a helper used to get a attr element if exists of a node structure

- @elem => a string representing an element
- @array => an array containing the node

Returns the attr if exists, otherwise null

2.2.6   Add Node helper

```
function __array_addnode($path, &$array, $value)
```

This function is used to add data into a xml structure

- @path => the desired path where do you want to add the data
- @array => the array with the xml structure
- @value => the value that do you want to add

true if the function can add the data, false otherwise

2.2.7   Set Node helper

```
function __array_setnode($path, &$array, $value)
```

This function is used to set data into a xml structure

- @path => the desired path where do you want to put the data,
- @array => the array with the xml structure
- @value => the value that do you want to put

Returns true if the function can set the value, false otherwise

2.2.8   Del Node helper

```
function __array_delnode($path, &$array)
```

This function is used to remove data of the xml structure

- @path => the desired path where do you want to remove
- @array => the array with the xml structure

Returns true if the function can remove the path, false otherwise

2.2.9   Filter helper

```
function __array_filter($array, $filter, $eval = false)
```

This function tries to apply a filter to a tree array, too allow to use the evaluation system to allow to pass as filter an expression like this A=M23

- @array => the tree array that you want to apply the filter
- @filter => the filter to apply
- @eval => set to true if you want to enable the eval feature

2.2.10   Filter Recursive helper

```
function __array_filter_rec($node, $filter, $eval, $parent = [])
```

This function is a helper of the previous function and is able to to the same but with recursivity

- @node => the tree node that you want to filter
- @filter => the filter to apply
- @eval => set to 1 if you want to enable the eval feature
- @parent => this parameter is intended to be used internaly by the function

2.2.11   Apply Patch

```
function __array_apply_patch(&$array, $key, $val)
```

This function is able to apply a patch in the tree array, this allow to update the desired branch of the tree using a xpath notation

- @array => the array that you want to apply the patch
- @key => the xpath where you want to apply the patch
- @val => the val that you want to put in the desired xpath

2.2.12   Apply Patch Recursive helper

```
function __array_apply_patch_rec(&$array, $key, $val)
```

This function is a helper of the previous function and is able to to the same but with recursivity

- @array => the array that you want to apply the patch
- @key => the xpath where you want to apply the patch
- @val => the val that you want to put in the desired xpath

2.2.13 Is attr value

```
function is_attr_value($array)
```

This function return true if the data argument is an array with #attr and value

- @array => the array that wants to check

2.2.14 Arrays to array

```
function arrays2array()
```

This function internally implements the old xml_join feature that allow to merge multiple files into one using the fix_key of the keys in the first level as key to join.

$args => This function process dynamically all arguments

Notes:

- This function is derived from the xmlfiles2array that get all files and process the contents to join with a certain logics
- In a commit, this function become to be the main joining function and xmlfiles2array uses it to do the real job

2.2.15 Xpath search array

```
function xpath_search_array($xpath, $array)
```

This function is intended to do searches using the xpath notation like list[id=table]/actions, the main idea is that this function can returns an array with all occurrences because the same xpath can choose more that one result

- @xpath => the string containing the search xpath
- @array => the array that contains the search data

2.2.16 Xpath search first

```
function xpath_search_first($xpath, $array)
```

This function is intended to returns the first result of the array returned by the xpath_search_first function

- @xpath => the string containing the search xpath
- @array => the array that contains the search data

Notes:

In case of not occurrences, null is returned

2.2.17 Xpath search first

```
function xpath_search_first_value($xpath, $array)
```

This function is intended to returns the value of the first result obtained from the xpath_search_first function

- @xpath => the string containing the search xpath
- @array => the array that contains the search data

Notes:

In case of not occurrences, null is returned

2.2.18 Array transpose

```
function array_transpose($input)
```

This function returns a transposed array, intended to be used when needs an array of cols instead of rows, for example, ideal to be used in the excel widget

The expected output must to be the same input array but swaping the first and second indeces level, in other words, this function is able to convert arrays from A*B to B*A dimensions

- @input => the input array

2.2.19 Array lowercase

```
function array_lowercase($array)
```

This function do the same that strtolower but apply it to all array values

- @array => the array that you want to convert to lower case

2.2.20 Array key lowercase

```
function array_key_lowercase($array)
```

This function do the same that the array_lowercase but only apply to all keys of the array

- @array => the array that you want to convert to lower case

2.2.21 Array key search

```
function array_key_search($needed, $array)
```

This function allow to search keys in the array using a case insensitive search

- @needed => the desired key used in the case insensitive search
- @array => the array where do you want to found the key

Notes:

In case of not found the desired key, the original needed param is returned

2.2.22 Explode With Quotes

```
function explode_with_quotes($separator, $str, $limit = 0)
```

This function tries to do the same things that the original explode but add the quotes feature, don't break an string contained in a single or double quotes, this allow to implement features like a search that forces specific strings with spaces

- @separator => the delimiter character used in the explode feature
- @str => the string that you want to explode
- @limit => the number of elements that can contains the result

2.2.23 Grep helper

```
function array_grep($input, $pattern, $invert = false)
```

This function emulates the grep command, is able to invert the pattern selection and returns the same array with the grep applied, tries to do the grep ignoring case and ignoring extended chars and is able to search words ignoring accents

- @input => the input array
- @pattern => the search pattern
- @invert => default to false to search, true to invert the selection

2.2.24 Convert a multidimensional associative matrix into a purely indexed matrix.

```
function matrix_values($matrix)
```

This function recursively transforms a 2D associative array (matrix) into a fully numeric-indexed matrix by applying 'array_values()' at both levels: the rows and the matrix itself.

For example, it converts: ['row1' => 'a' => 1, 'b' =>, 'row2' => 'x' => 3, 'y' =>] Into: [1,, 3,]

- @matrix => A 2D associative array

Return a 2D numeric-indexed array

2.2.25 Check Real Matrix helper

```
function check_real_matrix($array)
```

This function checks that the argument is a matrix, to do this, checks that the argument is an array, that all keys are numeric and that all entries of the main array is another array, and for each another array, checks that the keys are numeric and that all values are non arrays

- @array => the array to check

2.3 Compatibility helper module

```
api/php/autoload/compat.php
```

This file add some functions used by SaltOS that can not be found in all allowed versions of PHP

2.3.1 Array Key Last

```
function array_key_last(array $array)
```

This function appear in PHP 7.3, and for previous version SaltOS uses this code

- @array => the array where you want to obtain the last key

Notes:

Code copied from the follow web: https://www.php.net/manual/es/function.array-key-last.php#124007

2.3.2 Array Key First

```
function array_key_first(array $arr)
```

This function appear in PHP 7.3, and for previous version SaltOS uses this code

- @array => the array where you want to obtain the first key

Notes:

Code copied from the follow web: https://www.php.net/manual/es/function.array-key-last.php#124007

2.4 Config helper module

```
api/php/autoload/config.php
```

This fie contains useful functions related to configuration features

2.4.1 Get config

```
function get_config($key, $user_id = -1)
```

This function is intended to be used to retrieve values from the config system, as first level, the function try to get the value from the tbl_config, and if it is not found, then the function try to get the values from the config file.

- @key => the key that you want to retrieve the value
- @user_id => the user_id used in the first search step

Notes:

This function is a new release of the olds getConfig and getDefault, depending of the user_id argument, it tries to search in the config file or in the database, for negative values the function uses the config file and for zero or positive values, tries to search it in the database

To prevent errors in case of duplicates, we are using the ORDER BY id ASC LIMIT 1, this allow to get and set only the first register found by the select query

2.4.2 Set config

```
function set_config($key, $val, $user_id = -1)
```

This function sets a value to a config key, the data will be stored in the database using the tbl_config for zero or positive values of user_id, and in the memory of the config file for negative user_id values

- @key => the key that you want to set

- @val => the value that you want to set

- @user_id => the user_id used as filter

Notes:

If null val is passed as argument, then the entry of the config or database is removed, the main idea is to use the same method used by the setcookie that allow to remove entries by setting the value to null

To prevent errors in case of duplicates, we are using the ORDER BY id ASC LIMIT 1, this allow to get and set only the first register found by the select query

2.4.3 Detect config files

```
function detect_config_files($file)
```

This function returns the files found in the main path, in the apps path and in the files path

- @file => the pattern used to search files

2.4.4 Get config array

```
function get_config_array($prefix, $user_id)
```

This function is intended to retrieve the configuration associated to an user using a prefix for all keys.

- @prefix => the prefix used in the keys of the config
- @user_id => the user_id used in the search query

2.4.5 Prepare Config Files

```
function prepare_config_files($array)
```

This function tries to join all config files into one unique structure, to do it joins nodes with the same key, for example, you can use the data/files/config.xml file to set the specific database configuration and overwrite the db/pdo_mysql or the db/type, the main idea is to replace the nodes of the second level because the arrays2array function join previously the differents files into one structure usin the first level to do the join, and only adds the repeated overload without take decisions about overwrite the contents like this function do

- @array => the main array that you want to prepare as config array

2.5 Database helper module

```
api/php/autoload/database.php
```

This fie contains useful functions related to database, allow to connect, check queries, execute queries, disconnect, retrieve rows and manipulate resultsets of the database

2.5.1 DB Connect

```
function db_connect($args = null)
```

This function is intended to stablish the connection to the database

- @args => is an array with key val pairs
- @host => the host for the connection
- @port => the port used for the connection
- @name => name of the database for the connection
- @user => user used to stablish the connection
- @pass => pass used to stablish the connection
- @file => the file that contains the database

Notes:

The parameters can be different depending of each database driver, in general the arguments can be the host, port, name, user and pass for database's servers and only file for the sqlite database

If the args argument is null, the the function try to use the configuration stored in the config file and depending of the argument presense, it will return the database object or store it in the config to be used by the nexts functions of this php file

2.5.2 DB Check

```
function db_check($query, $params = [])
```

This function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

2.5.3 DB Escape

```
function db_escape($str)
```

This function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

2.5.4 DB Query

```
function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute
- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

2.5.5 DB Fetch Row

```
function db_fetch_row(&$result)
```

This function returns the next row of the resultset queue

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.6 DB Fetch All

```
function db_fetch_all(&$result)
```

This function returns all rows of the resultset queue

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.7 DB Num Rows

```
function db_num_rows($result)
```

This function returns the total number of the results in the resultset queue

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.8 DB Num Fields

```
function db_num_fields($result)
```

This function returns the number of fields of the resultset queue

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.9 DB Field Name

```
function db_field_name($result, $index)
```

This function returns the name of the field identified by the index field

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.10 DB Last Insert ID

```
function db_last_insert_id()
```

This function returns the last insert id

2.5.11 DB Free

```
function db_free(&$result)
```

This function releases all memory used by the resultset queue

- @result => this argument is passed by reference and contains the resultset queue obtained by the db_query

2.5.12 DB Disconnect

```
function db_disconnect()
```

This function close the database connection and sets the link to null

2.6 Datetime helper module

```
api/php/autoload/datetime.php
```

This fie contains useful functions related to datetime, allow to get current timestamp in various formats, allow to evaluate timestamps and some helper more

2.6.1 Current Date

```
function current_date($offset = 0)
```

This function returns the current date in a YYYY-MM-DD format, this is used by a lot of functions in SaltOS, allow to specify a bias used to move the current time mark

- @offset => the bias added to the current time mark

2.6.2 Current Time

```
function current_time($offset = 0)
```

This function returns the current time in a HH:II:SS format, this is used by a lot of functions in SaltOS, allow to specify a bias used to move the current time mark

- @offset => the bias added to the current time mark

2.6.3 Current DateTime

```
function current_datetime($offset = 0)
```

This function returns the current date and time in a YYYY-MM-SS HH:II:SS format, this is used by a lot of functions in SaltOS, allow to specify a bias used to move the current time mark

- @offset => the bias added to the current time mark

2.6.4 Current Decimals

```
function current_decimals($offset = 0, $size = 4)
```

This function returns the current decimals to be added to the seconds as a decimal part, this function uses the microtime function to get this level of precision that can not be obtained using the original date and time functions

- @offset => the bias added to the current time mark
- @size => the size of the returned decimal part

Notes:

This function is used by current_datetime_decimals, and don't have more uses that provice more precision in the logs files

2.6.5 Current DateTime Decimals

```
function current_datetime_decimals($offset = 0, $size = 4)
```

This function returns the current date and time with decimals in the seconds in a YYYY-MM-DD HH:II:SS.XXXX format, useful when do you want to log information more accuracy to debug issues, for example

- @offset => the bias added to the current time mark
- @size => the size used by the decimal part

2.6.6 Dateval

```
function dateval($value)
```

This function try to do the same thing that intval or strval, but for date values, to do this, this function try to separate all elements and identify the year position and the other elements, the result will be of the format YYYY-MM-DD

- @value => the input value to validate

Notes:

This function try to cast the year, month and day from 0000-00-00 to valid values, this is because the databases accepts the 0000-00-00 date and is used as emulated null, the month are limited to 12 and the day is limited to the days of the month and year, this is useful because the dates that are more greather that zero, will have a valid and an existing value

2.6.7   Day of a Month helper

```
function __days_of_a_month($year, $month)
```

This function is a helper used by other date and datetime functions, this is useful because allow to fix problems in dates that use days out of range

- @year => year that you want to use in the validation
- @month => month that you want to use in the validation

2.6.8   Timeval

```
function timeval($value)
```

This function try to do the same thing that intval or strval, but for time values, to do this, this function try to separate all elements and identify the elements, the result will be of the format HH:II:SS

- @value => the input value to validate

2.6.9   Datetimeval

```
function datetimeval($value)
```

This function try to do the same thing that intval or strval, but for datetime values, to do this, this function try to separate all elements and identify the year position and the other elements, the result will be of the format YYYY-MM-DD HH:II:SS

- @value => the input value to validate

Notes:

This function try to cast the year, month and day from 0000-00-00 to valid values, this is because the databases accepts the 0000-00-00 date and is used as emulated null, the month are limited to 12 and the day is limited to the days of the month and year, this is useful because the dates that are more greather that zero, will have a valid and an existing value

2.6.10   Time to Seconds

```
function __time2secs($time)
```

This function converts the time format into seconds

- @time => time to be converted into seconds, the format will be HH:II:SS

2.6.11 Seconds to Time

```
function __secs2time($secs)
```

This function converts the seconds into time format

- @secs => seconds to be converted to time format, the format will be a number

2.6.12 Current Day Of Week

```
function current_dow($offset = 0)
```

This function returns the current day of week as integer between 1 and 7 range, this is used by some functions in SaltOS, allow to specify a bias used to move the current time mark

- @offset => the bias added to the current time mark

2.6.13 Datetime format

```
function datetime_format($x)
```

This function returns the datetime using the europe format for d/m/Y instead of Y-m-d

- @x => the datetime string that you want to convert

2.7 Error helper module

```
api/php/autoload/error.php
```

This fie contains useful functions related to error management, allow to trigger and manage errors, too contains the code used for the error and exception handlers

2.7.1 Show PHP Error

```
function show_php_error($array)
```

This function allow to SaltOS to trigger the errors in a multiple levels:

1) Check if the error is caused by a memory allocation error, and in this case, try to setup more memory to continue executing the error function, sometimes if the system is using all memory, this function can not do all tasks and don't know whats be happening

2) Add some extra traces if they are not found in the input array

3) Create a human readable message in text and json format, the text will be used to log the error using a regular file and the json will be used as stdout response

4) If the error is caused by a deprecation, the error will be logged in the log file but the execution of the code will continue (if it can continue!!!)

5) Try to categorize the error and log the text in the specific log file, this part is optimized to prevent the addition of repeated errors using a hash as a trick

6) Send a json to the stdout using the output handler.

THs input @array can contain pairs of key val:

- @dberror => The text used in the DB Error section
- @phperror => The text used in the PHP Error section
- @xmlerror => The text used in the XML Error section
- @jserror => The text used in the JS Error section
- @dbwarning => The text used in the DB Warning section
- @phpwarning => The text used in the PHP Warning section
- @xmlwarning => The text used in the XML Warning section
- @jswarning => The text used in the JS Warning section
- @source => The text used in the Source section
- @exception => The text used in the Exception section
- @details => The text used in the Details section
- @query => The text used in the Query section
- @backtrace => The text used in the Backtrace section
- @debug => The text used in the Debug section

Notes:

The unset for the pid and the time keys of the debug array is justificate because each execution modify the pid and the time entries and break the optimization of the hash with the checklog to prevent repetitions in the log file

2.7.2 Do Message Error

```
function do_message_error($array)
```

This function acts as a helper of the show_php_error, is intended to build the text and the json messages used to the log file and for the stdout channel

THs input @array can contain pairs of key val:

- @dberror => The text used in the DB Error section
- @phperror => The text used in the PHP Error section
- @xmlerror => The text used in the XML Error section
- @jserror => The text used in the JS Error section
- @dbwarning => The text used in the DB Warning section
- @phpwarning => The text used in the PHP Warning section
- @xmlwarning => The text used in the XML Warning section
- @jswarning => The text used in the JS Warning section
- @source => The text used in the Source section
- @exception => The text used in the Exception section
- @details => The text used in the Details section

- @query => The text used in the Query section
- @backtrace => The text used in the Backtrace section
- @debug => The text used in the Debug section

Returns an array with the text and the json formated output ready to be used in the log file and in the stdout channel

2.7.3 Program Handlers

```
function program_handlers()
```

This function program all error handlers

2.7.4 Error Handler

```
function __error_handler($type, $message, $file, $line)
```

This function is the callback function used by the set_error_handler

Ths arguments are defined by the set_error_handler:

- @type => The code of the error
- @message => The descriptive message of the error
- @file => The filename of the file that trigger the error
- @line => The line where the error will occurred

2.7.5 Exception Handler

```
function __exception_handler($e)
```

This function is the callback function used by the set_exception_handler

Ths arguments are defined by the set_exception_handler:

- @e => object that contains the getMessage, getCode, getFile, getLine and getTrace methods

2.7.6 Shutdown Handler

```
function __shutdown_handler()
```

This function is the callback function used by the register_shutdown_function, try to detect if an error is the cause of the shutdown of the script, note that a correct execution will execute this function and only it must to trigger an error if a real error is in the stack of the errors events, to do it this function uses the error_get_last to check if the value in in the list of typified errors

2.7.7 Get code from trace

```
function __get_code_from_trace()
```

This function acts as helper of the show_json_error, and try to get the filename and the line where the error will be triggered, for example, an error triggered from the index.php at line 23 will generate a code index:23, this information will be useful for our technical service to help the users when live issues with our API

- @trace => the array returned by the debug_backtrace function
- @index => the position of the array used to get the filename and the line

Notes:

This function gets the arguments dinamically, this allow to send for example the array with the trace, the desired index or both datas in your prefered order, the trick to detect each param is to expect an array for the trace and a number for the index.

2.7.8 Show JSON Error

```
function show_json_error($msg, $logout = false)
```

This function is triggered from the code in a controlate errors, the idea is to have a simple way to send controled errors to the user using a json output channel, and to do it, we have this function that can be called with a simple message and the code is created automatically to help the backtrace of the issues

- @msg => this contains a simple text that is used in the json output
- @logout => this allow to send the logout flag to force to show the login screen

2.7.9 Get code from file and line

```
function __get_code_from_file_and_line($file, $line)
```

This function returns the string that contains the PATHINFO_FILENAME and the line to idenfify the launcher of an error, for example

- @file => filename used to obtain the first part of the code
- @line => line used to construct the last part of the code

2.7.10 Detect Recursion

```
function detect_recursion($fn)
```

This function allow to SaltOS to detect the recursión, to do it, uses the debug_backtrace function that returns all information about the execution of the current function, the main idea of this function is to detect in what lines of the backtrace appear the file or the function, and returns the count of times that appear

- @fn => the name of the function or file, can be multiples functions or files separated by a comma

2.7.11 Overload error handler

```
function overload_error_handler($words)
```

This function allow to add a new error handler that is able to ignore certain errors that contains the words used as argument, this is usefull in cases where php trigger an uncontrolable error that not are possible to control using other techniques like try/catch or using the documented specification

- @words => the list of words that must to contain the error desired to be ignored

2.8 Execution helper module

```
api/php/autoload/exec.php
```

This fie contains useful functions related to execution of external programs, allow to execute, check commands and manage some features as the cache usage or the timeout used in each execution

2.8.1 OB Passthru

```
function ob_passthru($cmd, $expires = 0)
```

This function is a join of ob_start and passthru, the main idea is to execute a program using the command line and get the output (only stdout and not stderr) and return the data

- @cmd => the command line that you want to execute
- @expires => the expires time used to compute if the cache is valid

This funtion tries to execute the command using some of the provided methods, passthru, system, exec and shell_exec, another feature is that the function detects what command are available to use

2.8.2 Check Commands

```
function check_commands($commands, $expires = -1)
```

This function tries to validate if the commands are available in the system, to do it, uses the unix command witch

- @commands => the commands that you want to check if are they available
- @expires => the expires time used to compute if the cache is valid

2.8.3 Is Disabled Function

```
function is_disabled_function($fn)
```

This function check if the argument contains a disabled function, this feature uses the variables disable_functions and suhosin.executor.func.blacklist to get the list of all disabled functions

- @fn => the function that you want to check if is it disabled

Notes:

As an extra feature, this function can receive two arguments to add and del functions to the static $array, this is usefull for utest to check the correctness of the function

2.8.4 Exec Timeout

```
function __exec_timeout($cmd)
```

This helper function allow to execute commands using the external command timeout, this unix command allow to define the timeout for an execution of other command, and when the timeout is reached, then break the execution killing the process

- @cmd => the command that you want to execute with a timeout control

Returns the string that contains the command with ths timeout control

2.9 File utils helper module

```
api/php/autoload/file.php
```

This fie contains useful functions related to the file usage, allow to manage directories, files, caches, paths, permissions, remote files and more

2.9.1 Get Directory

```
function get_directory($key)
```

This function returns the directory configured to the key requested, too can define a default value, useful when the configuration still not loaded and SaltOS need some directory to do something as store data in the log file, for example

- @key => the key used in get_config to request the configured directory

2.9.2 Get Temp File

```
function get_temp_file($ext = '')
```

This function is intended to get a unique temporary file, used for temporary purposes as put contents to be used as input in a command

- @ext => the extension of the temporary file, useful for some commands that try to detect the contents using the extension

Notes:

This function uses the dirs/tempdir config key

2.9.3 Get Cache File

```
function get_cache_file($data, $ext = '')
```

This function is intended to get a cache filename, used for caching purposes

- @data => data used to compute the hash used by the cache, can be an string or an array with lot of contents
- @ext => extension of the cache filename

Notes:

This function uses the dirs/cachedir config key

2.9.4 Cache Exists

```
function cache_exists($cache, $files)
```

This function check the existence of valid cache by comparing the timestamp of the filemtime between the cache file and all files of the second argument

- @cache => cache filename
- @files => array of files that are considered as dependencies of the cache

2.9.5 URL Get Contents

```
function url_get_contents($url)
```

This file is an equivalent of the file_get_contents but intended to be used for request remote files using protocols as http or https

- @url => the url that you want to retrieve

Notes:

This function only returns the body of the request, if you are interested to get the headers of the request, try to use the __url_get_contents

2.9.6 URL Get Contents helper

```
function __url_get_contents($url, $args = [])
```

This file is an equivalent of the file_get_contents but intended to be used for request remote files using protocols as http or https

- @url => the url that you want to retrieve
- @args => Array of arguments, explained in the follow lines
- @cookies => an array with the cookies to be restored before send the request
- @_method => method used in the request
- @values => an array with the post values, useful when you want to send a POST request with pairs of variables and values
- @headers => an array with the headers to be send in the request
- @body => the full body used of the request, useful when you want to send a json file in the body instead of pairs of keys vals

This function returns an array with four elements, body, headers, cookies and code

2.9.7 Extension

```
function extension($file)
```

This function returns the PATHINFO_EXTENSION of the file

- @file => file used in the pathinfo call

2.9.8 Encode Bar Chars File

```
function encode_bad_chars_file($file)
```

This function is equivalent to encode_bad_chars but intended to be used with filenames, in this case, the extension and the rest of the filename will be encoded separately and the return value will contain the dot separating the filename with the extension

- @file => filename used in the encode process

2.9.9 Realpath Protected

```
function realpath_protected($path)
```

This function returns the realpath of the path, this version of the function allow to return the path of an unexistent file, this is useful when do you want to get the realpath of a unexistent file, for example, to the output of a command that must to generate the file but at the moment of the execution of this function the file is not found

- @path => path used in the realpath call

2.9.10 Getcwd Protected

```
function getcwd_protected()
```

This function returns the same result that the getcwd function but checking that the result is not an slash, this is an issue in some cases caused by permissions problems, and a good solution for this cases is to get the directory of the script as current work directory

2.9.11 Glob Protected

```
function glob_protected($pattern)
```

This function returns the same result that the glob function but checking that the result is an array, if glob fails or not get a files by the pattern, can return other values that an array, and this can cause problems if you are expecting an array to iterate in each element, this function prevent this problem

- @pattern => pattern used in the glob command

2.9.12 Chmod Protected

```
function chmod_protected($file, $mode)
```

This function tries to change the mode of the file using the chmod function only if the fileperms of the file are different that the requested mode and the fileowner of the file is the same user that is executing the script

- @file => file used by the chmod function
- @mode => mode used by the chmod function

2.9.13 Fsockopen Protected

```
function fsockopen_protected($hostname, $port, &$errno = 0, &$errstr = '',
    $timeout = null)
```

This function is only used by the httpclient library to avois problems with the certificates validations

Ths arguments is the same that the fsockopen function, in this case, the function uses the stream_socket_client to emulate the original fsockopen

2.9.14 File with hash

```
function file_with_hash($file)
```

This function returns the name of the file adding as argument the hash of the file for the http/https requests, this allow to helps the browser to know when the file has changed

- @file => the file that you want to add the hash querystring argument

2.9.15 File with min

```
function file_with_min($file)
```

This function returns the name of the file adding the .min. between the filename and the extension of the file if the .min. file exists

- @file => the file that you want to add the .min. part if exists

2.9.16 File Get Contents Protected

```
function file_get_contents_protected(...$args)
```

This function call the original file_get_contents and returns the buffer returned by the original function, the main idea of this function is to protect the caller to prevent I/O errors like "nohup php index.php" when

- @args => the original arguments are passed to the file_get_contents

2.9.17 Get human size

```
function get_human_size($size, $pre = '', $post = '')
```

This function returns an string containing the size in human format

- @size => the number of bytes to convert to human format
- @pre => string added between the number and the unit letter post => string added after the unit letter at the end

2.10 Get data helper module

```
api/php/autoload/getdata.php
```

This fie contains useful functions related to the $_DATA global variable, allow to get and set values in the global $_DATA variable using xpath as key

2.10.1 Get data

```
function get_data($key)
```

This function is intended to be used to retrieve values from the data system

- @key => the key that you want to retrieve the value

Notes:

If you request the key rest/-1, then the function returns the last value of the rest, intended to get the last value of the rest array, too you can get the rest array using negative indexes beginning from the last position

2.10.2 Set data

```
function set_data($key, $val)
```

This function sets a value in the data system for the specified key

- @key => the key that you want to set
- @val => the value that you want to set

Notes:

If null val is passed as argument, then the entry of the data is removed, the main idea is to use the same method used by the setcookie that allow to remove entries by setting the value to null

2.11 Gettext helper module

```
api/php/autoload/gettext.php
```

This fie contains useful functions related to gettext funcionality, allow to manage the SaltOS translations using a merged system of the unix locales and the old SaltOS translations system.

2.11.1 Get Text function

```
function T($text = '', $app = '')
```

This function replaces the gettext abreviation _() using the SaltOS gettext feature, is based in the original system of the old SaltOS with improvements to do more open as the GNU gettext

- @text => The text that you want to translate
- @app => Optional, to force gettext to use an specific app instead of current_app

Notes:

This function uses multiples locales at same time, SaltOS provides a basic set of usefull strings and each application can add and overwrite more strings, this is the same feature that old SaltOS provides

If you call the function without argument, the function returns the gettext dictionary intended to populate the clients gettext module and contains the app, the lang and the locales for the app and lang.

2.11.2 Check lang format

```
function check_lang_format($lang)
```

This function checks the correctness of the lang and returns a valid string that can be used safely as lang in other sites

- @lang => the lang that you want to process

2.11.3 Current lang

```
function current_lang()
```

This function returns the current lang

2.12 Iniset helper module

```
api/php/autoload/iniset.php
```

This fie contains useful functions related to the evaluation of the iniset, puntenv and extra directives configures in the config.xml file

2.12.1 Eval Iniset

```
function eval_iniset($array)
```

This function evaluates the iniset section of the config file, is intended to execute all ini_set commands detecting the current values and determining if is needed to change or not the current setting, is able to understand boolean values as On/Off, and too is able to set keys as mbstring.internal_encoding or mbstring.detect_order that must to be set by using another mb_* functions

- @array => the array with the pairs of keys vals

2.12.2 Eval Putenv

```
function eval_putenv($array)
```

This function evaluates the putenv section of the config file, is intended to execute all putenv commands detecting the current values and determining if is needed to change or not the current setting

- @array => the array with the pairs of keys vals

2.12.3 According to the documentation, putenv must return false in error cases, but

```
if ($key == '' || putenv("$key=$val") === false) {
```

putenv(): Argument #1 ($assignment) must have a valid syntax (code 0)

For this reason, the show_php_error placed after the putenv that must to be executed when putenv returns false never can be executed.

As trick, I have added the void key condition to force a case that executes the show_php_error

2.12.4 Eval Extras

```
function eval_extras($array)
```

This function evaluates the extra init requirements, intended for the multibyte functions and for the gettext initialization process

- @array => the array with the pairs of keys vals

2.13 Json helper module

```
api/php/autoload/json.php
```

This fie contains useful functions related to colors

2.13.1 Terminal colors

```
define('__COLORS_MAP__', [
```

This define sets the colors array used in the next functions

2.13.2 Json colorize

```
function json_colorize($json)
```

This funcion is able to colorize a json fragment to dump into a tty terminal

- @json => the json code that you want to colorize

Notes:

This function uses a trick to convert numbers in scientific notation to an old decimal style, to do it, detects numbers with the e letter and print using the places, ensuring precision up to the typical limit of a double type in C, which supports approximately 15-17 significant digits

2.14 Log helper module

```
api/php/autoload/log.php
```

This fie contains useful functions related to the logs files, allowing to add and check contents to the logs file, useful for debug purposes

2.14.1 Check Log

```
function checklog($hash, $file)
```

This function is a helper for the show_php_error, allow to detect repetitions of the same text in the log file to prevent to add repeated lines, the usage is very simple, only requires a hash and a file to check that the hash is not found in the contents of the file, you can think in this function as a grep replacement that is able to found the hash in the file

- @hash => the pattern that you want to search in the file
- @file => the file where search the pattern

2.14.2 Add Log

```
function addlog($msg, $file = '')
```

This function add messages to the specified log file

- @msg => message that you want to add to the log file
- @file => the log file that you want to use without directory

Notes:

If not file is specified, the debug/logfile (saltos.log) is used by default

The logs files are stored in the logsdir (/data/logs)

This function performs the log rotation is the maxfilesize is reached

2.14.3 Add Trace

```
function addtrace($array, $file)
```

This function performs the addlog to the file using as input the array, the main idea is to pass the same array that the used in the show_php_error, the difference is that addtrace, only add the backtrace and debug to the array and then, saves the log to the specified file

- @array => the array that can contains the same info that show_php_error
- @file => the file where do you want to store the log contents

2.14.4 Get Trace

```
function gettrace(...$args)
```

This function get an array as show_php_error, add the backtrace and debug information and convert all array into a string

- @array => the array that can contains the same info that show_php_error
- @full => add the full debug backtrace instead of the generic partial

This differentiation is because the full debug backtrace can break the checklog feature because the time and pid can be different in each execution

2.14.5 Session Backtrace

```
function session_backtrace()
```

Returns a string with the pid, sessid and current datetime with decimals

Notes:

The fields of this array allow to do low level debug processes, this data is generally used by the semaphores and some forced addtrace calls, but causes problems in the error reporting because break the hash and checklog optimization

2.15 Memory helper module

```
api/php/autoload/memory.php
```

This fie contains useful functions related to memory and time usage, allow to control the usage of time and/or memory of a process, intended to prevent crashes in processes that requires more time or memory resources that the defined by the system limits

2.15.1 Memory Get Free

```
function memory_get_free($bytes = false)
```

This function returns the free memory in bytes or the percentage of the memory_limit

- @bytes => if true, returns the free bytes, if false, returns the percentage

2.15.2 Get Time Usage

```
function time_get_usage($secs = false)
```

This function returns the time usage in seconds or in percentage of the max_execution_time

- @secs => if true, returns the used seconds, if false, returns the percentage

2.15.3 Get Free Time

```
function time_get_free($secs = false)
```

This function returns the free time in seconds or in percentage of the max_execution_time

- @secs => if true, returns the used seconds, if false, returns the percentage

2.15.4 Init Time Get

```
function init_timer()
```

This function call the helper to initialize the static ini to the current microtime

2.15.5 Get Time helper

```
function __time_get_helper($fn, $secs)
```

This function is a helper of the time_get_usage and time_get_free functions, is used to check the time usage and the free time that remain to finish the execution of the script

2.15.6 Set Max Memory Limit

```
function set_max_memory_limit()
```

This function is intended to do a ini_set with a more greather value to allow an exceptionally amount of memory usage

2.15.7 Set Max Execution Time

```
function set_max_execution_time()
```

This function is intended to do a ini_set with a more greather value to allow an exceptionally amount of execution time

2.16 Mime helper module

```
api/php/autoload/mime.php
```

This file contains the mimetype feature provided by saltos using different techniques suck as the extension file, using the mime_content_type or the finfo_file functions.

2.16.1 SaltOS Content Type

```
function saltos_content_type($file)
```

This function is intended to returns the mime content-type string using different techniques.

- @file => the file of which you want to know the content-type

2.16.2 SaltOS Content Type first helper

```
function saltos_content_type0($mime)
```

This function returns the first part of the content-type, for example, if you pass the string image/jpeg, this function will returns the string image.

- @mime => the mime that you want to process

2.16.3 SaltOS Content Type second helper

```
function saltos_content_type1($mime)
```

This function returns the second part of the content-type, for example, if you pass the string image/jpeg, this function will returns the string jpeg.

- @mime => the mime that you want to process

2.16.4 Mime inline

```
function mime_inline($type, $data)
```

This function returns the inline mime fragment of string that contains the mime and the encoded in base64 data, intended to embed it in img tags, for example.

- @type => the mime type (image/png for example)
- @data => the contents of the data that must to be encoded in base64

2.16.5 SaltOS Content Type from string

```
function saltos_content_type_from_string($buffer)
```

This function tries to return content-type of the buffer contents

- @buffer => the data that contains the image or other thing that you want to know the content-type.

2.16.6 Mime extract

```
function mime_extract($data)
```

This function tries to do the invert action that mime_inline, the main idea is to return the type and data used in the inline image, for example

- @data => an inline image in the format of data:image/png;base64,xxxxx

2.16.7 Mime to name

```
function mime2name($type)
```

This function returns a valid filename for the type used in the argument

- @type => the content-type that you want to use to get the filename

Notes:

- This function contains some special cases that contains special chars

2.17 Output helper module

```
api/php/autoload/output.php
```

This fie contains useful functions related to the output of the SaltOS, allow to send contents to the clients using the specified format and configuration, useful to return contents, too implement a specific output for the json format that is the most format used by the new SaltOS

2.17.1 Output Handler

```
function output_handler($array)
```

This function is intended to send data to the output channel, and can have the follow arguments:

- @array => array with the follow pairs of key val
- @file => file that contains the contents that you want to send
- @data => contents that you want to send to the output channel
- @type => content type header used
- @cache => boolean to enable the cache usage, includes the etag algorithm
- @name => the filename used in the content disposition attachment header
- @extra => headers that you can add to the transfer

2.17.2 Output header helper

```
function __output_header($header, $replace = true)
```

This function is a filter to ignore the headers when CLI SAPI is detected

- @header => The header that do you want to send to the client
- @replace => The boolean used as replace in the original function, true by default

2.17.3 Output Handler JSON

```
function output_handler_json($array)
```

This function allow to quickly send json output, the unique argument that it requires is the data that you want to send

- @array => content to convert to json and send to the output channel

Notes:

This function is able to generate a pretty output when stdout is connected to a terminal, intended to be used by humans, in other cases, the output will be minified.

2.18 PCOV helper module

```
api/php/autoload/pcov.php
```

This fie contains useful functions related to the pcov module used to measure the coverage of the unit tests

2.18.1 PCOV start

```
function pcov_start()
```

This function start the pcov recording to allow the measuring of the coverage in the unit tests.

2.18.2 PCOV stop

```
function pcov_stop()
```

This function stop the pcov recording that allow the measuring of the coverage in the unit tests and puts in the output file the collected data in a serialized format

2.19 Permissions helper module

```
api/php/autoload/perms.php
```

This fie contains useful functions related to permissions, allow to apply permissions in php core or in sql queries, to do it, uses all permissions tables and predefined configurations, more info in each function

2.19.1 Check User

```
function check_user($app, $perm)
```

This function checks the permissions using the tables apps_perms, users_apps_perms and groups_apps_perms, to do it, this function uses the user_id and groups_id (note that groups_id contains all groups where the user is associated), and try to check that the permissions permutations exists in the apps_perms, if some permission is found in the users and groups tables and it is not found in the apps_perms, an integrity error is launched.

- @app => the app to check
- @perm => the perm to check

2.19.2 Check SQL

```
function check_sql($app, $perm)
```

This function returns the fragment of SQL intended to filter by app and perm for the current user

- @app => the app to check
- @perm => the perm to check

Notes:

This function returns the portion of sql used to check permissions associated to an user with a specific permission and to an specific register, as an optimization, it detects if the all owner is on and return a true expression to improve the performance

2.19.3 Perms helper function

```
function __perms($fn, $arg)
```

This function is used by the XXX2YYY functions as helper, it stores the dictionary of all conversions and resolves the data using it

- @fn => the caller function
- @arg => the argument passed to the function

2.19.4 Id to Perm

```
function id2perm($id)
```

This function resolves the code of the perm from the perm id

- @id => the id used to resolve the perm

2.19.5 Perm to Id

```
function perm2id($perm)
```

This function resolves the id of the perm from the perm code

- @perm => the perm code used to resolve the id

Notes:

This function can return an integer or an array of integers, depending if the app is using the owner parameter or not

2.19.6 Perm Exists

```
function perm_exists($perm)
```

This function detect if a perm exists

- @perm => the perm that you want to check if exists

Notes:

This function returns true if a perm exists, and in case of the usage of the owner parameter, the function will return true for a perm that contains the owner and for the perm without the owner, for exampe, this function returns true for perm list and form perm list|user

2.19.7 Check App Perm Id

```
function check_app_perm_id($app, $perm, $id = null)
```

This function returns true if the app, the perm and the id accomplishes the expected level of permissions, it is intended to be used before the execution of each action, to guarantee the security

- @app => the app to check
- @perm => the perm to check
- @id => the id to check, if needed, you can omit in the create case

2.19.8 User is admin

```
function __user_is_admin($app)
```

This function returns true if the current user has all perms for the app

2.19.9 Merge data actions

```
function merge_data_actions($data, $actions)
```

This function merge the rows of a table or list with the specified actions

- @data => the data of the table or list widget
- @actions => the desired actions to use in the table or list widget

2.19.10 App has perm

```
function __app_has_perm($app, $perm)
```

This function checks that the app has the requested perm, used by the merge_data_actions to validate the existence of a permission in the application.

- @app => the app to check
- @perm => the perm to check

2.20 Random helper module

```
api/php/autoload/random.php
```

This fie contains useful functions related to random number generator, currently only initialize the internal generator, but in the future we can add more features if it is needed

2.20.1 Init Random

```
function init_random()
```

This function initialize the random number generator

Notes:

This function previously sets the seed using the microtime, but reading the srand php documentation, I see that the seed is not needed because if it is not provided, a randomly seed is used by default

2.21 Semaphore helper module

```
api/php/autoload/semaphores.php
```

This fie contains useful functions related to semaphores, allow to create and use semaphores for acquiring and release semaphores that guaranty the exclusivity of the code execution

2.21.1 Semaphore Acquire

```
function semaphore_acquire($name = '', $timeout = INF)
```

This function implement the acquire of a semaphore

- @name => the name of the semaphore
- @timeout => the timeout used in waiting operations

2.21.2 Semaphore Release

```
function semaphore_release($name = '')
```

This function implement the release of the semaphore

- @name => the name of the semaphore

2.21.3 Semaphore Shutdown

```
function semaphore_shutdown()
```

This function implement the shutdown of all semaphores, to do it, the function will iterate in each semaphore to release and set to null the semaphore pointer

2.21.4 Semaphore File

```
function semaphore_file($name = '')
```

This function returns the associated semaphore file used by the named semaphore, useful for debug purposes

- @name => the name of the semaphore

2.21.5 Semaphore helper

```
function __semaphore_helper($fn, $name, $timeout)
```

This function implements the real semaphore functionalities, includes the code to do an acquire, the release, the shutdown and to get the file, is programmed as a function instead of a class by historical motivation, in reality, the statics fds acts as a properties of a class and each if stripos acts as a methods of a class

- @fn => the function name that call the helper, to detect the feature
- @name => the name of the semaphore
- @timeout => the timeout used in waiting operations

2.21.6 Semaphore USleep helper

```
function __semaphore_usleep($usec)
```

This function implements an usleep (micro sleeper) using sockets, this allow to break the execution of the function if a signal is received by the process, in reality, the feature is powered by the socket_select that is allowed to wait for read and write operations with a very precise timeout.

The returned value will be the difference between the end less the start, in other words, the returned value is the ellapsed time sleeped by the function

- @usec => the micro seconds that you want to sleep

2.22   Server helper module

`api/php/autoload/server.php`

This fie contains useful functions related to the $_SERVER global variable, currently only publish a getter function, but in the future, can store more features if it is needed

2.22.1   Get Server

`function get_server($key)`

This function returns the server variable requested by index if it exists

- @key => the index key used to get the value of the server

2.22.2   Set Server

`function set_server($key, $val)`

This function is intended to replace some server variabe in runtime mode

- @key => the index key used to get the value of the server
- @val => the value that you want to set in the server array

2.22.3   Current hash

`function current_hash()`

This function tries to do the same like current_user but for the hash parameter obtained from the QUERY_STRING server variable

2.23   SQL utils helper module

`api/php/autoload/sql.php`

This fie contains useful functions related to SQL queries, allow to help modules that requires the entire management of the database (create tables, drop tables, create indexes, and more), too allow to prepare sql queries suck inserts, updates or wheres fragments that are procected to external injections, for example, by escaping all special characters.

Too it provides functions to do subparts of the where queries suck as special likes combinations or match again combinations for the fulltext search engine, see all detailed information by reading the list of functions of this module

2.23.1   Parse Query

`function parse_query($query, $type = '')`

This function is intended to apply the query filters defined by the users when write queries for multiples db engines as MySQL and/or SQLite, for example, if you want to write a fragment of SQL with one version for MySQL and another version for SQLite, you can do / *MYSQL ... * // *SQLite ... * /

Note that the previous example add a spaces between the bar and the asterisc because we can not put comments inside another comment!!!

- @query => the query that must be parsed

- @type => the db type that you want to allow by the filters

2.23.2 Parse Query Type helper

```
function __parse_query_type()
```

This function returns the type used by parse_query using as detector the dbtype of the config file, currently only allow to return MYSQL and/or SQLITE

2.23.3 Parse Query Strpos helper

```
function __parse_query_strpos($haystack, $needle, $offset = 0)
```

This function is the same that strpos, but with some improvements required by the parse_query funcion, the idea is to use the strpos functionality, but controlling that the found position must acomplish some constraints as the number of simple and double quotes must to be even

The arguments are the same that the strpos function

- @haystack => string where search the needle

- @needle => the needle text that must be found in the haystack

- @offset => bias applied to begin the search of the needle

2.23.4 Execute Query

```
function execute_query($query, $params = null)
```

This function executes the query and depending in the result, returns the resultset trying to do the more good combination in the return data

- @query => the SQL query that you want to execute

Note that the db_query is executed with the "auto" fetch mode, this causes that the db_query returns an array with one dimension if the query only generates a resultset with only one column, or returns an array with two dimensions if the query generates a resultest with more that one column

To be more practice:

If you execute a query that select one field and only returns one row, the return value will be the value of the field

If you execute a query that select one field and returns more that one row, the return value will be an array of one dimension with all values of this field

If you execute a query that select multiples fields and only return one row, the return value will be an array of one dimension with all fields

Ig you execute a query that select multiples fields and returns more that one row, the return value will be an array of two dimensions with all rows and each row with all fields

Be carefully to use the output of this command in an foreach, for example because you can get for the same query differents output types, if you need to be more standarized in the output types, see the execute_query_array

2.23.5 Execute Query Array

```
function execute_query_array($query, $params = null)
```

This function is the same that execute_query but guarantee that for the same query, you get the same output type if the resultet contains one row or more rows, useful is you want to use the output of this function in a foreach, for example

- @query => the SQL query that you want to execute

2.23.6 Get Fields

```
function get_fields($table)
```

This function returns the fields of the requested table

- @table => the table where that you want to know the fields

2.23.7 Get Indexes

```
function get_indexes($table)
```

This function returns the indexes of the requested table

- @table => the table where that you want to know the indexes

2.23.8 Get Tables

```
function get_tables()
```

This function returns the tables of the database

2.23.9 Get Field Type

```
function get_field_type($type)
```

This function returns an standarized type for the specific types used in the real database, for example, returns string if the field is of TEXT type

- @type => the real type in the database

2.23.10 Get Field Size

```
function get_field_size($type)
```

This function returns the size for the types used in the database, for example, returns 65535 if the field is of TEXT type

- @type => the real type in the database

Notes:

In general, type1 is used to detect the size of text fields, the type2 is used for VARCHAR(X) that specify the size of the VARCHAR in the type2 parameter.

2.23.11 Has Engine

```
function __has_engine($engine)
```

This function allow to SaltOS to ask to the database if an enxine is availabie

- @engine => the engine that you want to get information about existence

2.23.12 Make Insert Query

```
function make_insert_query($table, $array)
```

Returns the insert query for the table with all fields specified by the array param

- @table => table where you want to add the register
- @array => array with key val pairs that represent the field and the value of the field

Notes:

This function tries to cast each value to their data type getting this information from dbschema config, you can pass in array all fields that you want and not is needed to put all fields of the table, only the fields that appear in the array will be used in the insert, if some field is not a part of the fields of the table, an error will be triggered

This function uses the array_key_exists instead of isset because the check of the $array[$name] fails when the item exists but is false or null, for example

2.23.13 Make Update Query

```
function make_update_query($table, $array, $where)
```

Returns the update query for the table with all fields specified by the array param and using the specified where

- @table => table where you want to update the register
- @array => array with key val pairs that represent the field and the value of the field
- @where => array with key val pairs that represent the field and the value of the field used in the where part of the query

Notes:

This function tries to cast each value to their data type getting this information from dbschema config, you can pass in array all fields that you want and not is needed to put all fields of the table, only the fields that appear in the array will be used in the update, if some field is not a part of the fields of the table, an error will be triggered

This function uses the array_key_exists instead of isset because the check of the $array[$name] fails when the item exists but is false or null, for example

2.23.14   Make Where Query

```
function make_where_query($table, $array)
```

This function allow to create where sentences joinin all fields by AND

- @table => table where you want to apply the where
- @array => array with key val pairs that represent the field and the value of the field

Notes:

The keys normally contains the name of the field, but if you need to use a different comparison operator, you can use the field name and add the operator that you want to use in the comparison, the allowed comparison operators are $>, <, =, >=, <=, !=$

2.23.15   Escape Reserved Word

```
function escape_reserved_word($word)
```

This function tries to escape the reserved words that can not be used in sql queries as field names or table names, currently is only used to escape field names but in a future, if it is needed, can be added to escape table names too

- @word => the word that must to be escape if needed

Notes:

If you use an array as argument, then the function is applied to all elements of the array.

2.23.16   Make Like Query

```
function make_like_query($keys, $values, $args = [])
```

This function is intended to returns the sql fragment to be added to the where condition to filter for the specified keys and values

- @keys => an string with comma separated field names
- @values => the value of the input search
- @minsize => the minimal size of the length used in each like
- @default => sql fraement returned if some thing was wrong

Notes:

This function generates a sequence of (like or like) and (like and like) and is able to understand the prefix plus or minus in each word of the search string, this allow to the function to use the like or not like depending the sign of the word, and too to use the disjunction or conjunction in each like group

2.23.17   Make Fulltext Query Helper

```
function __make_fulltext_query_helper($values, $args = [])
```

This function is similar to the make_like_query, but uses the match agains clausule instead of the like clausule, the match agaings is used for fulltext searches and generally, this function is not intended to be used directly, it must acts as a helper of the make_fulltext_query

- @values => the value of the input search

- @minsize => the minimal size of the length used in each like

- @default => sql fraement returned if some thing was wrong

Notes:

This function differs between the make_like_query in the idea that this function only is used to search using fulltext indexes and in one unique field named search

2.23.18   Make Fulltext Query

```
function make_fulltext_query($values, $app, $args = [])
```

While the two version returns the fragment that must to be added to the query that search in the table that contains the search field, this function allow to specify the same that the two version with two fields more, the app and the prefix to be added to the id field of the in subquery

- @values => the value of the input search

- @app => the app used to detect the indexing table

- @prefix => the prefix added to the id used in the in subquery

- @minsize => the minimal size of the length used in each like

- @default => sql fraement returned if some thing was wrong

2.23.19   Prepare Helper Query

```
function __prepare_helper_query($table, $array)
```

This function allow to get the names and values used internaly by the prepare_insert_query, prepare_update_query and prepare_where_query, the main idea is to get the fields of the table and computes the intersect with the array provided with data, and using the spec of the table prepare the values with the needed type cast

- @table => the table where to you want to use the data

- @array => the array that contains the data

2.23.20   Prepare Insert Query

```
function prepare_insert_query($table, $array)
```

Returns the prepared insert query for the table with all fields specified by the array param

- @table => table where you want to add the register
- @array => array with key val pairs that represent the field and the value of the field

2.23.21 Prepare Update Query

```
function prepare_update_query($table, $array, $where = [])
```

Returns the prepared update query for the table with all fields specified by the array param and using the specified where

- @table => table where you want to update the register
- @array => array with key val pairs that represent the field and the value of the field
- @where => array with key val pairs that represent the field and the value of the field used in the where part of the query

2.23.22 Prepare Where Query

```
function prepare_where_query($table, $array)
```

This function allow to create where sentences joinin all fields by AND

- @table => table where you want to apply the where
- @array => array with key val pairs that represent the field and the value of the field

2.24 String utils helper module

```
api/php/autoload/strings.php
```

This fie contains useful functions related to strings manipulations, suck as get the test in UTF8, remove bad chars, of apply filters to the text, manipulate texts of other related tasks with strings

2.24.1 Remove Bar Chars

```
function remove_bad_chars($temp, $pad = '')
```

This function removes chars from keycodes 0 to 31 except 9, 10, 13 (tab, newline, return)

- @temp => input string that you want to fix
- @pad => padding string used as replacement for bar chars (void by default)

2.24.2 Encode Bar Chars

```
function encode_bad_chars($cad, $pad = '_', $extra = '')
```

This function tries to replace accenteds chars and other extended chars into an ascii chars, to do it, they define an array with the pairs of chars to do a quick replace, too is converted all to lower and are removed

all chars that are out of range (valid range are from 0-9 and from a-z), the function allow to specify an extra parameter to add extra chars that must to be allowed in the output, all other chars will be converted to the padding argument, as a bonus extra, all padding repetitions will be removed to only allow one pading char at time

- @cad => the input string to encode

- @pad => the padding char using to replace the bar chars

- @extra => the list of chars allowed to appear in the output

2.24.3 Prepare Words

```
function trim_words($cad, $pad = ' ')
```

This function allow to prepare words removing repetitions in the padding char

- @cad => the input string to prepare

- @pad => the padding char using to replace the repetitions

Notes:

Apart of remove repetitions of the padding char, the function will try to remove padding chars in the start and in the end of the string

2.24.4 Sprintr

```
function sprintr($array)
```

This function is an improved version of the print_r, allow to convert an array into a string removing some extra lines that not contain information, lines that contains only contains an open or close parenthesis, or nothing, are removed, optimizing the output string

- @array => the array that do you want to convert into string

2.24.5 Get Unique ID MD5

```
function get_unique_id_md5()
```

This function returns an unique hash using the random generator

2.24.6 Intelligence Cut

```
function intelligence_cut($txt, $max, $end = '...')
```

This function allow to cut text by searching spaces to prevent to break words

- @txt => the text that you want to cut

- @max => the size of the expected output text

- @end => the suffix added if the text is cutted

2.24.7 Normalize Value

```
function normalize_value($value)
```

This function allow to detect the last letter to detect what magnitude is using (K, M or G) and multiply the numeric part by the needed factor to get the number without factor

- @value => the string that contain the number, for example "123k"

2.24.8 HTML to Text

```
function html2text($html)
```

This function uses the html2text roundcube function to convert html to plain text, this code have the issue that requires the error_reporting(0) because it have a lot of errors causes by use nondefined variables, for example

- @html => the html code that you want to convert to plain text

2.24.9 Get UTF-8

```
function getutf8($str)
```

This function returns the string codified in a UTF-8 encoding

- @str => the input string that you want to covnert to UTF-8

2.24.10 Words Exists

```
function words_exists($words, $buffer)
```

This function check that all words exists in the buffer

- @words => the string that contains words separated by spaces
- @buffer => the string where we must to found the words

2.24.11 String Replace Assoc

```
function str_replace_assoc($array, $cad)
```

This function do the same that str_replace, but using only one associative array, using the keys as search and the values as replace, intended only to do more prerry the code

- @array => the associative array with the pairs keys vals
- @cad => the string that you want to apply the replacement

Notes:

This function can be replaced by strtr in the two arguments option, the difference is that str_replace is more efficient that strtr because the strtr tries to prepare the replacement list by ordering by size the replace_pairs and too to prevent repeated replacements in previously replacements, you can see test_strtr.php for more info

2.24.12 Get Part From String

```
function get_part_from_string($input, $delim, $index)
```

This function explodes de input using delim and returns the element of the index position, if the index is negative, then returns the element beginning from the end

- @input => the string that you want to cut in parts
- @delim => the delimiter char used to cut in parts
- @index => the index that you want to request of the explode result

2.24.13 Check IDS Helper

```
function __check_ids_helper()
```

This function checks the correctness of the list of ids and returns a valid list available to be used in sql queries or as an array of valid ids

- @ids => the string containing the list of ids

2.24.14 Check IDS

```
function check_ids(...$args)
```

This function checks the correctness of the list of ids and returns a valid list available to be used in sql queries

- @ids => the string containing the list of ids

2.24.15 Check IDS Array

```
function check_ids_array(...$args)
```

This function checks the correctness of the list of ids and returns a valid list available to be used as array of valid ids

- @ids => the string containing the list of ids

2.24.16 String Replace One

```
function str_replace_one($from, $to, $cad)
```

This function tries to do the same that str_repalce but only for the first occurrence

- @from => the string used as search
- @to => the string that you want to use as replacement
- @cad => the string that you want to modify

Notes:

This function can be replaced by preg_replace, but this implementation is more efficient, see the test_replace.php for more info

2.24.17 Get String From Quotes

```
function get_string_from_quotes($val)
```

This string tries to return the string contained in a single or double quotes, indended to be used for example in the construction of the where used by the search engine

- @val => the string that you want to process

2.25 System helper module

```
api/php/autoload/system.php
```

This fie contains useful functions related to system checks, allow to detect dependencies not installed on the system, or misconfigurations on the SaltOS installation

2.25.1 Check System

```
function check_system()
```

This function checks the system to detect if all knowed dependencies are found in the system, to do it, defines an array with the type (class or function), the name and some extra info for the error message that is triggered if the dependency is not satisfied

2.25.2 Check Directories

```
function check_directories()
```

Check all directories of the data directory to validate that the process can write inside it

2.25.3 Exec Check System

```
function exec_check_system()
```

This function executes the check system function and trigger an error if needed

2.26 Tokens helper module

```
api/php/autoload/tokens.php
```

This fie contains the functions related to the tokens usage and manipulations

2.26.1 Get Unique Token

```
function get_unique_token()
```

This function returns a string with a new and unique token

2.26.2   Check token format

```
function check_token_format($token)
```

This function checks the correctness of the token and returns a valid string that can be used safely as token in sql queries

- @token => the token that you want to process

2.27   User helper module

```
api/php/autoload/user.php
```

This file contains the functions associated to the user validation, to improve the performance, all functions are using a cache based trick that performs an important speed up

2.27.1   Current Token

```
function current_token()
```

This function returns the id of the current token, this info is retrieved using the token of the request, for security reasons, this validation only can be performed by the same origin that execute the login action

2.27.2   Current User

```
function current_user()
```

This function returns the id of the current user, this info is retrieved using the token of the request

Notes:

This function allow to authenticate using the token_id provided by the current_token function and using the server/user variable in the data structure that contains the desired login when executed using the cli sapi.

2.27.3   Current Group

```
function current_group()
```

This function returns the id of the current group, this info is retrieved using the token of the request

2.27.4   Current Groups

```
function current_groups()
```

This function returns the id of all current groups, this info is retrieved using the token of the request and the main idea of this function is to returns the list of all groups associated to the current user to facily the permissions checks

2.27.5 Crontab Users

```
function crontab_users()
```

This function executes the maintenance queries to update the active field in the passwords and tokens tables, it's intended to be used as helper

Notes:

This function uses an internal static variable to detect repeated executions and only accepts the first execution, this is to prevent that multiples calls to other actions and functions that requires the integrity of the passwords and tokens

2.28 Version helper module

```
api/php/autoload/version.php
```

This fie contains useful functions related to the version of the SaltOS of the php engine

2.28.1 Get Name Version Revision

```
function get_name_version_revision($full = false)
```

This function returns a string with the SaltOS name, version, revision and copyright if needed

- @full => boolean to specify if you want to add the copyright to the output

2.28.2 SVN Version

```
function svnversion($dir = null)
```

This function tries to return the svn version of the project

- @dir => allow to specify where do you want to execute the svnversion command

2.28.3 SVN Version helper

```
function __svnversion_helper($dir)
```

This function tries to return the svn version of the project

- @dir => allow to specify where do you want to execute the svnversion command

2.28.4 GIT Version

```
function gitversion($dir = null)
```

This function tries to return the git version of the project

- @dir => allow to specify where do you want to execute the gitversion command

2.28.5 GIT Version helper

```
function __gitversion_helper($dir)
```

This function tries to return the git version of the project

- @dir => allow to specify where do you want to execute the gitversion command

2.29 XML to Array helper module

```
api/php/autoload/xml2array.php
```

This fie contains useful functions related to the xml manipulation, this code is a part of the main version of the SaltOS because the main idea defined some years ago continues active, if you compare this code, you can see more accuracy in the specification to be more specific and precise but this code is a part of all the SaltOS versions

2.29.1 Set Array

```
function set_array(&$array, $name, $value)
```

This function allow to specify multiples entries in an array with the same key, to do this, the function will add #num where num is a unique number, in reality if you want to set multiples entries for the key "test", you get in reality an array with entries as "test", "test#1", "test#2"

This function works in concordance of the fix_key, that is able to get the key as "test#1" and return only "test" that is the original key without the suffix added to allow multiples instances of the same key in an associative array

- @array => array that you want to add the key with the value (by reference)
- @name => the key used in the array, if exists, it will try to add the suffix to prevent collisions
- @value => the value that you want to set in this position of the array

2.29.2 Unset Array

```
function unset_array(&$array, $name)
```

This function remove all entries of the array that matches with the name of the key, for example, if you specify the name "test", the function unset all entries as "test" or begin by "test#", in the example of the previous function will remove "test", "test#1" and "test#2"

- @array => array that you want to remove the key (by reference)
- @name => the key used in the array and as prefix of the entries of the array

2.29.3 Fix Key

```
function fix_key($arg)
```

This function returns the "real" part of the key removing the suffix added to prevent collisions in the associative array, for the above example, if you request the fix_key of the "test#2", the function will returns "test"

- @arg => the name of the key that you want to remove the suffix part (if exists)

2.29.4 XML Files to Array

```
function xmlfiles2array($files, $usecache = true)
```

This function allow to convert all XML files to an array, allow to use cache to optimize repetitive calls of the same file

As an special mention, this function internally implements the old xml_join feature that allow to merge multiple files into one using the fix_key of the keys in the first level as key to join.

- @files => the files that you want to convert from xml to array
- @usecache => if do you want to enable the cache feature

Notes:

- The arrays2array function is derived from this function that get all arguments and process the contents to join with a certain logics

2.29.5 XML File to Array

```
function xmlfile2array($file, $usecache = true)
```

This function allow to convert a XML file to an array, allow to use cache to optimize repetitive calls of the same file

As an special mention, this function internally uses semaphores to prevent multiple instances of the same execution with the same file, too uses a cache management to optimize the usage

- @file => the file that you want to convert from xml to array
- @usecache => if do you want to enable the cache feature

2.29.6 XML to Array

```
function xml2array($xml, $file = '')
```

This function allow to convert a XML string to an array

- @xml => xml code to be converted to an array
- @file => filename of the contents, only used when an errors occurs

2.29.7 XML to Struct

```
function xml2struct($xml, $file = '')
```

This function is a helper of the xml2array function, the main purpose of this function is to convert the xml string into a struct to be processed by the struct2array function

The motivation to use the xml_parse_into_struct function is that this function is the more quick to parse xml files, after a lot of tests, the more quickly execution is to use the xml_parse_into_struct, reverse the array and then program a simple recursive function that convert a unidimensional array into a tree

At the begining of this function, we will try to detect the enconding of the xml file, the main objective is to convert all xml to UTF-8 that is the default enconding of SaltOS

The returned value is the result of the xml_parse_into_struct function, that is the key of this feature and this function

- @xml => xml fragment that must be converted into struct

- @file => the source filename, it is used only to generate error reports

2.29.8 Struct to Array

```
function struct2array(&$data, $file = '')
```

This function is the second part in the xml2array convertsion, here, the function receives an unidimensional array with commands to open, close, and their respective values and attributes, with this information, this function is able to generate a tree with the xml converted to an array tree

- @data => the struct array, by reference

- @file => the source filename, it is used only to generate error reports

Notes:

This function uses recursivity to accomplish the objetive, returns the portion of xml between the open and close command, in each call, the data array passed by reference will decrement in size because the array_pop removes the last element of the array

Remember that previously of call this function, the array is reversed, this is because is more efficient to do a reverse and then pops instead of use directly the array_shift to get the next element, the reason is that array_shift must to reorder all keys of the resulted array and this add a very big cost if the xml is big, this problem was detected in 2014 and was optimized by add the reverse and the pop instead of only shift

2.29.9 Eval Attributes

```
function eval_attr($array, $recursive = true)
```

This function is very special in SaltOS, is part of the initial code an is used by a lot of parts of the program, currently are using a simplified version of the original function and have improvements that allow to return arrays with attributes without evaluate and without causing an error, this allow to define xml with attributes that can be used by other processes and SaltOS only interpret three attributes

- @array => the array that contains a tree representation of the xml

The three attributes are:

- @eval => this attribute must be a boolean and allow to evaluate the value of the node

- @ifeval => this attribute must contains an expression that must evaluate as true or false, and allow to maintain or remove the entire node thas contains the ifeval attribute, this is useful when you need a node in some conditions

- @require => this attribute allow to SaltOS to add code to the current exec context and is intended to load code that will then be used by the eval feature

The great change between the eval_attr of the previous versions of SaltOS is that this version only accepts three internal commands and the other attributes can be maintained in order to be used by other processes (internally or externally)

2.29.10 Eval Bool

```
function eval_bool($arg)
```

This function returns a boolean depending on the input evaluation, the main idea is to get an string, for example, and determine if must be considered true or false otherwise will finish in an error

The valid inputs are the strings one, zero, void, true, false, on, off, yes and no

- @arg => the value that do you want to evaluates as boolean

2.30 Yaml helper module

```
api/php/autoload/yaml.php
```

This file provide the functions provided by the php-yaml package, intended to be used by setups that can not install this package.

2.30.1 Yaml Parse

```
function yaml_parse(string $yaml)
```

Parse a YAML string into a PHP array.

- @yaml => The YAML string.

2.30.2 Yaml Parse File

```
function yaml_parse_file(string $filename)
```

Parse a YAML file into a PHP array

- @filename => The path to the YAML file

2.30.3 Yaml Emit

```
function yaml_emit(array $data, int $inline = 2, int $indent = 4)
```

Emit an array as a YAML string.

- @data => The data to convert to YAML.
- @inline => The level at which to start inlining YAML (default: 2).
- @indent => The number of spaces to use for indentation (default: 4).

2.30.4 Yaml Emit File

```
function yaml_emit_file(string $filename, array $data, int $inline = 2, int
    $indent = 4)
```

Emit an array as a YAML file

- @filename => The path to save the YAML file

- @data => The data to convert to YAML

- @inline => The level at which to start inlining YAML (default: 2)

- @indent => The number of spaces to use for indentation (default: 4)

2.31   Main execution module

`api/php/autoload/zindex.php`

This file contains the old index.php file, this was moved here to simplify the index.php and to allow some php checks found in the current index.php

This code implements the main method to access to the SaltOS API using rest and json requests, to use it, you can use the follow methods:

1) Rest using GET requests

This kind of requests requires that you send a GET request with a querystring of the follow form:

- @https://127.0.0.1/saltos/code4/?app/invoices/view/2

And the system process it of the follow form:

- @rest/1 => invoices

- @rest/2 => view

- @rest/3 => 2

And you can programm any action that uses these parameters to do the desired task

2) Json using POST requests

This other kind of requests requires that you send a POST request with the appropiate header for the content-type as application/json and a json in the body of the request, with this call, saltos can map all contents of the json to the json/????? variables.

As an extra bonus, this module defines some useful server variables used in a lot of features of saltos, like the follow vars:

- @request_method => can be GET or POST

- @content_type => used to check the content type for the JSON requests

- @token => used to validate the HTTP_TOKEN send as authentication

- @remote_addr => used internally for security reasons

- @user_agent => used internally for security reasons

As a brief resume, you can use the follow keys in get_data or set_data:

- @rest => to get an array with all rest data, for the above example they must return some thing like this: ["app", "invoices", "view", "2"]

- @rest/1 => to get only the element that contains "invoices"

- @rest/2 => to get only the element that contains "view"

- @rest/3 => to get only the element that contains "2"

- @json => to get an array with all json data, for the above example they must return some thing like

80

this: ["user"=>"xxx", "pass"=>"xxx"]

- @json/user => to get only the element that contains the user
- @json/pass => to get only the element that contains the pass
- @server => to get an array with all server data
- @server/request_method => can be GET or POST
- @server/content_type => used to check the content type for the JSON requests
- @server/token => used to validate the HTTP_TOKEN send as authentication
- @server/remote_addr => used internally for security reasons
- @server/user_agent => used internally for security reasons

3 Database

3.1 SQLite3 functions library

```
api/php/database/libsqlite.php
```

SQLite's database allow to define external functions that can be used from the SQL language, this is a great feature that allow to use SQLite as MySQL, and using this feature of the database, the SQLite drivers use the libsqlite to add a lot of features found in MySQL and used in a lot of queries by SaltOS

More info about this feature by searching:

- @PDO => sqliteCreateFunction/sqliteCreateAggregate
- @SQLite3 => createFunction/createAggregate

3.1.1 GROUP_CONCAT

```
function __libsqlite_group_concat_step($context, $rows, $string, $separator
    = ',')
```

This function add the GROUP_CONCAT to the SQLite database

3.1.2 GROUP_CONCAT

```
function __libsqlite_group_concat_finalize($context, $rows)
```

This function add the GROUP_CONCAT to the SQLite database

3.1.3 REPLACE

```
function __libsqlite_replace($subject, $search, $replace)
```

This function add the REPLACE to the SQLite database

3.1.4 LPAD

```
function __libsqlite_lpad($input, $length, $char)
```

This function add the LPAD to the SQLite database

3.1.5 CONCAT

```
function __libsqlite_concat(...$args)
```

This function add the CONCAT to the SQLite database

3.1.6 CONCAT_WS

```
function __libsqlite_concat_ws(...$args)
```

This function add the CONCAT_WS to the SQLite database

3.1.7 UNIX_TIMESTAMP

```
function __libsqlite_unix_timestamp($date)
```

This function add the UNIX_TIMESTAMP to the SQLite database

3.1.8 FROM_UNIXTIME

```
function __libsqlite_from_unixtime($timestamp)
```

This function add the FROM_UNIXTIME to the SQLite database

3.1.9 YEAR

```
function __libsqlite_year($date)
```

This function add the YEAR to the SQLite database

3.1.10 MONTH

```
function __libsqlite_month($date)
```

This function add the MONTH to the SQLite database

3.1.11 WEEK

```
function __libsqlite_week($date, $mode = 0)
```

This function add the WEEK to the SQLite database

3.1.12 TRUNCATE

```
function __libsqlite_truncate($n, $d)
```

This function add the TRUNCATE to the SQLite database

3.1.13 DAY

```
function __libsqlite_day($date)
```

This function add the DAY to the SQLite database

3.1.14 DAYOFYEAR

```
function __libsqlite_dayofyear($date)
```

This function add the DAYOFYEAR to the SQLite database

3.1.15 DAYOFWEEK

```
function __libsqlite_dayofweek($date)
```

This function add the DAYOFWEEK to the SQLite database

3.1.16 HOUR

```
function __libsqlite_hour($date)
```

This function add the HOUR to the SQLite database

3.1.17 MINUTE

```
function __libsqlite_minute($date)
```

This function add the MINUTE to the SQLite database

3.1.18 SECOND

```
function __libsqlite_second($date)
```

This function add the SECOND to the SQLite database

3.1.19 MD5

```
function __libsqlite_md5($temp)
```

This function add the MD5 to the SQLite database

3.1.20 REPEAT

```
function __libsqlite_repeat($str, $count)
```

This function add the REPEAT to the SQLite database

3.1.21 FIND_IN_SET

```
function __libsqlite_find_in_set($str, $strlist)
```

This function add the FIND_IN_SET to the SQLite database

3.1.22 IF

```
function __libsqlite_if($condition, $value_if_true, $value_if_false)
```

This function add the IF to the SQLite database

3.1.23 POW

```
function __libsqlite_pow($base, $exp)
```

This function add the POW to the SQLite database

3.1.24 DATE_FORMAT

```
function __libsqlite_date_format($date, $format)
```

This function add the DATE_FORMAT to the SQLite database

3.1.25 NOW

```
function __libsqlite_now()
```

This function add the NOW to the SQLite database

3.2 MySQL improved driver

```
api/php/database/mysqli.php
```

This file implements the MySQL improved driver. This driver was coded later that the initial mysql driver and contains lots of improvements respect to the old mysql driver. The old mysql driver was removed in the PHP 7.0 release.

3.2.1 Database MySQL improved class

```
class database_mysqli
```

This class allow to SaltOS to connect to MySQL databases using the MySQL improved driver

3.2.2 Private link variable

```
private $link = null;
```

This private variable contains the link to the database

3.2.3 Constructor

```
public function __construct($args)
```

This public function is intended to stablish the connection to the database

- @args => is an array with key val pairs
- @host => the host for the connection
- @port => the port used for the connection
- @name => name of the database for the connection
- @user => user used to stablish the connection
- @pass => pass used to stablish the connection

Notes of this driver:

MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT are the same that PDO::ERRMODE_EXCEPTION
MYSQLI_USE_RESULT is the same that PDO::MYSQL_ATTR_USE_BUFFERED_QUERY = false

3.2.4 DB Check

```
public function db_check($query, $params = [])
```

This public function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

3.2.5 DB Escape

```
public function db_escape($str)
```

This public function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

3.2.6 DB Query

```
public function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute
- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

3.2.7   DB Last Insert ID

```
public function db_last_insert_id()
```

This function returns the last insert id

3.2.8   DB Disconnect

```
public function db_disconnect()
```

This function close the database connection and sets the link to null

3.3   PDO MsSQL driver

```
api/php/database/pdo_mssql.php
```

This file implements the MySQL improved driver. This driver uses the dblib library to connect with SQL servers.

3.3.1   Database PDO MsSQL class

```
class database_pdo_mssql
```

This class allow to SaltOS to connect to MsSQL databases using the PDO driver

3.3.2   Private link variable

```
private $link = null;
```

This private variable contains the link to the database

3.3.3 Constructor

```
public function __construct($args)
```

This public function is intended to stablish the connection to the database

- @args => is an array with key val pairs
- @host => the host for the connection
- @port => the port used for the connection
- @name => name of the database for the connection
- @user => user used to stablish the connection
- @pass => pass used to stablish the connection

3.3.4 DB Check

```
public function db_check($query, $params = [])
```

This public function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

3.3.5 DB Escape

```
public function db_escape($str)
```

This public function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

Notes:

This driver adds a simple quotes at the beginning and at the finish of ths string, by this reason this function returns the substr(1, -1) of the quoted string.

3.3.6 DB Query

```
public function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute
- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

3.3.7 DB Last Insert ID

```
public function db_last_insert_id()
```

This function returns the last insert id

3.3.8 DB Disconnect

```
public function db_disconnect()
```

This function close the database connection and sets the link to null

3.4 PDO MySQL driver

```
api/php/database/pdo_mysql.php
```

This file implements the MySQL improved driver. This is the recommended driver when you want to use MySQL servers, it uses the PDO extension and alow to connect to MySQL servers with a good performance.

3.4.1 Database PDO MySQL class

```
class database_pdo_mysql
```

This class allow to SaltOS to connect to MySQL databases using the PDO driver

3.4.2 Private link variable

```
private $link = null;
```

This private variable contains the link to the database

3.4.3 Constructor

```
public function __construct($args)
```

This public function is intended to stablish the connection to the database

- @args => is an array with key val pairs

- @host => the host for the connection

- @port => the port used for the connection

- @name => name of the database for the connection

- @user => user used to stablish the connection

- @pass => pass used to stablish the connection

3.4.4   DB Check

```
public function db_check($query, $params = [])
```

This public function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

3.4.5   DB Escape

```
public function db_escape($str)
```

This public function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

Notes:

This driver adds a simple quotes at the beginning and at the finish of ths string, by this reason this function returns the substr(1, -1) of the quoted string.

3.4.6   DB Query

```
public function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute

- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

3.4.7 DB Last Insert ID

```
public function db_last_insert_id()
```

This function returns the last insert id

3.4.8 DB Disconnect

```
public function db_disconnect()
```

This function close the database connection and sets the link to null

3.5 PDO SQLite driver

```
api/php/database/pdo_sqlite.php
```

This file implements the MySQL improved driver. This is the recommended driver when you want to use SQLite3 files as database server and it uses the PDO extension to do it, this driver solves the concurrence problem using POSIX semaphores, generally it is a good option for setups that don't require a fulltext search optimizations suck as mroonga, intended for a personal usage or demos.

3.5.1 Database PDO SQLite class

```
class database_pdo_sqlite
```

This class allow to SaltOS to connect to SQLite databases using the PDO driver

3.5.2 Private link variable

```
private $link = null;
```

This private variable contains the link to the database

3.5.3 Constructor

```
public function __construct($args)
```

This public function is intended to stablish the connection to the database

- @args => is an array with key val pairs
- @file => the file that contains the database

Notes:

This database allow to define external functions that can be used from the SQL language, this is a great feature that allow to use SQLite as MySQL, and using this feature of the database, this driver uses the libsqlite to add a lot of features found in MySQL and used in a lot of queries by SaltOS

3.5.4 DB Check

```
public function db_check($query, $params = [])
```

This public function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

3.5.5 DB Escape

```
public function db_escape($str)
```

This public function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

Notes:

This driver adds a simple quotes at the beginning and at the finish of ths string, by this reason this function returns the substr(1, -1) of the quoted string.

3.5.6 DB Query

```
public function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute
- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

3.5.7 DB Last Insert ID

```
public function db_last_insert_id()
```

This function returns the last insert id

3.5.8   DB Disconnect

```
public function db_disconnect()
```

This function close the database connection and sets the link to null

3.6   SQLite3 driver

```
api/php/database/sqlite3.php
```

This file implements the MySQL improved driver. This is the recommended driver when you want to use SQLite3 files as database server, this driver solves the concurrence problem using POSIX semaphores, generally it is a good option for setups that don't require a fulltext search optimizations suck as mroonga, intended for a personal usage or demos.

3.6.1   Database SQLite3 class

```
class database_sqlite3
```

This class allow to SaltOS to connect to SQLite databases using the SQLite3 driver

3.6.2   Private link variable

```
private $link = null;
```

This private variable contains the link to the database

3.6.3   Constructor

```
public function __construct($args)
```

This public function is intended to stablish the connection to the database

- @args => is an array with key val pairs
- @file => the file that contains the database

Notes:

This database allow to define external functions that can be used from the SQL language, this is a great feature that allow to use SQLite as MySQL, and using this feature of the database, this driver uses the libsqlite to add a lot of features found in MySQL and used in a lot of queries by SaltOS

3.6.4   DB Check

```
public function db_check($query, $params = [])
```

This public function is intended to check that the query execution will not trigger an error

- @query => the query that you want to validate

3.6.5 DB Escape

```
public function db_escape($str)
```

This public function is intended to escape the special chars to sanitize the string to be used in a sql query

- @str => the string that you want to sanitize

3.6.6 DB Query

```
public function db_query($query, ...$args)
```

This public function is intended to execute the query and returns the resultset

- @query => the query that you want to execute
- @fetch => the type of fetch that you want to use, can be auto, query, column or concat

Notes:

The fetch argument can perform an speed up in the execution of the retrieve action, and can modify how the result is returned

auto: this fetch method try to detect if the resultset contains one or more columns, and sets the fetch to column (if the resultset only contains one column) or to query (otherwise)

query: this fetch method returns all resultset as an array of rows, and each row contain the pair of key val with the name of the field and the value of the field

column: this fetch method returns an array where each element is each value of the field of the each row, this is useful when for example do you want to get all ids of a query, with this method you can obtain an array with each value of the array is an id of the resultset

concat: this fetch method is an special mode intended to speed up the retrieve of large arrays, this is useful when you want to get all ids of a query and you want to get a big sized array, in this case, is more efficient to get an string separated by commas with all ids instead of an array where each element is an id

3.6.7 DB Last Insert ID

```
public function db_last_insert_id()
```

This function returns the last insert id

3.6.8 DB Disconnect

```
public function db_disconnect()
```

This function close the database connection and sets the link to null

4 Libraries

4.1 Actions module

```
api/php/lib/actions.php
```

This file provides the basic actions used by all apps like insert, update and delete, this modules uses features from others modules as controol, log, version, indexing, upload, depend and trash.

4.1.1 Insert action

```
function insert($app, $data)
```

This action allow to insert registers in the database associated to each app

- @app => the application involved in the action
- @data => an array with the data that you want to use for the operation

Notes:

The data array must contains the fields of the main table, an object with the subtables, and with two fields (addfiles and addnotes)

4.1.2 Update action

```
function update($app, $id, $data)
```

This action allow to update registers in the database associated to each app

- @app => the application involved in the action
- @data => an array with the data that you want to use for the operation

Notes:

The data array must contains the fields of the main table, an object with the subtables, and with four fields (addfiles, delfiles, addnotes and delnotes)

4.1.3 Delete action

```
function delete($app, $id)
```

This action allow to delete registers in the database associated to each app

- @app => the application involved in the action
- @data => an array with the data that you want to use for the operation

Notes:

This action removes the register identified by the id in the main table of the app and the subtables, too removes the related data of the files and notes tables, as an excepcion, the version table maintain all data to allow the recovery action, too, the files of the files feature will be moved to the trash folder to allow the previous recovery from the version table

4.2 Array to XML helper module

```
api/php/lib/array2xml.php
```

This fie is a part of the most old files of the SaltOS, accomplish the task to convert an array to an XML string, currently it is little used because the most important module compared to this is the inverse xml2array module

4.2.1 Check Node Name

```
function __array2xml_check_node_name($name)
```

This function acts as helper of the array2xml function, is intended to return if the node name is valid

- @name => the name that you want to validate

4.2.2 Check Attr Name

```
function __array2xml_check_node_attr($name)
```

This function acts as helper of the array2xml function, is intended to return if the attribute name is valid

- @name => the name that you want to validate

4.2.3 Write Nodes array2xml helper

```
function __array2xml_write_nodes(&$array, $level = null)
```

This function acts as helper of the array2xml function, is intended to return a string with the tree array

- @array => the tree array that you want to convert to XML
- @level => can be null to minify the output zero to indent the XML contents

4.2.4 Array to XML

```
function array2xml($array, $indent = false)
```

This function returns a string with the contents of array converted into a XML language file, to do it, uses some helpers as __array2xml_* functions

- @array => the array that contains the tree structure that you want to convert to XML
- @indent => a boolean to enable or disable the indent (the old minify) feature

4.3 Make Table ASCII

```
api/php/lib/ascii.php
```

This table is intended for debug purposes and is able to paint in ascii mode the contents of a matrix

4.3.1 Make Table ASCII

```
function __ascii_make_table_ascii($array)
```

This table is intended for debug purposes and is able to paint in ascii mode the contents of a matrix

- @rows => the contents of the matrix to paint
- @head => set to true if you want to use the first row as header
- @compact => set to true if you want to minify the ascii table

4.4 Login functions

```
api/php/lib/auth.php
```

This file contain all functions needed by the logins app

4.4.1 Authentication token action

```
function authtoken($user, $pass)
```

This file implements the login action, allowing to authenticate users using the pair of login/password to validate the credentials and get a valid token to operate in SaltOS

- @user => username used in the authentication process
- @pass => password used in the authentication process

This action not requires a valid token, all valid tokens associated to the user will be revoked when a new token is assigned, as the result of this action is a flag that indicates the validity of the token, this action returns a json with the status of te token instead of returns a json with an error in case of non validity

4.4.2 Deauthentication token action

```
function deauthtoken()
```

This file implements the logout action, allowing to deauthenticate users using a valid token, for security reasons, the deauth action only can be performed by the same actor that execute the login action

The unique requirement to execute this action is to have a valid token

4.4.3 Check token action

```
function checktoken()
```

This file implements the check action, allowing to check token's validity, the check action only can be performed by the same actor that execute the login action

The unique requirement to execute this action is to have a token to be checked, as the result of this action is a flag that indicates the validity of the token, this action returns a json with the status of te token instead of returns a json with an error in case of non validity

4.4.4 Authentication update action

```
function authupdate($oldpass, $newpass, $renewpass)
```

This file implements the update password action, allowing to authenticated users by a token, and providing the old password to update a new password

- @oldpass => Old password, must to validate the active password of the user associated to the token used in the action

- @newpass => New password, must to be new, must to pass the score check and never must to be used in the system for the user

- @renewpass => The repeated new password, to prevent writing errors

This action requires a valid token associated to the user that wants to do the password update

4.4.5   Score check

```
function score_check($newpass)
```

This function checks the score quality of the provided password

- @newpass => the password thay you want to check

4.4.6   Old password check

```
function oldpass_check($user_id, $oldpass)
```

This function checks that the provided password is valid for the user_id

- @user_id => the user_id to use in the check task
- @oldpass => the password to use in the check task

4.4.7   New password check

```
function newpass_check($user_id, $newpass)
```

This function checks that the provided password has never been used by the user_id

- @user_id => the user_id to use in the check task
- @newpass => the password to use in the check task

4.4.8   Old password disable

```
function oldpass_disable($user_id)
```

This function disable all passwords associated to the user_id

- @user_id => the user_id to use in the check task

4.4.9   New password insert

```
function newpass_insert($user_id, $newpass)
```

This function inserts a new password record to the database

- @user_id => the user_id to use in the insert task
- @newpass => the password to use in the insert task

Notes:

This function returns the created and expires timestamps

4.5 Barcode helper module

```
api/php/lib/barcode.php
```

This fie contains useful functions related to barcodes

4.5.1 BarCode image function

```
function __barcode_image($msg, $w, $h, $m, $s, $t)
```

This function allow to generate a barcode, you can pass the desired message that you want to convert in barcode and it returns an image with the data

- @msg => Contents of the barcode
- @w => width of each unit's bar of the barcode
- @h => height of the barcode (without margins and text footer)
- @m => margin of the barcode (white area that surround the barcode)
- @s => size of the footer text, not used if zero
- @t => type of the barcode, C128 is the most common type used

Notes:

The normal behavior is returns a png image, but if something was wrong, the function can returns an empty string

4.5.2 BarCode image png function

```
function __barcode_image_png($msg, $w, $h, $t)
```

This function allow to generate a barcode, you can pass the desired message that you want to convert in barcode and it returns an image with the data

- @msg => Contents of the barcode
- @w => width of each unit's bar of the barcode
- @h => height of the barcode (without margins and text footer)
- @t => type of the barcode, C128 is the most common type used

Notes:

The normal behavior is returns a png image without margins and text, but if something was wrong, the function can returns an empty string

4.5.3 BarCode image svg function

```
function __barcode_image_svg($msg, $w, $h, $t)
```

This function allow to generate a barcode, you can pass the desired message that you want to convert in barcode and it returns an image with the data

- @msg => Contents of the barcode
- @w => width of each unit's bar of the barcode
- @h => height of the barcode (without margins and text footer)
- @t => type of the barcode, C128 is the most common type used

Notes:

The normal behavior is returns a svg image without margins and text, but if something was wrong, the function can returns an empty string

4.6 Browser helper module

```
api/php/lib/browser.php
```

This file contain useful browser helper functions

4.6.1 Get Browser Platform Device Type

```
function get_browser_array($user_agent = null)
```

This function gets the browser, platform and device_type form the user_agent header

4.6.2 Get Browser Platform Device Type

```
function get_browser_string($user_agent = null)
```

This function gets the browser, platform and device_type form the user_agent header

4.7 Captcha helper module

```
api/php/lib/captcha.php
```

This fie contains useful functions related to captchas

4.7.1 Captcha Image

```
function __captcha_image($code, $args = [])
```

This function returns an image with the code drawed in a background that contains white noise to prevent that robots read the code

- @code => the code that you want to paint

- @width => the width of the generated image
- @height => the height of the generated image
- @letter => the size of the letters of the generated image
- @number => the size of the numbers of the generated image
- @angle => the angle allowed to rotate the letters and numbers
- @color => the color user to paint the code
- @bgcolor => the background color of the image
- @fgcolor => the color used to paint the letters of the background of the image
- @period => parameter for the wave transformation
- @amplitude => parameter for the wave transformation
- @blur => true or false to enable or disable the blur effect

Notes:

The main idea to program this captcha was obtained from this post:

- http://sentidoweb.com/2007/01/03/laboratorio-ejemplo-de-captcha.php

Too appear in ther posts if you search for it in google:

- http://www.google.es/search?q=captcha+alto_linea

4.7.2 Captcha Make Number function

```
function __captcha_make_number($length)
```

This function returns a random number of the desired length and as trick, checks that the output is a prime number

- @length => the length of the desired output string

4.7.3 Captcha Make Math Operation function

```
function __captcha_make_math($length)
```

This function returns a random math operation of the desired length and as trick, checks that the output operation is performed by prime numbers

- @length => the length of the desired output string

4.8   Color helper module

```
api/php/lib/color.php
```

This fie contains useful functions related to colors

4.8.1   Color To Dec function

```
function color2dec($color, $component)
```

This function is a helper that allow to get from a RGB hex color the value in decimal of the specified component, useful to get the amount of color red, green or blue in decimal base from an string

Is able to understand colors with the formats #abcdef, abcdef, #000, #fff

- @color => The color that you want to parse
- @component => The component that you want to retrieve their value

4.9 Control helper module

```
api/php/lib/control.php
```

This fie contains useful functions related to the control and version system, they allow to relationate registers with users and groups, and to add and retrieve the versions of a register

4.9.1 Make Control function

```
function make_control($app, $reg_id)
```

This function allow to insert and delete the control registers associacted to any application and to any register of the application

- @app => code of the application that you want to index
- @reg_id => register of the app that you want to index

Notes:

This function returns an integer as response about the control action:

- +1 => insert executed, this is because the app register exists and the control register not exists
- +2 => delete executed, this is because the app register not exists and the control register exists
- -1 => app not found, this is because the app requested not have a table in the apps config
- -2 => control table not found, this is because the has_control feature is disabled by dbstatic
- -3 => data not found, this is because the app register not exists and the control register not exists
- -4 => control exists, this is because the app register exists and the control register too exists

As you can see, negative values denotes an error and positive values denotes a successfully situation

4.9.2 Integrity

```
function integrity()
```

This function tries to execute some periodic task intended to fix issues with the integrity of internal relationships, to do it tries to search not found registers in the control table in the first loop and tries to search not found registers in the app table in the second loop, with all found not found registers the function executes the make_control that add or remove the needed control register to maintain the integrity.

4.10 Cron utils helper module

```
api/php/lib/cron.php
```

This fie contains useful functions related to cron operations

4.10.1 Cron compare helper

```
function __cron_compare($val, $now)
```

This function provide the compare feature to the cron system, it is able to evaluate the first argument to check the posible cases like true case, list of options, using module, using range and direct case

- @val => the value that you want to evaluate
- @now => the now value used in the evaluation

4.10.2 Cron is now helper

```
function __cron_is_now($minute, $hour, $day, $month, $dow)
```

This function uses the __crom_compare to validate if the five items of the cron configuration triggers the execution of the cron task

4.10.3 Cron users helper

```
function __cron_users($arg)
```

This function allow to get an array of users from the arg

- @arg => the list of users, can be * from all users

4.10.4 Cron GC

```
function cron_gc()
```

This function tries to monitorize the cron processes and store in the database all stdout and stderr with the timestamps of the start and stop processes, too removes all unused files

4.10.5 Cron Exec

```
function cron_exec()
```

This function tries to detect what cron tasks must to execute and with what users must to execute each cron task, too detects if a previous instance of the commands was launched previously and is still running in this moment

4.11   Database schema helper module

```
api/php/lib/dbschema.php
```

This fie contains useful functions related to database schema, allow to manage the entire database schema, and too, allow to maintain contents of some tables using the dbstatic feature

4.11.1   DB Schema

```
function db_schema()
```

This function try to maintain the database structure, to do it, this feature uses the dbschema.xml file to store the database structure.

4.11.2   DB Schema hash

```
function __dbschema_hash()
```

This function returns the hash used by db_schema

4.11.3   DB Schema check

```
function __dbschema_check()
```

This function returns the comparison between the old hash and the new hash

4.11.4   DB Static

```
function db_static()
```

This function try to maintain the database contents, to do it, this feature uses the dbstatic.xml file to store the database contents that must to be maintaned.

This version of the db_static allow you to use a comma separated values in fields as "id", start by "id_" or end by "_id"

4.11.5   DB Static hash

```
function __dbstatic_hash()
```

This function returns the hash used by db_static

4.11.6   DB Static check

```
function __dbstatic_check()
```

This function returns the comparison between the old hash and the new hash

4.11.7 DB Static insert

```
function __dbstatic_insert($table, $row)
```

This function is a helper of previous function, is intended to be used by db_static and allow to use a comma separated values in fields as "id", start by "id_" or end by "_id"

- @table => the table that you want to use in the insert process
- @row => the row that you want to add in the table

Notes:

This feature allow to you to use comma separated lists of values, commonly used for id fields as user_id, perms_id, or similar.

4.11.8 Get Tables from DB Schema

```
function get_tables_from_dbschema()
```

This function returns the tables from the DB Schema file

4.11.9 Get Fields from DB Schema

```
function get_fields_from_dbschema($table)
```

This function returns the fields from the DB Schema file

- @table => the table that you want to request the fields

4.11.10 Get Indexes from DB Schema

```
function get_indexes_from_dbschema($table)
```

This function returns the indexes from the DB Schema file

- @table => the table that you want to request the indexes

4.11.11 Get Ignores from DB Schema

```
function get_ignores_from_dbschema()
```

This function returns the ignores tables from the DB Schema file

4.11.12 Get Fulltext from DB Schema

```
function get_fulltext_from_dbschema()
```

This function returns the fulltext tables from the DB Schema file

4.11.13 Get Fkeys from DB Schema

```
function get_fkeys_from_dbschema($table)
```

This function returns the fkeys from the DB Schema file

- @table => the table that you want to request the fkeys

4.11.14 DB Schema helper

```
function __dbschema_helper($fn, $table)
```

This function is a helper for the previous functions, is intended to be used to returns the tables of the DB Schema or the fields of a table

- @fn => the caller function name
- @table => the table used by some features

4.11.15 DB Schema Auto Apps

```
function __dbschema_auto_apps($dbschema)
```

This function is a helper to the dbschema functions, to create an indexing table for each app

- @dbschema => the dbschema array

Notes:

This feature creates a table and try to use Mroonga storage engine with one field, the main idea of this tables is to store all contents of the register to do quick searchs using a fulltext search engine

4.11.16 DB Schema Auto Fkey

```
function __dbschema_auto_fkey($dbschema)
```

This function is a helper to the dbschema functions, to create an index for each fkey

- @dbschema => the dbschema array

Notes:

By default, MariaDB creates an index for each foreign key, but SQLite not does is by default and for this reason, SaltOS creates an index automatically, to improve the performance

This function checks that the field not exists in the defined indexes to prevent error in duplicates indexes

4.11.17 DB Schema Auto Name

```
function __dbschema_auto_name($dbschema)
```

This function is a helper to the dbschema functions, to auto name the indexes

- @dbschema => the dbschema array

Notes:

This function allow to specify indexes only specifying the fields that you want to conform the index, but the engines as MariaDB and SQLite, requires that each index have a unique name, and for this reason, we add this feature to automate this part of the process

You can see how the name of the index is different for MySQL and SQLite, this is because in MySQL, the name can be repeated in different tables, but in SQLite, the name must be unique in the database

4.11.18   Get Apps From DBStatic

```
function get_apps_from_dbstatic()
```

This function returns the list of apps that have a table and field defined in the dbstatic file

4.11.19   Get Tables From DBStatic

```
function get_tables_from_dbstatic()
```

This function returns the list of tables that have a table and field defined in the dbstatic file

4.11.20   Get Field From DBStatic

```
function get_field_from_dbstatic($table, $field = 'field')
```

This function return the field associated to the table in the dbstatic file and associated to the apps table

- @table => the table of the dbstatic that want to convert to field
- @field => the field name, field by default

Notes:

This function uses the special feature in the helper that allow to use as table parameter an app code to retrieve the field, this is useful if you want some field of the app table and you want to use the app code instead of the app table to identify what row do you want to use

4.11.21   DB Static helper

```
function __dbstatic_helper($fn, $table, $field)
```

This function is intended to act as helper of the dbstatic ecosystem, this function can return the apps that contain table and field definitions and too, can return the field associated to a apps table, useful for the indexing feature

- @fn => the caller function name
- @table => the table used by some features
- @field => the field used by some features

4.11.22   Manifest to dbstatic

```
function __manifest2dbstatic($files)
```

This function returns the equivalent dbstatic data using as input the contents of the manifests files.

- @files => An array with all the manifests files

4.11.23   DB Schema Create Table

```
function __dbschema_create_table($tablespec)
```

This function returns the SQL needed to create the table defined in the tablespec argument

- @tablespec => the specification for the create table, see the dbschema file to understand the tablespec structure

This function creates the table, supports the primary key, supports the foreign key, and detect fulltext indexes with mroonga engines

4.11.24   DB Schema Alter Table

```
function __dbschema_alter_table($orig, $dest)
```

This function returns the alter table command

- @orig => source table
- @dest => destination table

4.11.25   DB Schema Insert From Select

```
function __dbschema_insert_from_select($dest, $orig)
```

This function returns the insert from select command

- @orig => source table
- @dest => destination table

4.11.26   DB Schema Drop Table

```
function __dbschema_drop_table($table)
```

This function returns the drop table command

- @table => table that you want to drop

4.11.27   DB Schema Create Index

```
function __dbschema_create_index($indexspec)
```

This function returns the SQL needed to create the index defined in the indexspec argument

- @indexspec => the specification for the create index, see the dbschema file to understand the indexspec structure

This function creates the index, supports fulltext indexes

4.11.28   DB Schema Drop Index

```
function __dbschema_drop_index($index, $table)
```

This function returns the drop index command

- @index => index that you want to drop
- @table => table where the indes is part of

4.12   Dependencies feature

```
api/php/lib/depend.php
```

This file contains the code that has the ability to identify dependencies

4.12.1   Check Dependencies

```
function check_dependencies($app, $id)
```

This function allow to get all dependencies of an app and id, intended to check the if the sytem can delete some register or if the register has relations with other apps or tables

4.13   Export helper module

```
api/php/lib/export.php
```

This fie contains useful functions related to export data, allow to generate outputs in formats suck as excel, csv, edi, json and xml

4.13.1   Export File main function

```
function export_file($args)
```

This function is intended to export data in the supported formats

- @type => can be xml, csv, xls, xlsx, edi or json
- @data => the matrix to export
- @sep => separator char used only by csv format
- @eol => enf of line char used by csv and xml format
- @encoding => charset used by csv and xml format
- @replace => array with two elements, from and to, used to do replacements of the matrix values

- @escape => array with two elements, char and mode, used to specify the escape character and the escape mode
- @title => title used only by excel format
- @file => local filename used to store the results
- @ext => extension used for the filename if provided
- @wrap => boolean argument used only for edi indentation
- @indent => boolean argument used only for json indentation
- @prefn => function executed between the load and the tree construction
- @notree => boolean to enable or disable the tree2array feature
- @postfn => function executed after the tree construction
- @novoid => boolean to enable or disable the removevoid feature

If file argument is specified, void string is returned If file argument is not specified, then they will returns all data

4.13.2 Export File XML

```
function __export_file_xml($matrix, $eol = "\n", $encoding = 'UTF-8')
```

This function is intended to export data in xml format

- @matrix => the matrix to export
- @eol => enf of line char
- @encoding => charset used

They will returns all data

4.13.3 Export File CSV

```
function __export_file_csv(
```

This function is intended to export data in csv format

- @matrix => the matrix to export
- @sep => separator char
- @eol => enf of line char
- @encoding => charset used
- @replace => array with two elements, from and to, used to do replacements of the matrix values
- @escape => array with two elements, char and mode, used to specify the escape character and the escape mode

They will returns all data

4.13.4 Export File Excel

```
function __export_file_excel($matrix, $title = '', $type = 'Xlsx')
```

This function is intended to export data in excel format

- @matrix => the matrix to export
- @title => title used in the excel file
- @type => can be Xls or Xlsx

They will returns all data

4.13.5 Export File JSON

```
function __export_file_json($matrix, $indent = false)
```

This function is intended to export data in json format

- @matrix => the matrix to export
- @indent => boolean argument to enable or disable the indent feature

They will returns all data

4.13.6 Tree to Array

```
function __export_tree2array($array)
```

This function convert a tree into a 2D matrix, it's intended to convert a tree structure into a csv, for example

- @array => the tree that you want to convert into a 2D matrix

4.13.7 Get Keys

```
function __export_getkeys($array)
```

This function tries to return an array with all the keys used internally in the tree

- @array => the tree array that you want to process

4.14 Files module

```
api/php/lib/files.php
```

This file provide some usefull functions for the files module

4.14.1 Check Files Old

```
function check_files_old($app, $action, $id = null)
```

This function returns true or false and is an utility to know if the ui must to shown the needed widgets related with the old files

- @app => app that you want to use

- @action => action that you want to do (create, view, edit)

- @id => register of the app that must contain files

4.14.2 Check Files New

This function returns true or false and is an utility to know if the ui must to shown the needed widgets related with the new files

- @app => app that you want to use

- @action => action that you want to do (create, view, edit)

4.14.3 Get cid

```
function files_cid($app, $id, $cid)
```

This function returns the requested attachment indentified by the cid argument

- @id => id of the email

- @cid => the cid of the content requested

4.14.4 Get viewpdf

```
function files_viewpdf($app, $id, $cid)
```

This function returns the requested attachment indentified by the cid argument in a pdf format for the viewpdf widget

- @id => id of the email

- @cid => the cid of the content requested

4.14.5 Get download

```
function files_download($app, $id, $cid)
```

This function returns the requested attachment indentified by the cid argument in an array format for the download feature

- @id => id of the email

- @cid => the cid of the content requested

4.15 Garbage collector helper module

```
api/php/lib/gc.php
```

This fie contains useful functions related to the garbaging unused resources, currently only implements the clear of temporary files

4.15.1 Garbage Collector Executor

```
function gc_exec()
```

This function tries to clean the directories of old files, the parameters that this function uses are defined in the config file, the timeout is getted from the server/cachetimeout config file key, too is able to detect hidden files and remove except the special files as current directory, parent directory and htaccess file

4.16 GD utils helper module

```
api/php/lib/gdlib.php
```

This fie contains useful functions related to the GD library

4.16.1 Compute width

```
function compute_width($text, $size)
```

This function uses the GD library to compute the width of a text, can contains newlines and the returned value is the width of the bounring box required to print the text using the font and size specified

- @text => the text that you want to compute the width
- @size => the size used in the render process

Notes:

As the default font in saltos is the Atkinson Hyperlegible, in this function was set as default font and you can not replace at the moment

4.16.2 Image resize

```
function image_resize($data, $size)
```

This function is a helper for the html functions, and is intended to get images less than 1000x1000 pixels, to do it, maintain the width and height relation, the main idea is to get images scaled less that the size parameter

- @data => the data of the image
- @size => the size used in the control (size x size)

4.17 GeoIP helper module

```
api/php/lib/geoip.php
```

This file contain useful geoip helper functions

4.17.1 Get GeoIP information

```
function get_geoip_array($remote_addr)
```

This function gets the geoip information an returns it

4.17.2 Get GeoIP information

```
function get_geoip_string($remote_addr)
```

This function gets the geoip information an returns it

4.18 Help feature

```
api/php/lib/help.php
```

This file contains the help functions used by SaltOS

4.18.1 Detect Help File

```
function detect_help_file($app, $lang)
```

This function is intended to return the name of the pdf file used as help for the app and lang, they use 4 checks to search the correct file that must to returns, the first in to search in the app folder for the specified lang, otherwise search for some othe lang, and if no app file is found, then the same process is used for the notfound.pdf file.

- @app => the application to search
- @lang => the prefered lang to search

4.19 Import file helper module

```
api/php/lib/import.php
```

This fie contains useful functions related to import contents using differents formats suck as excel, csv, edi, json, xml and bytes. Too this module allow to manipulate data using the tree array of the core of the SaltOS, allowing to add, modify and remove nodes, too can apply patch of the memory data and paint arrays as ascii tables

4.19.1 Import File main function

```
function import_file($args)
```

This function is intended to import data in the supported formats

- @data => contents used as data instead of file
- @file => local filename used to load the data
- @type => can be xml, csv, xls, bytes, edi or json
- @sep => separator char used only by csv format
- @sheet => sheet that must to be read
- @map => map used as dictionary for each field, pos and length
- @offset => the offset added to the start position in each map field
- @nomb => boolean to disable or enable the multibyte support

- @novoid => boolean to enable or disable the removevoid feature

- @prefn => function executed between the load and the tree construction

- @notree => boolean to enable or disable the array2tree feature

- @nodes => an array with the fields that define each nodes used in the tree construction

- @nohead => if the first row doesn't contains the header of the data, put this field to one

- @noletter => if you want to use numeric index instead of excel index, put this field to one

- @postfn => function executed after the tree construction

This function returns an array with the loaded data from file Can return a matrix or tree, depending the nodes parameter

4.19.2 UTF8 BOM helper

```
function __import_utf8bom($data)
```

This function remove the bom header of the string

- @data => the data that must to be checked

Returns the data without the bom characters

4.19.3 XML to Array

```
function __import_xml2array($file)
```

This function convert an xml into an array

- @file => the file that contains the xml

Returns an array with the contents of the xml

4.19.4 Special Chars helper

```
function __import_specialchars($arg)
```

This function is a helper used by the csv2array function

- @arg => a string or array

Returns the input with the expected replacements

4.19.5 CSV to Array helper

```
function __import_csv2array($file, $sep)
```

This function is a helper of the __import_xml2array

- @file => the filename and the sheet that do you want to retrieve

- @sep => the separator field used in the csv file

Returns a matrix with the contents

4.19.6 XLS to Array helper

```
function __import_xls2array($file, $sheet)
```

This fuction can convert an excel file into a matrix structure, it has some additional features as:

- If the file exceds the 1Mbyte and the server has the xlsx2csv executable, it tries to convert the xslx to an excel to use less memory

- Do some internals trics to solve some knowed issues

- @file => the filename and the sheet that do you want to retrieve

- @sheet => the second parameter can be a number or a sheet name

Returns a matrix with the contents

4.19.7 Bytes to Array helper

```
function __import_bytes2array($file, $map, $offset, $nomb)
```

This function can read files as blocks of bytes, they can use a map, can specify an offset and can be used using multibyte if it is needed

- @file => local filename used to load the data

- @map => map used as dictionary for each field, pos and length

- @offset => the offset added to the start position in each map field

- @nomb => boolean to disable or enable the multibyte support

Returns a matrix with the contents

Notes:

The map must be an array of strings of the follow form: ["field1;0;10", "field2;10;20", "field3;20;40"]

4.19.8 Edi to Array helper

```
function __import_edi2array($file)
```

This fuction can convert an edi file into a tree structure

- @file => local filename used to load the data

4.19.9 JSON to Array helper

```
function __import_json2array($file)
```

This fuction can convert an excel file into a tree structure

- @file => local filename used to load the data

4.19.10 Remove Void helper

```
function __import_removevoid($array)
```

This function is able to remove an entire row or column if it is void

- @array => the array to fix

4.19.11   Array to Tree helper

```
function __import_array2tree($array, $nodes, $nohead, $noletter)
```

This function tries to convert the array into a tree using the nodes, specification

- @array => the matrix that you want to convert into a tree
- @nodes => the dictionary used to the conversion, must to be an array with the fields used by each node, for example ["A,B,C","D,E,F"]
- @nohead => set it to true to prevent the usage of the first row of the matrix as header, this option uses the letter as id of each element of the tree
- @noletter => set it to true to prevent the usage of letters, if the previous option is set to true

4.19.12   Array Intersect

```
function __import_array_intersect($data, $filter)
```

This function returns the same result that array_intersect_key($data,array_flip($filter)) maintaining the order of the filter array.

- @data => the array that you want to apply the filter
- @filter => the array where obtain the keys to apply the filter

4.19.13   Array to Tree Set helper

```
function __import_array2tree_set(&$result, $parts)
```

This function tries to set values in a tree structure, to do it, it uses the parts array that contains a list of paired keys and values used to move by the tree setting the values of each pair of key val

- @result => the array where do you want to put the parts
- @parts => an array with pairs of key val

4.19.14   Array to Tree Clean helper

```
function __import_array2tree_clean($array)
```

This function tries to clean the tree by setting an automatic indexes

- @array => the array to clean

4.19.15 Column to Name helper

```
function __import_col2name($n)
```

This function returns the name of the column from the position n

- @n => the position number

Notes:

This function was copied from:

- http://www.php.net/manual/en/function.base-convert.php#94874

4.19.16 Name to Column helper

```
function __import_name2col($a)
```

This function returns the position number of the column from the name

- @a => the column name

Notes:

This function was copied from:

- http://www.php.net/manual/en/function.base-convert.php#94874

4.19.17 Is Name helper

```
function __import_isname($name)
```

This function returns true if the name argument contains only valid letters used in the name of the column

- @name => the name that you want to check

4.19.18 Cell to Column and Row helper

```
function __import_cell2colrow($cell)
```

This function extract the column part and the row part from a cell name

- @cell => the cell that you want to process

Notes:

This function tries to retusn an array with two elements, for example, for the cell AX23, the function returns [AX,23]

4.20 Make index helper module

```
api/php/lib/indexing.php
```

This fie contains useful functions related to the indexing feature that internally uses the mroonga engine to search in the fulltext string generated by this feature

4.20.1 Make Index main function

```
function make_index($app, $reg_id)
```

This function implements the make index feature of SaltOS, this consists in a concatenation of fields and subqueries to retrieve all data related to the tables involved in the desired application and the register reg_id

- @app => code of the application that you want to index
- @reg_id => register of the app that you want to index

Notes:

This function returns an integer as response about the index action:

- +1 => insert executed, this is because the app register exists and the index register not exists
- +2 => update executed, this is because the app register exists and the index register too exists
- +3 => delete executed, this is because the app register not exists and the index register exists
- -1 => app not found, this is because the app requested not have a table in the apps config
- -2 => index table not found, this is because the has_index feature is disabled by dbstatic
- -3 => data not found, this is because the app register not exists and the indexting register not exists

As you can see, negative values denotes an error and positive values denotes a successfully situation

4.20.2 Make Index helper

```
function __make_index_helper($table, $id = '')
```

This function allow the make_index to retrieve all data of the fiels and all data of the related fields of the related tables, this is done by using the fkey information of the dbschema, this function uses some features of the dbschema functions to get the fields, types, fkeys and too, the dbstatic information of the app table

This function uses a cache technique to improve the performance, returns an array with all fields and subqueries to allow to retrieve all data related to the app register

4.20.3 Get Field helper

```
function __get_field_helper($table)
```

This function return the field associated to the table in the dbstatic file and associated to the apps table

- @table => the table of the dbstatic that want to convert to field

Notes:

This function uses the dbschema.php library

4.20.4 Get Fkeys helper

```
function __get_fkeys_helper($table)
```

This function returns the fkeys from the DB Schema file

- @table => the table that you want to request the fkeys

Notes:

This function uses the dbschema.php library

4.20.5 Indexing Files

```
function indexing_files()
```

This function tries to index al unindexed files, and to do it tries to search the pending files and tries to index using the unoconv2txt, this function needs a file and returns all contents in a text format, to protect this function of posible fails, the system first increment the retry field and then tries to index and update the file register, if some thing is wrong in the index time the script must unexpectedly finish and in the next executions can continue in the same file, but only for three times, the requirement to index a file is that indexed = 0 and retries < 3

4.20.6 Indexing Apps

```
function indexing_apps()
```

This function tries to execute some periodic task intended to fix issues with the indexing relationships, to do it tries to search not found registers in the indexing table in the first loop and tries to search not found registers in the app table in the second loop, with all found not found registers the function executes the make_index that add or remove the needed index register to maintain the integrity with the indexing feature.

4.21 Log helper module

```
api/php/lib/log.php
```

This fie contains useful functions related to the log feature

4.21.1 Make Log function

```
function make_log($app, $log, $reg_ids, $extra_ids = '')
```

This function adds a log register to the associated log table for each application.

- @app => code of the application where you want to add the log
- @log => the log message that you want to add to the log register
- @reg_id => register ids of the app where you want to add the log
- @extra_id => extra ids of the app where you want to add the log

Notes:

This function returns an integer as response about the control action:

- +1 => insert executed, this is because the app register exists and the control register not exists
- -1 => app not found, this is because the app requested not have a table in the apps config
- -2 => log table not found, this is because the has_log feature is disabled by dbstatic

4.21.2 Make Log Bypass function

```
function make_log_bypass($app, $log, $data)
```

This function is intended to be used as wrapper between the caller and the execute_query or execute_query_array function, the main idea is to do the same that make_log but uses the reg_id from the array data, and this array can be an array with the contents of one register with an id field, or an array of rows where each item must contain an id field

- @app => code of the application where you want to add the log

- @data => data with the register or registers of the app where you want to add the log, remember that an id field is needed

- @log => the log message that you want to add to the log register

Notes:

This function always returns the input data

4.21.3 Get Logs

```
function get_logs($app, $reg_id)
```

This function allow to get all logs associated to an app and reg_id, to do it, the app must to have enabled the has_log feature, the results will be tunned in the query to disallow the zeroes in some fields that are replaced by void strings, too implement the search in the reg_id or inside the reg_ids using the FIND_IN_SET feature available as native feature in MariaDB and added by SaltOS to the sqlite drivers thansk to the createFunction feature that allow to define sql commands from php functions

- @app => code of the application that you want to get the logs

- @reg_id => register of the app that you want to get the logs

4.21.4 Delete Log function

```
function del_log($app, $reg_id)
```

This function allow to delete the last log to a reg_id of an app, to do it, the function requires to specify the app and reg_id

- @app => code of the application that you want to delete the last log

- @reg_id => register of the app that you want to delete the last log

Notes:

This function returns an integer as response about the control action:

- +1 => delete executed, this is because the app register exists and they can delete the last register

- -1 => app not found, this is because the app requested not have a table in the apps config

- -2 => log table not found, this is because the has_version feature is disabled by dbstatic

- -3 => data not found, this is because the app register not exists and the version register not exists

As you can see, negative values denotes an error and positive values denotes a successfully situation

4.22   Math utils helper module

```
api/php/lib/math.php
```

This fie contains useful functions related to math operations

4.22.1   Sign function

```
function sign($n)
```

This function returns 1 for positive, -1 for negative and 0 for zero.

- @n => the number that you want to be processed

4.22.2   Is Prime Number

```
function is_prime($num)
```

This function is a detector of prime numbers, uses some optimizations and ideas from www.polprimos.com, please, see the previous web to understand the speedup of this function in the prime numbers validation

- @num => the number that you want to check if it is a primer numner

Notes:

See www.polprimos.com for understand this algorithm

4.23   Notes module

```
api/php/lib/notes.php
```

This file provide some usefull functions for the notes module

4.23.1   Check Notes Old

```
function check_notes_old($app, $action, $id = null)
```

This function returns true or false and is an utility to know if the ui must to shown the needed widgets related with the old notes

- @app => app that you want to use
- @action => action that you want to do (create, view, edit)
- @id => register of the app that must contain notes

4.23.2   Check Notes New

```
function check_notes_new($app, $action)
```

This function returns true or false and is an utility to know if the ui must to shown the needed widgets related with the new notes

- @app => app that you want to use
- @action => action that you want to do (create, view, edit)

4.24 Password helper module

```
api/php/lib/password.php
```

This fie contains useful functions related to password, currently only publish one feature to check the password strength, but is open to be used to add more password features if it is needed

4.24.1 Password Strength

```
function password_strength($pass)
```

This fucntion returns a number between 0 and 100 that try to categorize the quality of the pass checked, this is useful to known if the new password is a good option or maybe is needed to request another new password

- @pass => password that do you want to check

4.25 PDF helper module

```
api/php/lib/pdf.php
```

This file contains useful functions related to PDF generation using TCPDF library including custom PDF class extensions and various PDF manipulation utilities

4.25.1 Custom PDF class extending TCPDF

```
class MyPDF extends TCPDF
```

Provides enhanced functionality for header/footer management and page checks

4.25.2 Initialize PDF document settings

```
public function Init()
```

Resets all internal variables to their default state

4.25.3 Set header content and data

```
public function set_header($arr, $row)
```

- @arr => Header template array
- @row => Data row for header evaluation

4.25.4 Set footer content and data

```
public function set_footer($arr, $row)
```

- @arr => Footer template array
- @row => Data row for footer evaluation

4.25.5 Override TCPDF header method

```
public function Header()
```

Processes and renders the header content while temporarily disabling Y checks

4.25.6 Override TCPDF footer method

```
public function Footer()
```

Tracks page numbers where footers need to be rendered

4.25.7 Render all accumulated footers

```
public function render_footers()
```

Processes footer content on all tracked pages while temporarily disabling Y checks

4.25.8 Check Y position and add new page if needed

```
public function check_y($offset = 0)
```

- @offset => Additional offset to consider in Y position check

4.25.9 Evaluate dynamic value in PDF context

```
function __pdf_eval_value($input, $row, $pdf)
```

- @input => Expression to evaluate
- @row => Data row for variable substitution
- @pdf => PDF object reference

Returns the evaluated result

4.25.10 Evaluate array values in PDF context

```
function __pdf_eval_array($array, $row, $pdf)
```

- @array => Input array with expressions
- @row => Data row for variable substitution

- @pdf => PDF object reference

Returns array with evaluated values

4.25.11 Advanced string explosion with quote and parentheses awareness

```
function __pdf_eval_explode($separator, $str, $limit = 0)
```

- @separator => Delimiter character
- @str => Input string to explode
- @limit => Maximum number of splits

Returns array of exploded parts

4.25.12 Process PDF template tags and generate PDF content

```
function __pdf_eval_pdftag($array, $row = [])
```

- @array => PDF template definition array
- @row => Data row for template evaluation

Returns the generated PDF object or output array

4.25.13 Generate PDF from template file

```
function pdf($file, $row = [])
```

- @file => Path to PDF template file
- @row => Data row for template evaluation

Returns array containing PDF name, type and base64 encoded data

4.25.14 Convert various file types to PDF

```
function __pdf_all2pdf($input)
```

- @input => Path to input file to convert

Returns PDF content as string

4.25.15 Build the expected PDF template file path for the given app.

```
function detect_pdf_file($app)
```

Given an application ID, this function resolves the folder name using 'detect_app_folder($app)' and returns the full relative path to its associated PDF XML definition file, expected at:

apps/<folder>/xml/<app>_pdf.xml

This path is used to define the PDF layout when exporting records from the app.

- @app => The application code (e.g., 'invoices', 'quotes')

Return the relative path to the PDF XML file

4.25.16 Check if the PDF layout file exists for the given app.

```
function exists_pdf_file($app)
```

This function uses 'detect_pdf_file()' to resolve the expected path of the XML file used to generate PDF output for a given app, and returns whether the file actually exists.

- @app => The application code (e.g., 'invoices', 'quotes')

Return true if the PDF file exists, false otherwise

4.26 Push utils helper module

```
api/php/lib/push.php
```

This fie contains useful functions related to push feature

4.26.1 Push insert

```
function push_insert($type, $message)
```

This function adds an entry to the push system using the type and message

- @type => the type (one of this: success, danger or event)
- @message => the desired message that you want to put in the queue

4.26.2 Push select

```
function push_select($timestamp)
```

This function returns the push data found after the timestamp used

- @timestamp => the timestamp used to begin the search

Notes:

- This function returns the entries found without repetitions, to be usefull, only uses the last entries removing the repeated entries and using only the type and message to detect repetitions

4.27 QRCode helper module

```
api/php/lib/qrcode.php
```

This fie contains useful functions related to QRCodes

4.27.1   QRCode image function

```
function __qrcode_image($msg, $s, $m, $l)
```

This function allow to generate a qrcode with the SaltOS logo embedded in the center of the image, you can pass the desired message that you want to convert in qrcode and it returns an image with the data

- @msg => Contents of the qrcode
- @s => size of each pixel used in the qrcode
- @m => margin of the qrcode (white area that that surround the qrcode)
- @l => level error correction: L (low), M (medium), Q (better), H (best)

Notes:

The normal behavior is returns a png image, but if something was wrong, the function can returns an empty string

4.27.2   QRCode image function

```
function __qrcode_image_png($msg, $s, $l)
```

This function allow to generate a qrcode with the SaltOS logo embedded in the center of the image, you can pass the desired message that you want to convert in qrcode and it returns an image with the data

- @msg => Contents of the qrcode
- @s => size of each pixel used in the qrcode
- @l => level error correction: L (low), M (medium), Q (better), H (best)

Notes:

The normal behavior is returns a png image without margins, but if something was wrong, the function can returns an empty string

4.27.3   QRCode image function

```
function __qrcode_image_svg($msg, $s, $l)
```

This function allow to generate a qrcode with the SaltOS logo embedded in the center of the image, you can pass the desired message that you want to convert in qrcode and it returns an image with the data

- @msg => Contents of the qrcode
- @s => size of each pixel used in the qrcode
- @l => level error correction: L (low), M (medium), Q (better), H (best)

Notes:

The normal behavior is returns a svg image without margins, but if something was wrong, the function can returns an empty string

4.28   Score image helper module

```
api/php/lib/score.php
```

This fie contains useful functions related to score images

4.28.1 Score Image function

```
function __score_image($score, $width, $height, $size)
```

This function generates an image with a gradient from red to yellos and then, to green, depending of the score passed to the function, the params allos to define the size of the image or the size of the font used to write the score percent

- @score => a number between 0 and 100
- @width => the width of the generated image
- @height => the height of the generated image
- @size => the size of the font of the generated image

4.29 Security helper module

```
api/php/lib/security.php
```

This file contain useful securiry helper functions

4.29.1 Get Browser Platform Device Type

```
function get_connection_detected($remote_addr, $user_agent = null)
```

This function gets the browser, platform and device_type form the user_agent header

4.30 Setup helper module

```
api/php/lib/setup.php
```

This file contains useful functions related to the setup process

4.30.1 Setup function

```
function setup()
```

This function allow to create the minimal user and group information to do SaltOS usable by the admin user, only do things if the keys tables of the array are void, too is able to maintain the integrity of the tbl_{users,groups}_apps_perms tables by removing the unused registers, usefull when you modify the tbl_apps_perms rows

4.30.2 Load and initialize sample data for all apps in the given directory.

```
function __setup_helper($dir)
```

This function is used during the setup or development phase to populate the database with sample data stored in '.sql.gz' files under 'apps/<dir>/sample/sql/'. For each file:

- It infers the corresponding table and app.

- If the table is empty, it loads the data from the SQL file.

- It then generates control/version/index/log metadata for each inserted record.

- It ensures that subtable and main table mappings are respected.

The function returns timing information and the number of records processed per app, which can be used for diagnostics or logging.

- @dir => The directory under 'apps/' (e.g., "crm", "sales", ...)

Return an associative array with total execution time and per-app counts

4.31 Send file to trash

```
api/php/lib/trash.php
```

This function tries to implement the send to trash feature, to do it, move the requested files to the trash folder, execute the insert in the tbl_trash with the new file and delete the file from the files app table.

4.31.1 Garbage Collector Trash

```
function gc_trash()
```

This function tries to clean the trash database of old files, the parameters that this function uses is defined in the config file, only uses the timeout that is getted from the server/trashtimeout

4.32 Unoconv library

```
api/php/lib/unoconv.php
```

This file contains all functions that allow conversions between formats like docx, xlsx and more to pdf, too includes the ocr code that allow to get text from images.

4.32.1 Unoconv to PDF

```
function unoconv2pdf($input)
```

This function allow to convert all input files into their equivalent pdf file

- @input => the file that you want to process

4.32.2 Unoconv to TXT

```
function unoconv2txt($input)
```

This function allow to convert all input files into their equivalent txt file

- @input => the file that you want to process

4.32.3 Unoconv list

```
function __unoconv_list()
```

This function returns an array with all suported extensions by libreoffice

4.32.4 PDF to TXT

```
function __unoconv_pdf2txt($input, $output)
```

This function convert files between pdf to txt using the pdftotext

- @input => the file that you want to process
- @output => the file where you want to store the result

4.32.5 All to PDF

```
function __unoconv_all2pdf($input, $output)
```

This function convert all formats to pdf using libreoffice

- @input => the file that you want to process
- @output => the file where you want to store the result

4.32.6 Convert

```
function __unoconv_convert($input, $output, $format)
```

This function convert between formats using libreoffice

- @input => the file that you want to process
- @output => the file where you want to store the result
- @format => the desired destination format

4.32.7 Image to OCR

```
function __unoconv_img2ocr($file)
```

This file uses tesseract to extract all text from the file, if the file is not a tiff image, then is converted to a tiff to be used as input in the tesseract process.

- @file => the file that you want to process

4.32.8 PDF to OCR

```
function __unoconv_pdf2ocr($pdf)
```

This function uses the pdftoppm command to generate one image per page, and then, extract the text of each page to finish the task.

- @pdf => the file that you want to process

4.32.9 Calculate histogram value

```
function __unoconv_histogram($values, $usage1, $usage2)
```

This function calculates a representative value from a histogram based on given usage thresholds. It finds the highest percentage where at least a certain portion of values and unique values are included.

- @values => array of values to analyze
- @usage1 => minimum percentage of total values to include (0-1)
- @usage2 => minimum percentage of unique values to include (0-1)

Returns the calculated representative value

4.32.10 Rotate coordinates

```
function __unoconv_rotate($posx, $posy, $angle)
```

This function rotates a point around the origin by a given angle in degrees.

- @posx => x coordinate of the point
- @posy => y coordinate of the point
- @angle => rotation angle in degrees

Returns the array with new x and y coordinates

4.32.11 Extract attributes from OCR node

```
function __unoconv_node2attr($node)
```

This function processes a node from OCR output to extract its attributes, specifically focusing on the bounding box information.

- @node => the OCR node to process

Returns the array containing node ID and bounding box coordinates

4.32.12 Extract text value from OCR node

```
function __unoconv_node2value($node)
```

This function extracts the text content from an OCR node, handling nested arrays.

- @node => the OCR node to process

Returns the extracted text content

4.32.13   Convert OCR lines to character matrix

```
function __unoconv_lines2matrix($lines, $width, $height)
```

This function converts OCR-detected lines and words into a 2D character matrix for text reconstruction and analysis.

- @lines => array of OCR-detected lines and words
- @width => width divisor for coordinate normalization
- @height => height divisor for coordinate normalization

Returns the 2D character matrix or index of problematic line if error occurs

4.32.14   Reorder line coordinates

```
function __unoconv_fixline($line, $pos1, $pos2, $pos3, $pos4)
```

This function reorders the coordinates of a line based on specified positions, used for correcting orientation in OCR results.

- @line => original line coordinates
- @pos1 => target position for first coordinate
- @pos2 => target position for second coordinate
- @pos3 => target position for third coordinate
- @pos4 => target position for fourth coordinate

Returns the reordered line coordinates

4.32.15   Convert HOCR to plain text

```
function __unoconv_hocr2txt($hocr)
```

This function processes HOCR (HTML OCR) output to extract and reconstruct the text content while maintaining spatial relationships.

- @hocr => HOCR content to process

Returns the extracted plain text

4.32.16   Proportional substring extraction

```
function __unoconv_substr($string, $start, $length, $reference)
```

This function extracts a substring based on proportional positions relative to a reference length, useful for working with scaled text representations.

- @string => input string to extract from
- @start => starting position (relative to reference)
- @length => length to extract (relative to reference)
- @reference => reference length for proportional calculation

Returns the extracted substring

4.32.17 2D proportional substring extraction

```
function __unoconv_substr2d($page, $x1, $x2, $x3, $y1, $y2, $y3)
```

This function extracts a 2D region from a text page based on proportional coordinates, maintaining spatial relationships in the extracted content.

- @page => array of text lines representing the page
- @x1 => starting x position (relative to x3)
- @x2 => width to extract (relative to x3)
- @x3 => reference width for x coordinates
- @y1 => starting y position (relative to y3)
- @y2 => height to extract (relative to y3)
- @y3 => reference height for y coordinates

Returns the array of extracted lines

4.32.18 Remove margins from text page

```
function __unoconv_remove_margins($page)
```

This function trims empty margins from a text page, removing leading/trailing whitespace and empty lines from the top and bottom.

- @page => text content to process (multiple lines separated by newlines)

Returns the text content with margins removed

4.33 Add upload file

```
api/php/lib/upload.php
```

This function is intended to add files to the upload mecano

- @val => is an associative array with the follow entries:
- @id => the uniq id used to identify the file in the uploads system
- @name => local name of the file
- @size => size of the contents of the file
- @type => mime type of the file

- @data => the inline data in base64 form, with the mime type prefix

4.33.1 Del upload file

```
function del_upload_file($val)
```

This function is intended to del files in the upload mecano

- @val => is an associative array with the follow entries:
- @id => the uniq id used to identify the file in the uploads system
- @name => local name of the file
- @size => size of the contents of the file
- @type => mime type of the file
- @file => the inline data in base64 form, with the mime type prefix
- @hash => the hash of the binary data contents

4.33.2 Check upload file

```
function check_upload_file($val)
```

This function is intended to check files in the upload mecano

- @val => is an associative array with the follow entries:
- @user_id => the user id intended to identify the property of the file
- @uniqid => the uniq id used to identify the file in the uploads system
- @name => local name of the file
- @size => size of the contents of the file
- @type => mime type of the file
- @file => the inline data in base64 form, with the mime type prefix
- @hash => the hash of the binary data contents

4.33.3 Rename upload file

```
function rename_upload_file($val, $app, $id)
```

This function copy the file contained in the val parameter to the app folder and create the files register using the id argument

- @val => the array with the file data
- @app => the app code that identify the application
- @id => the id of the application register

4.33.4 Garbage Collector Upload

```
function gc_upload()
```

This function tries to clean the upload database of old files, the parameters that this function uses is defined in the config file, only uses the timeout that is getted from the server/cachetimeout

4.34   Version helper module

```
api/php/lib/version.php
```

This fie contains useful functions related to the control and version system, they allow to relationate registers with users and groups, and to add and retrieve the versions of a register

4.34.1   Make Version function

```
function make_version($app, $reg_id)
```

This function allow to add a new version to a reg_id of an app, to do it, the function requires to specify the app, reg_id, the original data and the new data to compute the diff patch that must to be stored in the data field and create the register for the new version

To do this, the function validate the input data, checks the existence of registers of data and versions, prepare the data patch to store, get the old hash to do the blockchain, get the last ver_id and compute all needed things to do the insert of the new version register

- @app => code of the application that you want to add a new version
- @reg_id => register of the app that you want to add a new version

Notes:

This function returns an integer as response about the control action:

- +1 => insert executed, this is because the app register exists and they can add a new version register
- -1 => app not found, this is because the app requested not have a table in the apps config
- -2 => version table not found, this is because the has_version feature is disabled by dbstatic
- -3 => data not found and version not found, app register not exists and version register not exists
- -4 => data not found but version found, app register not exists and version register exists

As you can see, negative values denotes an error and positive values denotes a successfully situation

4.34.2   Get Version

```
function get_version($app, $reg_id, $ver_id = null)
```

This function allow to get an specific version of a register and app, intended to get the data used in a specific version to compare with other versions and to restore data to the requested version

- @app => code of the application that you want to add a new version
- @reg_id => register of the app that you want to add a new version
- @ver_id => the version that you want to get

Notes:

This function is not a simple select of the register that matches with the ver_id requested, it does an accumulative merge to get the register data in the moment where the version will be stored, to do it, they must to restore versions from 1 to ver_id, and must to discard the next versions

This function returns an array or an integer as response about the control action:

- -1 => app not found, this is because the app requested not have a table in the apps config

- -2 => version table not found, this is because the has_version feature is disabled by dbstatic

- -3 => data not found, this is because the version requested not exists

As you can see, negative values denotes an error and positive values denotes a successfully situation

4.34.3   Delete Version function

```
function del_version($app, $reg_id)
```

This function allow to delete the last version to a reg_id of an app, to do it, the function requires to specify the app and reg_id

- @app => code of the application that you want to add a new version

- @reg_id => register of the app that you want to add a new version

Notes:

This function returns an integer as response about the control action:

- +1 => delete executed, this is because the app register exists and they can delete the last register

- -1 => app not found, this is because the app requested not have a table in the apps config

- -2 => version table not found, this is because the has_version feature is disabled by dbstatic

- -3 => data not found, this is because the app register not exists and the version register not exists

As you can see, negative values denotes an error and positive values denotes a successfully situation