

Applications Documentation

SaltOS 4.0 r1681

January 2025

Contents

1	Common	6
1.1	Matrix functions	6
1.1.1	TODO	6
2	Dashboard	6
2.1	Dashboard application	6
2.1.1	Main object	6
2.1.2	TODO	6
2.1.3	TODO	6
3	Emails	6
3.1	Email application	6
3.1.1	Driver emails object	7
3.1.2	TODO	7
3.1.3	TODO	7
3.2	Email application	7
3.2.1	Main object	7
3.2.2	TODO	7
3.2.3	TODO	7
3.2.4	TODO	7
3.2.5	TODO	8
3.2.6	TODO	8
3.2.7	TODO	8
3.2.8	TODO	8
3.2.9	TODO	8
3.2.10	TODO	8
3.3	TODO	8
3.3.1	TODO	8
3.4	Get email library	9
3.4.1	Requires section	9
3.4.2	Defines section	9
3.4.3	Remove all body	9

3.4.4	Process message	9
3.4.5	Process plain html	9
3.4.6	Process file	9
3.4.7	Check permissions	10
3.4.8	Get source	10
3.4.9	Mime decode protected	10
3.4.10	Get mime	10
3.4.11	Get Node	10
3.4.12	Get type	11
3.4.13	Get disposition	11
3.4.14	Get files	11
3.4.15	Get human size	11
3.4.16	Get info	11
3.4.17	Fix string	11
3.4.18	Get text body	12
3.4.19	Get full body	12
3.4.20	Get cid	12
3.4.21	Insert	12
3.4.22	Update	12
3.4.23	Add bcc	13
3.4.24	Gzfile size	13
3.4.25	Get email body	13
3.4.26	TODO	13
3.4.27	TODO	13
3.4.28	Get email source	14
3.4.29	Get email files	14
3.4.30	Get cid	14
3.4.31	Is outbox	14
3.4.32	Server	14
3.4.33	Delete	14
3.4.34	Get viewpdf	15
3.4.35	Get download	15
3.4.36	TODO	15

3.4.37	TODO	15
3.4.38	TODO	15
3.5	Make indexing action	15
3.6	Send email library	15
3.6.1	Used libraries	16
3.6.2	Sendmail	16
3.6.3	Debugoutput helper	16
3.6.4	Parser	16
3.6.5	Message Id	16
3.6.6	Eml saver	16
3.6.7	Obj saver	17
3.6.8	TODO	17
3.6.9	TODO	17
3.6.10	TODO	17
3.6.11	TODO	17
3.6.12	TODO	17
4	Invoices	18
4.1	invoices application	18
4.1.1	Main object	18
4.1.2	TODO	18
4.1.3	TODO	18
4.1.4	TODO	18
4.1.5	TODO	18
4.1.6	TODO	18
4.1.7	TODO	18
5	Tester	19
5.1	Tester application	19
5.1.1	Tester object	19
5.1.2	Campo 8	19
5.1.3	Campo 9	19
5.1.4	Campo 10	19
5.1.5	Campo 11	19

5.1.6	Campo 22	19
6	Users	20
6.1	Login application	20
6.1.1	Main object	20
6.1.2	Authenticate login function	20
6.2	Days functions	20
6.2.1	Days to bin	20
6.2.2	Bin to days	20
6.2.3	Fix for days	21
6.3	Groups functions	21
6.3.1	Insert group action	21
6.3.2	Update group action	21
6.3.3	Delete group action	21
6.4	Matrix functions	21
6.4.1	Make matrix	22
6.4.2	Make matrix	22
6.4.3	Make matrix	22
6.5	Users functions	22
6.5.1	Insert user action	22
6.5.2	Update user action	22
6.5.3	Delete user action	22

1 Common

1.1 Matrix functions

```
apps/common/php/matrix.php
```

This file contain all functions needed by the excel widget

1.1.1 TODO

```
function make_matrix_version($app, $id)
```

TODO

2 Dashboard

2.1 Dashboard application

```
apps/dashboard/js/dashboard.js
```

This application implements the tipical features associated to dashboard

2.1.1 Main object

```
saltos.dashboard = {};
```

This object contains all SaltOS code

2.1.2 TODO

```
saltos.dashboard.init = arg
```

TODO

2.1.3 TODO

```
saltos.dashboard.authupdate = ()
```

TODO

3 Emails

3.1 Email application

```
apps/emails/js/driver.js
```

This application implements the tipical features associated to emails

3.1.1 Driver emails object

```
saltos.driver.__types.emails = {};
```

This object stores the functions used by the emails driver

3.1.2 TODO

```
saltos.driver.__types.emails.template = arg
```

TODO

3.1.3 TODO

```
saltos.driver.__types.emails.init = saltos.driver.__types.type5.init;
```

TODO

3.2 Email application

```
apps/emails/js/emails.js
```

This application implements the typical features associated to emails

3.2.1 Main object

```
saltos.emails = {};
```

This object contains all SaltOS code

3.2.2 TODO

```
saltos.emails.init = arg
```

TODO

3.2.3 TODO

```
saltos.emails.server = ()
```

TODO

3.2.4 TODO

```
saltos.emails.delete1 = ()
```

TODO

3.2.5 TODO

```
saltos.emails.delete2 = ()
```

TODO

3.2.6 TODO

```
saltos.emails.send = ()
```

TODO

3.2.7 TODO

```
saltos.emails.setter = what
```

TODO

3.2.8 TODO

```
saltos.emails.signature = ()
```

TODO

3.2.9 TODO

```
saltos.emails.viewpdf = ()
```

TODO

3.2.10 TODO

```
saltos.emails.download = ()
```

TODO

3.3 TODO

```
apps/emails/php/filter.php
```

TODO

3.3.1 TODO

```
function make_where_query_emails($json)
```

TODO

3.4 Get email library

```
apps/emails/php/getmail.php
```

This library provides the necessary functions to download, parse and manage emails.

3.4.1 Requires section

```
require_once 'apps/emails/lib/mimeparser/mime_parser.php';
```

This requires loads the external libraries needed to run this library.

3.4.2 Defines section

```
// phpcs:disable Generic.Files.LineLength
```

This defines allow to define some useful standards to do html pages and more.

3.4.3 Remove all body

```
function __getmail_removebody($array)
```

This function removes the body entry in the array, it is only for debug purposes

- @aarray => The array that you want to process

3.4.4 Process message

```
function __getmail_processmessage($disp, $type)
```

This function returns a boolean that identify if the disposition and the type allow the node to be processed.

- @disp => disposition, can be inline or attachment
- @type => type, can be message, html or plain

3.4.5 Process plain html

```
function __getmail_processplainhtml($disp, $type)
```

This function returns a boolean that identify if the disposition and the type allow the node to be processed.

- @disp => disposition, can be inline or attachment
- @type => type, can be message, html or plain

3.4.6 Process file

```
function __getmail_processfile($disp, $type)
```

This function returns a boolean that identify if the disposition and the type allow the node to be processed.

- @disp => disposition, can be inline or attachment
- @type => type, can be message, html or plain

3.4.7 Check permissions

```
function __getmail_checkperm($id)
```

This function allow to check if the current user has permissions to view the message identified by the id argument

- @id => id of the email

3.4.8 Get source

```
function __getmail_getsource($id, $max = 0)
```

This function returns the original RFC822 message as string

- @id => id of the email
- @max => max size that can be processed

3.4.9 Mime decode protected

```
function __getmail_mime_decode_protected($input)
```

This function decodes the input string that contains the RFC822 message using the mime_parser_class to do it, and returns the decoded array.

- @input => the RFC822 string that contains the message

3.4.10 Get mime

```
function __getmail_getmime($id)
```

This function returns the decoded array of the email identified by the id argument, to do this more optimal, this function uses an internal cache file to improve the performance for repeated executions.

- @id => id of the email

3.4.11 Get Node

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

This function returns the node using a xpath notation

- @path => xpath that identify the desired path that must to be returned
- @array => the decoded message in an array format

3.4.12 Get type

```
function __getmail_gettype($array)
```

This function tries to unify the different content-type to standardize it into the following formats: html, plain, message, alternative, multipart or other.

- @array => the decoded message in an array format

3.4.13 Get disposition

```
function __getmail_getdisposition($array)
```

This function tries to unify the different content-disposition to standardize it into the following formats: attachment, inline or other.

- @array => the decoded message in an array format

3.4.14 Get files

```
function __getmail_getfiles($array, $level = 0)
```

This function returns an array with the attachment files of the message

- @array => the decoded message in an array format
- @level => this parameter is internally used to detect recursion

3.4.15 Get human size

```
function __getmail_gethumansize($size)
```

This function returns a string containing the size in human format

- @size => the number of bytes to convert to human format

3.4.16 Get info

```
function __getmail_getinfo($array)
```

Returns all information of the decoded message in a structured format

- @array => the decoded message in an array format

3.4.17 Fix string

```
function __getmail_fixstring($arg)
```

This function is a helper used by all functions that process the headers of the decoded message.

- @arg => the string that must be checked and fixed if needed

3.4.18 Get text body

```
function __getmail_gettextbody($array, $level = 0)
```

This function returns all text body concatenated as an unique string

- @array => the decoded message in an array format
- @level => this parameter is internally used to detect recursion

3.4.19 Get full body

```
function __getmail_getfullbody($array)
```

This function returns all body and attachments information as an array

- @array => the decoded message in an array format

3.4.20 Get cid

```
function __getmail_getcid($array, $hash)
```

This function returns the requested attachment identified by the hash argument

- @array => the decoded message in an array format
- @hash => the hash of the content requested

3.4.21 Insert

```
function __getmail_insert(
```

This function do the insert in the app_emails table, and

- @file => the gzfile that contains the message in RFC822 format
- @messageid => the id of the message (account.id/uidl)
- @state.new => the 0/1 that sets the state new flag
- @state.reply => the 0/1 that sets the state reply flag
- @state.forward => the 0/1 that sets the state forward flag
- @state.wait => the 0/1 that sets the state wait flag
- @id_correo => the id of the related email (used to create relations between emails)
- @is_outbox => the 0/1 that sets the is outbox flag
- @state.sent => the 0/1 that sets the state sent flag
- @state.error => the string that contains the error (if exists an error)

3.4.22 Update

```
function __getmail_update($field, $value, $id)
```

This function updates the field with the value of the app_emails for the register identified by the id argument.

- @field => field that you want to update
- @value => value that you want to set
- @id => id of the register to do the update

3.4.23 Add bcc

```
function __getmail_add_bcc($id, $bcc)
```

This function adds the bcc to the database, this is because the messages does not contains the bcc field (is hidden in theory), and only is available if the current execution is the sender of the message.

- @id => id of the email
- @bcc => an array with the addresses of the emails

3.4.24 Gzfile size

```
function gzfilesize($filename)
```

This function is copied from <http://php.net/manual/es/function.gzread.php#110078> and allow to know the file size of the file after a gzip descompression.

- @filename => the gzip filename that you want to know the size

3.4.25 Get email body

```
function getmail_body($id, $images = false)
```

This function returns the string that contains the body of the email intended to be rendered in an iframe, for example

- @id => id of the email

3.4.26 TODO

```
function __getmail_head_helper($decoded, $email_id)
```

TODO

3.4.27 TODO

```
function __getmail_body_helper($decoded, $images = false)
```

TODO

3.4.28 Get email source

```
function getmail_source($id)
```

This function returns the string that contains the source of the email intended to be rendered in an iframe, for example

- @id => id of the email

3.4.29 Get email files

```
function getmail_files($id)
```

This function returns an array that contains the files of the email intended to be rendered in a table, for example

- @id => id of the email

3.4.30 Get cid

```
function getmail_cid($id, $cid)
```

This function returns the requested attachment identified by the cid argument

- @id => id of the email
- @cid => the cid of the content requested

3.4.31 Is outbox

```
function getmail_field($field, $id)
```

Returns the field of the email identified by the id argument

- @field => field requested
- @id => id of the email

3.4.32 Server

```
function getmail_server()
```

This function implements the old getmail action of the old saltos.

3.4.33 Delete

```
function getmail_delete($ids)
```

This function implements the old delete action of the old saltos.

- @ids => array with the emails id

3.4.34 Get viewpdf

```
function getmail_viewpdf($id, $cid)
```

This function returns the requested attachment identified by the cid argument in a pdf format for the viewpdf widget

- @id => id of the email
- @cid => the cid of the content requested

3.4.35 Get download

```
function getmail_download($id, $cid)
```

This function returns the requested attachment identified by the cid argument in an array format for the download feature

- @id => id of the email
- @cid => the cid of the content requested

3.4.36 TODO

```
function getmail_setter($ids, $what)
```

TODO

3.4.37 TODO

```
function getmail_pdf($ids)
```

TODO

3.4.38 TODO

```
function __iframe_srcdoc_helper($html)
```

TODO

3.5 Make indexing action

```
apps/emails/php/indexing.php
```

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

3.6 Send email library

```
apps/emails/php/sendmail.php
```

This library provides the necessary functions to send emails.

3.6.1 Used libraries

```
use PHPMailer\PHPMailer\PHPMailer;
```

This use loads the external libraries needed to run this library.

3.6.2 Sendmail

```
function sendmail($account_id, $to, $subject, $body, $files = '', $async = true)
```

This function send an email in synchronous and/or asynchronous mode

\$account_id => the account id used to detect the source of the email
\$to => can be an string with the destination email or an array with the follow prefixes => to:, cc:, bcc:, crt:, priority:, sensitivity:, replyto
\$subject => the subject string
\$body => the body string
\$files => an array with files

3.6.3 Debugoutput helper

```
function __sendmail_debugoutput_helper($str, $level)
```

This function tries to echoed all debug information with the SMTP Error: prefix, with this feature we can capture smtp errors with more details like the error stored in \$mail->ErrorInfo that always contains SMTP connect() failed, the signature of this function will accomplish the neested callback arguments

- @str => the debug trace string that can contain the errors
- @level => unused in this function

3.6.4 Parser

```
function __sendmail_parser($oldaddr)
```

This function gets an address and tries to detect the name part and the addr part of the argument. It's returns an array with two elements, the first is for the addr and the second is for the name.

- @oldaddr => the string that must to be processed

3.6.5 Message Id

```
function __sendmail_messageid($account_id, $from)
```

This function returns the message id for a new email, to do it, tries to detect the outbox directory, compute an aproximation to the newest value and checks that is unique in the system to prevent concurrence.

- @account_id => the account id used to send the new email
- @from => the from used to compute the crc32

3.6.6 Eml saver

```
function __sendmail_emlsaver($message, $messageid)
```


This function is intended to save the RFC822 message into the eml gzfile

- @message => the contents in RFC822 format of the message
- @messageid => the message id computed previously

3.6.7 Obj saver

```
function __sendmail_obj saver($mail, $messageid)
```

This function is intended to save the PHPMailer object into the obj file

- @mail => the PHPMailer object of the asynchronous transaction
- @messageid => the message id computed previously

3.6.8 TODO

```
function sendmail_prepare($action, $email_id)
```

TODO

3.6.9 TODO

```
function sendmail_action($json, $action, $email_id)
```

TODO

3.6.10 TODO

```
function sendmail_server()
```

TODO

3.6.11 TODO

```
function sendmail_files($action, $email_id)
```

TODO

3.6.12 TODO

```
function sendmail_signature($json)
```

TODO

4 Invoices

4.1 invoices application

```
apps/invoices/js/invoices.js
```

This application implements the typical features associated to invoices

4.1.1 Main object

```
saltos.invoices = {};
```

This object contains all SaltOS code

4.1.2 TODO

```
saltos.invoices.init = arg
```

TODO

4.1.3 TODO

```
saltos.invoices.compute_total = ()
```

TODO

4.1.4 TODO

```
saltos.invoices.add_item = ()
```

TODO

4.1.5 TODO

```
saltos.invoices.remove_item = (obj)
```

TODO

4.1.6 TODO

```
saltos.invoices.viewpdf = ()
```

TODO

4.1.7 TODO

```
saltos.invoices.download = ()
```

TODO

5 Tester

5.1 Tester application

```
apps/tester/js/tester.js
```

This application implements the typical features associated to tester

5.1.1 Tester object

```
saltos.testers = {};
```

This object stores all function used by this app

5.1.2 Campo 8

```
saltos.testers.campo8 = ()
```

TODO

5.1.3 Campo 9

```
saltos.testers.campo9 = ()
```

TODO

5.1.4 Campo 10

```
saltos.testers.campo10 = ()
```

TODO

5.1.5 Campo 11

```
saltos.testers.campo11 = ()
```

TODO

5.1.6 Campo 22

```
saltos.testers.campo22 = ()
```

TODO

6 Users

6.1 Login application

```
apps/users/js/login.js
```

This application implements the typical features associated to login

6.1.1 Main object

```
saltos.login = {};
```

This object contains all SaltOS code

6.1.2 Authenticate login function

```
saltos.login.authenticate = async ()
```

This function tries to authenticate the user using the user and pass fields of the form, to do it uses the authenticate function that send data to the authtoken action

6.2 Days functions

```
apps/users/php/days.php
```

This file contain all functions needed by the days feature

6.2.1 Days to bin

```
function days2bin($days)
```

This function tries to convert the days format used by the multiselect to the string expected by the database formed by ones and zeroes to represent if a day is operative for the user or not, for example, the selection 64,32,16,8,4 is returned like from monday to friday (1111100)

- @days => the string containing the days in power of two separated by comma

6.2.2 Bin to days

```
function bin2days($days)
```

This function tries to do the reverse action that the previous function, is able to get an string like 1111100 and returns the list of all bits in decimal like 64,32,16,8,4.

- @days => the string containing the days in binary format

6.2.3 Fix for days

```
function fix4days($data)
```

This function is intended to be used as wrapper in the result of the query that contains an element called days, in the database the days is stored using the binary notation like 1111100, and for the user interface, is needed to translate this string into a decimal string like 64,32,16,8,4.

- @data => the data obtained from an execute_query, for example, they must contain an entry called days.

6.3 Groups functions

```
apps/users/php/groups.php
```

This file contain all functions needed by the groups app

6.3.1 Insert group action

```
function insert_group($data)
```

This action allow to insert registers in the database associated to the groups app and only requires the data.
TODO

6.3.2 Update group action

```
function update_group($group_id, $data)
```

This action allow to update registers in the database associated to the groups app and requires the group_id and data.

TODO

6.3.3 Delete group action

```
function delete_group($group_id)
```

This action allow to delete registers in the database associated to the groups app and only requires the group_id.

TODO

6.4 Matrix functions

```
apps/users/php/matrix.php
```

This file contain all functions needed by the excel widget

6.4.1 Make matrix

```
function make_matrix_data($perms, $apps, $main, $user)
```

TODO

6.4.2 Make matrix

```
function make_matrix_cell($perms, $apps, $main, $user)
```

TODO

6.4.3 Make matrix

```
function unmake_matrix_data($perms, $apps, $main, $json)
```

TODO

6.5 Users functions

```
apps/users/php/users.php
```

This file contain all functions needed by the users app

6.5.1 Insert user action

```
function insert_user($data)
```

This action allow to insert registers in the database associated to the users app and only requires the data.

TODO

6.5.2 Update user action

```
function update_user($user_id, $data)
```

This action allow to update registers in the database associated to the users app and requires the user_id and data.

TODO

6.5.3 Delete user action

```
function delete_user($user_id)
```

This action allow to delete registers in the database associated to the users app and only requires the user_id.

TODO