Portale v2

Documentazione tema

https://adminlte.io/docs/3.2/

https://github.com/ColorlibHQ/AdminLTE

https://adminlte.io/themes/v3/

Webservice REST

Handler: include/Webservices/CustomerPortal.php

Headers:
• Authorization: Basic base64_encode("username:accesskey")
• Portal-Session-Id: il parametro è opzionale e viene restituito dalla "portal.login". Utilizzare nelle chiamate successive per ottimizzare le performance del portale.
• Content-Type: application/json

Nome	Parametri	Descrizione
portal.info		Permette di ottenere le informazioni sulla licenza e i loghi utilizzati dall'installazione di vtenext collegata.
portal.login	**Body** . username: string (*) . password: string (*) . language: string	Permette di eseguire il login nel portale fornendo l'email (username) e la password del contatto. Il servizio restituisce anche i valori dell'accesskey e dell'id di sessione da utilizzare nelle chiamate successive come header (Authorization e Portal-Session-Id)
portal.logout	**Headers** . Authorization	Permette di eseguire il logout del contatto dal portale.
portal.send_mail_for_password	**Body** . email: string (*) . language: string	Permette il recupero della password di un contatto del portale.
portal.modules_list	**Headers** . Authorization **Body** . language: string (*)	Permette di ottenere una lista di moduli abilitati per il profilo del contatto.

portal.get_list	**Headers** . Authorization **Body** . module: string (*) . language: string . folderid: int . search: array	Permette di ottenere la lista dei record di un modulo filtrati per la visibilità del profilo del contatto. Il parametro "folderid" viene utilizzato dal modulo Documents per ottenere i documenti di una cartella specifica. Il parametro "search" permette di eseguire la paginazione della lista server-side. Es. ['length' => 50, // Numero di record da restituire 'start' => 0, // Offset della paginazione 'search' => '', // Ricerca globale 'search_columns' => [// Ricerca per colonne ['index' => 0, // Indice del campo 'column' => '', // Nome del campo 'search' => '', // Valore del campo]], 'ordering' => [// Ordinamento ['index' => 0, // Indice del campo 'column' => '', // Nome del campo 'dir' => '', // asc/desc]],] Note: 1. Tutte le ricerche vengono eseguite SOLO sulle colonne della lista. 2. La ricerca per colonna viene eseguita in modalità "LIKE". 3. L'ordinamento può essere eseguito solo su una colonna.
portal.get_blocks	**Headers** . Authorization **Body** . module: string (*) . language: string . mode: string (edit, create, detail, list) (*) . app_data: array	Permette di ottenere la lista dei blocchi e dei campi di un modulo filtrati per la visibilità del profilo del contatto. Il parametro "app_data" rappresenta il record che si sta modificando (array con nome campo e valore) ed è utilizzato dalle view SDK per stabilire la visibilità dei campi. Questo parametro verrà utilizzato anche in futuro per la gestione dei campi condizionali.
portal.get_record	**Headers** . Authorization **Body** . module: string (*) . id: int (*)	Permette di ottenere i dati del record indicato se il contatto ha il permesso di visualizzarlo.

portal.save_record	**Headers** . Authorization **Body** . module: string (*) . id: int (*) . values: encoded (*)	Permette il salvataggio del record indicato se il contatto ha il permesso di modificarlo.
portal.delete_record	**Headers** . Authorization **Body** . module: string (*) . id: int (*)	Permette l'eliminazione del record indicato se il contatto ha il permesso di cancellarlo.
portal.write_ticket_comment	**Headers** . Authorization **Body** . id: int (*) . comment: string (*)	Permette la scrittura di un commento all'interno del ticket indicato.
portal.get_attachments	**Headers** . Authorization **Body** . id: int (*)	Permette di ottenere la lista dei documenti relazionati al record indicato filtrati per la visibilità del profilo del contatto.
portal.download_attachment	**Headers** . Authorization **Body** . relid: int (*) . docid: int	Permette il download dell'allegato (relid) se il contatto ha il permesso di visualizzarlo.
portal.upload_attachment	**Headers** . Authorization **Body** . relid: int (*) . title: string (*) **File**	Permette il caricamento di un allegato relazionato al record specificato (relid). Si può indicare il nome (title) del documento che verrà generato.
portal.provide_confidential_info	**Headers** . Authorization **Body** . id: int (*) . comments: string . data: string (*) . request_commentid: int (*)	Permette di rispondere ad una richiesta di informazioni confidenziali. Il parametro "id" indica l'id del ticket, il parametro "comments" indica il commento non cifrato, il parametro "data" indica la risposta confidenziale e il parametro "request_commentid" indica l'id del commento a cui fornire la risposta confidenziale.
portal.get_home_widgets	**Headers** . Authorization **Body** . language: string	Permette di ottenere i widget configurati nel profilo del contatto.
portal.save_authenticate_cookie	**Headers** . Authorization **Body** . contactid: int (*)	Fornisce un hash da memorizzare in un cookie per ricordare l'accesso del contatto.

portal.check_authenticate_cookie	**Headers** . Authorization **Body** . contactid: int (*) . hash: string (*)	Permette di verificare l'hash utilizzato per ricordare l'accesso del contatto.
portal.change_password	**Headers** . Authorization **Body** . username: string (*) . old_password: string (*) . password: string (*) . language: string	Permette di cambiare la password del contatto.

Registrare un nuovo webservice REST

Creare un nuovo file ed eseguirlo (es. plugins/script/script.php).

```
<?php
require('../../config.inc.php');
chdir($root_directory);
require_once('include/utils/utils.php');
require_once('vtlib/Vtecrm/Module.php');
$Vtiger_Utils_Log = true;
global $adb, $table_prefix;
VteSession::start();

SDK::setClass('CustomerPortalRestApi', 'CustomerPortalRestApi2',
'modules/SDK/src/CustomerPortalRestApi2.php');

$parameters = ['param1' => 'string', 'param2' => 'encoded', 'param3' => 'encoded'];
$perm = 'read'; // read, write, readwrite
SDK::setRestOperation('portal.foo', 'modules/SDK/src/CustomerPortalRestApi2.php',
'CustomerPortalRestApi2.foo', $parameters, $perm);
```

Creare un nuovo file che contiene la classe estesa CustomerPortalRestApi2 (es. modules/SDK/src/CustomerPortalRestApi2.php).

```
<?php

require_once('include/Webservices/CustomerPortal.php');

class CustomerPortalRestApi2 extends CustomerPortalRestApi {
```

```
	public function foo($param1, $param2, $param3) {
		$data = [1, 2, 3, 4, 5];
		// ...
		return $data;
	}

}
```

Estendere un webservice REST esistente

Creare un nuovo file ed eseguirlo (es. plugins/script/script.php).

```
<?php
require('../../config.inc.php');
chdir($root_directory);
require_once('include/utils/utils.php');
require_once('vtlib/Vtecrm/Module.php');
$Vtiger_Utils_Log = true;
global $adb, $table_prefix;
VteSession::start();

SDK::setClass('CustomerPortalRestApi', 'CustomerPortalRestApi2',
'modules/SDK/src/CustomerPortalRestApi2.php');
```

Creare un nuovo file che contiene la classe estesa CustomerPortalRestApi2 (es. modules/SDK/src/CustomerPortalRestApi2.php).

```
<?php

require_once('include/Webservices/CustomerPortal.php');

class CustomerPortalRestApi2 extends CustomerPortalRestApi {

	public function get_list($module, $language, $folderid = 0, $search = []) {
		$ret = parent::get_list($module, $language, $folderid, $search);
		// your code here ...
		return $ret;
	}

}
```

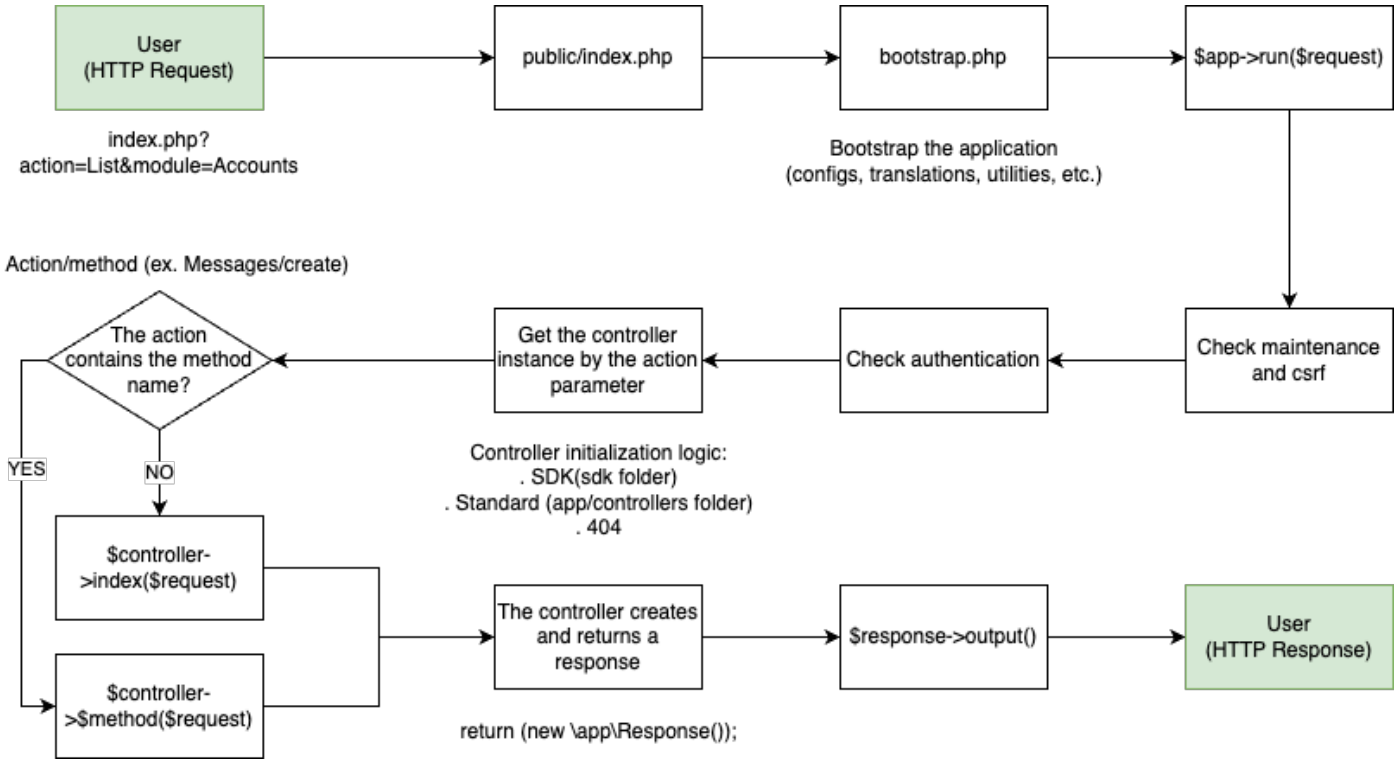
Struttura delle cartelle/file principali

> La cartella del portale si trova in: **VTE_ROOT/portal/v2**

Le cartelle/file principali del nuovo portale sono:

Nome cartella/file	**Descrizione**
app	Contiene i file e le logiche per il funzionamento del portale.
app/controllers	Contiene i file che gestiscono le azioni di default del portale (es. Login, Logout, Edit, Detail, ecc.).
app/fields	Contiene i file che gestiscono i campi dei moduli del portale.
app/modules	Contiene delle logiche custom di alcuni moduli di vtenext (es. Documents, Processes, HelpDesk, ecc.).
app/PortalModule.php	Classe che gestisce le azioni dei moduli del portale (es. Create, Detail, Edit, ecc.). La classe può essere estesa per modificare i comportamenti di default di un modulo.
config	Contiene i file di configurazione del portale.
public	Contiene i file pubblici del portale (es. index.php, css, javascript, immagini, ecc.).
resources	Contiene le risorse utilizzate dal portale come i file per le traduzioni (lang) e i template (templates).
sdk	La cartella è utilizzata per inserire nuove personalizzazioni per il cliente.
storage	Contiene i file temporanei (es. cache, logs, ecc.).
vendor	Contiene le librerie esterne utilizzate dal portale.

Request lifecycle



Configurazione del portale

Il file di configurazione del portale si trova in "config/portal.config.php".

> Per **sovrascrivere i parametri** deve essere utilizzato il file "config/sdk.config.php" altrimenti, in caso di aggiornamento della versione di vtenext, le modifiche potrebbero essere perse.

> **Come spostare il Business Portal in un'altra macchina/cartella**
> Valorizzare i parametri "portal_url", "vte_url" e "csrf_secret" nel file "config/sdk.config.php". Eventualmente cambiare anche il parametro "default_timezone" se la macchina ha un timezone diverso da quello in cui risiede il vte. Cambiare la prop "portal.url" con il link che punta alla cartella v2. es. https://ticket.vtenext.com/v2

Ecco la lista dei parametri supportati dal nuovo portale:

Parametro	Tipo	Default	Descrizione

portal_url	String	$PORTAL_URL (config.inc.php)	Indica l'URL del portale clienti. Se la cartella del portale si trova all'interno della root directory di vtenext, la variabile verrà impostata con il valore della variabile $PORTAL_URL impostata nel file config.inc.php.
vte_url	String	$site_URL (config.inc.php)	Indica l'URL di vtenext e viene utilizzato per ottenere i relativi dati tramite le rest API. Se la cartella del portale si trova all'interno della root directory di vtenext, la variabile verrà impostata con il valore della variabile $site_URL impostata nel file config.inc.php.
default_language	String	it_it	Indica la lingua predefinita utilizzata nel portale clienti. Il valore può essere sostituito con una lingua supportata (vedi parametro "languages").
languages	Array	['en_us' => 'US English', 'it_it' => 'IT Italiano']	Indica le lingue supportate nel portale clienti. Per aggiungere una nuova lingua si deve creare un nuovo file nella cartella resources/lang.
production	Bool	false	Questa configurazione indica se gli errori devono essere visualizzati o meno. Se l'ambiente è di produzione, il valore verrà impostato su true per disabilitare la visualizzazione degli errori. Se l'ambiente è di sviluppo, il valore verrà impostato su false per consentire la visualizzazione degli errori.

default_module	String		Indica il modulo predefinito da caricare dopo l'accesso al portale. Il valore può essere sostituito con il nome del modulo desiderato (deve essere abilitato da profilo). Se il parametro è vuoto, verrà caricata la home del portale.
favicon	String	assets/img/VTENEXT_favicon.ico	Indica il percorso della favicon da caricare nel portale clienti. Il valore è relativo alla cartella public.
login_logo	String	assets/img/VTENEXT_login.png	Indica il percorso del logo da caricare all'interno della pagina di login. Il valore è relativo alla cartella public. [UPDATE] Il logo deve essere caricato nelle impostazioni di vtenext > Loghi.
login_background	String		Indica il percorso dello sfondo da caricare all'interno della pagina di login. Il valore è relativo alla cartella public.
header_logo_sm	String	assets/img/VTENEXT_toggle.png	Indica il percorso dell'icona da caricare nella barra laterale minimizzata. Il valore è relativo alla cartella public. [UPDATE] L'icona deve essere caricata nelle impostazioni di vtenext > Loghi.
header_logo_lg	String	assets/img/VTENEXT_header.png	Indica il percorso dell'icona da caricare nella barra laterale allargata. Il valore è relativo alla cartella public. [UPDATE] L'icona deve essere caricata nelle impostazioni di vtenext > Loghi.
helpdesk_logo	String	assets/img/helpdesk.png	Indica il logo utilizzato per visualizzare la risposta data dall'assistenza clienti. Il valore è relativo alla cartella public.

sidebar_theme	String	sidebar-dark-primary	Indica la classe della barra laterale principale. Può avere una luminosità "dark" o "light". Può avere anche una variante di colore, come "primary", "success", "warning", "info", "danger".
enable_sidebar_search	Bool	false	Consente di attivare/disattivare la barra di ricerca nella barra laterale principale.
csrf_secret	String	$csrf_secret (config.inc.php)	Indica la chiave segreta utilizzata per generare un token csrf. Se la cartella del portale si trova all'interno della root directory di vtenext, la variabile verrà impostata con il valore della variabile $csrf_secret impostata nel file config.inc.php.
upload_dir	String		Indica il nome della cartella utilizzata per il caricamento dei file.
browser_title_prefix	String		Indica l'etichetta del prefisso da utilizzare per il titolo del browser. Il valore può essere sostituito con l'etichetta desiderata.
browser_title_suffix	String	customer_portal	Indica l'etichetta del suffisso da utilizzare per il titolo del browser. Il valore può essere sostituito con l'etichetta desiderata.
remember_cookie_name	String	portal_login_hash	Indica il nome del cookie utilizzato per ricordare l'autenticazione dell'utente.
login_expire_time	Int	2592000 (one month)	Indica la scadenza del cookie utilizzato per ricordare l'autenticazione dell'utente. Il valore può essere sostituito con il numero di secondi desiderato.

default_timezone	String	Europe/Rome	Indica il fuso orario predefinito utilizzato nel portale clienti. Questa configurazione deve essere uguale alla variabile $default_timezone impostata nel file config.inc.php di vtenext.
module_icons	Array	[]	Con questa configurazione è possibile sovrascrivere le icone predefinite utilizzate per i moduli abilitati nel portale clienti. Le icone predefinite dei moduli si trovano in app/layouts/PortalLayout.php. I nomi delle icone si possono trovare qui https://fonts.google.com/icons.
sdk_languages	Array	[]	Con questa configurazione si possono aggiungere nuove etichette o modificare quelle esistenti. Si deve creare un nuovo file nella cartella sdk.
sdk_global_php	Array	[]	Con questa configurazione si possono aggiungere file php da caricare in ogni pagina (dovrebbero contenere classi/funzioni). Si deve creare un nuovo file nella cartella sdk.
sdk_global_js	Array	[]	Con questa configurazione si possono aggiungere js da caricare a livello globale. Si deve creare un nuovo file nella cartella public/assets/sdk.
sdk_module_js	Array	[]	Con questa configurazione si possono aggiungere js da caricare per un modulo specifico. Si deve creare un nuovo file nella cartella public/assets/sdk.
sdk_global_css	Array	[]	Con questa configurazione si possono aggiungere css da caricare a livello globale. Si deve creare un nuovo file nella cartella public/assets/sdk.

sdk_controllers	Array	[]	Con questa configurazione si possono aggiungere azioni personalizzate (campo "action" nell'url). L'array rappresenta un'associazione tra il nome dell'azione e il file che contiene la classe controller per gestire la richiesta. Si deve creare un nuovo file nella cartella sdk.
sdk_module	Array	[]	Con questa configurazione si possono aggiungere personalizzazioni su un modulo specifico. L'array rappresenta un'associazione tra il nome del modulo e il file che contiene la classe estesa del modulo. Si deve creare un nuovo file nella cartella sdk.
sdk_menu	Array	[]	Con questa configurazione si possono aggiungere aggiungere voci di menu personalizzate nella barra laterale. Si deve creare un nuovo file nella cartella sdk.

API Reference - Classi, metodi, funzioni e variabili principali

\app\Request

Metodo	**Argomenti**	**Descrizione**

get()	$keys = null, $purify = false	Permette di ottenere uno o più parametri dalla variabile globale $_GET. Se il secondo argomento è impostato a "true", i parametri vengono purificati attraverso la libreria HTML Purifier. Esempio:
		`$request->get('foo');`
		`$request->get(['foo', 'bar']);`
		`$request->get('foo', true);`

post()	$keys = null, $purify = false	Permette di ottenere uno o più parametri dalla variabile globale $_POST. Se il secondo argomento è impostato a "true", i parametri vengono purificati attraverso la libreria HTML Purifier. Esempio:
		`$request->post('foo');`
		`$request->post(['foo', 'bar']);`
		`$request->post('foo', true);`

files()	$key	Permette di ottenere i file caricati attraverso il metodo HTTP POST e organizzati tramite la variabile globale $_FILES. `$request->files('attachments');`
cookie()	$keys = null, $purify = false	Permette di ottenere uno o più parametri dalla variabile globale $_COOKIE. Se il secondo argomento è impostato a "true", i parametri vengono purificati attraverso la libreria HTML Purifier. Esempio: `$request->cookie('foo');`
server()	$keys = null, $purify = false	Permette di ottenere uno o più parametri dalla variabile globale $_SERVER. Se il secondo argomento è impostato a "true", i parametri vengono purificati attraverso la libreria HTML Purifier. Esempio: `$request->server('');`
isGet()		Ritorna true se il metodo della richiesta è GET.
isPost()		Ritorna true se il metodo della richiesta è POST.
isAjax()		Ritorna true se la richiesta è AJAX.
purify()	$input	Purifica la variabile $input attraverso la libreria HTML Purifier.

\app\Response

Metodo	Argomenti	Descrizione
__construct()	$content = '', $statusCode = 200, $headers = []	Inizializza un nuovo oggetto \app\Response() con il contenuto della risposta ($content), il codice di ritorno ($statusCode) e gli header di default ($headers).
setContent()	$content	Imposta il contenuto della risposta.
setStatusCode()	$statusCode	Imposta il codice di ritorna della risposta.
setHeader()	$header, $replace = true	Imposta un nuovo header nella risposta. Con il secondo argomento è possibile indicare se l'header deve sostituire oppure no un header precedente già impostato.
setMimeType()	$mimeType = 'text/html'	Imposta il mime del contenuto della risposta.
json()	$data	Imposta il contenuto della risposta con $data convertito in formato JSON e mime 'application/json'.
redirect()	$page	Esegue un redirect verso $page.
downloadFile()	$fullpath	Permette lo scaricamento di un file posizionato in $fullpath.
output()		Esegue l'output del contenuto, del codice di risposta e degli header impostati.

\app\Session

Metodo	Argomenti	Descrizione
set()	$key, $value = ''	Imposta il valore $value con chiave $key in $_SESSION.
get()	$key	Permette di ottenere il valore con chiave $key da $_SESSION.
flash()	$key	Permette di ottenere il valore con chiave $key da $_SESSION. Successivamente, la chiave $key verrà eliminata da $_SESSION.
remove()	$key	Elimina la chiave $key da $_SESSION.
hasKey()	$key	Ritorna true se esiste la chiave $key in $_SESSION.

setMulti()	$keys	Permette di scrivere valori multipli in $_SESSION.
removeMulti()	$keys	Elimina valori multipli in $_SESSION.
append()	$key, $value = ''	Imposta la chiave $key come un array in $_SESSION e il valore $value viene aggiunto a quest'ultimo.

\app\Config

Metodo	**Argomenti**	**Descrizione**
has()	$key	Ritorna true se esiste la chiave $key nella configurazione globale.
get()	$key	Permette di ottenere il valore con chiave $key dalla configurazione globale.
set()	$key, $value	Imposta il valore $value con chiave $key nella configurazione globale.
getAll()		Permette di ottenere una lista chiave-valore con tutta la configurazione globale.
setMulti()	$values	Permette di scrivere valori multipli nella configurazione globale.
clear()	$key	Elimina la chiave $key dalla configurazione globale.
clearAll()		Elimina tutti i valori dalla configurazione globale.

\app\PortalModule

Variabile	**Default**	**Descrizione**
$hasComments	false	Indica se il modulo supporta il blocco commenti.
$hasAttachments	false	Indica se il modulo supporta gli allegati.
$enableEdit	true	Indica se il modulo può essere modificato (modalità edit).
$formColumns	3	Indica il numero di colonne da utilizzare per la visualizzazione dei campi in Create, Edit e Detail.
$listTemplate	List.tpl	Indica il template utilizzato per la visualizzazione di una lista (action List).

$referenceListTemplate	sections/ReferenceList.tpl	Indica il template utilizzato per la visualizzazione di una lista relazionata (uitype 10).
$detailTemplate	Detail.tpl	Indica il template utilizzato per la visualizzazione del dettaglio di un record (action Detail).
$editTemplate	Edit.tpl	Indica il template utilizzato per la visualizzazione della modifica di un record (action Edit).
$notAuthorizedTemplate	PageNotAuthorized.tpl	Indica il template utilizzato per visualizzare un errore di permesso (es. record non trovato o errori di permessi).

Metodo	**Argomenti**	**Descrizione**
__construct()	$module	$module indica il nome del modulo per ottenere un'instanza della classe. Se il modulo è stato esteso tramite sdk allora verrà ritornata un'istanza della classe estesa.
prepareList()	$viewer, $request	Metodo utilizzato per la visualizzazione di una lista (action List).
postProcessList()	$viewer, $request	Questo metodo può essere utilizzato dalle classi estese per inserire/modificare i dati impostati nella prepareList().
prepareEdit()	$viewer, $request	Metodo utilizzato per la visualizzazione della modifica di un record (action Edit).
postProcessEdit()	$viewer, $request	Questo metodo può essere utilizzato dalle classi estese per inserire/modificare i dati impostati nella prepareEdit().
prepareDetail()	$viewer, $request	Metodo utilizzato per la visualizzazione del dettaglio di un record (action Detail).
postProcessDetail()	$viewer, $request	Questo metodo può essere utilizzato dalle classi estese per inserire/modificare i dati impostati nella prepareDetail().
saveRecord()	$request	Metodo utilizzato per il salvataggio di un record (action Save).

postProcessSaveValues()	$request, &$values	Questo metodo può essere utilizzato dalle classi estese per inserire/modificare i dati impostati nella saveRecord().
isPermitted()	$module, $action = self::ACTION_LIST, $recordValues = []	Ritorna se il contatto ha il permesso di eseguire una determinata azione. Lista azioni supportate: . ACTION_LIST . ACTION_CREATE . ACTION_EDIT . ACTION_DETAIL . ACTION_DELETE . ACTION_SAVE . ACTION_ADD_COMMENTS . ACTION_UPLOAD_ATTACHMENTS . ACTION_CHANGE_PWD . ACTION_SOLVE_TICKET

\app\clients\PortalRestClient

Metodo	**Argomenti**	**Descrizione**
get()	$restName, $queryParameters = [], $headers = []	Permette di eseguire una richiesta GET verso vtenext.
post()	$restName, $queryParameters = [], $bodyParameters = [], $headers = []	Permette di eseguire una richiesta POST verso vtenext.
postMultipart()	$restName, $queryParameters = [], $bodyParameters = [], $fileParameters = [], $headers = []	Permette di eseguire una richiesta POST multipart verso vtenext.
patch()	$restName, $queryParameters = [], $bodyParameters = [], $headers = []	Permette di eseguire una richiesta PATCH verso vtenext.
delete()	$restName, $queryParameters = [], $headers = []	Permette di eseguire una richiesta DELETE verso vtenext.
postDownload()	$restName, $queryParameters = [], $bodyParameters = [], $headers = []	Permette di eseguire una richiesta POST per scaricare un file di vtenext in modalità stream.

Helpers

Funzione	**Argomenti**	**Descrizione**
preprint()	$var	print_r formattato con tag <pre>
predump()	$var	var_dump formattato con tag <pre>

encodeForHtml()	$value, $charset = 'UTF-8'	Codifica il valore $value per essere inserito in una pagina HTML.
encodeForHtmlAttr()	$value, $enclosing = '"'	Codifica il valore $value per essere inserito all'interno di un attributo di un tag HTML.
encodeForJs()	$value, $enclosing = '"'	Codifica il valore $value per essere inserito all'interno di uno <script> javascript.
htmlAttr()	$attributes	Codifica una lista di attributi per essere inseriti all'interno di un tag HTML.
config()	$key	Permette di ottenere il valore con chiave $key dalla configurazione globale.
trans()	$key, $args = []	Permette la traduzione di un'etichetta $key.
listUrl()	$module, $extraParams = []	Genera un link per aprire una lista.
createUrl()	$module, $extraParams = []	Genera un link per aprire la creazione di un record.
createDocUrl()	$module, $folderId, $extraParams = []	Genera un link per aprire la creazione di un documento.
detailUrl()	$module, $record, $extraParams = []	Genera un link per aprire il dettaglio di un record.
editUrl()	$module, $record, $extraParams = []	Genera un link per aprire la modifica di un record.
downloadUrl()	$record, $documentId, $extraParams = []	Genera un link per eseguire il download di un documento.
docFolderUrl()	$module, $folderId, $extraParams = []	Genera un link per aprire una cartella specifica nel modulo documenti.
returnUrl()	$request	Genera un link di ritorno indietro (es. azione "Annulla").
portalLanguage()		Ritorna la lingua utilizzata nel portale.
setPortalLanguage()	$language	Imposta la lingua $language nel portale.
getBrowserTitle()	$title	Ritorna il titolo da impostare in una pagina HTML del portale.
getModuleLabel()	$module	Ritorna la traduzione del modulo $module.
getSingleModuleLabel()	$module	Ritorna la traduzione singolare del modulo $module.

getMaxUploadSize()		Ritorna la dimensione massima di upload nel portale.
setPortalCookie()	$name, $value = "", $expires_or_options, $httponly = false	Imposta un cookie nel portale.
unsetPortalCookie()	$name, $httponly = false	Rimuove un cookie dal portale.
csrfToken()		Ritorna il token csrf.
csrfInputName()		Ritorna il nome dell'input per l'invio del token csrf.
flashPortalError()	$error	Permette di visualizzare un messaggio di errore temporizzato.
flashPortalMessage()	$message	Permette di visualizzare un messaggio temporizzato.
customerId()		Ritorna l'id del contatto autenticato.
customerEmail()		Ritorna l'email del contatto autenticato.
customerUsername()		Ritorna il nome e cognome del contatto autenticato.
resourcever()	$filename	Permette il versionamento di file css e javascript.
basePath()	$path = ''	Ritorna il percorso della cartella base del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella base.
appPath()	$path = ''	Ritorna il percorso della cartella "app" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "app".
configPath()	$path = ''	Ritorna il percorso della cartella "config" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "config".
resourcesPath()	$path = ''	Ritorna il percorso della cartella "resources" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "resources".
langPath()	$lang	Ritorna il percorso del file della lingua $lang indicata.

storagePath()	$path = ''	Ritorna il percorso della cartella "storage" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "storage".
publicPath()	$path = ''	Ritorna il percorso della cartella "public" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "public".
vtePath()	$path = ''	Ritorna il percorso della cartella di vtenext. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella di vtenext.
sdkPath()	$path = ''	Ritorna il percorso della cartella "sdk" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "sdk".
sdkAssetsPath()	$path = ''	Ritorna il percorso della cartella "public/assets/sdk" del portale. Se viene indicato $path viene creato e restituito un percorso a partire dalla cartella "public/assets/sdk".
publicRelPath()	$assetPath	Ritorna un percorso relativo a partire dalla cartella "public".

Esempi

View SDK

Ecco un esempio di come modificare la visibilità dei campi del portale tramite view SDK.

```
<?php
require('../../config.inc.php');
chdir($root_directory);
require_once('include/utils/utils.php');
require_once('vtlib/Vtecrm/Module.php');
$Vtiger_Utils_Log = true;
global $adb, $table_prefix;
VteSession::start();

$module = '';
$src = '';
$mode = 'constrain';
```

```
$success = 'continue';
SDK::addView($module, $src, $mode, $success);
```

```
<?php

global $sdk_mode, $table_prefix;

switch ($sdk_mode) {
	case 'portal.create':
		$readonly = 100;
		$success = true;
		break;
	case 'portal.edit':
	case 'portal.detail':
		if ($col_fields['field3'] !== 'Open') {
			$readonly = 99;
			$success = true;
		}
		if (in_array($fieldname, ['field1', 'field2'])) {
			$readonly = 100;
			$success = true;
		}
		break;
}
```

Creazione di un nuovo controller

Ecco un esempio di come creare un nuovo controller per gestire un'azione custom.

- Modificare "config/sdk.config.php" inserendo il nuovo controller

```
return [
	'sdk_controllers' => [
		'SampleVte' => 'controllers/SampleVteController.php',
	]
];
```

- Implementare la classe SampleVteController in "sdk/controllers/SampleVteController.php"

```
<?php
```

```
class SampleVteController extends \app\controllers\BaseController {
	
	public function index($request) {
		return $this->displaySomething($request);
	}

	protected function displaySomething($request) {
		$parameter1 = $request->get('parameter1', true);
		$parameter2 = $request->get('parameter2', true);

		$this->viewer->assign('PARAMETER1', $parameter1);
		$this->viewer->assign('PARAMETER2', $parameter2);

		$layout = \app\LayoutFactory::getPortalLayout($this->viewer, $this->client, $request);
		$output = $this->fetchWithLayout('sdk/SampleVte.tpl', $layout);

		return new \app\Response($output);
	}
	
}
```

- Creare un nuovo template in "resources/templates/sdk/SampleVte.tpl"

```
{extends file='layouts/PortalLayout.tpl'}

{block name=content}
	<h1>Sample Vte</h1>
{/block}
```

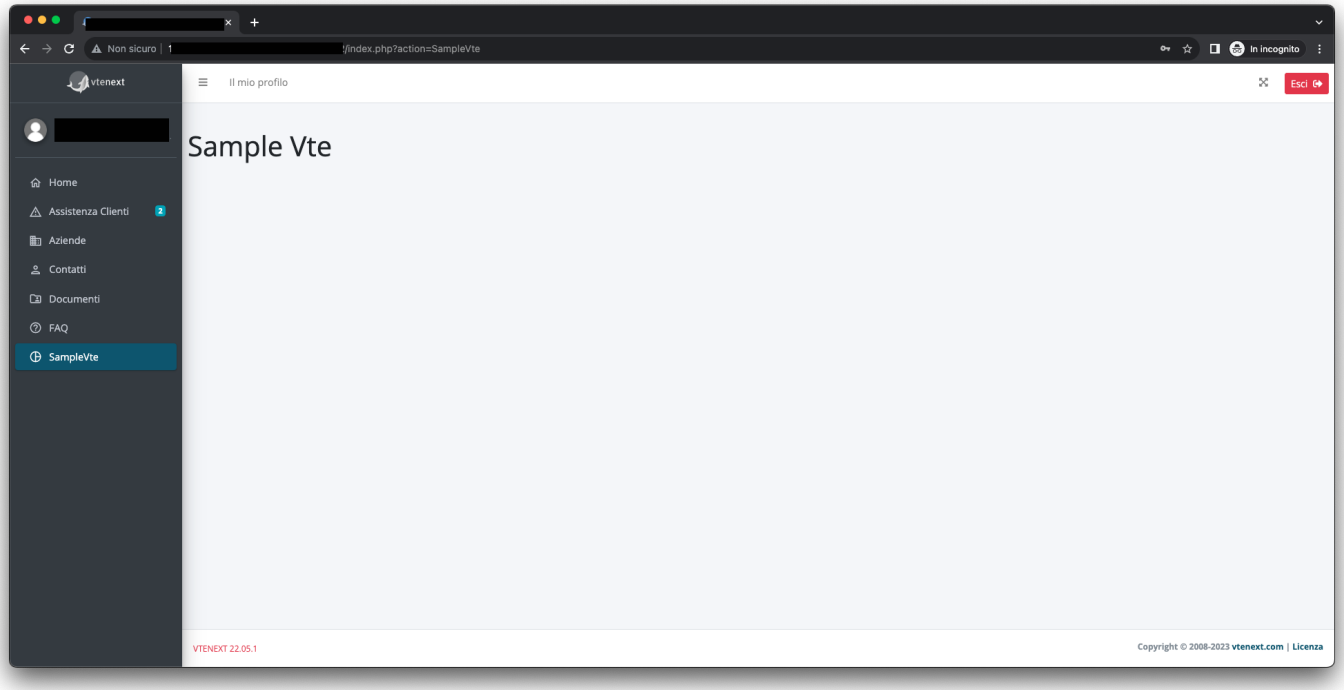
- Modificare "config/sdk.config.php" indicando il file sdk per l'inserimento delle nuovi voci nel menù laterale

```
return [
	'sdk_controllers' => [
		'SampleVte' => 'controllers/SampleVteController.php',
	],
	  'sdk_menu' => [
		'samplevte-menu.php',
	]
];
```

- Modificare il file "sdk/samplevte-menu.php"

```
<?php

return [
    [
        'text' => trans('SampleVte'),
        'active' => false,
        'prefix' => [
            'type' => 'icon',
            'icon_style' => 'material',
            'icon_name' => 'pie_chart',
        ],
        'action' => [
            'type' => 'link',
            'link_href' => "index.php?action=SampleVte",
        ],
    ],
];
```



Estensione di un modulo

Ecco un esempio di come estendere le funzionalità di un modulo.

- Modificare "config/sdk.config.php"

```
return [
    'sdk_module' => [
        'Contacts' => 'modules/ContactsModule.php',
    ]
];
```

- Implementare la classe ContactsModule in "sdk/modules/ContactsModule.php". In questo esempio viene attivato il blocco per inserire i commenti e viene cambiato il layout passando ad una visualizzazione a due colonne.

```
<?php

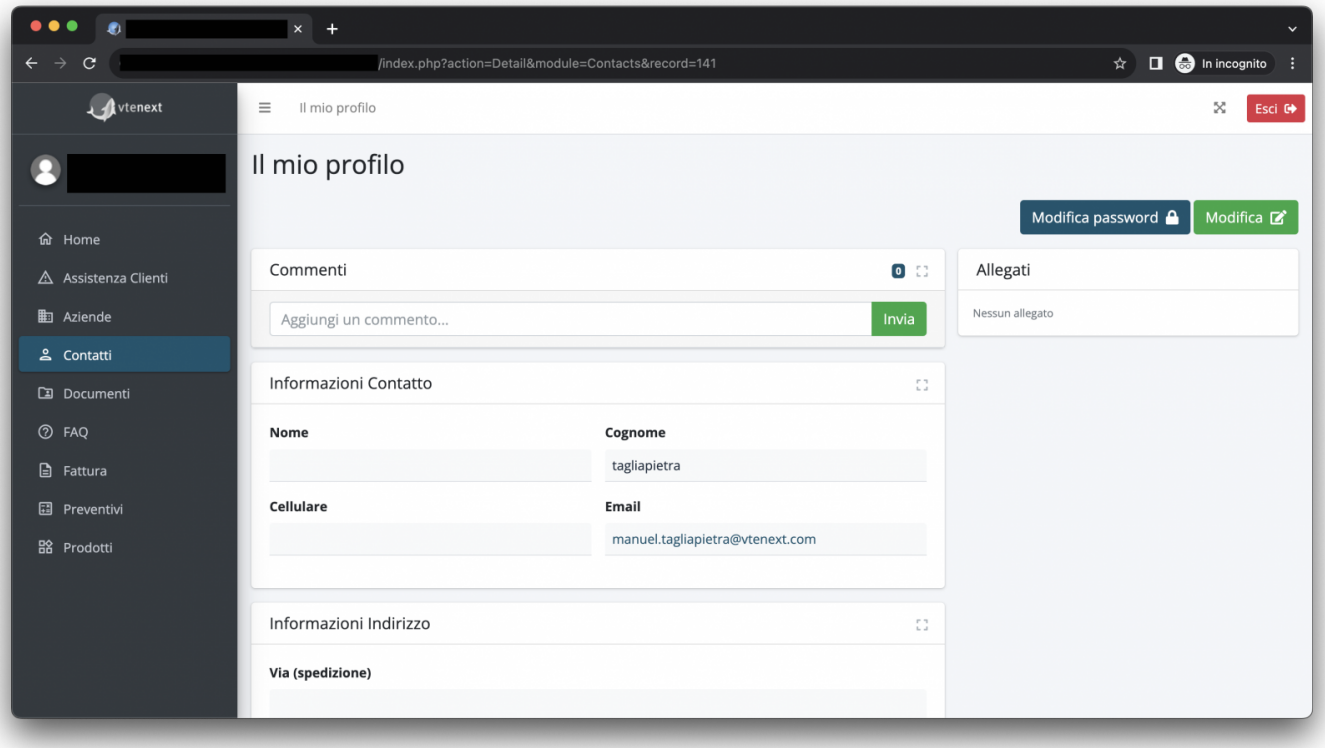
class ContactsModule extends \app\PortalModule {

    public $hasComments = true;

    public $formColumns = 2;

    public function canAddComments() {
        return true;
    }

}
```



Revision #65
Created 2023-09-12 09:45:41 UTC by manuel.tagliapietra
Updated 2025-03-04 15:51:45 UTC by m.maporti