

## SugarCRM - Bug # 337: New Errors after fresh install.

<b>Status:</b>	Resolved	<b>Priority:</b>	High
<b>Author:</b>	Ryan Schuster	<b>Category:</b>	
<b>Created:</b>	05 Dec 2013	<b>Assignee:</b>	Rahul Bhattacharya
<b>Updated:</b>	31 Jul 2014	<b>Due date:</b>	
<b>Subject:</b>	New Errors after fresh install.		
<b>Description:</b>	<p>Hello, I have two errors coming now. I've posted a while ago about moving to virtual hardware. I have the app installed now with no known issues with the installation. when I go to run a sync, I get the following errors:</p> <pre>Notice Trying          to          get          property          of          non-object (C:/inetpub/apps/helpdesk/___apps/sugarcrm/cron/class.Controller_SugarCRMFetchAccountMinute.php:66) ===== =====  Warning Invalid          argument          supplied          for          foreach() (C:/inetpub/apps/helpdesk/___apps/sugarcrm/cron/class.Controller_SugarCRMFetchAccountMinute.php:66) ===== =====</pre>		

### History

12/13/2013 07:04 am - Anjali Sharma

Greetings,

Please replace the code as i mentioned below for the fix:

1) \*file: class.SWIFT\_SugarCRMRestAPIRequest.php\*

\*

line : #145 and #190

\*replace:\*\*

```
return $this->CurlResponse($_requestParameters);
```

\*with:\*

```
$_response = $this->CurlResponse($_requestParameters);
```

```
if (!empty($_response->entry_list)) {
```

```
    return $_response->entry_list;
```

```
} else {
```

```
    return false;
```

```
}
```

\*2) \*file : class.Controller\_SugarCRMFetchAccountMinute.php\*

\*line\* #60

replace:\*

else

```
{
```

```
    $_sugarcrmOrganizationResponse = $this->SugarCRMRestAPIRequest->FetchAccounts($_sugarcrmSessionResponse, $_sugarcrmApiurl);
```

```

$this->Load->Model('SugarCRMAccount:SugarCRMAccount');

foreach($_sugarcrmOrganizationResponse->entry_list as $key=>$value)
{
    $_organizationDetails = array(
        'id' => $value->name_value_list->id->value,
        'name' => $value->name_value_list->name->value,
        'billing_address_street' => $value->name_value_list->billing_address_street->value,
        'billing_address_city' => $value->name_value_list->billing_address_city->value,
        'billing_address_state' => $value->name_value_list->billing_address_state->value,
        'billing_address_postalcode' => $value->name_value_list->billing_address_postalcode->value,
        'billing_address_country' => $value->name_value_list->billing_address_country->value,
        'phone_office' => $value->name_value_list->phone_office->value,
        'website' => $value->name_value_list->website->value,
        'date_entered' => $value->name_value_list->date_entered->value,
        'date_modified' => $value->name_value_list->date_modified->value,
    );

    $this->SugarCRMAccount->InsertOrganizations($_organizationDetails);
    }
}

*With:*
else
    {
        $_sugarcrmOrganizationResponse = $this->SugarCRMRestAPIRequest->FetchAccounts($_sugarcrmSessionResponse, $_sugarcrmApiurl);

    if (is_array($_sugarcrmOrganizationResponse)) {

        $this->Load->Model('SugarCRMAccount:SugarCRMAccount');

        foreach($_sugarcrmOrganizationResponse as $key=>$value)
        {
            if(!isset($value->name_value_list)) {
                continue;
            }

            $_organizationDetails = array(
                'id' => $value->name_value_list->id->value,
                'name' => $value->name_value_list->name->value,
                'billing_address_street' => $value->name_value_list->billing_address_street->value,
                'billing_address_city' => $value->name_value_list->billing_address_city->value,
                'billing_address_state' => $value->name_value_list->billing_address_state->value,
                'billing_address_postalcode' => $value->name_value_list->billing_address_postalcode->value,
                'billing_address_country' => $value->name_value_list->billing_address_country->value,
                'phone_office' => $value->name_value_list->phone_office->value,
                'website' => $value->name_value_list->website->value,
                'date_entered' => $value->name_value_list->date_entered->value,
                'date_modified' => $value->name_value_list->date_modified->value,
            );

            $this->SugarCRMAccount->InsertOrganizations($_organizationDetails);

```

```
}  
}  
}
```

\*3) file: class.Controller\_SugarCRMFetchUserMinute.php

line\* #48

\*replace:\*

```
} else {  
    $this->Load->Model('SugarCRMAccount:SugarCRMAccount');  
    $this->Load->Model('SugarCRMAccount:SugarCRMUser');  
  
    $_organizationList = $this->SugarCRMAccount->FetchOrganizations();  
    foreach ($_organizationList as $_accountList) {  
        //Sync sugar contacts with kayako user  
        $_sugarcrmContactResponse = $this->SugarCRMRestAPIRequest->FetchContacts($_sugarcrmSessionResponse, $_sugarcrmApiUrl,  
$_accountList['accountid']);  
  
        foreach ($_sugarcrmContactResponse->entry_list as $key => $value) {  
            $_userDetails = array(  
                'id' => $value->name_value_list->id->value,  
                'userorganizationid' => $_accountList['parentaccountid'],  
                'fullname' => $value->name_value_list->name->value,  
                'email' => $value->name_value_list->email1->value,  
                'phone' => $value->name_value_list->phone_work->value,  
                'userdesignation' => $value->name_value_list->title->value,  
                'dateline' => time(),  
            );  
  
            $this->SugarCRMUser->InsertUsers($_userDetails);  
        }  
    }  
}
```

```
*With:*
```

```
} else {  
    $_sugarcrmSessionResponse = $this->SugarCRMRestAPIRequest->LoginRequest($_sugarcrmUsername, $_sugarcrmPassword,  
$_sugarcrmApiUrl);  
  
    // Verify the login credentials for SugarCRM Rest API.  
    if ($_sugarcrmSessionResponse == 'fail') {  
        throw new SWIFT_Exception(SWIFT_INVALIDDATA);  
    } else {  
        $this->Load->Model('SugarCRMAccount:SugarCRMAccount');  
        $this->Load->Model('SugarCRMAccount:SugarCRMUser');  
  
        $_organizationList = $this->SugarCRMAccount->FetchOrganizations();  
        foreach ($_organizationList as $_accountList) {  
            //Sync sugar contacts with kayako user  
            $_sugarcrmContactResponse = $this->SugarCRMRestAPIRequest->FetchContacts($_sugarcrmSessionResponse, $_sugarcrmApiUrl,  
$_accountList['accountid']);
```

```

if (!is_array($_sugarcrmContactResponse)) {
    continue;
}

foreach ($_sugarcrmContactResponse as $key => $value)
{
    if(!isset($value->name_value_list)) {
        continue;
    }

    $_userDetails = array(
        'id'          => $value->name_value_list->id->value,
        'userorganizationid' => $_accountList['parentaccountid'],
        'fullname'    => $value->name_value_list->name->value,
        'email'       => $value->name_value_list->email1->value,
        'phone'       => $value->name_value_list->phone_work->value,
        'userdesignation' => $value->name_value_list->title->value,
        'dateline'    => time(),
    );

    $this->SugarCRMUser->InsertUsers($_userDetails);
}
}
}
}

```

Thanks and Regards,  
Anjali Sharma

**07/31/2014 02:06 am - Anjali Sharma**

- Status changed from New to Resolved
- Target version deleted (v1.1)

Greetings Ryan,

Issue has been resolved in SugarCRM v1.5.

Supported Versions:

Kayako Helpdesk : 4.64 and above

SugarCRM : 6.5.16

Thanks and Regards,  
Anjali Sharma