

Bug Report — VTENEXT 26.04 — Calendar: invitati email persi al salvataggio

Reporter: Pantarei SRL **Data:** 13 maggio 2026 **Componente:** Core VTE — modulo `Calendar` **File coinvolto:** `Smarty/templates/modules/Calendar/EventInviteUI.tpl` (funzione `addEmail`) **Versione VTE in cui è stato riprodotto:** 26.04 (build `vte23083_2711`) su PHP 8.3 + FPM **Severità:** Alta — perdita silenziosa di dati immessi dall'utente **Riproducibilità:** 100%

1. Sintesi (TL;DR)

Nel form di creazione/modifica di un evento del modulo **Calendar** (tab **Invita** → **Email**), quando l'utente aggiunge **due o più** indirizzi email come invitati premendo il pulsante **+** (o **Invio**) per ciascuno, al salvataggio dell'evento **viene memorizzata solo l'ultima email**. Le precedenti vengono visualizzate correttamente come "bubble" nel form ma vanno perse silenziosamente al submit.

La causa è una regressione nella funzione JavaScript `addEmail` del template `EventInviteUI.tpl`: l'algoritmo che ricostruisce la lista nascosta `CalendarCustomMail_idlist` sovrascrive le entry esistenti, perdendo gli `inviteeid` numerici già abbinati alle email precedenti. Solo l'ultima entry — quella appena aggiunta — viene formattata come `id::email`; tutte le altre diventano solo `email` (senza id), e di conseguenza vengono scartate dal backend.

Il bug è 100% lato client (JS), in un file `.tpl` del core VTE. Nessun warning lato UI e nessun errore lato server: l'utente non riceve alcun feedback sul fatto che gli invitati sono stati persi.

2. Step di riproduzione (minimal)

- VTE 26.04 / PHP 8.3 / FPM, login utente con accesso al modulo Calendar.
- Calendario → nuovo evento (oppure modifica un evento esistente).
- Tab **Invita** → **Email**.
- Digita nel campo input `bbit-cal-txtEmails` la prima email, es. `a@example.com`, e premi **+** (o Invio).
 - Atteso: appare la bubble `a@example.com` nella lista.
 - OK.
1. Digita la seconda email, es. `b@example.com`, e premi **+** (o Invio).
 - Atteso: appare la bubble `b@example.com` nella lista (la prima resta presente).
 - OK visivamente.
1. **Salva** l'evento.

2. Apri l'evento in **DetailView**, tab **Invita → Email**.

- Atteso: lista mostra entrambe le email.
- **Osservato**: lista mostra solo `b@example.com`. La prima è sparita.

1. Verifica diretta in DB:

```
``sql SELECT activityid, inviteeid, email FROM vte_invitees_emails WHERE activityid = <ID_EVENTO>;``
```

Una sola riga, quella corrispondente all'ultima email digitata.

3. Causa tecnica

3.1 Codice originale (buggy)

File: `Smarty/templates/modules/Calendar/EventInviteUI.tpl` Riferimento esatto: la stessa versione è presente in `vte_updater/3001/files3367/Smarty/templates/modules/Calendar/EventInviteUI.tpl` (distribuita come parte dell'aggiornamento a 26.04).

```
addEmail(id, mode, item) {
    // try to decode it
    if (typeof item == 'string') {
        try {
            item = JSON.parse(item);
        } catch (e) {
            return false;
        }
    }

    var idlist = this.$('#Calendar' + mode + '_idlist').val().split('|');
    idlist.forEach((value, index) => {
        if (value.includes('::')) {
            idlist[index] = value.split('::')[1];    // (A)
        }
    });

    // check for duplicates
    if (idlist.indexOf(item.value) >= 0) return false;

    // add the selected item
    var bubble = '<div id="Calendar' + mode + '_list_'+id+'" class="userSelectorBubble">';
    // ... (DOM) ...

    this.$('#Calendar' + mode + '_idlist').val(idlist.join('|') + id + '::' + item.value + '|');    // (B)
}
```

3.2 Cosa fa effettivamente

`#CalendarCustomMail_idlist` è il campo hidden che conserva la lista degli invitati email come stringa pipe-delimited di pair `inviteeid::email`, es.:

```
|112::a@example.com|113::b@example.com|
```

Al submit del form, `check_form()` in `modules/Calendar/script.js` legge quel campo e lo trasferisce in `inviteesid_cem`, che il backend (`CalendarHandler.php::insertIntoInviteeTable`) splitta con `explode("::", $entry)` per ottenere `inviteeid` ed `email` di ciascuna riga.

Il bug è nel blocco **(A)**: per ogni entry esistente con il separatore `::`, si fa `idlist[index] = value.split('::')[1]`, cioè si **sostituisce** la stringa `"id::email"` con la sola `email`, **perdendo l'id**. Effetto:

- entry pre-esistenti (1, 2, ..., N-1) → da `"id::email"` diventano `"email"` (senza id);
- entry nuova (N) → viene concatenata bene come `id::email` nel passaggio **(B)**.

3.3 Stringa risultante (osservata in runtime)

Dopo l'aggiunta di `a@example.com` (id 112) e poi `b@example.com` (id 113), il campo nascosto contiene:

```
|a@example.com|113::b@example.com|
```

invece del corretto:

```
|112::a@example.com|113::b@example.com|
```

3.4 Effetto lato backend

`CalendarHandler.php` riceve come POST:

```
inviteesid_cem = "a@example.com;113::b@example.com"
```

e per ogni entry esegue:

```
list($inviteeid, $customEmail) = explode("::", $inviteeidemail);
```

- entry `"a@example.com"` → `$inviteeid = "a@example.com"`, `$customEmail = null` → la riga viene scartata o inserita malformata (a seconda dei controlli upstream);
- entry `"113::b@example.com"` → ok.

Risultato: in `vte_invitees_emails` viene salvata **solo** la riga dell'ultima email.

3.5 Evidenza runtime catturata

POST intercettato lato handler `vtiger.entity.beforesave` (sessione di debug):

```
{
  "event": "vtiger.entity.beforesave",
  "activityid": 14413,
  "post_matching_keys": {
    "inviteesid_cem": "::;112::m.melucci@pantareisrl.com"
  }
}
```

(Nel caso reale `::` rappresenta un'entry pre-esistente già "rotta" da un save precedente; il pattern è

analogo al caso `email|113::email` descritto sopra.)

Riga ottenuta in DB dopo il save:

```
vte_invitees_emails:
  activityid | inviteeid | email
-----+-----+-----
14413      | 112      | m.melucci@pantareisrl.com ← unica riga
```

4. Fix applicato (lato client, retro-compatibile)

Si conservano gli `id::email` esistenti, si deduplica per **email**, e si appende il nuovo pair preservando la forma corretta della lista.

4.1 Codice corretto

```

addEmail(id, mode, item) {
  // try to decode it
  if (typeof item == 'string') {
    try {
      item = JSON.parse(item);
    } catch (e) {
      return false;
    }
  }

  var rawBefore = this.$('#Calendar' + mode + '_idlist').val();
  // FIX: conserva gli "id::email" esistenti, deduplica per email.
  // Bug originale: il forEach sovrascriveva idlist[i] con la sola email
  // perdendo l'id; il backend salvava solo l'ultima riga in vte_invitees_emails.
  var rawList = rawBefore.split('|').filter(function(x){ return x != ''; });
  var emailsExisting = rawList.map(function(v){
    return v.indexOf('::') >= 0 ? v.split('::')[1] : v;
  });
  if (emailsExisting.indexOf(item.value) >= 0) return false;

  // add the selected item bubble (invariato)
  var bubble = '<div id="Calendar' + mode + '_list_' + id + '" class="userSelectorBubble">';
  bubble += '<div class="userSelectorBubbleImg">';
  bubble += '<i class="vteicon md-sm md-link">email</i>';
  bubble += '</div>';
  bubble += '<div class="userSelectorBubbleInfo">';
  bubble += '<div class="userSelectorBubbleTitle">' + item.value + '</div>';
  bubble += '</div>';
  bubble += '<div class="userSelectorBubbleRemove" id="CalendarEmail_list_' + id + '_remove">'
    + '<i class="vteicon md-sm md-link userSelectorBubbleRemoveIcon" title="Delete"
onClick="removeInvite(\'' + id + '\', \' ' + mode + '\');">clear</i>'
    + '</div>';
  bubble += '</div>';
  this.$allList.append(bubble);

  rawList.push(id + '::' + item.value);
  var rawAfter = '|' + rawList.join('|') + '|';
  this.$('#Calendar' + mode + '_idlist').val(rawAfter);
}

```

4.2 Differenze sostanziali rispetto all'originale

Aspetto	Originale (buggy)	Fix
Parsing lista	`val().split('`	`)'` (entry vuote agli estremi)
Lista in memoria	<code>idlist</code> con stringhe <code>id::email</code> modificate inline a <code>email</code>	<code>rawList</code> array di stringhe <code>id::email</code> immutate
Calcolo "esistenti"	implicito (dopo aver perso gli id)	<code>emailsExisting</code> derivato senza alterare <code>rawList</code>

Aspetto	Originale (buggy)	Fix					
Append nuova entry	<code>`idlist.join('</code>	<code>') + id + '::' + item.value + '</code>	<code>`</code> (concatenazione senza separatore pulito)	<code>rawList.push(id + '::' + item.value)</code> poi <code>`</code>	<code>' + rawList.join('</code>	<code>) + '</code>	<code>`</code>
Side effect su lista	Sì (perdita id pre-esistenti)	No					

4.3 Patch unificato

```

--- Smarty/templates/modules/Calendar/EventInviteUI.tpl (original)
+++ Smarty/templates/modules/Calendar/EventInviteUI.tpl (fixed)
@@ -550,30 +550,29 @@
    addEmail(id, mode, item) {
        // try to decode it
        if (typeof item == 'string') {
            try {
                item = JSON.parse(item);
            } catch (e) {
                return false;
            }
        }

-
-        var idlist = this.$('#Calendar' + mode + '_idlist').val().split('|');
-        idlist.forEach((value, index) => {
-            if (value.includes('::')) {
-                idlist[index] = value.split('::')[1];
-            }
-        });
-
-        // check for duplicates
-        if (idlist.indexOf(item.value) >= 0) return false;
+
+        var rawBefore = this.$('#Calendar' + mode + '_idlist').val();
+        // FIX: conserva gli "id::email" esistenti, deduplica per email.
+        var rawList = rawBefore.split('|').filter(function(x){ return x !== ''; });
+        var emailsExisting = rawList.map(function(v){
+            return v.indexOf('::') >= 0 ? v.split('::')[1] : v;
+        });
+        if (emailsExisting.indexOf(item.value) >= 0) return false;

        // add the selected item bubble
        var bubble = '<div id="Calendar' + mode + '_list_'+id+'"'
class="userSelectorBubble">';
        bubble += '<div class="userSelectorBubbleImg">';
        bubble += '<i class="vteicon md-sm md-link">email</i>';
        bubble += '</div>';
        bubble += '<div class="userSelectorBubbleInfo">';
        bubble += '<div class="userSelectorBubbleTitle">' + item.value + '</div>';
        bubble += '</div>';
        bubble += '<div class="userSelectorBubbleRemove"
id="CalendarEmail_list_'+id+'_remove">'
            + '<i class="vteicon md-sm md-link userSelectorBubbleRemoveIcon" title="Delete"
onClick="removeInvite(\''+id+'\',' + mode + '\');">clear</i>'
            + '</div>';
        bubble += '</div>';
        this.$allList.append(bubble);

-        this.$('#Calendar' + mode + '_idlist').val(idlist.join('|') + id + '::' + item.value
+ '|');
+        rawList.push(id + '::' + item.value);
+        var rawAfter = '|' + rawList.join('|') + '|';
+        this.$('#Calendar' + mode + '_idlist').val(rawAfter);
    }

```

5. Verifica post-fix (runtime)

Stesso flusso, con il fix applicato.

POST intercettato a `vtiger.entity.beforesave`:

```
{
  "event": "vtiger.entity.beforesave",
  "activityid": 14413,
  "post_matching_keys": {
    "inviteesid_cem": "112::m.melucci@pantareisrl.com;113::mrt.mazzini@gmail.com"
  }
}
```

Stato della tabella `vte_invitees_emails` dopo `aftersave`:

activityid	inviteeid	email
14413	112	m.melucci@pantareisrl.com
14413	113	mrt.mazzini@gmail.com

DetailView dell'evento (riproduzione visiva):

Email	m.melucci@pantareisrl.com	Partecipazione: Pendente
	mrt.mazzini@gmail.com	Partecipazione: Pendente

Entrambi gli invitati conservati. Fix CONFERMATO.

6. Note per il team VTE

- Il file modificato è **core**: il fix non è distribuibile in un modulo SDK senza override (`SDK::setSmartyTemplate`). Sarebbe preferibile applicarlo upstream nel prossimo build VTE.
- Il fix è **retro-compatibile**: non cambia firma di `addEmail`, non altera il payload atteso dal backend, non tocca altri rami (`Users`, `Contacts`, `Leads`).
- Il fix non introduce dipendenze nuove.
- Suggerirei in più di:
- validare il formato di ogni entry letta da `#CalendarCustomMail_idlist` (skip se manca `::` o se l'`id` non è numerico);
- aggiungere un controllo equivalente lato server in `CalendarHandler.php::insertIntoInviteeTable` per non inserire righe con `email = NULL` quando `inviteeid` non è numerico.

7. Allegati disponibili su richiesta

- log NDJSON pre-fix e post-fix dell'handler `vtiger.entity.beforesave` / `aftersave`;
- screenshot DetailView pre-fix e post-fix;
- dump SQL di `vte_invitees_emails` pre-fix e post-fix.