OpSense Bildirimleri

Mopuan entegrasyonu için kullanım kılavuzu

opsenseapi · Sürüm 1.0 · 6 Kasım 2026

1. Bu servis ne işe yarar

OpSense, bir çalışanın ilgilenmesi gereken bir durum olduğunda o çalışan için bir bildirim oluşturur. Örneğin:

- yeni bir iş emri,
- çalışana atanmış bir iş emrine eklenen not,
- incelenmesi gereken bir iş emri denetimi,
- ekibinden birinin gönderdiği panik uyarısı.

`SRVGetNotifications` bu bildirimleri döndürür, böylece onları Mopuan'da gösterebilirsiniz. Servis yalnızca okuma yapar. Çağırmak hiçbir şeyi değiştirmez.

Push bildirimi yalnızca bir uyarıdır. Tam kayıt bu servisten gelen listedir. Bildirim ekranı açıkken bir push gelirse listeyi yeniden yükleyin.

2. Kimlik doğrulama

Şirketin `api_key` değerini, diğer OpSense servislerinde olduğu gibi normal bir istek parametresi olarak, POST alanı şeklinde gönderin. HTTP başlığında veya URL'de göndermeyin.

Anahtar, çağrının hangi şirket için yapıldığını belirler. Yalnızca o şirketin bildirimlerini alırsınız.

Durum	Yanıt
`api_key` eksik	`error_code` 10015
`api_key` geçersiz veya pasif	`error_code` 10016

Her yanıt aynı şekilde sarılır:

```
{"result": { ... başarılı yanıt veya hata gövdesi ... }}
```

3. İstek

```
POST https://<api-host>/v1/SRVGetNotifications
Content-Type: application/x-www-form-urlencoded
```

Parametreleri **form alanları** olarak gönderin. JSON istek gövdesi okunmaz: alanlar JSON olarak gönderilirse servis onları dikkate almaz ve şirketin tüm bildirimlerini döndürür.

Alan	Zorunlu	Notlar

`api_key`	**Zorunlu**	Bkz. Kimlik doğrulama.
`mopuan_user_id`	İsteğe bağlı	Yalnızca bu çalışanın bildirimlerini döndürür. Sizin Mopuan çalışan id'niz.
`user_id`	İsteğe bağlı	Yalnızca bu çalışanın bildirimlerini OpSense kullanıcı id'si ile döndürür. `mopuan_user_id` gönderilirse dikkate alınmaz.
`limit`	İsteğe bağlı	Sayfa boyutu: yalnızca en yeni `limit` bildirimi, en yeniden eskiye doğru döndürür. 0'dan büyük bir tam sayı.
`before_id`	İsteğe bağlı	Yalnızca bu id'den daha eski bildirimleri döndürür. Sonraki sayfayı almak için önceki sayfanın `next_before_id` değerini gönderin.

Ne dönüyor

Gönderdiğiniz	Aldığınız
Yalnızca `api_key`	Şirketin tüm bildirimleri, en yeniden eskiye doğru.
`mopuan_user_id`	O çalışanın tüm bildirimleri, en yeniden eskiye doğru.
`mopuan_user_id`, `limit=30`	İlk sayfa: o çalışanın en yeni 30 bildirimi, en yeniden eskiye doğru, ayrıca `has_more` ve `next_before_id`.
`mopuan_user_id`, `limit=30`, `before_id=<next_before_id>`	Sonraki sayfa: önceki sayfadan sonraki 30 bildirim, ayrıca `has_more` ve `next_before_id`.

`limit` gönderilmezse, 0 ise veya sayı değilse ilk iki satırdaki gibi tüm bildirimler döner ve yanıtta `has_more` veya `next_before_id` bulunmaz.

Sayfalama

1. İlk sayfayı `limit` ile ve `before_id` olmadan isteyin.
2. Çalışan listenin sonuna geldiğinde ve `has_more` `true` ise, sonraki sayfayı aynı `limit` ile ve `before_id` değerini `next_before_id` yaparak isteyin. Yeni bildirimleri listenin sonuna ekleyin.
3. `has_more` `false` olduğunda daha eski bildirim yoktur. İstemeyi bırakın.
4. Yenilemek için ilk sayfayı yeniden isteyin (`before_id` olmadan) ve listeyi değiştirin.

Çalışan listeyi kaydırırken oluşan bildirimler sayfaları kaydırmaz. Hiçbir bildirim tekrarlanmaz veya atlanmaz. Yeni bildirimler bir sonraki yenilemede görünür.

Örnekler

İlk sayfa, en yeni 30 bildirim:

```
curl -X POST "https://<api-host>/v1/SRVGetNotifications" \
 -d "api_key=<api_key>" \
 -d "mopuan_user_id=345" \
 -d "limit=30"
```

Sonraki sayfa:

```
curl -X POST "https://<api-host>/v1/SRVGetNotifications" \
 -d "api_key=<api_key>" \
 -d "mopuan_user_id=345" \
 -d "limit=30" \
 -d "before_id=25481"
```

Çalışanın tüm bildirimleri, sayfalama olmadan:

```
curl -X POST "https://<api-host>/v1/SRVGetNotifications" \
 -d "api_key=<api_key>" \
 -d "mopuan_user_id=345"
```

4. Yanıt

`limit` içeren bir isteğin yanıtı:

```
{
 "result": {
   "result": "success",
   "data": [
     {
       "id": 25510,
       "user_id": 87,
       "mopuan_user_id": 345,
       "link": "https://<portal-host>/calendar",
       "title": "New Note on Work Order",
       "description": "New note added by Jane Smith",
       "added_date": "2026-11-06 14:02:05"
     }
   ],
   "has_more": true,
   "next_before_id": 25481
 }
}
```

Alan	Notlar
`data`	Bildirimler. Hiç bildirim yoksa boş bir liste.
`has_more`	Yalnızca `limit` gönderildiğinde. Daha eski bildirim varsa `true`.
`next_before_id`	Yalnızca `limit` gönderildiğinde. Sonraki sayfa için `before_id` olarak gönderin. `has_more` `false` ise `null`.

`data` içindeki her bildirim:

Alan	Notlar
`id`	Bildirim id'si. Daha yeni bir bildirimin id'si her zaman daha büyüktür.
`user_id`	Bildirimin ait olduğu çalışanın OpSense kullanıcı id'si.
`mopuan_user_id`	Aynı çalışanın Mopuan çalışan id'si. Çalışan Mopuan'a bağlı değilse boştur.

`link`	Bildirimin ilgili olduğu OpSense portal sayfası.
`title`	Kısa başlık. Düz metin.
`description`	Tek satırlık açıklama. Düz metin.
`added_date`	Bildirimin oluşturulma zamanı, `YYYY-MM-DD HH:MM:SS`, şirketin saat diliminde.

5. Öneriler

- **Her zaman `mopuan_user_id` gönderin.** Göndermezseniz şirketin tüm bildirimlerini alırsınız. Bu hem yavaştır hem de çalışanlara kendilerine ait olmayan bildirimleri gösterir.
- **Liste ekranlarında sayfalama kullanın.** `limit` gönderin (bir ekran için 30 ile 50 arası yeterlidir) ve daha eski sayfaları `before_id` ile yükleyin. Çalışanın kaç bildirimi olursa olsun her yanıt küçük ve hızlı kalır.
- **Yanıttaki sırayı koruyun.** Bildirimler her zaman `id`'ye göre, en yeniden eskiye doğru gelir. Kendiniz sıralarsanız `added_date`'e göre değil, `id`'ye göre sıralayın: birden fazla bildirim aynı saniyeye düşebilir.
- **`title` ve `description` alanlarını metin olarak gösterin.** HTML olarak işlemeyin.