مستندات فنی افزونه
ns-didar

سیستم مدیریت درخواست‌های وردپرس و یکپارچه‌سازی با Didar CRM

نسخهٔ افزونه: `1.7.2`

حداقل `6.4` :WordPress | حداقل `7.4` :PHP

تاریخ تولید مستندات: ۲۷ مرداد ۱۴۰۵

فهرست مطالب

1. 1. معرفی و نمای کلی
2. 2. معماری
3. 3. فرم‌ها و درخواست‌ها
4. 4. کاربران و کنترل دسترسی
5. 5. پروفایل کاربر و سازگاری Digits
6. 6. یکپارچگی با Didar CRM
7. 7. جریان‌های همگام‌سازی
8. 8. Webhook
9. 9. گردش‌کار و وضعیت‌ها
10. 10. فایل‌ها
11. 11. رخدادها و diagnostics
12. 12. تنظیمات
13. 13. انتقال تنظیمات
14. 14. Cron، پردازش پس‌زمینه، retry و lock
15. 15. امنیت
16. 16. پایگاه‌داده و ذخیره‌سازی
17. 17. Hookها و نقاط توسعه
18. 18. استقرار
19. 19. رفع اشکال
20. 20. راهنمای توسعه‌دهنده
21. ۲۱. محدودیت‌های فعلی

فصل 1

معرفی و نمای کلی

`ns-didar` سامانهٔ درخواست با فرم‌های ازپیش‌تعریف‌شده است؛ form builder عمومی نیست. فرم‌ها از `Didar_Form_Registry` خوانده می‌شوند و هر ثبت یک `didar_submission` است. کاربر واردشده با `[didar_form type="…"]` ثبت و با `[didar_submissions]` سابقهٔ خود را می‌بیند.

همگام‌سازی Didar فقط با API پیکربندی‌شده اجرا می‌شود. حذف Deal در Didar انجام نمی‌شود، زیرا API حذفِ تأییدشده در کد وجود ندارد؛ حذف یا trash محلی فقط ثبت رویداد/diagnostic می‌شود.

فصل 2

معماری

نمودار جریان

```
B[didar.php]-->P[Didar_Plugin]
P-->R[Didar_Form_Registry]
P-->S[Didar_Submission_Service]
P-->F[Didar_File_Service]
P-->M[Didar_Sync_Manager]
M-->A[Didar_Api_Client]
M-->W[Didar_Workflow_Manager]
S-->E[Didar_Event_Log]
M-->L[Didar_Logger]
```

کلاس	مسئولیت
`Didar_Plugin`	boot، lifecycle و سلامت workerها
`Didar_Post_Type`	CPT `didar_submission`
`Didar_Form_Registry` / `Didar_Reference_Data`	فرم‌ها و دادهٔ مرجع
`Didar_Field_Renderer` / `Didar_Validator`	نمایش و اعتبارسنجی
`Didar_Submission_Service`	ایجاد، تغییر، workflow و دسترسی
`Didar_File_Service`	رکورد و ذخیرهٔ فایل خصوصی
`Didar_Sync_Manager` / `Didar_Field_Mapper`	صف و تبدیل داده به Didar
`Didar_Event_Log` / `Didar_Logger`	رخداد کسب‌وکاری و diagnostic

activation نقش‌ها، schemaها و scheduleها را آماده می‌کند؛ deactivation schedulehای افزونه را پاک می‌کند. bootstrap عادی `maybe_repair` / `maybe_upgrade` و بازیابی schedule را اجرا می‌کند.

فصل 3

فرم‌ها و درخواست‌ها

فرم‌ها در `Didar_Form_Registry` تعریف شده‌اند؛ renderer و validator ورودی مرورگر را مرجع نمی‌دانند. نوع‌های فعلی `complaint_suggestion`، `traveler_evaluation`، `embassy_appointment`، `consultation` و `visa_request` هستند.

ثبت در `Didar_Submission_Service` یک پست `didar_submission` با `post_author` کاربر جاری می‌سازد. `_didar_form_type` نوع و `_didar_fields` مقادیر اعتبارسنجی‌شده را نگه می‌دارد. ویرایش فرانت‌اند فقط برای درخواست قابل‌دسترسی و قابل‌ویرایش انجام می‌شود؛ درخواست تکمیل‌شده قابل‌ویرایش نیست.

`Didar_Request_Search` عبارت جست‌وجو را تا ۱۰۰ نویسه پاک‌سازی می‌کند. فهرست فرانت‌اند مالکیت را در query اعمال و تعداد هر صفحه را از تنظیمات، بین ۱ تا ۱۰۰، می‌گیرد.

کاربران و کنترل دسترسی

عملیات	مشتری	didar_colleague	didar_broker	Administrator
ثبت/دیدن درخواست خود	بله، پس از ورود	بله	بله	بله
پنل Didar	خیر	خیر	بله	بله
درخواست‌های تیم	خیر	خیر	بله	بله
ویرایش فرانت‌اندِ درخواست خود	تا پیش از وضعیت عمومی completed	تا پیش از completed	با didar_edit_requests و edit_post	با همان capabilityها
وضعیت داخلی، مسئول و یادداشت داخلی	خیر	فقط مشاهدهٔ اختیاری history/workflow خود؛ بدون تغییر	بله	بله
تنظیمات	خیر	خیر	بله	بله

کنترل‌ها capabilityمحورند. didar_colleague فقط didar_colleague_access، read، didar_view_own_internal_workflow و didar_view_own_request_history می‌گیرد؛ مشاهدهٔ workflow/history داخلیِ درخواست خودش نیز منوط به colleague_can_view_internal_history است. didar_broker capabilityهای مدیریتی Didar را می‌گیرد. افزونه کاربر WordPress را از webhook Person نمی‌سازد و هیچ کاربری را Administrator نمی‌کند.

فصل 5

پروفایل کاربر و سازگاری Digits

`Didar_User_Profile` فرم پروفایل کاربر جاری را ارائه می‌کند. وضعیت هر فیلد (`readonly`، `editable` یا `disabled`) از `profile_field_states` می‌آید: `gender`، `email`، `display_name`، `last_name`، `first_name`، `mobile` و `profile_image`.

`first_name` و `last_name` در WordPress canonical هستند؛ فقط اگر خالی باشند، mapper به‌ترتیب از `billing_first_name` و `billing_last_name` (سازگاری Digits/WooCommerce) می‌خواند. موبایل ابتدا از `digits_phone` و سپس از ترکیب `digt_countrycode` و `digits_phone_no` خوانده می‌شود؛ افزونه این کلیدها را نمی‌نویسد. اگر موبایل readonly باشد، تغییر آن از فرم پروفایل پذیرفته نمی‌شود.

فرم `display_name` را ویرایش می‌کند، نه `nickname`. هنگام `nickname`، `wp_update_user` از مقدار موجود، سپس display name و در نهایت login تعیین می‌شود؛ `user_login` تغییر نمی‌کند. `user_nicename` نامعتبر قدیمی با `user-{id}` ترمیم می‌شود.

تصویر فقط در حالت editable، با فرمت JPG/PNG/GIF/WebP و سقف ۵MB در Media Library ذخیره می‌شود. پس از ذخیرهٔ پروفایل، Person sync فوری تلاش و در خطا برای retry صف می‌شود.

یکپارچگی با Didar CRM

`Didar_Api_Client` درخواست‌های POST به `/api/contact/getbyphonenumber`، `/api/contact/save`، `/api/deal/search_v2` و `/api/deal/save_v2` ارسال می‌کند. کلید API در `didar_settings` است؛ آن را در مستندات یا log عمومی قرار ندهید.

Person

هویت محلی WordPress User است و شناسهٔ خارجی در `_didar_person_id` user meta پایدار می‌ماند. sync به owner پیش‌فرض Didar و موبایل نیاز دارد. lookup دقیق موبایل و بررسی شناسهٔ ذخیره‌شده پیش از create/update انجام می‌شود؛ ابهام Person، ایجاد Deal را متوقف می‌کند.

Deal

شناسهٔ اصلی `_didar_deal_id` است. در نبود آن، بازیابی تنها از custom field شناسهٔ WordPress submission انجام می‌شود؛ matching با نام Deal، ایمیل، موبایل یا Person به‌عنوان هویت Deal اجرا نشده است. Deal فقط بعد از تأیید صفر match ساخته می‌شود. نگاشت فیلد، pipeline/stage، owner، وضعیت و لینک فایل از تنظیمات و `Didar_Field_Mapper` می‌آید.

محدودیت lifecycle: تغییر به trash یا حذف درخواست در WordPress ثبت می‌شود، اما افزونه برای حذف/archive Deal در Didar درخواست remote نمی‌فرستد؛ مسیر API امن و تأییدشده‌ای برای آن در پیاده‌سازی فعلی استفاده نشده است.

فصل 7

جریان‌های همگام‌سازی

نمودار جریان

```
participant U as WordPress User
participant Q as didar_process_user_sync
participant D as Didar Person
U->>Q: user_register / تغییر meta پروفایل / موبایل
Q->>D: resolve سپس save
D-->>Q: Person ID
Q->>U: _didar_person_id و state
```

نمودار جریان

```
participant S as Submission
participant Q as didar_process_sync
participant D as Didar Deal
S->>Q: state=pending + trace_id
Q->>Q: lock 120s
Q->>D: Person، سپس resolve/create/update Deal
D-->>Q: Deal ID
Q->>S: _didar_deal_id; state=synced
```

ثبت، تغییر، workflow change و ذخیرهٔ admin sync را trigger می‌کنند. `manual_sync` همان مسیر مرکزی را فوری اجرا می‌کند. retry تا ۱۰ تلاش با تأخیر خطیِ حداقل ۶۰ ثانیه و سقف یک ساعت است؛ sweep پنج‌دقیقه‌ای رویداد گم‌شده را بازیابی می‌کند. webhook با suppression مانع بازصف‌شدن update ورودی می‌شود.

snapshot، Webhook Deal فیلدهای نگاشت‌شده، status داخلی/عمومی و owner را در درخواستِ متصل به‌روزرسانی می‌کند. اگر رویداد از نوع create باشد و `form_type` و WordPress user از فیلدهای سیستمی resolve شوند، یک درخواست محلی ساخته می‌شود؛ webhook Person فقط ثبت diagnostic دارد و پروفایل/کاربر WordPress را تغییر یا ایجاد نمی‌کند.

Webhook

دو route ثبت می‌شود: `POST /wp-json/didar/v1/webhook/{secret}` که `{secret}` دقیقاً ۶۴ نویسهٔ hex است، و route قدیمی `POST /wp-json/didar/v1/webhook` . مسیر عملیاتی باید URL تولیدشده از تنظیمات باشد. route قدیمی فقط وقتی `didar_webhook_legacy_enabled` فعال باشد و `x-didar-webhook-token` header با secret پیکربندی‌شده برابر باشد، authenticate می‌شود.

نمودار جریان

```
participant D as Didar
participant W as Webhook REST
participant S as درخواست WordPress
D->>W: POST JSON و secret
W->>W: authentication، rate limit، dedupe
W->>S: اعمال snapshot Deal با suppression
W-->>D: 200 received یا خطای REST
```

ورودی `meta` ، `data` ، `application/json` و شناسه/نوع موجودیت لازم را اعتبارسنجی می‌کند، rate limit برابر ۱۲۰ درخواست در ۶۰ ثانیه دارد و event ID را در `didar_seen_webhooks` (حداکثر ۵۰۰ مورد) dedupe می‌کند. فقط actionهای create/update برای Person و Deal پشتیبانی می‌شوند. webhook Person صرفاً diagnostic ثبت می‌کند و کاربر یا پروفایل WordPress ایجاد/ویرایش نمی‌کند. Deal ورودی ابتدا با `_didar_deal_id` و سپس custom field شناسهٔ submission متصل می‌شود؛ در create، در صورت resolve شدن form/user از فیلدهای سیستمی، درخواست محلی می‌سازد. payload بدفرم، secret نامعتبر، content type نادرست، نرخ زیاد یا رویداد پشتیبانی‌نشده با خطای REST رد می‌شود؛ رویداد تکراری پاسخ ۲۰۰ با `duplicate` می‌گیرد.

فصل 9

گردش‌کار و وضعیت‌ها

`Didar_Workflow_Manager` برای هر workflow `form_type` شامل `pipeline_id` و statusها را می‌خواند. هر status کلید پایدار، `stage_id`، label، ترتیب و امکان `is_default` دارد. mapping معکوس pipeline/stage نیز برای webhook وجود دارد.

وضعیت داخلی در `_didar_internal_status` و وضعیت عمومی در `_didar_public_status` (با سازگاری `_didar_status`) نگهداری می‌شود. مسئول در `_didar_assigned_user_id` است؛ امکان تخصیص به capability نیاز دارد. اگر workflow همان فرم ناقص باشد، sync به fallback قدیمی نمی‌رود و pending/retry می‌ماند.

فصل 10

فایل‌ها

فایل فرم در جدول اختصاصی `Didar_File_Service::table_name()` و زیرشاخهٔ `didar-private` از `wp_upload_dir()` ذخیره می‌شود؛ attachment Media Library برای فایل‌های درخواست ساخته نمی‌شود. فایل موقت حداکثر یک روز عمر دارد و worker روزانه پاک‌سازی می‌کند. حذف submission فایل‌های وابسته را پاک می‌کند.

AJAXهای `didar_upload_file` و `didar_remove_file` ورود، nonce، فرم/فیلد معتبر، مالکیت، نوع/اندازه و record را کنترل می‌کنند. دانلود امن از `admin-post.php?action=didar_download_file` و مجوز دسترسی انجام می‌شود. حالت URL `direct` مستقیم می‌دهد؛ بنابراین دسترسی فایل به حفاظت وب‌سرور وابسته است. حالت پیش‌فرض `secure` است و برای دادهٔ حساس توصیه می‌شود.

رخدادها و diagnostics

`Didar_Event_Log` تاریخچهٔ افزایشی کسب‌وکاری هر درخواست را در جدول اختصاصی نگه می‌دارد؛ نمونه‌ها `request_trashed`، `didar_sync_failed` و `request_deleted` هستند. `_didar_last_updated_at` برای مرتب‌سازی/نمایش به‌روز می‌شود.

diagnostic `Didar_Logger` عملیاتی و `trace_id`/correlation ثبت می‌کند. رویدادهای sync شامل `api_request`، `sync_execute`، `sync_worker_started`، `sync_immediate_started`، `sync_queue_persisted`، `sync_retry_exhausted`، `sync_retry_scheduled`، `api_response` و `sync_permanent_failure` هستند. endpointهای Person payload شخصی را ساختاریافته و بدون مقادیر پروفایل log می‌کنند.

برای «درخواست ذخیره شد اما Deal نیست»، ابتدا state/meta، سپس صف، worker، خطای mapping/Person و پاسخ API را با trace یکسان بررسی کنید.

فصل 12

تنظیمات

تنظیمات اصلی در `didar_settings` option هستند: owner، secret webhook، `didar_api_key`، field پیش‌فرض، mappingهای workflow، `didar_broker_user_map`، Person/Dealهای فرم، ID فیلدهای سیستمی، states پروفایل، mode دانلود فایل، pagination و اجازهٔ مشاهدهٔ history داخلی همکار.

cacheهای custom field، pipeline و user Didar runtime هستند و منبع پیکربندی قابل‌انتقال نیستند. API key، pipeline/stage، secret و user mapping محیط‌وابسته‌اند؛ پیش از تغییر، metadata دیدار را refresh و mapping را اعتبارسنجی کنید.

انتقال تنظیمات

export/import `Didar_Settings_Transfer` پیکربندی portable را با allowlist انجام می‌دهد و preview پیش از apply دارد. حالت‌های `merge` و `replace` پشتیبانی می‌شوند و پیش از اعمال نسخهٔ پشتیبان تنظیمات می‌سازند.

نوع	Export/Import
API key و webhook secret	خیر
cacheهای runtime	خیر
state همگام‌سازی، Person ID و Deal ID	خیر
workflow و mappingهای قابل‌حمل	بله، با اعتبارسنجی

نگاشت کاربران در export به descriptor شامل login/email و شناسهٔ Didar تبدیل می‌شود و import فقط user موجود را ابتدا با login و سپس email resolve می‌کند؛ user نمی‌سازد. انتقال Administrator تخصیص نمی‌دهد و خارج از allowlist در `wp_options` یا usermeta نمی‌نویسد.

Cron، پردازش پس‌زمینه، retry و lock

`didar_process_sync` و `didar_process_user_sync` هم رویداد تک‌بارهٔ شناسه‌دار و هم worker بدون آرگومان هر پنج دقیقه دارند. sweep در هر نوبت حداکثر ۱۰ submission یا user pending را پردازش می‌کند. `Didar_Plugin::ensure_runtime_workers()` در bootstrap عادی نیز scheduleهای گم‌شده و cleanup روزانهٔ `didar_cleanup_temporary_uploads` را بازمی‌سازد.

نمودار جریان

```
[*] --> pending: queue
pending --> synced: API موفق
pending --> pending: خطای retryable / retry
pending --> failed: خطای دائمی یا اتمام ۱۰ تلاش
pending --> pending: sweep پنج‌دقیقه‌ای
```

lock هر submission در option با پیشوند `didar_submission_sync_lock_` و TTL ۱۲۰ ثانیه است. `spawn_cron()` فقط تلاش سریع پس از persist است؛ پایداری به `didar_sync_state_` و sweep متکی است. برای WP-production، Cron واقعی/سیستمی را طوری تنظیم کنید که بازدید سایت شرط اجرای job نباشد.

نمودار جریان

```
P[pending state] --> O[رویداد یکباره]
O --> R{موفق؟}
R -->|بله| S[synced]
R -->|خیر، کمتر از ۱۰ تلاش| B[backoff و retry]
B --> P
P --> W[sweep پنج‌دقیقه‌ای]
W --> R
R -->|خطای دائمی یا اتمام تلاش| F[failed]
```

امنیت

کنترل‌های فعلی برای کاهش ریسک طراحی شده‌اند؛ تضمین «امنیت صددرصد» نیستند.

تهدید	کنترل
CSRF	nonceهای مشخص در فرم، AJAX و admin
دسترسی غیرمجاز	ورود، capability و بررسی مالکیت/نوع پست
ورودی جعلی	sanitize/validate، allowlist، registry سمت‌سرور
XSS	escaping متناسب هنگام render
دانلود فایل	record معتبر، مجوز و nonce در mode امن
path traversal/upload	نام و path سرور، MIME/type/size allowlist
webhook جعلی/تکراری	dedupe، rate limit، secret و suppression loop
Deal تکراری	lookup، external ID دقیق submission ID و lock
نشت secret در انتقال	allowlist و حذف key/secret/runtime
نشت PII در log Person	ثبت ساختار payload به‌جای مقدار پروفایل

حالت دانلود `direct` سطح ریسک بیشتری دارد؛ برای درخواست‌های حساس `secure` را نگه دارید. استقرار کثیف می‌تواند فایل قدیمی و رفتار امنیتی ناسازگار باقی بگذارد؛ جایگزینی تمیز الزامی عملیاتی است.

پایگاه‌داده و ذخیره‌سازی

کلید/محل	موجودیت	کاربرد
`didar_submission`	`wp_posts`	درخواست
`_didar_form_type`	post meta	نوع فرم
`_didar_fields`	post meta	دادهٔ فرم
`_didar_deal_id`	post meta	شناسهٔ Deal
`_didar_sync_state`	post meta	state، trace، attempts و زمان‌ها
`_didar_assigned_user_id`	post meta	مسئول
`_didar_public_status` / `_didar_internal_status`	post meta	وضعیت‌ها
`_didar_person_id`	user meta	شناسهٔ Person
`_didar_person_sync_state`	user meta	وضعیت Person sync
`didar_settings`	option	تنظیمات
`didar_seen_webhooks`	option	dedupe webhook

جدول event log و جدول فایل توسط schema manager ایجاد/verify می‌شوند. منبع authoritative برای هویت کاربر، WordPress User است؛ Person/Deal صرفاً شناسهٔ خارجی پایدار دارند. قفل sync و cacheهای Didar optionهای runtimeاند.

Hookها و نقاط توسعه

Hook مصرف‌شده	کاربرد
`updated_user_meta`، `added_user_meta`، `register_new_user`، `user_register`	صف Person sync
`rest_api_init`	ثبت webhook
`cron_schedules`	interval پنج‌دقیقه‌ای
`pre_trash_post`، `pre_delete_post`، `before_delete_post`	حفاظت/ثبت lifecycle
`posts_search`	جست‌وجوی درخواست
`update_option_didar_settings`	حفاظت storage فایل

Hook منتشرشده	پارامتر
`didar_submission_created`	`$post_id`
`didar_submission_updated`	`$post_id`
`didar_submission_workflow_changed`	`$post_id, $changed_keys`

Cronهای افزونه `didar_cleanup_temporary_uploads`، `didar_process_user_sync`، `didar_process_sync` و `didar_backfill_last_updated` هستند. shortcodeها و AJAX endpointها public extension API رسمی اعلام نشده‌اند؛ به signature داخلی آن‌ها وابستگی سخت ندهید.

استقرار

1. از پوشهٔ فعلی افزونه و پایگاه‌داده backup بگیرید.
2. پوشهٔ قدیمی `ns-didar` را rename یا به‌طور کنترل‌شده حذف کنید.
3. بستهٔ تمیز را در `wp-content/plugins/ns-didar/` استخراج کنید.
4. وجود `wp-content/plugins/ns-didar/didar.php` و نبود پوشهٔ تودرتوی `ns-didar/ns-didar/` را تأیید کنید.
5. تنظیمات را بازبینی کنید؛ key و secret را در deployment log چاپ نکنید.
6. پنل، فرم، پروفایل/webhook، submission/Deal sync، Person sync و diagnostics را تست عملیاتی کنید.

جایگزینی روی پوشهٔ کثیف توصیه نمی‌شود. نسخهٔ فعال پس از bootstrap معمول، workerهای لازم را self-heal می‌کند؛ بااین‌حال سلامت WP-Cron را بررسی کنید. rollback: پوشهٔ جدید را کنار بگذارید، نسخهٔ قبلیِ backup را با همان مسیر بازگردانید، سپس schema/settings/workerها را بررسی کنید. دادهٔ پایگاه‌داده را بدون برنامهٔ مهاجرت overwrite نکنید.

رفع اشکال

درخواست ثبت شد اما Deal ساخته نشد

`_didar_sync_state` و سپس `sync_execute` / `sync_worker_started`، `sync_queue_persisted`، `person_conflict`، mapping/`settings_check` و `api_request` / `api_response` را با trace بررسی کنید. API key، owner پیش‌فرض و pipeline/stage فرم را تأیید کنید؛ DB را مستقیم ویرایش نکنید.

Person همگام نمی‌شود

وجود موبایل canonical (از جمله meta Digits)، owner پیش‌فرض، `_didar_person_sync_state` و خطاهای `didar_mobile_missing` یا conflict را بررسی کنید. state pending با worker بازیابی می‌شود.

Deal تکراری، webhook ردشده یا دانلود ممنوع

`_didar_deal_id` و custom field شناسهٔ webhook، URL/secret، submission یا (فقط legacy فعال) header `x-` event ID، rate limit، format JSON، `didar-webhook-token`، تکراری، ورود، مالکیت درخواست، record فایل و nonce را بررسی کنید. secret واقعی را در ticket عمومی نگذارید.

Worker یا import

schedule، WP-Cronهای دو hook و `sync_schedule_failed` diagnostic را بررسی کنید. برای import ابتدا preview، version و allowlist را کنترل کنید.

فصل 20

راهنمای توسعه‌دهنده

برای فرم جدید، definition را فقط در `Didar_Form_Registry` اضافه کنید؛ `Didar_Field_Renderer`، `Didar_Validator` و `Didar_Submission_Service` همان registry را مصرف می‌کنند. داده‌های مرجع مشترک در `Didar_Reference_Data` قرار می‌گیرند.

برای mapping Didar از `Didar_Field_Mapper` و settingهای validateشده استفاده کنید؛ برای status/pipeline از diagnostic. `Didar_Workflow_Manager` را با `Didar_Logger::log()` و context شامل `local_id`، `entity_type` و در `trace_id`، sync بنویسید. worker جدید باید bounded batch، idempotency، schedule و cleanup/deactivation روشن داشته باشد.

برای تنظیم `Didar_Settings_Transfer` allowlist، portable را تغییر دهید؛ usermeta، runtime state، secret و option دلخواه را وارد نکنید. Checklist امنیت: capability جدا از unslash/sanitize/validate، ownership، nonce، escaping، عدم اعتماد به ID/مسیر مرورگر، و عدم log کردن secret/PII.

فصل ۲۱

محدودیت‌های فعلی

حذف یا archive کردن Deal در Didar در پیاده‌سازی فعلی پشتیبانی نمی‌شود. حذف یا انتقال محلی درخواست ممکن است در event log و diagnostics ثبت شود، اما افزونه برای حذف یا archive remote Deal درخواست API ارسال نمی‌کند.