DEXPRO AWF Externe Genehmigungen — API-Integrationshandbuch

> **Zielgruppe:** Entwickler, die benutzerdefinierte Genehmigungsoberflächen oder Integrationen aufbauen, die die DEXPRO-AWF-API für externe Genehmigungen direkt konsumieren.
>
> **Suchen Sie die SharePoint-/Teams-Referenzimplementierung?** Siehe `SETUP-GUIDE.de-DE.md` (Admin-Einrichtung) und `SHAREPOINT-CONTRACT.de-DE.md` (SharePoint-Listen-Vertrag).

Übersicht

Die API für externe Genehmigungen macht ausstehende Genehmigungseinträge direkt als OData-Ressourcen innerhalb des Standard-API-V2-Frameworks von Business Central verfügbar. Jedes System, das HTTPS-Aufrufe machen kann — ein benutzerdefiniertes Portal, eine mobile App, ein Drittanbieter-Benachrichtigungsdienst, eine Azure Function — kann:

- Genehmigungsanfragen abfragen, die bestimmten Personen oder Gruppen zugewiesen sind
- Den Datensatzkontext dem Genehmiger anzeigen
- Genehmigungs- / Ablehnungsentscheidungen mit einem optionalen Kommentar übermitteln
- Gruppenbasierte (übernahmebasierte) Genehmigungen verwalten

Kein SharePoint-Setup erforderlich. Die API funktioniert unabhängig vom SharePoint-Integrations-Toggle (`SP Approvals Enabled`). Solange die DEXPRO-AWF-Erweiterung installiert und externe Genehmiger in BC konfiguriert sind, sind die API-Seiten aktiv.

Wann die API vs. die SharePoint-Integration verwenden

	API-Integration	SharePoint-Integration
Einrichtung	Keine außer BC-API-Zugang	App-Registrierung, SP-Website, Assistent

	API-Integration	SharePoint-Integration
Benachrichtigung	Ihr System übernimmt dies	Power Automate → Teams Adaptive Card
Antworterfassung	Ihre UI ruft die BC-API direkt auf	Genehmiger antwortet in Teams; PA schreibt in SP; BC fragt SP ab
Lizenz (Genehmiger)	Kostenpflichtige BC-Benutzerlizenz (siehe Lizenzhinweis)	Kostenpflichtige BC-Benutzerlizenz, plus M365 für Teams (siehe Lizenzhinweis)
Geeignet für	Benutzerdefinierte Portale, mobile Apps, Lieferanten-Self-Service	Schnelles Teams-Rollout, minimaler Entwicklungsaufwand

> **Lizenzierung — Verantwortung des Kunden/Partners:** Das Bearbeiten von Business-Central-Daten — ob direkt über die API oder indirekt über die SharePoint-Weiterleitung — erfordert für jeden Genehmiger eine **geeignete kostenpflichtige Business-Central-Benutzerlizenz**. Eine Microsoft-365-Lizenz allein gewährt **keinen** API- oder Schreibzugriff auf BC, und das Weiterleiten von Aktionen über ein Dienstkonto hebt die Lizenzpflicht pro Benutzer **nicht** auf (Microsoft-Multiplexing- bzw. Regeln zum indirekten Zugriff). Eine **Team-Member**-Lizenz kann für reine Genehmigungsnutzung ausreichen, jedoch beschränkt Microsoft Team Member auf vorgesehene Szenarien — für benutzerdefinierte oder Drittanbieter-Oberflächen ist dies **nicht garantiert**, und ein vollwertiger **Essentials/Premium**-Benutzer kann erforderlich sein. **DEXPRO gibt keine lizenzrechtliche Zusicherung.** Der Kunde und sein Microsoft-Lizenzierungspartner sind allein für die Ermittlung und Einhaltung der korrekten Lizenzierung verantwortlich; prüfen Sie den aktuellen Microsoft Dynamics 365 Licensing Guide.

Voraussetzungen

Anforderung	Details
Business Central	DEXPRO-AWF-Erweiterung installiert, BC 25 oder höher
Entra-App-Registrierung	Eine Entra-ID-App mit `Financials.ReadWrite.All` (oder ein delegierter Benutzerflow) für den BC-API-Zugang
BC-Berechtigungssatz	Das Dienstkonto des API-Aufrufers benötigt einen Berechtigungssatz, der Lesezugriff auf `DXP AWF Ext. Approval Entry` und Schreibzugriff für die gebundenen Aktionen (`approve`, `reject`, `claim`, `releaseClaim`, `markNotified`, `setTeamsMessageId`) gewährt. Der Berechtigungssatz **DXP AWF Admin** deckt dies ab.

Anforderung	Details
Externe Genehmiger	Mindestens ein externer Genehmiger in BC konfiguriert (AWF-Einrichtung → Externe Genehmiger)

Authentifizierung

Die BC-API-V2 verwendet OAuth 2.0 mit Microsoft Entra ID. Beziehen Sie ein Bearer-Token von:

```
POST https://login.microsoftonline.com/{tenantId}/oauth2/v2.0/token


grant_type=client_credentials
&client_id={clientId}
&client_secret={clientSecret}
&scope=https://api.businesscentral.dynamics.com/.default
```

Fügen Sie das Token in jeder Anfrage ein:

```
Authorization: Bearer {access_token}
```

> **Hinweis:** Der Client-Credentials-Flow eignet sich für Backend-Service-Integrationen. Für benutzer-delegierte Flows (z. B. ein Webportal, bei dem sich jeder Genehmiger selbst authentifiziert), verwenden Sie stattdessen Authorization Code + PKCE — das Token trägt dann die Benutzeridentität, was für Prüfzwecke nützlich ist.

Basis-URL

```
https://api.businesscentral.dynamics.com/v2.0/{tenantId}/{environmentName}/api/dexpro/advanced
Workflow/v1.0
```

Alle Beispiele unten verwenden `{baseUrl}` als Kurzform für diese vollständige Basis-URL.

So finden Sie Ihre Unternehmens-ID (eine GUID, die in den meisten Anfragen erforderlich ist):

```
GET {baseUrl}/companies
```

Antwort (gekürzt):

```
{
  "value": [
    {
      "id": "5e9f4c3a-0001-ef11-9f8a-6045bd028c9f",
      "name": "CRONUS International Ltd.",
      ...
    }
  ]
}
```

Verwenden Sie den `id`-Wert als `{companyId}` in nachfolgenden Anfragen.

Entitäten

Es stehen drei schreibgeschützte Entitätenmengen zur Verfügung. Alle verwenden **`SystemId`** (eine plattformzugewiesene GUID, als `id` exponiert) als OData-Schlüssel — der BC-API-V2-Standardkonvention folgend.

Entitätenmenge	OData-URL-Segment	OData-Schlüsselfeld	Beschreibung
Genehmigungseinträge	`externalApprovalEntries`	`id` (SystemId)	Die aktionsfähigen Einträge — einer pro Genehmiger pro Genehmigungsanfrage. Hauptentität für Integrationen.
Genehmiger	`externalApprovers`	`id` (SystemId)	Die registrierten externen Genehmigerdatensätze. Nützlich für Bootstrap / Vorabfüllung.
Gruppenmitglieder	`externalApproverMembers`	`id` (SystemId)	Mitglieder externer Genehmigergruppen.
Eintragsdokument	`externalApprovalEntryDocuments`	`id` (SystemId)	Quelldokument-Datensatz als JSON. Separater Endpunkt — nur auf Anfrage abgerufen.
Eintragsanhänge	`externalApprovalEntryAttachments`	`id` (SystemId)	Standard-BC-Dokumentanhänge für den Quelldatensatz. Separater Endpunkt — nur auf Anfrage abgerufen.

> **Warum GUIDs, keine Integer?** Das `entryNo`-Integer-Feld ist im Antwort-Body für Anzeige und Korrelation noch vorhanden, ist aber nicht der OData-Schlüssel. Die Verwendung von SystemId (GUID) als Schlüssel folgt Microsofts offiziellen BC-API-Richtlinien und stellt die Kompatibilität mit Power Automate, Power Apps und Logic Apps Connectors sicher. Verwenden Sie immer `id` in URL-Pfaden.

Mit Genehmigungseinträgen arbeiten

Alle ausstehenden Einträge für einen Genehmiger auflisten

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=approverEmail eq 'hans.mueller@contoso.com' and status eq 'Pending'
    &$orderby=createdDateTime asc
```

Antwort:


```
{
  "@odata.context": "...",
  "value": [
    {
      "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
      "entryNo": 42,
      "approvalEntryNo": 17,
      "workflowInstanceId": "a1b2c3d4-0000-0000-0000-000000000001",
      "templateCode": "EINKAUFSGENEHMIGUNG",
      "stageCode": "EXT-PRUEFUNG",
      "stageDescription": "Externe Prüfung",
      "documentNo": "EK-00123",
      "documentType": "Order",
      "documentTableId": 38,
      "approverEmail": "hans.mueller@contoso.com",
      "approverDisplayName": "Hans Müller",
      "status": "Pending",
      "isGroupApproval": false,
      "isRelatedApproval": false,
      "description": "Fabrikam GmbH",
      "recordDescription": "Einkaufsbestellung EK-00123",
      "amount": 15000.00,
```

```
      "dueDate": "2026-05-15",
      "senderUserId": "EINKAUF",
      "languageCode": "DEU",
      "processed": false,
      "createdDateTime": "2026-05-04T09:12:33Z",
      "errorMessage": "",
      "spListItemId": "",
      "spAttachmentsUrl": ""
    }
  ]
}
```

Einzelnen Eintrag abrufen

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)
```

Häufige Filtermuster

```
# Alle ausstehenden Einträge für einen Genehmiger (Einzel + Gruppe)
$filter=approverEmail eq 'anna@contoso.com' and processed eq false and status ne 'Cancelled'

# Alle ausstehenden Einträge für eine Gruppe (allen Mitgliedern anzeigen, bevor eines übernimmt)
$filter=extApproverGroupCode eq 'FINANZ-GRUPPE' and status eq 'Pending'

# Einträge mit Verarbeitungsfehlern (zur Admin-Überwachung)
$filter=errorMessage ne '' and processed eq false

# Alle aktionsfähigen Einträge für ein Gruppenmitglied (ausstehend oder von ihm übernommen)
$filter=approverEmail eq 'anna@contoso.com'
    and (status eq 'Pending' or status eq 'Notified' or status eq 'Claimed')
    and processed eq false
```

Gebundene Aktionen

Aktionen werden als OData-gebundene Aktionen über `POST` aufgerufen. Das URL-Muster ist:

```
POST
{baseUrl}/companies({companyId})/externalApprovalEntries({id})/Microsoft.NAV.{aktionsname}
Content-Type: application/json
```

`{id}` ist das `id`-Feld (SystemId-GUID) aus dem Eintrag. Ein erfolgreicher Aufruf gibt `200 OK` mit dem aktualisierten Eintrag im Antwort-Body zurück.

approve (Genehmigen)

Zeichnet eine Genehmigungsentscheidung auf und bringt den BC-Workflow voran.

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)/Microsoft.NAV.approve
Content-Type: application/json

{
  "approverEmail": "hans.mueller@contoso.com",
  "comment": "Sieht gut aus. Im Rahmen des Q2-Budgets genehmigt."
}
```

`approverEmail` ist **erforderlich** — übergeben Sie die E-Mail des Benutzers, der die Genehmigung tatsächlich durchgeführt hat. Dies wird im BC-Protokoll und im Genehmigungseintrag aufgezeichnet. Da die API typischerweise von einem Backend-Dienst (Clientanmeldeinformationen) aufgerufen wird, hat BC keine andere Möglichkeit, die echte Identität des Akteurs zu kennen.

`comment` ist optional. Weglassen oder `""` übergeben, um ohne Kommentar zu genehmigen.

Antwort `200 OK`:

```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "approverEmail": "hans.mueller@contoso.com",
  "status": "Approved",
  "responseComment": "Sieht gut aus. Im Rahmen des Q2-Budgets genehmigt.",
  "responseDateTime": "2026-05-04T10:35:12Z",
  "processed": true,
  "processedDateTime": "2026-05-04T10:35:12Z",
  ...
}
```

reject (Ablehnen)

Zeichnet eine Ablehnungsentscheidung auf und steuert den BC-Ablehnungsflow (einschließlich Neuzuweisung für verknüpfte Genehmigungen).

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)/Microsoft.NAV.reject
Content-Type: application/json

{
  "approverEmail": "hans.mueller@contoso.com",
  "comment": "Betrag überschreitet Abteilungsvollmacht. An CFO eskalieren."
}
```

`approverEmail` ist aus denselben Gründen wie bei `approve` **erforderlich**.

Antwort `200 OK`:

```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "approverEmail": "hans.mueller@contoso.com",
  "status": "Rejected",
  "responseComment": "Betrag überschreitet Abteilungsvollmacht. An CFO eskalieren.",
  "responseDateTime": "2026-05-04T10:38:44Z",
  "processed": true,
  ...
}
```

markNotified (Als benachrichtigt markieren)

Rufen Sie dies auf, nachdem Ihr System die Genehmigungsbenachrichtigung erfolgreich an den Genehmiger zugestellt hat (z. B. eine E-Mail gesendet oder die Karte in Ihrem Portal angezeigt hat). Wechselt den Status von `Pending` → `Notified`. Dies ist optional, aber empfohlen — es ermöglicht BC-Administratoren, zwischen Einträgen zu unterscheiden, die nie zugestellt wurden, und Einträgen, die auf eine Antwort warten.

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)/Microsoft.NAV.markNotified
Content-Type: application/json

{}
```

Antwort `200 OK`:

```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "status": "Notified",
  ...
}
```

> Nur wirksam, wenn der aktuelle Status `Pending` ist. No-Op bei jedem anderen Status.

setTeamsMessageId (Teams-Message-ID setzen)

Rufen Sie dies **anstelle von** `markNotified` auf, wenn Ihr Transport die gepostete Nachricht später aktualisieren/schließen können muss (z. B. eine Microsoft-Teams-Adaptive-Card). Es tut alles, was `markNotified` tut (Pending → Notified), **und** speichert die Message-ID des Transports auf dem Eintrag (BC-Feld `Teams Message ID`) und spiegelt sie in die SharePoint-Spalte `TeamsMessageId`. Der Power-Automate-v2-Flow nutzt dies, damit eine Peer-Übernahme/-Entscheidung die noch offenen Karten anderer Gruppenmitglieder über *Adaptive Card aktualisieren* ersetzen (schließen) kann.

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)/Microsoft.NAV.setTeamsMessageId
Content-Type: application/json

{ "teamsMessageId": "1700000000000" }
```

Antwort `200 OK`:

```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "status": "Notified",
  "teamsMessageId": "1700000000000",
  ...
}
```

> Übergeben Sie die Message-ID, die Ihr Transport beim Posten der Nachricht zurückgibt — bei der *Karte in einem Chat oder Kanal posten*-Aktion des Teams-Connectors ist das `body/id`. Leere Eingabe wird ignoriert (der Eintrag wird trotzdem als `Notified` markiert). Erneuter Aufruf mit derselben ID ist ein No-Op.

claim (Übernehmen) — *nur Gruppengenehmigungs-Anfragen*

Übernimmt einen Gruppengenehmigungs-Eintrag. Dies ist der „Erster gewinnt"-Schritt: Sobald übernommen, werden alle Einträge der anderen Gruppenmitglieder für dieselbe Genehmigungsanfrage storniert, und nur der Übernehmer kann dann genehmigen oder ablehnen.

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(c4e7d1a2-0002-ef11-bf8d-6045bd028c9f)/Microsoft.NAV.claim
Content-Type: application/json

{
  "claimerEmail": "anna.schmidt@contoso.com"
}
```

`claimerEmail` ist **erforderlich** — übergeben Sie die E-Mail des Benutzers, der den Eintrag übernimmt. Dies wird im Prüfprotokoll aufgezeichnet und in den BC-Genehmigungseintrag als Identität des Übernehmers geschrieben.

Antwort `200 OK`:

```
{
  "id": "c4e7d1a2-0002-ef11-bf8d-6045bd028c9f",
  "entryNo": 43,
  "status": "Claimed",
  "claimedByEmail": "anna.schmidt@contoso.com",
  "claimedByDisplayName": "Anna Schmidt",
  "claimedDateTime": "2026-05-04T10:41:02Z",
  ...
}
```

> **Race Condition:** Wenn zwei Gruppenmitglieder gleichzeitig `claim` aufrufen, gewinnt nur eines. Der Verlierer erhält `400 Bad Request` mit dem Fehler-Body: *„Dieser Eintrag wurde bereits von anna.schmidt@contoso.com übernommen."* Ihre UI sollte dies abfangen und den Eintrag für den verlierenden Aufrufer aktualisieren.

releaseClaim (Freigeben) — *nur Gruppengenehmigungs-Anfragen*

Gibt einen zuvor übernommenen Eintrag frei und setzt ihn auf den Status `Notified` zurück, sodass ein anderes Gruppenmitglied ihn übernehmen kann.

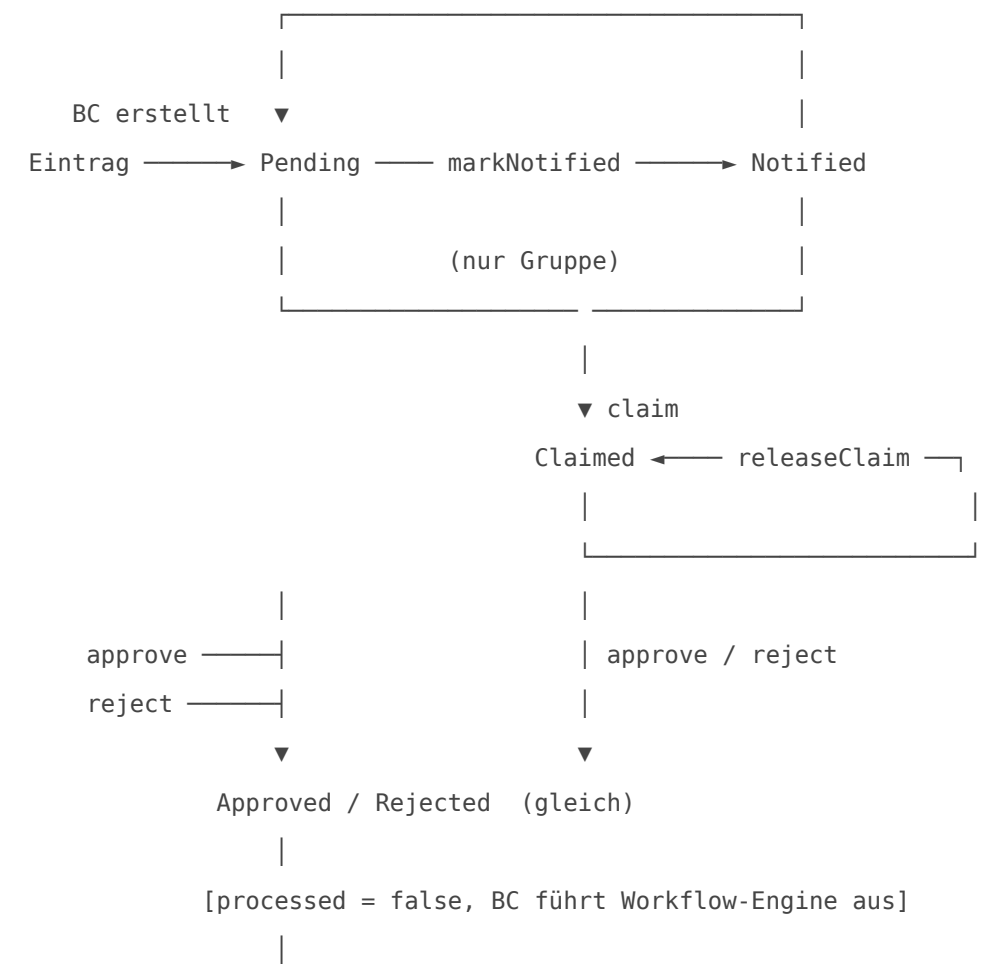
```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(c4e7d1a2-0002-ef11-bf8d-
6045bd028c9f)/Microsoft.NAV.releaseClaim
Content-Type: application/json

{}
```

Antwort `200 OK`:

```
{
  "id": "c4e7d1a2-0002-ef11-bf8d-6045bd028c9f",
  "entryNo": 43,
  "status": "Notified",
  "claimedByEmail": "",
  "claimedByDisplayName": "",
  ...
}
```

Status-Zustandsautomat



```
                [processed = true]
                      |
                      ▼
```

Status	Bedeutung	Aktionsfähig?
`Pending`	Erstellt, noch nicht an den Genehmiger zugestellt	`markNotified`, `approve`, `reject`, `claim`
`Notified`	Ihr System hat bestätigt, dass der Genehmiger benachrichtigt wurde	`approve`, `reject`, `claim`
`Claimed`	Ein Gruppenmitglied hat diesen Eintrag übernommen	`approve`, `reject` (nur Übernehmer), `releaseClaim`
`Approved`	Genehmiger hat genehmigt; BC verarbeitet möglicherweise noch	—
`Rejected`	Genehmiger hat abgelehnt	—
`Cancelled`	Von BC storniert (Peer hat übernommen/entschieden, Workflow wurde neu zugewiesen, oder Dokument wurde wieder geöffnet)	—

> **`processed` vs. `status`:** `status` ist die Entscheidung des Genehmigers. `processed = true` bedeutet, dass die Workflow-Engine von BC die Entscheidung erfolgreich konsumiert hat. Ein Eintrag kann `status = Approved` mit `processed = false` haben, wenn der Workflow-Engine-Aufruf fehlgeschlagen ist — prüfen Sie `errorMessage` für Details. BC-Administratoren können von der Seite „Externe Genehmigungseinträge" aus erneut versuchen.

Gruppengenehmigungs-Anfragen

Wenn eine Genehmigungsstufe auf eine **Externe Genehmigergruppe** abzielt, erstellt BC einen `externalApprovalEntry` für jedes Gruppenmitglied. Sie teilen dieselbe `workflowInstanceId` und `approvalEntryNo`, haben aber unterschiedliche `id`-, `approverEmail`- und `externalApproverCode`-Werte.

`isGroupApproval = true` gilt für alle.

Integrationsflow für Gruppengenehmigungs-Anfragen

1. **Alle Mitglieder benachrichtigen** — Einträge nach `extApproverGroupCode` + `status eq 'Pending'` abfragen. Jedem Mitglied seinen eigenen Eintrag anzeigen und die `id` des Eintrags für nachfolgende Aktionsaufrufe speichern.

2. **Mitglied übernimmt** — Wenn ein Mitglied auf „Übernehmen" tippt, rufen Sie `Microsoft.NAV.claim` auf der `id` **seines** Eintrags auf. BC storniert die Einträge der anderen Mitglieder automatisch.
3. **Übernehmer entscheidet** — Zeigen Sie dem Genehmiger seinen übernommenen Eintrag. Er ruft `approve` oder `reject` auf derselben `id` auf, die er übernommen hat.
4. **Auf Stornierungen prüfen** — Nach einer Übernahme wechseln die Einträge anderer Gruppenmitglieder auf `Cancelled`. Ihre UI sollte damit elegant umgehen (z. B. „Diese Genehmigung wurde von Anna Schmidt übernommen").

Peer-Stornierungen erkennen

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=approverEmail eq 'bob@contoso.com' and status eq 'Cancelled' and processed eq
false
    &$orderby=createdDateTime desc
    &$top=50
```

Einträge im Status `Cancelled` mit `processed = false` wurden storniert, weil ein Peer zuerst übernommen oder entschieden hat (nicht weil der Workflow von einem BC-Benutzer widerrufen wurde).

Gruppen mit einem Mitglied

BC übernimmt Einträge für Gruppen mit einem Mitglied automatisch bei der Erstellung — `status` ist beim ersten Abruf bereits `Claimed` und `claimedByEmail` ist vorausgefüllt. Überspringen Sie den Übernahmeschritt und gehen Sie direkt zu `approve` / `reject`.

Feldreferenz — externalApprovalEntry

JSON-Feld	Typ	Beschreibung
`id`	GUID	**OData-Schlüssel.** SystemId — verwenden Sie diesen in allen Aktions-URLs und Einzeldatensatz-GETs.
`entryNo`	Integer	Interne BC-Sequenznummer. Nützlich für Anzeige, Korrelation und Lesen verwandter SharePoint-Elemente (`BCEntryNo`-Spalte). Nicht der OData-Schlüssel.
`approvalEntryNo`	Integer	Schlüssel des verknüpften BC-Standard-Genehmigungspostens.

JSON-Feld	Typ	Beschreibung
`workflowInstanceId`	GUID	BC-Workflow-Instanz. Von allen Einträgen (inkl. verknüpfter Zeilen) in derselben Workflow-Ausführung geteilt.
`templateCode`	String	AWF-Workflowvorlagen-Code.
`stageCode`	String	AWF-Workflow-Schritt-Code.
`stageDescription`	String	Für Menschen lesbare Schrittbeschreibung. Dem Genehmiger anzeigen.
`documentTableId`	Integer	BC-Tabellen-ID des Quelldatensatzes (z. B. `38` für Einkaufskopf, `36` für Verkaufskopf). Nützlich für Routing oder tabellenspezifische UI in Ihrem Portal.
`documentNo`	String	Dokumentnummer, die genehmigt wird (z. B. `EK-00123`).
`documentType`	String	Beschriftung des Dokumenttyp-Enums (z. B. `Order`, `Invoice`, `""` für benutzerdefinierte Tabellen).
`externalApproverCode`	String	Code des externen Genehmigerdatensatzes in BC.
`extApproverGroupCode`	String	Code der externen Genehmigergruppe, falls dies eine Gruppengenehmigungs-Anfrage ist. Leer bei Einzelgenehmigungen.
`approverEmail`	String	E-Mail des diesem spezifischen Eintrag zugewiesenen Genehmigers. Für Routing verwenden.
`approverDisplayName`	String	Anzeigename des zugewiesenen Genehmigers.
`status`	String	Siehe Status-Zustandsautomat oben.
`isGroupApproval`	Boolesch	Ob dies Teil einer übernahmebasierten Gruppengenehmigungs-Anfrage ist.
`claimedByEmail`	String	E-Mail des Gruppenmitglieds, das den Eintrag übernommen hat. Leer bei Einzelgenehmigungen.
`claimedByDisplayName`	String	Anzeigename des Übernehmers.
`claimedDateTime`	Datum/Uhrzeit	Wann der Eintrag übernommen wurde (UTC).
`processed`	Boolesch	Ob die Workflow-Engine von BC diese Entscheidung erfolgreich konsumiert hat.

JSON-Feld	Typ	Beschreibung
`processedDateTime`	Datum/Uhrzeit	Wann BC die Entscheidung verarbeitet hat (UTC).
`errorMessage`	String	Nicht leer, wenn `processed = false` nach einer Genehmigungs-/Ablehnungsentscheidung — der Workflow-Engine-Aufruf ist fehlgeschlagen.
`description`	String	Kontextbezogene Beschreibung (z. B. Kreditorenname, Debitorenname). Gut für die Anzeige in einer Karte oder Liste.
`recordDescription`	String	Vollständige Datensatzkennung (z. B. `"Tabelle 38 (Einkaufskopf): Bestellung, EK-00123"`). Für detaillierte Anzeige verwenden.
`amount`	Dezimalzahl	Genehmigungsbetrag.
`dueDate`	Datum	Fälligkeitsdatum der Genehmigung (ISO-8601-Datum, z. B. `2026-05-15`).
`senderUserId`	String	BC-Benutzer-ID der Person, die das Dokument zur Genehmigung gesendet hat.
`languageCode`	String	BC-Sprachcode des zugewiesenen Genehmigers (z. B. `ENU`, `DEU`). Für die Lokalisierung Ihrer Benachrichtigung verwenden.
`responseComment`	String	Vom Genehmiger eingegebener Kommentar. Nach Genehmigung/Ablehnung befüllt.
`responseDateTime`	Datum/Uhrzeit	Wann der Genehmiger geantwortet hat (UTC).
`isRelatedApproval`	Boolesch	Wahr für zeilenebene (verknüpfte) Einträge, die mit einer Kopfgenehmigung verknüpft sind.
`createdDateTime`	Datum/Uhrzeit	Wann BC diesen Eintrag erstellt hat (UTC).
`spListItemId`	String	SharePoint-Listenelement-ID. Nur befüllt, wenn SP-Genehmigungen aktiviert sind. Für reine API-Integrationen ignorieren.
`spAttachmentsUrl`	String	SharePoint-Freigabelink für Dokumentanhänge. Nur befüllt, wenn SP-Genehmigungen aktiviert sind und Anhänge hochgeladen wurden.

Feldreferenz — externalApprover

JSON-Feld	Typ	Beschreibung
`id`	GUID	**OData-Schlüssel.** SystemId.
`code`	String	Interner BC-Genehmiger-Code (Code[20]).
`displayName`	String	Vollständiger Name.
`email`	String	E-Mail / UPN.
`languageCode`	String	BC-Sprachcode für Benachrichtigungen.
`entraObjectId`	GUID	Entra-ID-Objekt-ID (befüllt für aus Entra synchronisierte Genehmiger).
`blocked`	Boolesch	Blockierte Genehmiger können keine neuen Einträge erhalten.

Feldreferenz — externalApproverMember

JSON-Feld	Typ	Beschreibung
`id`	GUID	**OData-Schlüssel.** SystemId.
`groupCode`	String	Der Code der externen Genehmigergruppe.
`lineNo`	Integer	Zeilennummer innerhalb der Gruppe (nur informativ).
`externalApproverCode`	String	Genehmiger-Code des Mitglieds.
`displayName`	String	Anzeigename des Mitglieds.
`email`	String	E-Mail des Mitglieds.

Vollständiges Beispiel: Einzelner Genehmigungsflow

Die folgende Sequenz implementiert ein minimales Genehmigungsportal für Einzelgenehmiger.

Schritt 1 — Authentifizieren und ausstehende Einträge abrufen

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=approverEmail eq 'hans.mueller@contoso.com'
```

```
        and status ne 'Cancelled'
        and processed eq false

&$select=id,entryNo,documentNo,description,stageDescription,amount,dueDate,status,isGroupAppro
val,createdDateTime
    &$orderby=createdDateTime asc
```

Schritt 2 — Eintragsdetails dem Genehmiger anzeigen

Verwenden Sie `description`, `stageDescription`, `amount`, `dueDate` und `senderUserId`, um eine Genehmigungskarte aufzubauen. Speichern Sie die `id`-GUID des Eintrags — Sie benötigen sie für Aktionsaufrufe.

Schritt 3 — Als benachrichtigt markieren

Nachdem die Karte dem Genehmiger angezeigt wurde:

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-
6045bd028c9f)/Microsoft.NAV.markNotified
Content-Type: application/json

{}
```

Schritt 4 — Entscheidung erfassen

Genehmiger klickt auf Genehmigen mit einem Kommentar:

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(b9f3a2c1-0001-ef11-bf8d-
6045bd028c9f)/Microsoft.NAV.approve
Content-Type: application/json

{
  "approverEmail": "hans.mueller@contoso.com",
  "comment": "Im Rahmen des Budgets, genehmigt."
}
```

Vollständiges Beispiel: Gruppengenehmigungs-Flow

Schritt 1 — Alle Gruppenmitglieder benachrichtigen

Einträge für die Gruppe abrufen, einen pro Mitglied:

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=extApproverGroupCode eq 'FINANZ-GRUPPE'
        and (status eq 'Pending' or status eq 'Notified')
        and processed eq false
```

Jedes Mitglied mit einer Benachrichtigung benachrichtigen, die die `id` seines Eintrags enthält.

Schritt 2 — Mitglied übernimmt

Anna öffnet das Portal und tippt auf „Übernehmen" (mit der `id` aus ihrem eigenen Eintrag):

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(c4e7d1a2-0002-ef11-bf8d-
6045bd028c9f)/Microsoft.NAV.claim
Content-Type: application/json

{
  "claimerEmail": "anna.schmidt@contoso.com"
}
```

BC storniert alle Geschwistereinträge (die Einträge der anderen Gruppenmitglieder für dieselbe Genehmigung).

Race Condition behandeln: Wenn der Eintrag bereits von jemand anderem übernommen wurde, gibt BC `400 Bad Request` zurück:

```
{
  "error": {
    "code": "Internal",
    "message": "Dieser Eintrag wurde bereits von bob.mueller@contoso.com übernommen.
CorrelationId: ..."
  }
}
```

Aktualisieren und „Diese Genehmigung wurde von Bob Müller übernommen" für Anna anzeigen.

Schritt 3 — Übernehmer entscheidet

Anna ruft genehmigen mit derselben `id` auf:

```
POST {baseUrl}/companies({companyId})/externalApprovalEntries(c4e7d1a2-0002-ef11-bf8d-
6045bd028c9f)/Microsoft.NAV.approve
```

```
Content-Type: application/json

{
  "approverEmail": "anna.schmidt@contoso.com",
  "comment": "Zeilenpositionen geprüft, alle richtlinienkonform."
}
```

Fehlerbehandlung

HTTP-Statuscodes

Code	Wann
`200 OK`	Aktion erfolgreich. Antwort-Body enthält den aktualisierten Eintrag.
`400 Bad Request`	Geschäftslogikfehler (Eintrag bereits verarbeitet, bereits von jemand anderem übernommen, nicht in einem übernahmbaren Zustand). `error.message` für Details lesen.
`401 Unauthorized`	Fehlendes oder abgelaufenes Bearer-Token.
`403 Forbidden`	Das BC-Benutzerkonto des API-Aufrufers fehlt der erforderliche Berechtigungssatz.
`404 Not Found`	Eintrag in diesem Unternehmen nicht gefunden, oder falsche `id`-GUID.

Idempotenz

- `approve` und `reject` sind gesichert: Sie auf einem bereits verarbeiteten Eintrag aufzurufen, gibt `400` mit *„Dieser externe Genehmigungseintrag wurde bereits verarbeitet...*" zurück. Sicher nur zu wiederholen, wenn der vorherige Aufruf eine Nicht-`200`-Antwort zurückgegeben hat.
- `markNotified` ist ein No-Op, wenn der Status nicht `Pending` ist — sicher mehrfach aufzurufen.
- `claim` ist absichtlich nicht idempotent: Es zweimal von zwei verschiedenen Aufrufern aufzurufen ist die Race Condition, die Sie behandeln müssen.

SharePoint/Teams-Race-Condition

Wenn die SharePoint-Integration aktiv ist, kann der Eintrag zwischen Ihrer letzten Abfrage und Ihrem `approve`/`reject`/`claim`-Aufruf über Teams entschieden worden sein. Die API synchronisiert den Live-SP-Zustand vor der Verarbeitung (`SyncFromSPIfNeeded`); ein inzwischen geschlossener Eintrag wird daher mit `400` und einer der tatsächlichen Guard-Meldungen abgelehnt:

> *„This external approval entry has already been processed by Business Central and cannot be acted on again."*
>
> *„This external approval entry has been cancelled (a peer in the group decided first, or the workflow was reassigned). Refresh the list to see the current state."*

Dies ist kein wiederholbarer Fehler — behandeln Sie ihn wie einen `Cancelled`-Status und aktualisieren Sie die Eintrags-Liste für den Benutzer.

Einträge, die verschwinden

Der Status eines Eintrags kann jederzeit auf `Cancelled` wechseln aufgrund externer Ereignisse:

- Ein BC-Benutzer weist den zugrundeliegenden Genehmigungsposten neu zu oder storniert ihn
- Das Dokument wird wieder geöffnet (storniert den gesamten Workflow)
- Ein Peer in derselben Gruppe hat zuerst übernommen (nur bei Gruppeneinträgen)

Prüfen Sie immer auf den Status `Cancelled`, bevor Sie einem Genehmiger einen Eintrag anzeigen, und behandeln Sie `400`-Antworten elegant mit einer Aktualisierung.

Abfrageempfehlungen

Es gibt keinen Webhook- oder Push-Mechanismus — Ihre Integration fragt die API ab. Empfohlene Intervalle:

Anwendungsfall	Intervall
Benachrichtigungszustellung (neue `Pending`-Einträge abrufen)	1–5 Minuten
Genehmigungsportal-Aktualisierung (angemeldeter Benutzer)	Auf Anfrage (Benutzeraktualisierung) + beim Seitenaufruf
Gruppenmitgliedschaftsaktualisierung	`externalApproverMembers` bei jedem Portal-Login abfragen

Um neu erstellte Einträge seit der letzten Abfrage zu erkennen, filtern Sie auf `createdDateTime`:

```
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=status eq 'Pending' and createdDateTime gt 2026-05-04T09:00:00Z
    &$orderby=createdDateTime asc
```

Datensatz-JSON

Jeder Genehmigungseintrag hat einen entsprechenden `externalApprovalEntryDocuments`-Endpunkt, der die vollständige JSON-Darstellung des Quelldokument-Datensatzes zurückgibt — die eigentliche BC-Tabellenzeile, die genehmigt wird (z. B. die Einkaufsbestellung, das Verkaufsangebot oder der benutzerdefinierte Datensatz).

Dies ist ein **separater Endpunkt** von `externalApprovalEntries`. Er wird nur abgerufen, wenn explizit aufgerufen, damit Listenabfragen schnell bleiben.

```
GET {baseUrl}/companies({companyId})/externalApprovalEntryDocuments(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)
```

Die `id` ist dieselbe SystemId-GUID wie der entsprechende `externalApprovalEntry`. Beispielantwort:

```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "documentNo": "EK-00123",
  "recordJson": "{\"No_\":\"EK-00123\",\"Document_Type\":\"Order\",\"Buy-from_Vendor_No_\":\"K00010\",\"Buy-from_Vendor_Name\":\"Fabrikam GmbH\",\"Amount\":15000.00,\"Amount_Including_VAT\":17850.00,\"Due_Date\":\"2026-05-15\",\"TableNo\":38,\"TableName\":\"Purchase Header\",\"TableCaption\":\"Einkaufskopf\",\"Company\":\"CRONUS International Ltd.\", ...}"
}
```

> `recordJson` ist eine JSON-Zeichenfolge (kein verschachteltes Objekt) — mit `JSON.parse()` in Ihrem Client parsen.

Was das JSON enthält

Das JSON-Objekt wird von `DXP Json Helper.Rec2Json` erzeugt und enthält:

- **Alle Tabellenfelder** — Feldbeschriftungen (mit durch `_` ersetzten Leerzeichen) als Schlüssel, Werte in ihre JSON-nativen Typen serialisiert
- **Metadatenfelder**, die automatisch vom Helper hinzugefügt werden:
  - `TableNo` — numerische Tabellen-ID
  - `TableName` — interner Tabellenname
  - `TableCaption` — lokalisierte Tabellenbeschriftung
  - `Company` — Unternehmensname
  - `RecordId` — BC-RecordId-Zeichenfolge
  - `SysLink` — SystemId + TableNo-Komposit

Leistungshinweis

`recordJson` wird bei jedem Datensatzlesen berechnet — er navigiert zum Quelldokument und serialisiert alle seine Felder. Da er auf einem dedizierten Endpunkt liegt, wird er nur abgerufen, wenn Sie `externalApprovalEntryDocuments` explizit aufrufen. Verwenden Sie den Eintragsendpunkt für Listenabfragen und rufen Sie den Dokumentendpunkt nur auf, wenn Sie den vollständigen Datensatz benötigen:

```
# Schnelle Liste – kein Dokumentnachschlag
GET {baseUrl}/companies({companyId})/externalApprovalEntries
    ?$filter=approverEmail eq 'hans@contoso.com' and status eq 'Pending'

# Auf Anfrage – vollständigen Datensatz nur für den vom Genehmiger geöffneten Eintrag abrufen
GET {baseUrl}/companies({companyId})/externalApprovalEntryDocuments({id})
```

Fallback

Wenn das Quelldokument gelöscht wurde oder der Genehmigungspostens-Link fehlt, gibt `recordJson` `"{}"` (eine leere JSON-Objektzeichenfolge) statt eines Fehlers zurück.

Datensatzanhänge

Jeder Genehmigungseintrag hat einen entsprechenden `externalApprovalEntryAttachments`-Endpunkt, der alle Standard-BC-Dokumentanhänge für den Quelldatensatz als JSON-Array zurückgibt. Jedes Element enthält die Dateimetadaten und den vollständigen Dateiinhalt als Base64-Zeichenfolge, bereit zur clientseitigen Darstellung oder zum Download.

```
GET {baseUrl}/companies({companyId})/externalApprovalEntryAttachments(b9f3a2c1-0001-ef11-bf8d-6045bd028c9f)
```

Beispielantwort:


```
{
  "id": "b9f3a2c1-0001-ef11-bf8d-6045bd028c9f",
  "entryNo": 42,
  "documentNo": "EK-00123",
  "recordAttachmentsJson": "[{\"fileName\":\"EK-00123 Spezifikation\",\"fileExtension\":\"pdf\",\"attachedDate\":\"2026-05-03T14:22:00Z\",\"attachedBy\":\"Alice Johnson\",\"lineNo\":0,\"contentBase64\":\"JVBERi0xLjQK...\"}]"
}
```

> `recordAttachmentsJson` ist eine JSON-Zeichenfolge — mit `JSON.parse()` parsen.

Anhang-Objektfelder

Feld	Typ	Beschreibung
`fileName`	String	Dateiname ohne Erweiterung.
`fileExtension`	String	Dateierweiterung ohne Punkt (z. B. `pdf`, `xlsx`, `png`).
`attachedDate`	Datum/Uhrzeit	Wann die Datei angehängt wurde (UTC).
`attachedBy`	String	BC-Benutzer-ID der Person, die die Datei angehängt hat.
`lineNo`	Integer	`0` für Anhänge auf Dokumentebene (Kopf); ungleich null für Anhänge auf Zeilenebene.
`contentBase64`	String	Vollständiger Dateiinhalt, Base64-kodiert. Leere Zeichenfolge, wenn die Datei keinen Inhalt hat. Mit `atob()` im Browser oder dem Base64-Decoder Ihrer Plattform dekodieren.

Clientseitiges Dekodierungsbeispiel (JavaScript)

```
const attachments = JSON.parse(entry.recordAttachmentsJson);
attachments.forEach(a => {
  const bytes = Uint8Array.from(atob(a.contentBase64), c => c.charCodeAt(0));
  const blob  = new Blob([bytes], { type: mimeTypeFor(a.fileExtension) });
  const url   = URL.createObjectURL(blob);
  // Download-Link rendern oder inline öffnen
});
```

Verwandte Dokumentation

Dokument	Zweck
`SETUP-GUIDE.de-DE.md`	Admin-Einrichtungshandbuch für die SharePoint-/Teams-Integration
`SHAREPOINT-CONTRACT.de-DE.md`	Vertrag für Integrationen, die die SharePoint-Liste direkt lesen/schreiben
`POWERAUTOMATE-DEV-GUIDE.de-DE.md`	Internes Handbuch zur Pflege des Power-Automate-Referenzflows

---

Revision #5
Created 2026-05-06 15:08:13 UTC by Bernd Feddersen
Updated 2026-06-24 12:18:53 UTC by Bernd Feddersen