Abruf von Daten eines einzelnen Gastes (fetch)

Swagger-Referenz: Operation und Antwortschema öffnen.

Worum es geht

Mit `fetch` lesen Sie die vereinbarten Ausgabefelder eines einzelnen Gastes über dessen Ticketcode aus. **Die Administration legt immer die Ausgabefeldnamen und deren Inhalte fest.** Dazu gehören auch zusammengesetzte Feldinhalte, beispielsweise ein Feld `Name`, das Vor- und Nachname zusammen ausgibt. Die Felder und Filter werden in einer eigenen Datenansicht vom Typ **fetch** festgelegt.

Aufruf

```
curl --get 'https://gwatch.events/ext-api/fetch' \
  --header 'x-api-key: IHR_HAUPTSCHLUESSEL' \
  --header 'x-api-flavor: IHR_FETCH_SCHLUESSEL' \
  --header 'Accept: application/json' \
  --data-urlencode 'code=TEST0001'
```

Ersetzen Sie `TEST0001` durch den Ticketcode des gesuchten Gastes. Durch `--data-urlencode` werden auch Sonderzeichen korrekt als URL-Parameter übertragen.

Parameter	Vorgabe
`code`	Erforderliche, nicht leere Zeichenfolge mit dem Ticketcode. Keine Datenbank-ID.

Der Hauptschlüssel bestimmt die Veranstaltung. Ein Flavor vom Typ `pull` wird hier nicht akzeptiert. Das Weglassen von `x-api-flavor` setzt eine aktive Standardansicht mit leerem Schlüssel vom Typ `fetch` voraus.

Erfolgreiche Antwort

Bei HTTP 200 werden die vereinbarten Gastfelder **direkt** zurückgegeben, wie im Swagger-Modell `fetchData`:

```
{
  "First name": "Susi",
  "Surname": "Sorglos",
  "Access": "IT-Security Concepts"
}
```

Die Namen `First name`, `Surname` und `Access` im Beispiel sind von der Administration gewählte Bezeichnungen; Ihre Variante kann vollständig andere Feldnamen und zusammengesetzte Werte liefern. Verwenden Sie für Ihre Anbindung die mit der Administration abgestimmte Zuordnung.

Es gibt keinen zusätzlichen Bereich `guests`, `event` oder `meta` und kein automatisch ergänztes `error`-Feld. Anders als bei `pull` werden auch `id`, `created`, `updated` und `deleted` nicht automatisch angefügt. Benötigen Sie zusätzliche Angaben, müssen diese mit dem Teilnehmermanagement als Ausgabefelder vereinbart werden.

Ohne konfigurierte Gastfelder kann die erfolgreiche Antwort ein leeres Array `[]` sein.

Filter und Formate

Die Datenansicht kann den Abruf auf bestimmte Gäste begrenzen, beispielsweise anhand einer Zustimmung oder eines Teilnahmemerkmals. HTTP 403 bedeutet deshalb nicht zwingend eine fehlende Einwilligung, sondern allgemein: Der bestehende Gast erfüllt die konfigurierten Bedingungen nicht.

Wie bei `pull` hängen Werte und Formate von den konfigurierten Feldern und Platzhaltern ab. Eine pauschale Lokalisierung nach der Sprache des jeweiligen Gastes ist nicht zugesichert. Stimmen Sie erwartete Werte insbesondere für Listen, Ja/Nein-Felder und Zeitangaben ab.

`fetch` liefert keinen festen Löschstatus. Ein HTTP-200-Ergebnis allein ist daher kein verlässlicher Nachweis, dass ein Gast nicht gelöscht oder noch teilnahmeberechtigt ist. Für einen Bestandsabgleich verwenden Sie pull einschließlich `deleted`.

Statusmeldungen

HTTP-Status	Bedeutung und nächster Schritt
`200`	Datensatz wurde entsprechend der Datenansicht geliefert.
`401`	Hauptschlüssel, Archivstatus, aktiven Flavor vom Typ `fetch` und IP-Filter prüfen. Der Body kann leer sein.
`403`	Gast vorhanden, aber durch die Bedingungen der Datenansicht ausgeschlossen. Filter mit dem Teilnehmermanagement prüfen.
`404`	Kein entsprechender Gast gefunden. Veranstaltung und Ticketcode prüfen.
`422`	`code` fehlt, ist leer oder kein geeigneter Zeichenfolgenwert.

Für 403 und 404 wird keine fachliche Fehlerbeschreibung mitgeliefert; rechnen Sie mit einem leeren JSON-Ergebnis statt eines `message`-Objekts. Verarbeiten Sie diese Fälle anhand des HTTP-Status.

Beispiel einer Validierungsantwort bei fehlendem `code`:

```
{
  "error": true,
  "message": {"code": ["The code parameter is missing."]}
}
```

Der konkrete Meldungstext kann sprachabhängig abweichen.

Funktion prüfen

1. Einen bekannten Ticketcode abrufen und die vereinbarten Felder kontrollieren.
2. Einen nicht vorhandenen Testcode prüfen; erwartet wird HTTP 404.
3. Falls Filter eingerichtet sind, einen abgestimmten Testgast außerhalb dieser Bedingungen prüfen; erwartet wird HTTP 403.
4. Einen Aufruf ohne `code` prüfen; bei gültigem Zugang wird HTTP 422 erwartet.

Weiter: Gäste anlegen und aktualisieren.

Zurück: Gästeliste abrufen.

Version #8
Erstellt: 2026-04-24 11:45:17 CEST von Volker Lehnen
Zuletzt aktualisiert: 2026-09-18 15:16:57 CEST von KI Dokumentation