

QrPlatform API Specification

Document Version: 2.0 (Beta)

Date: August 31, 2025

1. Overview & Architecture

1.1. Introduction

This document provides the complete technical specification for the QrPlatform RESTful API. The platform offers a tiered "Freemium" model for event creation, automated invitation generation, and attendee management.

1.2. Core Workflow (Pro Tier)

The API is built around an asynchronous workflow to ensure a responsive user experience.

1. A user authenticates via a frontend client using Firebase Authentication.
2. The user's identity is synchronized with the backend via POST /api/account/sync.
3. The user creates a Pro event via POST /api/event/create.
4. The backend validates the request, creates the event, and immediately enqueues a background job to generate the invitation images.
5. The client receives an immediate **202 Accepted** response.
6. The client can then periodically poll GET /api/event/{id}/download to check the job's status and retrieve the final download link.

2. Global Elements

2.1. Authentication

The API uses a Firebase-centric authentication model. All protected API endpoints require a Firebase ID Token (JWT) in the Authorization header.

Header: Authorization: Bearer <firebase_id_token>

2.2. Standardized Error Response

All predictable errors (4xx status codes) are returned in a standardized ProblemDetails JSON format.

Standard Error Shape:

```
{  
  "status": 403,  
  "title": "Domain.ErrorType",
```

```
"detail": "A human-readable description of the error."
}
```

3. API Endpoints

3.1. Account Management

POST /api/account/sync

- **Summary:** Synchronizes a Firebase user with the backend database. Must be called after login.
- **Authorization:** Bearer Token.
- **Success Response (200 OK):** "User synced successfully."
- **Error Responses:**
 - **401 Unauthorized:** The Firebase token is malformed or invalid.

3.2. Event Management

POST /api/event/create-free

- **Summary:** Creates a "Free" tier event.
- **Authorization:** Bearer Token.
- **Request Body:** multipart/form-data
 - eventInfo (form field): A JSON string with the event's core details.

```
{
  "name": "Community Meetup",
  "eventDate": "2025-10-20"
}
```
 - attendeeFile (file): An .xlsx or .csv file.
- **Success Response (201 Created):**

```
{
  "id": 105,
  "name": "Community Meetup",
  "eventDate": "2025-10-20",
  "createdAt": "2025-08-31T10:00:00Z",
  "location": null,
  "backgroundImageUri":
    "[https://placeholder.co/600x400/020618/FFFFFF/png?text=Community+Meetup](https://placeholder.co/600x400/020618/FFFFFF/png?text=Community+Meetup)",
  "generatedInvitationsZipUri": null,
  "appUserId": "user-guid-123"
```

}

- **Error Responses:**

- **400 Bad Request:** {"status": 400, "title": "File.Validation", "detail": "Invalid attendee file type. Please upload an XLSX or CSV file."}
- **403 Forbidden:** {"status": 403, "title": "Event.LimitReached", "detail": "You have reached the maximum number of events you can create within your current subscription plan."}

POST /api/event/create

- **Summary:** Creates a "Pro" tier event and enqueues the invitation generation job.
- **Authorization:** Bearer Token.
- **Permissions:** Requires "Pro" subscription.
- **Request Body:** multipart/form-data with eventInfo, templateInfo, attendeeFile, and backgroundImage.
- **Success Response (202 Accepted):** The newly created Event object (same shape as create-free).
- **Error Responses:**
 - **400 Bad Request:** {"status": 400, "title": "File.Validation", "detail": "The uploaded file is missing one or more required column headers (e.g., 'Name', 'Email')."}}
 - **403 Forbidden:** {"status": 403, "title": "User.SubscriptionLimit", "detail": "You cannot apply this action within your current subscription plan."}

GET /api/event/all

- **Summary:** Retrieves a paginated list of the user's events.
- **Authorization:** Bearer Token.
- **Success Response (200 OK):**

```
[
  {
    "id": 105,
    "name": "IQGC",
    "eventDate": "2025-08-10",
    "attendeeCount": 103,
    "backgroundImageUri":
"[https://placeholder.co/600x400/2B2543/FFFFFF/png?text=IQGC](https://placeholder.co/600x400/2B2543/FFFFFF/png?text=IQGC)",
    "role": "Owner"
  }
]
```
- **Error Responses:**

- **404 Not Found:** {"status": 404, "title": "Event.NotFound", "detail": "You do not have any events yet"}

GET /api/event/{id}

- **Summary:** Retrieves the full details for a single event.
- **Authorization:** Bearer Token.
- **Permissions:** Any collaborator role except "Check-In Staff".
- **Success Response (200 OK):**

```
{
  "id": 105,
  "name": "IQGC",
  "eventDate": "2025-08-10",
  "location": "Conference Hall",
  "backgroundImageUri": "/uploads/event_images/guid.png",
  "generatedInvitationFullPath": "/downloads/CompressedFiles/IQGC_105.zip",
  "attendees": [
    {
      "id": 210,
      "name": "John Doe",
      "email": "john.doe@example.com",
      "checkedIn": false,
      "checkedInTimestamp": null,
      "phoneNumber": "1234567890",
      "customId": "VIP123",
      "category": "VIP",
      "invitationStatus": "Not Sent"
    }
  ],
  "templateElements": [],
  "role": "Owner"
}
```

- **Error Responses:**
 - **403 Forbidden:** {"status": 403, "title": "Collaborator.Forbidden", "detail": "As a Check-In staff You do not have permission to perform this action"}
 - **404 Not Found:** {"status": 404, "title": "Event.NotFound", "detail": "The event with the specific ID was not found."}

GET /api/event/{id}/download

- **Summary:** Provides a download link for generated invitations.
- **Authorization:** Bearer Token.

- **Permissions:** Requires "Pro" subscription. Any collaborator role except "Check-In Staff".
- **Success Responses:**
 - **200 OK:** Job is complete. Body: `"/downloads/CompressedFiles/IQGC_105.zip"`
 - **202 Accepted:** Job is still processing. Body: `"Processing... Please Check later"`
- **Error Responses:**
 - **403 Forbidden (Subscription):** `{"status": 403, "title": "User.SubscriptionLimit", "detail": "You cannot apply this action within your current subscription plan."}`
 - **403 Forbidden (Permission):** `{"status": 403, "title": "Collaborator.Forbidden", "detail": "As a Check-In staff You do not have permission to perform this action"}`

PUT /api/event/edit

- **Summary:** Updates an event's name and date.
- **Authorization:** Bearer Token.
- **Permissions:** Requires 'Owner' or 'Editor' role.
- **Request Body (application/json):**

```
{
  "id": 105,
  "eventName": "New Event Name",
  "date": "2025-12-25"
}
```
- **Success Response (200 OK):** The updated EventSummaryDto object.
- **Error Responses:** 403 Forbidden, 404 Not Found.

DELETE /api/event/delete

- **Summary:** Deletes an event.
- **Authorization:** Bearer Token.
- **Permissions:** Requires 'Owner' role.
- **Query Parameter:** id (int).
- **Success Response:** 204 No Content.
- **Error Responses:** 403 Forbidden, 404 Not Found.

3.3. Attendee Management

POST /api/event/{id}/add-attendee

- **Summary:** Adds new attendees to an event.
- **Permissions:** Requires 'Owner' or 'Editor' role.
- **Request Body (application/json):**

```
[
  {
    "name": "Jane Doe",
    "email": "jane.doe@example.com",
```

```

    "phoneNumber": "9876543210",
    "customId": "GEN456",
    "category": "General"
  }
]

```

- **Success Response (200 OK):** An array of the created Attendee objects.
- **Error Responses:** 403 Forbidden, 404 Not Found.

PUT /api/event/{id}/edit-attendee

- **Summary:** Updates existing attendees' information.
- **Permissions:** Requires 'Owner' or 'Editor' role.
- **Request Body (application/json):**

```

[
  {
    "id": 210,
    "name": "Jane Smith",
    "email": "jane.smith@example.com",
    "checkedIn": false,
    "phoneNumber": "9876543210",
    "customId": "GEN456",
    "category": "General",
    "invitationStatus": "Sent"
  }
]

```

- **Success Response (200 OK):** An array of the updated EditAttendeesRequestDto objects.
- **Error Responses:**
 - **400 Bad Request:** {"status": 400, "title": "Attendee.Validation", "detail": "Two or more attendees to edit has the same new emails"}
 - **404 Not Found:** {"status": 404, "title": "Attendee.NotFound", "detail": "The attendees could not be found for this event."}

POST /api/event/{id}/delete-attendee

- **Summary:** Deletes attendees from an event.
- **Permissions:** Requires 'Owner' or 'Editor' role.
- **Request Body (application/json):** [{ "id": 210 }]
- **Success Response:** 204 No Content.
- **Error Responses:** 404 Not Found.

POST /api/event/{id}/invitation-send

- **Summary:** Enqueues a job to send WhatsApp invitations to attendees.
- **Permissions:** Any collaborator role.
- **Success Response (200 OK):** "Sent"
- **Error Responses:** 404 Not Found.

3.4. Collaborator Management

(Details for collaborator endpoints would follow the same format)

4. Business Rules: Collaborator Management

Guiding Philosophy:

- **Ownership is Power:** The Owner role grants full administrative control over an event.
 - **The Creator is Special (Initially):** The user who originally creates the event has unique privileges regarding other Owners.
 - **Power Transfer:** If the original Creator leaves the event, their special authority dissolves, and the remaining Owners gain full, equal administrative power.
 - **The Safety Net:** An event can never be left without at least one Owner. This rule overrides all others.
1. **Adding Collaborators:**
 - Only Owners can add new collaborators.
 - If the invited user exists, they are added immediately. If not, an invitation is created, which they will claim on their next login/sync.
 2. **Editing Collaborator Roles:**
 - Only Owners can edit roles.
 - **The Last Owner Rule:** An Owner cannot be demoted if they are the last and only Owner of the event.
 - **The Creator Privilege Rule:**
 - If the original Creator IS an active collaborator: Only the Creator themselves can demote another Owner.
 - If the original Creator IS NOT an active collaborator: Any remaining Owner can demote any other Owner (as long as the Last Owner Rule is not violated).
 3. **Removing Collaborators:**
 - Only Owners can remove other collaborators.
 - **The Last Owner Rule:** The last Owner of an event cannot be removed.
 - **The Creator Privilege Rule:**
 - If the original Creator IS an active collaborator: Only the Creator themselves can remove another Owner.
 - If the original Creator IS NOT an active collaborator: Any remaining Owner can remove any other Owner.
 4. **Leaving an Event (Self-Removal):**
 - Any collaborator can choose to leave an event.

- **The Exception:** A collaborator cannot leave an event if they are the sole Owner. They must either delete the event or promote another collaborator to Owner before they can leave.

5. Webhooks (External Integrations)

POST /api/webhooks/whatsapp

- **Summary:** Receives real-time status updates for WhatsApp messages sent to attendees.
- **Caller:** Wasender (or a similar WhatsApp API provider).
- **Purpose:** To update the invitationStatus of an attendee.
- **Example Payload (Status Update):**

```
{
  "event": "message.update",
  "data": {
    "key": {
      "id": "alphanumeric_message_id_from_provider",
      "fromMe": true
    },
    "status": 4
  }
}
```

6. Data Models (DTOs)

(This section would define the structure of DTOs like EventSummaryDto, AttendeeDto, etc., in a clear, readable format.)

6.1. Event DTOs

- **EventSummaryDto** (Response for GET /api/event/all)
 - id (int)
 - name (string)
 - eventDate (DateOnly)
 - attendeeCount (int)
 - backgroundImageUri (string?)
 - role (string?)
- **EventDetailsDto** (Response for GET /api/event/{id})
 - id (int)

- name (string)
- eventDate (DateOnly)
- location (string?)
- backgroundImageUri (string?)
- generatedInvitationFullPath (string?)
- attendees (Array of AttendeeDto)
- templateElements (Array of TemplateElementsDto)
- role (string)
- **UpdateEventRequestDto** (Request for PUT /api/event/edit)
 - id (int)
 - eventName (string, required)
 - date (DateOnly, required)

6.2. Attendee DTOs

- **AttendeeDto** (Used in EventDetailsDto response)
 - id (int)
 - name (string)
 - email (string)
 - checkedIn (boolean)
 - checkedInTimestamp (DateTime?)
 - phoneNumber (string?)
 - customId (string?)
 - category (string?)
 - invitationStatus (string)
- **AddAttendeesRequestDto** (Request for POST /api/event/{id}/add-attendee)
 - name (string, required)
 - email (string, required)
 - phoneNumber (string?)
 - customId (string?)
 - category (string?)
- **EditAttendeesRequestDto** (Request for PUT /api/event/{id}/edit-attendee)
 - id (int, required)
 - name (string, required)

- email (string, required)
- checkedIn (boolean)
- phoneNumber (string?)
- customId (string?)
- category (string?)
- invitationStatus (string?)
- **DeleteAttendeesRequestDto** (Request for POST /api/event/{id}/delete-attendee)
 - id (int, required)

6.3. Collaborator DTOs

- **AddCollaboratorsRequestDto** (Request for POST /api/event/{id}/addteam)
 - email (string, required)
 - role (string, required)
- **GetCollaboratorsResponseDto** (Response for GET /api/event/{id}/getteam)
 - userId (string)
 - email (string)
 - role (string)
 - displayName (string?)
 - pictureUrl (string?)
- **EditCollaboratorRequestDto** (Request for PUT /api/event/{id}/editteam)
 - userId (string, required)
 - email (string, required)
 - role (string, required)

7. Query Parameters & Enumerations

7.1. GET /api/event/all Query Parameters

Name	Type	Description	Default
Name	string	Filters events by name (case-insensitive contains).	-
OrderBy	string	Allowed Values: "name", "eventdate"	"eventdate"
IsDescending	boolean	-	false
PageNumber	int	-	1
PageSize	int	-	8

7.2. Enumeration Values

Role (Used in Collaborator DTOs)

- "Owner"
- "Editor"
- "Check-In Staff"

InvitationStatus (Used in Attendee DTOs)

- "Not Sent"
- "Pending"
- "Sent"
- "Delivered"
- "Read"
- "Played"