

RESTful API Conceptual Documentation Sample

This writing sample introduces the API for a web-based event hosting app, "Reactivities." With "Reactivities," users create social events, join others' events, and chat with event hosts and attendees. I coded it myself!

This sample includes a general overview, common use cases, an architectural overview, authentication and authorization procedures, and tutorials with code snippets.

"Reactivities" API Documentation

Overview

Welcome to the "Reactivities" API, which allows developers to integrate our unique event hosting functionality into their own applications. "Reactivities" connects users through event hubs, letting them host, RSVP for, and chat within different events. With our API, external developers can build creative tools to enhance the "Reactivities" experience, or use "Reactivities" event hosting and attendance management functionality in their own applications.

Common Use Cases

1. **Event Management:** You can use POST and DELETE requests to embed an event creation and management experience in your app or website. You can also use GET requests to allow your users to check on the status of your event(s), view how many tickets are left, and if there are different times or venues at which the event(s) are held.
2. **Attendee Management:** You can use POST and DELETE requests to embed an attendee management interface in your app or website. You can also use GET requests to allow your users to see how many attendees are expected to attend their event(s), as well as whether there are any last-minute cancellations or special requests.
3. **Ticket Sales:** You can use POST and DELETE requests to embed a custom ticket purchasing experience in your app or website. You can also use GET requests to collect registration information and view other data analytics reports.

Requirements

- “Reactivities” account
- A free API key from the “Reactivities” [Developer Portal](#)
- [Authentication](#)

Key Concepts

- **“Profile”**: One of the “Reactivities” central data structures that holds a user ID, basic details, connected events (either as host or attendee), and a list of other users who the user is followed by/following.
- **“Event”**: Another central data structure in “Reactivities” which holds an event ID, date/time/venue information, host ID, and a list of attendees.

Architecture

“Reactivities” utilizes RESTful principles to manage and maintain user and event data.

Data Model & Workflow Diagrams

1. Data Model

Profile

- ↳ UserID
- ↳ DisplayName
- ↳ Bio
- ↳ ImageUrl
- ↳ Events
- ↳ Photos
- ↳ Following
- ↳ Followers

Event

- ↳ EventID
- ↳ Title
- ↳ DateTime
- ↳ Description
- ↳ Category
- ↳ IsCancelled

```
↳Attendees
↳Location
    ↳City
    ↳Venue
    ↳Latitude
    ↳Longitude
```

2. Workflow

User Authentication → Profile Creation → Event Creation/Search → Sign Up for Event → Purchase Tickets → Follow Other Users

Getting Started

To get started with the “Reactivities” API, follow the steps below.

1. Registration

- Sign up for a “Reactivities” account on our [Developer Portal](#).
- Complete the registration process and verify your email to activate the account.

2. Getting an App Key

- Log into the Developer Portal.
- Navigate to the **App Management** section.
- Create a new application and you will receive a unique **App Key**. This key will authenticate requests to the “Reactivities” API.

3. Authorization

- All API requests require an authorization header with your App Key:

```
Authorization: Bearer YOUR_APP_KEY
```

- Use HTTPS for all requests to ensure secure data transmission.

Tutorials

Tutorial 1: Creating a User Profile (Host or Attendee)

1. Authenticate with your App Key.
2. Use the **POST /user** endpoint to create a new user profile.
 - Required fields: username, email.
3. You'll receive a unique User ID in the response.

Tutorial 2: Creating an Event

1. Authenticate with your App Key.
2. Use the **POST /event** endpoint to create a new event.
 - Required fields: event name, date, location.
3. You'll receive a unique Event ID in the response.

Tutorial 3: Retrieving Attendees for an Event

1. Use the **GET /attendees?eventId={yourEventId}** endpoint to retrieve previous matches.
2. You'll receive a list of attendees for the given event, complete with the dates and times those users signed up to attend the event.

Sample Code

Below are sample code snippets showing how to create a new event via the API. You can use these examples to integrate event creation or ticketing features into your own app.

cURL:

```
curl -X POST https://api.reactivities.com/v1/events \
-H "Authorization: Bearer YOUR_APP_KEY" \
-H "Content-Type: application/json" \
-d '{
  "name": "Summer Launch Party",
  "startDate": "2025-06-21T19:00:00Z",
  "endDate": "2025-06-21T23:00:00Z",
  "location": "San Francisco, CA",
  "capacity": 150
}'
```

JavaScript (Node.js):

```
import axios from "axios";

const API_BASE = "https://api.reactivities.com/v1";
const API_KEY = "YOUR_APP_KEY";

async function createEvent() {
  const response = await axios.post(
    `${API_BASE}/events`,
    {
      name: "Summer Launch Party",
      startDate: "2025-06-21T19:00:00Z",
      endDate: "2025-06-21T23:00:00Z",
      location: "San Francisco, CA",
      capacity: 150
    },
    {
      headers: {
        Authorization: `Bearer ${API_KEY}`,
        "Content-Type": "application/json"
      }
    }
  );
  console.log(response.data);
```

```
}
```

```
createEvent();
```

Python:

```
import requests

API_BASE = "https://api.reactivities.com/v1"
API_KEY = "YOUR_APP_KEY"

headers = {
    "Authorization": f"Bearer {API_KEY}",
    "Content-Type": "application/json"
}

def create_event():
    url = f"{API_BASE}/events"
    data = {
        "name": "Summer Launch Party",
        "startDate": "2025-06-21T19:00:00Z",
        "endDate": "2025-06-21T23:00:00Z",
        "location": "San Francisco, CA",
        "capacity": 150
    }
    response = requests.post(url, headers=headers, json=data)
    print(response.json())

create_event()
```

(end of writing sample)