RunRoute - API Specification Document

1. Overview

This document defines the complete API specification for RunRoute, including authentication, route planning, run tracking, social features, and user management. The API follows RESTful principles with real-time WebSocket connections for live features.

Base Configuration

- **Base URL**: `https://your-project.supabase.co/rest/v1`
- **WebSocket URL**: `wss://your-project.supabase.co/realtime/v1`
- **Authentication**: Bearer token (JWT) in Authorization header
- **Content-Type**: `application/json`
- **API Version**: `v1`

Response Format Standards

```json
{
  "success": true,
  "data": {},
  "message": "Success message",
  "timestamp": "2024-01-15T10:30:00Z"
}
```

Error Response Format

```json
```

```
{
  "success": false,
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid input parameters",
    "details": {
      "field": "target_distance_meters",
      "reason": "Must be greater than 0"
    }
  },
  "timestamp": "2024-01-15T10:30:00Z"
}
```

2. Authentication Endpoints

2.1 User Registration

```http
POST /auth/v1/signup
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "securePassword123",
  "data": {
    "username": "runner_joe",
    "full_name": "Joe Runner"
  }
}
```

Response (201 Created):

```json
```

```
{
  "success": true,
  "data": {
    "user": {
      "id": "uuid",
      "email": "user@example.com",
      "email_confirmed_at": null,
      "user_metadata": {
        "username": "runner_joe",
        "full_name": "Joe Runner"
      }
    },
    "session": {
      "access_token": "jwt_token",
      "refresh_token": "refresh_token",
      "expires_in": 3600,
      "token_type": "bearer"
    }
  }
}
```

2.2 User Login

http

```http
POST /auth/v1/token?grant_type=password
Content-Type: application/json

{
  "email": "user@example.com",
  "password": "securePassword123"
}
```

2.3 Refresh Token

http

```
POST /auth/v1/token?grant_type=refresh_token
Content-Type: application/json

{
  "refresh_token": "refresh_token_here"
}
```

2.4 Logout

```
http

POST /auth/v1/logout
Authorization: Bearer jwt_token
```

3. User Management Endpoints

3.1 Get User Profile

```
http

GET /users?id=eq.{user_id}
Authorization: Bearer jwt_token
```

Response (200 OK):

```
json
```

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "username": "runner_joe",
      "full_name": "Joe Runner",
      "avatar_url": "https://example.com/avatar.jpg",
      "bio": "Marathon runner from NYC",
      "preferred_units": "metric",
      "privacy_level": "public",
      "total_runs": 45,
      "total_distance_meters": 450000,
      "total_duration_seconds": 162000,
      "created_at": "2024-01-01T00:00:00Z",
      "last_active_at": "2024-01-15T10:30:00Z"
    }
  ]
}
```

3.2 Update User Profile

http

```
PATCH /users?id=eq.{user_id}
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "username": "new_username",
  "full_name": "Updated Name",
  "bio": "Updated bio",
  "preferred_units": "imperial",
  "privacy_level": "friends",
  "hide_exact_location": true,
  "location_privacy_radius_meters": 200
}
```

3.3 Search Users

http

```
GET /users?username=ilike.*{query}*&privacy_level=eq.public&limit=20
Authorization: Bearer jwt_token
```

3.4 Get User Statistics

```
http

GET /rpc/get_user_statistics
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "user_id": "uuid",
  "date_from": "2024-01-01",
  "date_to": "2024-01-31"
}
```

Response (200 OK):

```
json

{
  "success": true,
  "data": {
    "total_runs": 12,
    "total_distance_meters": 60000,
    "total_duration_seconds": 18000,
    "average_pace_per_km": 300,
    "longest_run_meters": 10000,
    "fastest_pace_per_km": 240,
    "active_days": 8,
    "weekly_summary": [
      {
        "week_start": "2024-01-01",
        "runs": 3,
        "distance_meters": 15000,
        "duration_seconds": 4500
      }
    ]
  }
}
```

4. Route Planning Endpoints

4.1 Generate Route

```http
POST /rpc/generate_route
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "start_latitude": 40.7128,
  "start_longitude": -74.0060,
  "end_latitude": 40.7580,
  "end_longitude": -73.9855,
  "target_distance_meters": 5000,
  "is_loop": true,
  "route_preferences": {
    "avoid_highways": true,
    "prefer_parks": true,
    "surface_type": "mixed"
  }
}
```

Response (200 OK):

```json
```

```
{
  "success": true,
  "data": {
    "route_polyline": "u{~vFdnbnjX@B@D@F?F?FAFAFCFCFEFEHeEiFiIgJkK",
    "actual_distance_meters": 5150,
    "estimated_duration_seconds": 1800,
    "elevation_gain_meters": 45,
    "waypoints": [
      {
        "latitude": 40.7128,
        "longitude": -74.0060,
        "instruction": "Start running north"
      }
    ],
    "route_source": "google"
  }
}
```

4.2 Save Planned Route

```http
POST /planned_routes
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "name": "Morning Central Park Loop",
  "description": "Easy morning run through the park",
  "target_distance_meters": 5000,
  "actual_distance_meters": 5150,
  "is_loop": true,
  "start_latitude": 40.7128,
  "start_longitude": -74.0060,
  "end_latitude": 40.7128,
  "end_longitude": -74.0060,
  "waypoints": [],
  "route_polyline": "encoded_polyline_string",
  "elevation_gain_meters": 45,
  "difficulty_level": 2,
  "route_source": "google"
}
```

4.3 Get User's Planned Routes

```http
GET /planned_routes?user_id=eq.{user_id}&order=created_at.desc&limit=20
Authorization: Bearer jwt_token
```

4.4 Update Planned Route

```http
PATCH /planned_routes?id=eq.{route_id}
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "name": "Updated Route Name",
  "waypoints": [
    {
      "latitude": 40.7150,
      "longitude": -74.0070,
      "order": 1
    }
  ],
  "route_polyline": "updated_polyline",
  "actual_distance_meters": 5300
}
```

4.5 Delete Planned Route

```http
DELETE /planned_routes?id=eq.{route_id}
Authorization: Bearer jwt_token
```

4.6 Modify Route with Waypoints

```http
```

```
POST /rpc/modify_route_waypoints
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "original_polyline": "current_polyline",
  "waypoints": [
    {
      "latitude": 40.7150,
      "longitude": -74.0070,
      "order": 1
    }
  ],
  "target_distance_meters": 5000
}
```

5. Run Tracking Endpoints

5.1 Start Run Session

```http
POST /rpc/start_run_session
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "planned_route_id": "uuid", // optional
  "start_latitude": 40.7128,
  "start_longitude": -74.0060,
  "target_distance_meters": 5000
}
```

Response (200 OK):

```json
```

```
{
  "success": true,
  "data": {
    "session_id": "uuid",
    "started_at": "2024-01-15T07:00:00Z",
    "status": "active"
  }
}
```

5.2 Update Run Progress (Real-time)

```
http

POST /rpc/update_run_progress
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "session_id": "uuid",
  "latitude": 40.7130,
  "longitude": -74.0058,
  "altitude": 10.5,
  "accuracy": 5.0,
  "speed": 3.5,
  "timestamp": "2024-01-15T07:05:00Z",
  "distance_meters": 250,
  "duration_seconds": 300
}
```

5.3 Pause/Resume Run

```
http

POST /rpc/pause_run
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "session_id": "uuid",
  "timestamp": "2024-01-15T07:10:00Z"
}
```

http

```http
POST /rpc/resume_run
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "session_id": "uuid",
  "timestamp": "2024-01-15T07:12:00Z"
}
```

5.4 Complete Run

http

```
POST /runs
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "title": "Morning Run",
  "notes": "Great weather today!",
  "planned_route_id": "uuid",
  "distance_meters": 5250,
  "duration_seconds": 1800,
  "average_pace_seconds_per_km": 342,
  "max_pace_seconds_per_km": 280,
  "min_pace_seconds_per_km": 400,
  "start_latitude": 40.7128,
  "start_longitude": -74.0060,
  "end_latitude": 40.7135,
  "end_longitude": -74.0055,
  "route_polyline": "actual_gps_polyline",
  "waypoints": [
    {
      "latitude": 40.7128,
      "longitude": -74.0060,
      "timestamp": "2024-01-15T07:00:00Z",
      "accuracy": 5.0,
      "speed": 0
    }
  ],
  "elevation_gain_meters": 52,
  "elevation_loss_meters": 48,
  "gps_accuracy_meters": 5.2,
  "paused_duration_seconds": 120,
  "started_at": "2024-01-15T07:00:00Z",
  "completed_at": "2024-01-15T07:30:00Z",
  "temperature_celsius": 15,
  "weather_condition": "sunny"
}
```

5.5 Get Run History

```
http
```

```
GET /runs?user_id=eq.{user_id}&order=started_at.desc&limit=20&offset=0
Authorization: Bearer jwt_token
```

Response (200 OK):

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "title": "Morning Run",
      "distance_meters": 5250,
      "duration_seconds": 1800,
      "average_pace_seconds_per_km": 342,
      "started_at": "2024-01-15T07:00:00Z",
      "completed_at": "2024-01-15T07:30:00Z",
      "route_polyline": "encoded_polyline",
      "elevation_gain_meters": 52,
      "notes": "Great weather today!",
      "weather_condition": "sunny"
    }
  ],
  "pagination": {
    "total": 45,
    "page": 1,
    "per_page": 20,
    "has_more": true
  }
}
```

5.6 Get Run Details

```http
GET /runs?id=eq.{run_id}
Authorization: Bearer jwt_token
```

5.7 Update Run

```http
```

```
PATCH /runs?id=eq.{run_id}
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "title": "Updated Run Title",
  "notes": "Added some notes",
  "manual_adjustments": {
    "distance_correction_meters": -50,
    "reason": "GPS error at tunnel"
  }
}
```

5.8 Delete Run

```
http

DELETE /runs?id=eq.{run_id}
Authorization: Bearer jwt_token
```

6. Social Features Endpoints

6.1 Create Run Post

```
http

POST /run_posts
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "run_id": "uuid",
  "caption": "Amazing morning run in Central Park! 🏃 #running #nyc",
  "is_public": true,
  "image_urls": [
    "https://storage.supabase.co/run-images/img1.jpg"
  ]
}
```

6.2 Get Social Feed

```
http
```

```http
GET /rpc/get_social_feed?limit=20&offset=0
Authorization: Bearer jwt_token
```

Response (200 OK):

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "run_id": "uuid",
      "user": {
        "id": "uuid",
        "username": "runner_joe",
        "full_name": "Joe Runner",
        "avatar_url": "https://example.com/avatar.jpg"
      },
      "run": {
        "distance_meters": 5250,
        "duration_seconds": 1800,
        "average_pace_seconds_per_km": 342,
        "route_polyline": "encoded_polyline",
        "started_at": "2024-01-15T07:00:00Z"
      },
      "caption": "Amazing morning run! 🏃",
      "image_urls": ["https://example.com/image.jpg"],
      "likes_count": 12,
      "comments_count": 3,
      "is_liked_by_user": false,
      "created_at": "2024-01-15T07:45:00Z"
    }
  ],
  "pagination": {
    "has_more": true,
    "next_offset": 20
  }
}
```

6.3 Like/Unlike Post

```http
```

```
POST /post_likes
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "post_id": "uuid"
}
```

```http
DELETE /post_likes?post_id=eq.{post_id}&user_id=eq.{user_id}
Authorization: Bearer jwt_token
```

6.4 Add Comment

```http
POST /post_comments
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "post_id": "uuid",
  "parent_comment_id": "uuid", // optional for replies
  "comment_text": "Great job! What was your pace?"
}
```

6.5 Get Post Comments

```http
GET /rpc/get_post_comments?post_id={post_id}&limit=20&offset=0
Authorization: Bearer jwt_token
```

Response (200 OK):

```json
```

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "user": {
        "id": "uuid",
        "username": "friend_runner",
        "full_name": "Friend Runner",
        "avatar_url": "https://example.com/avatar2.jpg"
      },
      "comment_text": "Great job! What was your pace?",
      "parent_comment_id": null,
      "replies": [
        {
          "id": "uuid",
          "user": {
            "id": "uuid",
            "username": "runner_joe",
            "full_name": "Joe Runner"
          },
          "comment_text": "Thanks! About 5:42 per km",
          "created_at": "2024-01-15T08:15:00Z"
        }
      ],
      "created_at": "2024-01-15T08:10:00Z"
    }
  ]
}
```

6.6 Follow/Unfollow User

```http
POST /user_follows
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "following_id": "uuid"
}
```

```http
DELETE /user_follows?follower_id=eq.{user_id}&following_id=eq.{target_user_id}
Authorization: Bearer jwt_token
```

6.7 Get User's Followers/Following

```http
GET /rpc/get_user_followers?user_id={user_id}&limit=20&offset=0
Authorization: Bearer jwt_token
```

```http
GET /rpc/get_user_following?user_id={user_id}&limit=20&offset=0
Authorization: Bearer jwt_token
```

7. External API Integration Endpoints

7.1 Google Places Search

```http
POST /rpc/search_places
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "query": "Central Park",
  "latitude": 40.7128,
  "longitude": -74.0060,
  "radius_meters": 5000
}
```

Response (200 OK):

```json
```

```
{
  "success": true,
  "data": [
    {
      "place_id": "ChIJqaIXqBZYwokRa1VR_kYfkgQ",
      "name": "Central Park",
      "formatted_address": "New York, NY, USA",
      "latitude": 40.7829,
      "longitude": -73.9654,
      "types": ["park", "establishment"],
      "rating": 4.7,
      "user_ratings_total": 120000
    }
  ]
}
```

7.2 Geocode Address

```
http

POST /rpc/geocode_address
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "address": "1 Central Park West, New York, NY"
}
```

7.3 Reverse Geocode

```
http

POST /rpc/reverse_geocode
Authorization: Bearer jwt_token
Content-Type: application/json

{
  "latitude": 40.7829,
  "longitude": -73.9654
}
```

8. Utility Endpoints

8.1 Upload Image

```http
POST /storage/v1/object/run-images/{filename}
Authorization: Bearer jwt_token
Content-Type: image/jpeg

[binary image data]
```

8.2 Get App Configuration

```http
GET /app_config?select=key,value
Authorization: Bearer jwt_token
```

8.3 Health Check

```http
GET /health
```

Response (200 OK):

```json
{
  "status": "healthy",
  "timestamp": "2024-01-15T10:30:00Z",
  "version": "1.0.0",
  "services": {
    "database": "healthy",
    "storage": "healthy",
    "realtime": "healthy"
  }
}
```

9. WebSocket Real-time Events

Connection Setup

javascript

```javascript
const socket = new WebSocket('wss://your-project.supabase.co/realtime/v1/websocket?apikey=YOUR_API_KEY');

// Subscribe to real-time updates
const subscription = {
  topic: 'realtime:public:run_posts',
  event: '*',
  payload: {},
  ref: '1'
};
```

9.1 Live Run Tracking Events

javascript

```javascript
// Subscribe to run progress updates
{
  "topic": "run_tracking:{session_id}",
  "event": "progress_update",
  "payload": {
    "session_id": "uuid",
    "latitude": 40.7130,
    "longitude": -74.0058,
    "distance_meters": 1250,
    "duration_seconds": 600,
    "current_pace": 288,
    "timestamp": "2024-01-15T07:10:00Z"
  }
}
```

9.2 Social Feed Real-time Events

javascript

```
// New post in feed
{
  "topic": "social_feed",
  "event": "INSERT",
  "payload": {
    "new": {
      "id": "uuid",
      "user_id": "uuid",
      "run_id": "uuid",
      "caption": "Just finished a great run!",
      "created_at": "2024-01-15T08:00:00Z"
    }
  }
}

// New like on post
{
  "topic": "post_likes",
  "event": "INSERT",
  "payload": {
    "new": {
      "post_id": "uuid",
      "user_id": "uuid",
      "created_at": "2024-01-15T08:05:00Z"
    }
  }
}
```

10. Error Codes and Handling

Common Error Codes

Code	HTTP Status	Description
`VALIDATION_ERROR`	400	Invalid input parameters
`UNAUTHORIZED`	401	Missing or invalid authentication
`FORBIDDEN`	403	Insufficient permissions
`NOT_FOUND`	404	Resource not found
`ROUTE_GENERATION_FAILED`	422	Unable to generate valid route
`GPS_ACCURACY_LOW`	422	GPS accuracy below minimum threshold
`RATE_LIMIT_EXCEEDED`	429	Too many requests

Code	HTTP Status	Description
EXTERNAL_API_ERROR	502	Google/Mapbox API error
STORAGE_UPLOAD_FAILED	502	File upload failed
DATABASE_ERROR	500	Internal database error

Error Response Examples

json

```json
{
  "success": false,
  "error": {
    "code": "ROUTE_GENERATION_FAILED",
    "message": "Unable to generate route with specified parameters",
    "details": {
      "reason": "No valid path found between start and end points",
      "suggestions": [
        "Try increasing the target distance",
        "Choose different start/end locations"
      ]
    }
  },
  "timestamp": "2024-01-15T10:30:00Z"
}
```

11. Rate Limiting

Limits by Endpoint Category

Category	Requests per Hour	Burst Limit
Authentication	60	10
Route Generation	100	5
User Management	300	20
Run Tracking	1000	50
Social Features	500	30
File Uploads	50	5

Rate Limit Headers

http

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1642248000
X-RateLimit-Retry-After: 3600
```

12. API Versioning Strategy

Version Header

```http
API-Version: v1
Accept: application/vnd.runroute.v1+json
```

Backward Compatibility

- Minor versions (v1.1, v1.2) maintain backward compatibility
- Major versions (v2.0) may introduce breaking changes
- Deprecated endpoints include `Sunset` header with removal date
- Minimum supported version lifecycle: 12 months

This comprehensive API specification provides a complete contract for RunRoute's frontend and any future integrations.