

Wayfare Travel Planning API Documentation

Base URL: <https://api.wayfare.com>

Authentication: Bearer Token (JWT)

Content-Type: application/json

BREAKING CHANGES - Privacy System Implementation

Released: December 2024

IMPORTANT: The backend has been updated with a comprehensive privacy system that affects route creation and retrieval. Please update your code accordingly.

Migration Required:

1. Route Creation Changes:

```
// OLD - No privacy control
const route = {
  title: "My Trip",
  city: "Paris",
  start_date: "2025-06-01",
  end_date: "2025-06-07"
}

// NEW - Privacy control added
const route = {
  title: "My Trip",
  city: "Paris",
  start_date: "2025-06-01",
  end_date: "2025-06-07",
  is_public: false // NEW FIELD - Routes are PRIVATE by default
}
```

2. Response Schema Changes:

```
// All route responses now include:
{
  "route_id": "...",
  "title": "...",
  // ... other fields ...
  "is_public": true, // NEW FIELD in all route responses
  "stats": { ... }
}
```

3. Public Route Access Changes:

```
// BEHAVIOR CHANGE: These endpoints now only return PUBLIC routes
GET /routes/public      // Only shows is_public: true routes
GET /routes/search      // Only searches is_public: true routes

// UNCHANGED: User routes still return all routes (public + private)
GET /routes/user        // Shows all user's routes regardless of privacy
```

New Endpoints:

```
// Toggle route privacy
PATCH /routes/{route_id}/privacy?is_public=true  // Make public
PATCH /routes/{route_id}/privacy?is_public=false // Make private

// Advanced public route search
GET /routes/search?q=Rome&city=Paris&sort_by=popularity
```

Required Code Updates:

1. **Update route creation forms** to include optional `is_public` checkbox
2. **Handle `is_public` field** in all route response parsers
3. **Update route listing UI** to show privacy status (lock icon for private routes)
4. **Add privacy toggle buttons** in route management interfaces
5. **Test public route discovery** to ensure only public routes appear

Database Impact:

- **All existing routes** automatically set to `is_public: false` (private)
- **No data loss** - all routes preserved with privacy protection
- **Database migration** completed automatically

Configuration

Email Configuration

- **Location:** `backend/config/email.py`
- **Development Mode:** Email sending suppressed, codes printed to console
- **Production Options:** Gmail (App Password), Mailtrap, Outlook
- **Verification Codes:** 6-digit, 10-minute expiry, in-memory storage

To switch email providers: 1. Edit `backend/config/email.py` 2. Change `mail_config = your_chosen_config` 3. Update credentials and restart server

User Management Endpoints

@POST /user/register

It allows users to: - Create a new user account - Register with username, email, and personal information - Automatically hash passwords and set default preferences

Request Body:

```
{
  "username": "traveler123",
  "password": "securepass123",
  "email": "user@example.com",
  "first_name": "John",
  "last_name": "Doe"
}
```

Response:

```
{
  "message": "User registered successfully",
  "success": true,
  "user_id": "64a1b2c3d4e5f6789abcdef0"
}
```

@POST /user/login

It allows users to: - Authenticate with username and password - Receive access token for API authorization - Maintain secure session management

Request Body:

```
{
  "username": "traveler123",
  "password": "securepass123"
}
```

Response:

```
{
  "message": "User logged in successfully",
  "success": true,
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

@POST /user/addInfo

Requires Authentication

It allows users to: - Add travel preferences and interests - Set budget and travel style preferences - Define home city for personalized recommendations

Request Body:

```
{
  "preferences": {
    "interests": ["museums", "architecture", "food"],
    "budget": "medium",
    "travel_style": "moderate"
  },
  "home_city": "New York"
}
```

Response:

```
{
  "message": "User information updated successfully",
  "success": true,
  "status_code": 200
}
```

@GET /user/getCurrentUser

Requires Authentication

It allows users to: - Retrieve current user profile information - Access personal preferences and settings - View account details and travel preferences

Response:

```
{
  "user_id": "64a1b2c3d4e5f6789abcdef0",
  "email": "user@example.com",
  "username": "traveler123",
  "name": "John",
  "surname": "Doe",
  "preferences": {
    "interests": ["museums", "architecture", "food"],
    "budget": "medium",
    "travel_style": "moderate"
  },
  "home_city": "New York"
}
```

@POST /user/changePassword

Requires Authentication

It allows users to: - Update account password securely - Verify current password before change - Ensure password confirmation matches

Request Body:

```
{
  "current_password": "oldpassword123",
  "new_password": "newpassword456",
  "confirm_password": "newpassword456"
}
```

Response:

```
{
  "message": "Password changed successfully",
  "success": true,
  "status_code": 200
}
```

@DELETE /user/delete

Requires Authentication

It allows users to: - Permanently delete their account - Remove all associated data and routes - Verify identity with password confirmation

Request Body:

```
{
  "password": "userpassword123"
}
```

Response:

```
{
  "message": "User account deleted successfully",
  "success": true,
  "status_code": 200
}
```

@POST /user/sendVerification

It allows users to: - Send verification code to email address - Start email verification process during registration - Receive 6-digit verification code valid for 10 minutes

Request Body:

```
{
  "email": "user@example.com"
}
```

Response:

```
{
  "success": true,
  "message": "Code sent successfully",
  "status_code": 200
}
```

@POST /user/sendVerification/verifyCode

It allows users to: - Verify email verification code - Complete email verification process - Validate 6-digit code within 10-minute expiry

Request Body:

```
{
  "verification_code": "123456"
}
```

Response:

```
{
  "success": true,
  "message": "Code verified successfully",
  "status_code": 200
}
```

Error Responses:

// Invalid code

```
{
  "detail": "Invalid verification code"
}
```

// Expired code (after 10 minutes)

```
{
  "detail": "Verification code has expired"
}
```

Route Management Endpoints

@POST /route/create

Requires Authentication

It allows users to: - Create a customized daily schedule of activities (linked to places) - Specify must-visit places with notes and preferences - Generate AI-powered route recommendations based on travel style - Automatically store place images in activities during creation for optimal retrieval performance

Request Body:

```
{
  "title": "Trip to Rome",
  "city": "Rome",
  "start_date": "2025-06-01",
  "end_date": "2025-06-07",
  "category": "city_break",
  "season": "summer",
  "is_public": false,
  "must_visit": [
    {
      "place_id": "abc123",
      "place_name": "Colosseum",
      "address": "Rome",
      "notes": "Morning preferred",
      "source": "google",
      "coordinates": { "lat": 41.89, "lng": 12.49 }
    },
    {
      "place_name": "Hidden Bookstore",
      "notes": "TikTok place",
      "source": "user"
    }
  ],
  "days": [
    {
      "date": "2025-06-01",
      "activities": [
        {
          "place_id": "abc123",
          "place_name": "Colosseum",
          "time": "10:00",
          "notes": "Visit the Colosseum"
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

Request Body Parameters: - **title** (string, required): Route title/name - **city** (string, required): Destination city name - **start_date** (string, required): Trip start date in YYYY-MM-DD format - **end_date** (string, required): Trip end date in YYYY-MM-DD format - **category** (string, optional): Route category (default: "city_break") - **season** (string, optional): Travel season (auto-detected if not provided) - **is_public** (boolean, optional): Route visibility - **false = private** (default), **true = public** - **must_visit** (array, required): List of must-visit places with details

Privacy Control: - **Private Routes** (**is_public**: false): Only visible to the creator via `/routes/user` - **Public Routes** (**is_public**: true): Visible to all users via `/routes/public` and `/routes/search` - **Default:** All routes are **private** unless explicitly set to public

Response:

```

{
  "message": "Route created successfully",
  "success": true,
  "route_id": "string",
  "status_code": 200
}

```

@GET /routes/user

Requires Authentication

It allows users to: - Retrieve all routes created by the authenticated user - View route summaries and basic information
- Access personal travel planning history - Get stored place images for each activity in the daily itinerary (images are stored during route creation for optimal performance)

Response:

```

{
  "success": true,
  "message": "User routes retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "route_id": "64a1b2c3d4e5f6789abcdef1",
      "user_id": "64a1b2c3d4e5f6789abcdef0",

```



```

        "title": "Trip to Rome",
        "city": "Rome",
        "country": "Italy",
        "start_date": "2025-06-01",
        "end_date": "2025-06-07",
        "budget": "medium",
        "travel_style": "moderate",
        "category": "city_break",
        "season": "summer",
        "stats": {
            "views_count": 5,
            "copies_count": 2,
            "likes_count": 8
        },
        "created_at": "2024-12-01T10:00:00Z",
        "updated_at": "2024-12-01T10:00:00Z"
    }
}
]
}

```

@GET /routes/{route_id}

Requires Authentication

It allows users to: - Retrieve detailed information for a specific route - Access complete daily schedules and activities - View all must-visit places and their details

Path Parameters: - route_id (string): Unique identifier of the route

Response:

```

{
    "success": true,
    "message": "Route retrieved successfully",
    "status_code": 200,
    "data": {
        "route_id": "64a1b2c3d4e5f6789abcdef1",
        "user_id": "64a1b2c3d4e5f6789abcdef0",
        "title": "Trip to Rome",
        "city": "Rome",
        "country": "Italy",
        "start_date": "2025-06-01",
        "end_date": "2025-06-07",
        "budget": "medium",
        "travel_style": "moderate",
    }
}

```

```

    "category": "city_break",
    "season": "summer",
    "is_public": false,
    "stats": {
        "views_count": 5,
        "copies_count": 2,
        "likes_count": 8
    },
    "must_visit": [
        {
            "place_id": "abc123",
            "place_name": "Colosseum",
            "address": "Rome",
            "coordinates": { "lat": 41.89, "lng": 12.49 },
            "notes": "Morning preferred",
            "source": "google",
            "opening_hours": {
                "monday": "08:30-19:15",
                "tuesday": "08:30-19:15"
            }
        }
    ],
    "days": [
        {
            "date": "2025-06-01",
            "activities": [
                {
                    "place_id": "abc123",
                    "place_name": "Colosseum",
                    "time": "10:00",
                    "notes": "Visit the Colosseum",
                    "image": "https://dynamic-media-cdn.tripadvisor.com/media/photo-o/15
                }
            ]
        }
    ],
    "created_at": "2024-12-01T10:00:00Z",
    "updated_at": "2024-12-01T10:00:00Z"
}
}

```

@PUT /routes/{route_id}

Requires Authentication

It allows users to: - Update existing route information - Modify travel dates, destinations, or preferences - Add or remove must-visit places and activities

Path Parameters: - route_id (string): Unique identifier of the route

Request Body:

```
{
  "title": "Updated Trip to Rome",
  "city": "Rome",
  "start_date": "2025-06-15",
  "end_date": "2025-06-20",
  "category": "cultural",
  "season": "summer",
  "must_visit": [
    {
      "place_name": "Vatican Museums",
      "notes": "Book tickets in advance",
      "source": "user"
    }
  ]
}
```

Response:

```
{
  "message": "Route updated successfully",
  "success": true,
  "status_code": 200
}
```

@DELETE /routes/{route_id}

Requires Authentication

It allows users to: - Permanently delete a specific route - Remove all associated activities and schedules - Clean up travel planning history

Path Parameters: - route_id (string): Unique identifier of the route

Response:

```
{
  "message": "Route deleted successfully",
  "success": true,
  "status_code": 200
}
```

@PATCH /routes/{route__id}/privacy

Requires Authentication Owner Only

It allows users to: - Toggle route privacy settings between public and private
- Control route visibility and discoverability - Manage sharing preferences for travel plans - Update privacy settings in real-time

Path Parameters: - route_id (string): Unique identifier of the route

Query Parameters: - is_public (boolean, required): **true** = make public, **false** = make private

Request Examples:

```
PATCH /routes/64a1b2c3d4e5f6789abcdef1/privacy?is_public=true    # Make public
PATCH /routes/64a1b2c3d4e5f6789abcdef1/privacy?is_public=false # Make private
```

Response:

```
{
  "success": true,
  "message": "Route privacy updated to public",
  "status_code": 200
}
```

Error Responses:

// User doesn't own the route

```
{
  "success": false,
  "message": "Unauthorized: You can only modify your own routes",
  "status_code": 403
}
```

// Route not found

```
{
  "success": false,
  "message": "Route not found",
  "status_code": 404
}
```

Privacy Effects: - **Making Public:** Route becomes discoverable via /routes/public and /routes/search - **Making Private:** Route removed from public listings, only accessible to owner - **Ownership:** Users can only change privacy of their own routes

@GET /routes/public

Requires Authentication

It allows users to: - Browse **only public routes** shared by other users (excludes private routes) - Filter public routes by category, season, and budget - Discover travel inspiration and ideas from the community - Access routes with `is_public: true` setting only

Query Parameters: - `category` (string, optional): Filter by route category - `season` (string, optional): Filter by travel season - `budget` (string, optional): Filter by budget level - `limit` (int): Maximum number of routes to return (default: 10)

Response:

```
{
  "success": true,
  "message": "Public routes retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "route_id": "64a1b2c3d4e5f6789abcdef2",
      "user_id": "64a1b2c3d4e5f6789abcdef9",
      "title": "Rome in 5 Days",
      "city": "Rome",
      "country": "Italy",
      "category": "city_break",
      "season": "spring",
      "budget": "medium",
      "stats": {
        "views_count": 156,
        "copies_count": 23,
        "likes_count": 89
      }
    }
  ]
}
```

@GET /routes/search

Requires Authentication

It allows users to: - **Search only public routes** with advanced filtering options - Find routes by title/name, city, or country - Filter by category, season, budget, and travel style - Sort results by popularity, rating, recent, or alphabetically - Discover travel content through flexible search queries

Query Parameters: - `q` (string, optional): Search query for route title (minimum 2 characters) - `city` (string, optional): Filter by exact city name - `country` (string, optional): Filter by exact country name - `category` (string, optional):

Filter by route category - **season** (string, optional): Filter by travel season - **budget** (string, optional): Filter by budget level - **travel_style** (string, optional): Filter by travel style - **limit** (int, optional): Maximum results (default: 20, max: 50) - **sort_by** (string, optional): Sort order - “popularity” (default), “rating”, “recent”, “title”

Request Examples:

```
GET /routes/search?q=Rome&limit=10 # Search by title
GET /routes/search?city=Paris&sort_by=popularity # Search by city
GET /routes/search?country=Italy&budget=medium # Search by country + budget
GET /routes/search?q=cultural&city=Rome&travel_style=relaxed # Combined search
```

Response:

```
{
  "success": true,
  "message": "Found 3 public routes for: title containing 'Rome', budget 'medium'",
  "status_code": 200,
  "data": [
    {
      "route_id": "64a1b2c3d4e5f6789abcdef2",
      "user_id": "64a1b2c3d4e5f6789abcdef9",
      "title": "Rome in 5 Days",
      "city": "Rome",
      "country": "Italy",
      "category": "city_break",
      "season": "spring",
      "budget": "medium",
      "is_public": true,
      "stats": {
        "views_count": 156,
        "copies_count": 23,
        "likes_count": 89
      }
    }
  ]
}
```

Search Features: - **Public Routes Only:** Searches only routes with `is_public: true` - **Smart Messages:** Response message shows what filters were applied - **Flexible Sorting:** Multiple sort options for different discovery needs - **Exact Matching:** City and country filters use exact matches - **Text Search:** Route title search supports partial matching

Places Management Endpoints

@GET /places/city

Requires Authentication

It allows users to: - Retrieve all available places in a specific city - Access place details including coordinates and categories - Browse destinations for route planning

Query Parameters: - city (string): Name of the city to search places in

Response:

```
{
  "success": true,
  "message": "Places retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "place_id": "abc123",
      "name": "Colosseum",
      "address": "Piazza del Colosseo, 1, 00184 Roma RM, Italy",
      "coordinates": { "lat": 41.8902, "lng": 12.4922 },
      "category": "historical_site",
      "rating": 4.5,
      "price_level": 3,
      "opening_hours": {
        "monday": "08:30-19:15",
        "tuesday": "08:30-19:15"
      }
    }
  ]
}
```

@POST /places/id

Requires Authentication

It allows users to: - Retrieve detailed information for specific places by their IDs - Get multiple place details in a single request - Access comprehensive place data for route planning

Request Body:

```
{
  "place_ids": ["abc123", "def456", "ghi789"]
}
```

Response:

```
{
  "success": true,
  "message": "Places retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "place_id": "abc123",
      "name": "Colosseum",
      "address": "Piazza del Colosseo, 1, 00184 Roma RM, Italy",
      "coordinates": { "lat": 41.8902, "lng": 12.4922 },
      "category": "historical_site",
      "rating": 4.5,
      "price_level": 3,
      "opening_hours": {
        "monday": "08:30-19:15",
        "tuesday": "08:30-19:15"
      }
    }
  ]
}
```

@POST /places/search**Requires Authentication**

It allows users to: - Search for places using text queries - Filter results by city, category, and other criteria - Find specific attractions and points of interest

Request Body:

```
{
  "query": "museum",
  "city": "Rome",
  "category": "cultural",
  "limit": 20
}
```

Response:

```
{
  "success": true,
  "message": "Search results retrieved successfully",
  "status_code": 200,
  "data": [
    {
```



```

        "place_id": "mus123",
        "name": "Vatican Museums",
        "address": "00120 Vatican City",
        "coordinates": { "lat": 41.9065, "lng": 12.4536 },
        "category": "museum",
        "rating": 4.4,
        "price_level": 4
      }
    ]
  }

```

@GET /places/search-must-visit

Requires Authentication

It allows users to: - Search and discover places for must-visit selection in route creation - Browse places by category with autocomplete functionality - Get comprehensive place information including ratings, images, and coordinates - Reduce typos and improve place selection accuracy in the UI

Query Parameters: - **city** (string, required): Name of the city to search places in - **query** (string, optional): Search term for place names (autocomplete) - **category** (string, optional): Filter by wayfare_category (Cultural Sites, Entertainment, etc.) - **limit** (integer, optional): Maximum number of results (default: 20, max: 50)

Example Request:

GET /places/search-must-visit?city=Rome&query=colos&category=Cultural%20Sites&limit=10

Response:

```

{
  "success": true,
  "message": "Found 1 places in Rome matching 'colos' for category 'Cultural Sites'",
  "status_code": 200,
  "data": [
    {
      "place_id": "ChIJrRMgU7ZhLxMR...",
      "name": "Colosseum",
      "category": "historical_site",
      "wayfare_category": "museum",
      "rating": 4.7,
      "image": "https://example.com/colosseum.jpg",
      "coordinates": {
        "lat": 41.8902,
        "lng": 12.4922
      }
    }
  ]
}

```

```

    },
    "address": "Piazza del Colosseo, 1, Roma RM, Italy"
  }
]
}

```

Usage in UI: - **Autocomplete:** Call with `query` parameter as user types - **Browse:** Call with `city` only to show popular places - **Filter:** Use `category` to filter by place types - **Selection:** Use returned `place_id` and `name` directly in route creation

Integration with Route Creation: The response data can be directly used in the `/route/create` endpoint's `must_visit` array:

```

{
  "place_id": "ChIJrRMgU7ZhLxMR...",
  "place_name": "Colosseum",
  "source": "database",
  "notes": "Morning visit preferred"
}

```

@POST /places/autocomplete

Requires Authentication

It allows users to: - Get autocomplete suggestions for place names - Provide real-time search assistance - Improve user experience during place selection

Request Body:

```

{
  "query": "colo",
  "city": "Rome",
  "limit": 5
}

```

Response:

```

{
  "success": true,
  "message": "Autocomplete results retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "place_id": "abc123",
      "name": "Colosseum",
      "category": "historical_site"
    },
  ],
}

```

```

    {
      "place_id": "abc124",
      "name": "Colonna di Traiano",
      "category": "monument"
    }
  ]
}

```

@GET /places/top-rated

Requires Authentication

It allows users to:

- Retrieve the top 5 rated places based on aggregated user feedback
- Get places with calculated average ratings from multiple feedback entries
- Access complete place details including original ratings and feedback statistics
- View places sorted by average feedback rating (highest first) and alphabetically by name

Response:

```

{
  "success": true,
  "message": "Top rated places retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "place_id": "abc123",
      "name": "Vatican Museums",
      "city": "Rome",
      "category": "museum",
      "wayfare_category": "cultural",
      "price": "€17",
      "rating": 4.5,
      "wayfare_rating": 4.8,
      "total_feedback_count": 15,
      "image": "https://example.com/vatican.jpg",
      "detail_url": "https://example.com/vatican-museums",
      "opening_hours": {
        "monday": "09:00-18:00",
        "tuesday": "09:00-18:00"
      },
      "coordinates": {
        "lat": 41.9065,
        "lng": 12.4536
      },
      "address": "Viale Vaticano, 00165 Roma RM, Italy",
    }
  ]
}

```

```

        "source": "wayfare",
        "country": "Italy",
        "country_id": "IT",
        "city_id": "rome_001",
        "popularity": 1.2,
        "duration": 240,
        "created_at": "2024-01-15T10:30:00Z",
        "updated_at": "2024-01-15T10:30:00Z"
    },
    {
        "place_id": "def456",
        "name": "Colosseum",
        "city": "Rome",
        "category": "historical_site",
        "wayfare_category": "historical",
        "price": "€16",
        "rating": 4.3,
        "wayfare_rating": 4.7,
        "total_feedback_count": 12,
        "image": "https://example.com/colosseum.jpg",
        "detail_url": "https://example.com/colosseum",
        "opening_hours": {
            "monday": "08:30-19:15",
            "tuesday": "08:30-19:15"
        },
        "coordinates": {
            "lat": 41.8902,
            "lng": 12.4922
        },
        "address": "Piazza del Colosseo, 1, 00184 Roma RM, Italy",
        "source": "wayfare",
        "country": "Italy",
        "country_id": "IT",
        "city_id": "rome_001",
        "popularity": 1.5,
        "duration": 180,
        "created_at": "2024-01-15T10:30:00Z",
        "updated_at": "2024-01-15T10:30:00Z"
    }
]
}

```

Response Fields: - **place_id**: Unique identifier for the place - **name**: Name of the place - **city**: City where the place is located - **category**: Primary category of the place - **wayfare_category**: Wayfare-specific category classification - **price**: Price information for visiting the place - **rating**: Original place rat-

ing from the database - **wayfare_rating**: Calculated average rating from user feedback (rounded to 2 decimal places) - **total_feedback_count**: Number of feedback entries for this place - **image**: URL to place image - **detail_url**: URL to detailed information about the place - **opening_hours**: Opening hours for each day of the week - **coordinates**: Geographical coordinates (latitude and longitude) - **address**: Full address of the place - **source**: Source of the place data - **country**: Country where the place is located - **country_id**: Country identifier - **city_id**: City identifier - **popularity**: Popularity score (lower numbers indicate higher popularity) - **duration**: Recommended visit duration in minutes - **created_at**: Timestamp when the place was added to the database - **updated_at**: Timestamp when the place was last updated

Sorting Logic: - Primary sort: Wayfare rating (highest first) - Secondary sort: Place name (alphabetical order) in case of rating ties

Cities Management Endpoints

@GET /cities/search

Requires Authentication

It allows users to: - Search for cities by name with real-time autocomplete functionality - Get city suggestions as users type in the destination field - Retrieve city details including coordinates for UI integration - Ensure only valid cities can be selected for route creation

Query Parameters: - **q** (required): Search query string (minimum 2 characters) - **limit** (optional): Maximum number of results to return (default: 10, max: 20)

Request Examples:

```
GET /cities/search?q=Ro&limit=5
GET /cities/search?q=Rome
GET /cities/search?q=Par&limit=3
```

Response:

```
{
  "success": true,
  "message": "Found 2 cities matching 'Ro'",
  "status_code": 200,
  "data": [
    {
      "city_id": "68824f500658a953e0de8233",
      "name": "Rome",
      "country": "Italy",
      "country_id": "IT",
```

```

        "display_text": "Rome, Italy",
        "coordinates": {
            "lat": 41.9028,
            "lng": 12.4964
        }
    },
    {
        "city_id": "68824f506d75e9efdf51f44c",
        "name": "Rotterdam",
        "country": "Netherlands",
        "country_id": "NL",
        "display_text": "Rotterdam, Netherlands",
        "coordinates": {
            "lat": 51.9244424,
            "lng": 4.47775
        }
    }
]
}

```

Response Fields: - **city_id**: Unique identifier for the city - **name**: City name (use this for route creation) - **country**: Country where the city is located - **country_id**: ISO country code - **display_text**: Formatted text for UI display (e.g., “Rome, Italy”) - **coordinates**: Geographical coordinates (latitude and longitude)

Usage in UI: 1. User types in search box → Call this endpoint when query 2 characters 2. Display **display_text** in dropdown options 3. When user selects a city → Use **name** field for route creation 4. Store full city object for reference and validation

Error Responses: - 401: Invalid or missing authentication token - 400: Query parameter missing or too short - 500: Server error during search

@GET /cities/all

Requires Authentication

It allows users to: - Retrieve a complete list of all available cities - Access city information including country, coordinates, timezone, and status - Browse destinations for travel planning with geographical data

Response:

```

{
    "success": true,
    "message": "Cities retrieved successfully",
}

```

```

"status_code": 200,
"data": [
  {
    "city_id": "68824f500e3bd95431ee840c",
    "name": "Rome",
    "country": "Italy",
    "country_id": "IT",
    "active": true,
    "coordinates": {
      "lat": 41.9028,
      "lng": 12.4964
    },
    "timezone": "Europe/Rome",
    "created_at": "2025-07-30T15:34:43.551000",
    "updated_at": "2025-07-30T15:34:43.551000"
  },
  {
    "city_id": "68824f50960d81e2439af257",
    "name": "Paris",
    "country": "France",
    "country_id": "FR",
    "active": true,
    "coordinates": {
      "lat": 48.8566,
      "lng": 2.3522
    },
    "timezone": "Europe/Paris",
    "created_at": "2025-07-30T15:34:43.551000",
    "updated_at": "2025-07-30T15:34:43.551000"
  }
]
}

```

@POST /cities/specific

Requires Authentication

It allows users to: - Retrieve cities within a specific country - Filter destinations by country preference with complete geographical data - Narrow down location options for trip planning with coordinates and timezone info

Request Body:

```

{
  "country": "Italy"
}

```

Response:

```
{
  "success": true,
  "message": "Cities in Italy retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "city_id": "68824f500e3bd95431ee840c",
      "name": "Rome",
      "country": "Italy",
      "country_id": "IT",
      "active": true,
      "coordinates": {
        "lat": 41.9028,
        "lng": 12.4964
      },
      "timezone": "Europe/Rome",
      "created_at": "2025-07-30T15:34:43.551000",
      "updated_at": "2025-07-30T15:34:43.551000"
    },
    {
      "city_id": "68824f50b1c9d54783ee842f",
      "name": "Milan",
      "country": "Italy",
      "country_id": "IT",
      "active": true,
      "coordinates": {
        "lat": 45.4642,
        "lng": 9.1900
      },
      "timezone": "Europe/Rome",
      "created_at": "2025-07-30T15:34:43.551000",
      "updated_at": "2025-07-30T15:34:43.551000"
    }
  ]
}
```

Countries Management Endpoints

@GET /countries/all

Requires Authentication

It allows users to: - Retrieve a complete list of all available countries - Access country information with creation/update timestamps - Browse destinations by

country for travel planning

Response:

```
{
  "success": true,
  "message": "Countries fetched successfully",
  "status_code": 200,
  "data": [
    {
      "name": "Italy",
      "country_id": "IT",
      "region": "Europe",
      "active": true,
      "created_at": "2025-07-30T16:06:03.201000",
      "updated_at": "2025-07-30T16:06:03.201000"
    },
    {
      "name": "France",
      "country_id": "FR",
      "region": "Europe",
      "active": true,
      "created_at": "2025-07-30T16:06:03.201000",
      "updated_at": "2025-07-30T16:06:03.201000"
    }
  ]
}
```

@POST /countries/region

Requires Authentication

It allows users to: - Retrieve countries within a specific geographical region - Filter destinations by regional preferences - Explore travel options by geographical area

Request Body:

```
{
  "region": "Southern Europe"
}
```

Response:

```
{
  "success": true,
  "message": "Countries by region retrieved successfully",
  "status_code": 200,
}
```

```

    "data": [
      {
        "_id": "country123",
        "name": "Italy",
        "region": "Southern Europe"
      },
      {
        "_id": "country126",
        "name": "Spain",
        "region": "Southern Europe"
      }
    ]
  }

```

@POST /countries/search

Requires Authentication

It allows users to: - Search for countries using text queries - Find specific destinations quickly - Filter countries by name or other criteria

Request Body:

```

{
  "query": "ital",
  "limit": 10
}

```

Response:

```

{
  "success": true,
  "message": "Search results retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "_id": "country123",
      "name": "Italy",
      "region": "Southern Europe"
    }
  ]
}

```

@GET /countries/allRegions

Requires Authentication

It allows users to: - Retrieve a complete list of all geographical regions - Access regional classification for countries - Browse destinations by geographical regions

Response:

```
{
  "success": true,
  "message": "Regions retrieved successfully",
  "status_code": 200,
  "data": [
    "Northern Europe",
    "Southern Europe",
    "Western Europe",
    "Eastern Europe",
    "North America",
    "South America",
    "Asia",
    "Africa",
    "Oceania"
  ]
}
```

Authentication & Error Handling

Authentication

- All endpoints marked with `require` require a valid JWT token
- Include token in the Authorization header: `Bearer <your_jwt_token>`
- Tokens expire after 30 minutes and need to be refreshed

Common Error Responses

401 Unauthorized:

```
{
  "detail": "Could not validate credentials",
  "success": false,
  "status_code": 401
}
```

400 Bad Request:

```
{
  "detail": "Invalid input data",
  "success": false,
  "status_code": 400
}
```

404 Not Found:

```
{
  "detail": "Resource not found",
  "success": false,
  "status_code": 404
}
```

500 Internal Server Error:

```
{
  "detail": "Internal server error",
  "success": false,
  "status_code": 500
}
```

API Usage Guidelines

Rate Limiting

- 1000 requests per hour per authenticated user
- 100 requests per hour for unauthenticated endpoints

Data Formats

- All dates should be in YYYY-MM-DD format
- Coordinates use decimal degrees (latitude, longitude)
- Times are in 24-hour format HH:MM

Travel Styles

- **relaxed:** Slower pace, longer visits, more downtime
- **moderate:** Balanced schedule with reasonable activities
- **accelerated:** Fast-paced, maximum activities per day

Budget Levels

- **low:** Budget-conscious options
- **medium:** Balanced cost options
- **high:** Premium experiences

Categories

- **city_break:** Urban exploration and cultural sites
- **beach:** Coastal and water-based activities
- **mountain:** Outdoor and adventure activities

- road_trip: Multi-destination driving routes
-

Feedback Management Endpoints

@POST /feedback/place

It allows users to: - Submit feedback and ratings for places they've visited - Add comments about their experience - Provide visit date information - Prevent duplicate feedback for the same place

Request Body:

```
{
  "place_id": "ChIJyWEHuEmuEmsRm9hTkapTCrk",
  "rating": 5,
  "comment": "Amazing place! The view was breathtaking and the staff was very friendly.",
  "visited_on": "2024-12-01"
}
```

Response:

```
{
  "success": true,
  "message": "Place feedback submitted successfully",
  "status_code": 201,
  "feedback_id": "64f8a1b2c3d4e5f6789abc01",
  "created_at": "2024-12-01T10:30:00Z"
}
```

@GET /feedback/place/{place_id}

It allows users to: - Retrieve all feedback for a specific place - View ratings and comments from other users - Access feedback history and statistics

Response:

```
{
  "success": true,
  "message": "Retrieved 15 feedback entries for place",
  "status_code": 200,
  "data": [
    {
      "feedback_id": "64f8a1b2c3d4e5f6789abc01",
      "user_id": "64a1b2c3d4e5f6789abcdef0",
      "place_id": "ChIJyWEHuEmuEmsRm9hTkapTCrk",
      "rating": 5,

```

```

        "comment": "Amazing place! The view was breathtaking.",
        "visited_on": "2024-12-01",
        "created_at": "2024-12-01T10:30:00Z",
        "updated_at": "2024-12-01T10:30:00Z"
    }
]
}

```

@GET /feedback/place/{place_id}/user/{user_id}

It allows users to: - Retrieve specific user's feedback for a place - Check if a user has already provided feedback - Access user-specific feedback details

Response:

```

{
  "success": true,
  "message": "User feedback retrieved successfully",
  "status_code": 200,
  "data": [
    {
      "feedback_id": "64f8a1b2c3d4e5f6789abc01",
      "user_id": "64a1b2c3d4e5f6789abcdef0",
      "place_id": "ChIJyWEHuEmuEmsRm9hTkapTCrk",
      "rating": 5,
      "comment": "Amazing place!",
      "visited_on": "2024-12-01",
      "created_at": "2024-12-01T10:30:00Z",
      "updated_at": "2024-12-01T10:30:00Z"
    }
  ]
}

```

@PUT /feedback/place/{feedback_id}

It allows users to: - Update their existing place feedback - Modify rating, comment, or visit date - Maintain feedback history with updated timestamps

Request Body:

```

{
  "rating": 4,
  "comment": "Updated review - still great but had some issues.",
  "visited_on": "2024-12-02"
}

```

Response:

```
{
  "success": true,
  "message": "Place feedback updated successfully",
  "status_code": 200,
  "updated_at": "2024-12-01T11:45:00Z"
}
```

@DELETE /feedback/place/{feedback_id}

It allows users to: - Delete their own place feedback - Remove feedback from the system - Maintain data integrity with proper authorization checks

Response:

```
{
  "success": true,
  "message": "Place feedback deleted successfully",
  "status_code": 200,
  "deleted_at": "2024-12-01T12:00:00Z"
}
```

@GET /feedback/place/{place_id}/stats

It allows users to: - Get aggregated statistics for place feedback - View average ratings and rating distribution - Access comprehensive feedback analytics

Response:

```
{
  "success": true,
  "message": "Place feedback statistics retrieved successfully",
  "status_code": 200,
  "data": {
    "total_feedback": 120,
    "average_rating": 4.2,
    "rating_distribution": {
      "1": 5,
      "2": 10,
      "3": 25,
      "4": 35,
      "5": 45
    }
  }
}
```

@POST /feedback/route

It allows users to: - Submit feedback and ratings for routes they've followed - Share their experience with route planning - Help other users with route recommendations

Request Body:

```
{
  "route_id": "68877b97cd6e495f565e416c",
  "rating": 4,
  "comment": "Great route! Well planned with good timing between locations.",
  "visited_on": "2024-12-01"
}
```

Response:

```
{
  "success": true,
  "message": "Route feedback submitted successfully",
  "status_code": 201,
  "feedback_id": "64f8a1b2c3d4e5f6789abc02",
  "created_at": "2024-12-01T14:30:00Z"
}
```

@GET /feedback/route/{route_id}

It allows users to: - Retrieve all feedback for a specific route - View user experiences and recommendations - Access route quality information

Response:

```
{
  "success": true,
  "message": "Retrieved 8 feedback entries for route",
  "status_code": 200,
  "data": [
    {
      "feedback_id": "64f8a1b2c3d4e5f6789abc02",
      "user_id": "64a1b2c3d4e5f6789abcdef0",
      "route_id": "68877b97cd6e495f565e416c",
      "rating": 4,
      "comment": "Great route! Well planned.",
      "visited_on": "2024-12-01",
      "created_at": "2024-12-01T14:30:00Z",
      "updated_at": "2024-12-01T14:30:00Z"
    }
  ]
}
```



```
}  
]  
}
```

@GET /feedback/route/{route_id}/user/{user_id}

It allows users to: - Retrieve specific user's feedback for a route - Check if a user has already provided route feedback - Access user-specific route feedback details

Response:

```
{  
  "success": true,  
  "message": "User route feedback retrieved successfully",  
  "status_code": 200,  
  "data": [  
    {  
      "feedback_id": "64f8a1b2c3d4e5f6789abc02",  
      "user_id": "64a1b2c3d4e5f6789abcdef0",  
      "route_id": "68877b97cd6e495f565e416c",  
      "rating": 4,  
      "comment": "Great route! Well planned.",  
      "visited_on": "2024-12-01",  
      "created_at": "2024-12-01T14:30:00Z",  
      "updated_at": "2024-12-01T14:30:00Z"  
    }  
  ]  
}
```

@PUT /feedback/route/{feedback_id}

It allows users to: - Update their existing route feedback - Modify rating, comment, or visit date - Maintain feedback history with updated timestamps

Request Body:

```
{  
  "rating": 3,  
  "comment": "Updated review - route was okay but had some timing issues.",  
  "visited_on": "2024-12-02"  
}
```

Response:

```
{
  "success": true,
  "message": "Route feedback updated successfully",
  "status_code": 200,
  "updated_at": "2024-12-01T15:45:00Z"
}
```

@DELETE /feedback/route/{feedback_id}

It allows users to: - Delete their own route feedback - Remove feedback from the system - Maintain data integrity with proper authorization checks

Response:

```
{
  "success": true,
  "message": "Route feedback deleted successfully",
  "status_code": 200,
  "deleted_at": "2024-12-01T16:00:00Z"
}
```

@GET /feedback/route/{route_id}/stats

It allows users to: - Get aggregated statistics for route feedback - View average ratings and success metrics - Access route performance analytics

Response:

```
{
  "success": true,
  "message": "Route feedback statistics retrieved successfully",
  "status_code": 200,
  "data": {
    "total_feedback": 45,
    "average_rating": 4.1,
    "rating_distribution": {
      "1": 2,
      "2": 3,
      "3": 8,
      "4": 15,
      "5": 17
    }
  }
}
```

Database Collections Schema

The Wayfare API uses MongoDB with the following collections structure:

User Collection

```
{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "email": "user@example.com", // Required, unique
  "first_name": "John",
  "last_name": "Doe",
  "username": "johndoe", // Display name, required
  "hashed_password": "bcrypt_hashed_password_string",
  "preferences": {
    "interests": ["Museums and Art Galleries", "Food & Drinks", "Outdoors", "Hidden Gems"],
    "budget": "medium", // can be: low, medium, high
    "travel_style": "relaxed" // can be: relaxed, moderate, accelerated
  },
  "home_city": "New York"
}
```

Route Collection

```
{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "route_id": "string", // Application-level route ID
  "user_id": "64a1b2c3d4e5f6789abcdef0", // Reference to user who created route
  "title": "Trip to Rome",
  "city": "Rome",
  "city_id": "city_objectid_string",
  "country": "Italy",
  "country_id": "country_objectid_string",
  "start_date": "2025-06-01",
  "end_date": "2025-06-07",
  "budget": "medium", // low, medium, high - inherited from user preferences
  "travel_style": "moderate", // relaxed, moderate, accelerated
  "category": "city_break", // city_break, beach, mountain, road_trip
  "season": "summer", // spring, summer, autumn, winter
  "stats": {
    "views_count": 0,
    "copies_count": 0,
    "likes_count": 0
  },
  "must_visit": [
    {
      "place_id": "abc123", // Optional - from places collection
      "place_name": "Colosseum",
    }
  ]
}
```

```

        "address": "Piazza del Colosseo, 1, 00184 Roma RM, Italy",
        "coordinates": { "lat": 41.89, "lng": 12.49 },
        "notes": "Morning preferred",
        "source": "google", // google, user, database
        "opening_hours": {
            "monday": "08:30-19:15",
            "tuesday": "08:30-19:15",
            "wednesday": "08:30-19:15",
            "thursday": "08:30-19:15",
            "friday": "08:30-19:15",
            "saturday": "08:30-19:15",
            "sunday": "08:30-19:15"
        }
    },
    "days": [
        {
            "date": "2025-06-01",
            "activities": [
                {
                    "place_id": "abc123", // Optional - from places collection
                    "place_name": "Colosseum",
                    "time": "10:00", // 24-hour format
                    "notes": "Visit the ancient amphitheater"
                }
            ]
        }
    ],
    "created_at": "2024-12-01T10:00:00Z",
    "updated_at": "2024-12-01T10:00:00Z"
}

```

Places Collection

```

{
    "_id": ObjectId, // MongoDB auto-generated unique identifier
    "place_id": "abc123", // Unique place identifier (from Google Places API or generated)
    "city": "Rome",
    "city_id": "city_objectid_string", // Reference to cities collection
    "country": "Italy",
    "country_id": "country_objectid_string", // Reference to countries collection
    "name": "Colosseum",
    "category": "historical_site", // attraction category
    "wayfare_category": "must_see", // Optional - our internal categorization
    "price": "€15-25", // Price range or specific price
    "rating": "4.5", // User rating (string format)
}

```

```

"popularity": "4.8", // Popularity score (string format)
"image": "https://example.com/colosseum.jpg", // Optional image URL
"detail_url": "https://maps.google.com/place/...", // Optional detail URL
"opening_hours": {
  "monday": "08:30-19:15",
  "tuesday": "08:30-19:15",
  "wednesday": "08:30-19:15",
  "thursday": "08:30-19:15",
  "friday": "08:30-19:15",
  "saturday": "08:30-19:15",
  "sunday": "08:30-19:15"
},
"coordinates": {
  "lat": 41.8902,
  "lng": 12.4922
},
"address": "Piazza del Colosseo, 1, 00184 Roma RM, Italy",
"source": "google", // google, manual, scraping
"duration": 120, // Recommended visit duration in minutes
"created_at": "2024-01-01T00:00:00Z",
"updated_at": "2024-01-01T00:00:00Z"
}

```

Cities Collection

```

{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "name": "Rome",
  "country": "Italy",
  "country_id": "country_objectid_string", // Reference to countries collection
  "active": true, // Whether city is available for route planning
  "coordinates": { // Optional - city center coordinates
    "lat": 41.9028,
    "lng": 12.4964
  },
  "timezone": "Europe/Rome", // Optional timezone information
  "created_at": "2024-01-01T00:00:00Z",
  "updated_at": "2024-01-01T00:00:00Z"
}

```

Countries Collection

```

{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "name": "Italy",
  "country_id": "IT", // ISO country code identifier

```

```

    "region": "Europe", // Geographical region classification
    "active": true, // Whether country is available for route planning
    "created_at": "2025-07-30T16:06:03.201000",
    "updated_at": "2025-07-30T16:06:03.201000"
  }
}

```

Place Feedback Collection

```

{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "user_id": "64a1b2c3d4e5f6789abcdef0", // Reference to user who submitted feedback
  "place_id": "ChIJyWEHuEmuEmsRm9hTkapTCrk", // Google Places ID or internal place identifier
  "rating": 5, // Integer from 1-5 rating scale
  "comment": "Amazing place! The view was breathtaking and staff was very friendly.", // Optional comment
  "visited_on": "2024-12-01", // Date when user visited the place (YYYY-MM-DD format)
  "created_at": "2024-12-01T10:30:00Z", // Timestamp when feedback was submitted
  "updated_at": "2024-12-01T10:30:00Z" // Timestamp when feedback was last modified
}

```

Route Feedback Collection

```

{
  "_id": ObjectId, // MongoDB auto-generated unique identifier
  "user_id": "64a1b2c3d4e5f6789abcdef0", // Reference to user who submitted feedback
  "route_id": "68877b97cd6e495f565e416c", // Reference to route from routes collection
  "rating": 4, // Integer from 1-5 rating scale
  "comment": "Great route! Well planned with good timing between locations.", // Optional comment
  "visited_on": "2024-12-01", // Date when user followed the route (YYYY-MM-DD format)
  "created_at": "2024-12-01T14:30:00Z", // Timestamp when feedback was submitted
  "updated_at": "2024-12-01T14:30:00Z" // Timestamp when feedback was last modified
}

```

Collection Relationships

```

Countries (1) ↔ (many) Cities ↔ (many) Places
      ↑                               ↓
Users (1) ↔ (many) Routes ↔ (many) Must_Visit_Places
      ↓               ↓               ↓
(many) Place_Feedback + Route_Feedback ↔ Places

```

- Users create multiple **Routes** and submit **Feedback**
- **Routes** reference **Cities** and **Countries**
- **Routes** contain **Must_Visit** places which may reference **Places**
- **Places** belong to **Cities** which belong to **Countries**
- **Place Feedback** links **Users** to **Places** with ratings and comments
- **Route Feedback** links **Users** to **Routes** with ratings and comments
- All collections use MongoDB ObjectIds for relationships

Key Features

- **Flexible Schema:** MongoDB allows for optional fields and schema evolution
- **Geocoding Integration:** Places can be added from external APIs (Google Places)
- **User-Generated Content:** Routes can include custom places not in the places collection
- **Hierarchical Data:** Countries → Cities → Places relationship
- **Feedback System:** Users can rate and review places and routes with validation
- **Duplicate Prevention:** One feedback per user per place/route to maintain data integrity
- **Statistics:** Dynamic calculation of ratings, averages, and distributions
- **Audit Trail:** All collections include created_at and updated_at timestamps

Ready to start building amazing travel experiences? Check out our Getting Started Guide and API Examples for implementation details!