

Travel Management API Documentation

Overview

- **Base URL (development):** `http://127.0.0.1:8000`
- **Authentication:** OAuth2 password flow issuing JWT access tokens. Include `Authorization: Bearer <token>` on protected endpoints.
- **Content types:** JSON unless noted.
- **Version:** 2.0.0 (`src/main.py`).

Standard Responses

```
{  
    "detail": "Message describing the outcome"  
}
```

- 401 is returned when the JWT is missing, expired, or revoked.
- 403 is returned when the authenticated user lacks the required role.
- 404 is returned when the requested entity does not exist.

Authentication Endpoints (/auth)

POST /auth/token

- **Description:** Exchange employee identifier (email or employee_id) and password for an access token.
- **Auth:** none.
- **Body (application/x-www-form-urlencoded):**
 - `username`: identifier (email or employee_id).
 - `password`: password.
- **Response 200:**

```
{  
    "access_token": "<jwt>",  
    "token_type": "bearer",  
    "employee_id": "EMP001",  
    "role": "employee",  
    "name": "Riya Sen",  
    "expires_in": 3600  
}
```

POST /auth/register

- **Description:** Create a new user and return an access token.
- **Auth:** none.
- **Body (JSON):**

```
{  
    "employee_id": "EMP001",
```

```

    "name": "Riya Sen",
    "email": "riya@example.com",
    "password": "Secret#123",
    "grade": "E3",
    "role": "employee",
    "department": "Sales",
    "designation": "Account Manager",
    "manager_id": "EMP050",
    "city": "Indore",
    "gender": "F"
}

```

- **Responses:**

- 201 with the same payload format as /auth/token.
- 409 if employee ID or email already exists.

POST /auth/logout

- **Description:** Revoke the current JWT (blacklisted until expiry).
- **Auth:** Bearer token.
- **Response 200:** { "message": "Logged out" }

GET /auth/me

- **Description:** Return the authenticated user profile.
- **Auth:** Bearer token.
- **Response 200:**

```
{
    "employee_id": "EMP001",
    "name": "Riya Sen",
    "email": "riya@example.com",
    "grade": "E3",
    "role": "employee",
    "manager_id": "EMP050",
    "is_active": true
}
```

Employee Endpoints (/employee, role = employee)

POST /employee/create-indent

- **Description:** Create or submit a travel indent.
- **Body (JSON):**

```
{
    "purpose_of_booking": "Client review meet",
    "travel_type": "domestic",
}
```

```

    "travel_start_date": "2025-12-04",
    "travel_end_date": "2025-12-06",
    "from_city": "Indore",
    "from_country": "India",
    "to_city": "Pune",
    "to_country": "India"
}



- Optional field: indent_id to resubmit a draft.
- Response 200: { "message": "Travel ticket created successfully.", "indent_id": "IND-20251118120000-1A2B3C" }
- Errors: 400 for validation issues (date range, duplicates).

```

POST /employee/save-draft

- **Description:** Save or update a travel indent draft.
- **Body:** same as /employee/create-indent; initial_status="draft" is applied server side.
- **Response 200:** { "message": "Draft saved.", "indent_id": "IND-..." }

GET /employee/my-indents

- **Description:** List all indents for the employee, including drafts and history.
- **Response 200:**

```
{
  "items": [
    {
      "indent_id": "IND-20251118120000-1A2B3C",
      "travel_start_date": "2025-12-04",
      "travel_end_date": "2025-12-06",
      "from_city": "Indore",
      "to_city": "Pune",
      "travel_type": "domestic",
      "purpose_of_booking": "Client review meet",
      "status_code": "accepted_manager",
      "status": "Approved by manager (Pending HR)",
      "created_at": "2025-11-18T11:59:57.000Z"
    }
  ]
}
```

GET /employee/profile

- **Description:** Return employee master data sourced from the users table.
- **Response 200:**

```
{
  "employee_id": "EMP001",
  "name": "Riya Sen",
  "email": "riya@example.com",
  "grade": "E3",
  "department": "Sales",
  "designation": "Account Manager"
}
```

- **Errors:** 404 if profile missing.

GET /employee/frequent-routes

- **Description:** Fetch bookmarked city pairs ordered by last usage.
- **Response 200:**

```
{
  "items": [
    {
      "bookmark_id": "RT-12AB34CD",
      "from_city": "Indore",
      "to_city": "Pune",
      "from_country": "India",
      "to_country": "India",
      "label": "Quarterly review",
      "times_used": 4,
      "last_used_at": "2025-10-01T09:20:00Z",
      "created_at": "2025-08-14T07:10:00Z"
    }
  ]
}
```

POST /employee/frequent-routes

- **Description:** Create a route bookmark (duplicates rejected).
- **Body (JSON):**

```
{
  "from_city": "Indore",
  "to_city": "Pune",
  "from_country": "India",
  "to_country": "India",
  "label": "Client visits"
}

• Response 200: { "bookmark_id": "RT-12AB34CD" }
• Errors: 400 if fields missing or duplicate exists.
```

DELETE /employee/frequent-routes/{bookmark_id}

- **Description:** Delete a bookmark belonging to the employee.
- **Response 200:** { "message": "Bookmark deleted" }
- **Errors:** 404 if not found.

POST /employee/frequent-routes/{bookmark_id}/use

- **Description:** Increment usage counters for analytics.
- **Response 200:** { "message": "Bookmark usage recorded" }

POST /employee/policy/chat

- **Description:** Ask the travel policy RAG assistant for guidance.
- **Body (JSON):**

```
{  
  "message": "What is the hotel limit for Grade E4 in Mumbai?",  
  "session_id": null  
}  
  
• Response 200:  
  
{  
  "response": "Grade E4 employees can select hotels up to INR 8,000 per night in Mumbai...",  
  "session_id": "7f9f7a6e-9c3f-4b5a-86c7-3f4f7a5c1b0d",  
  "sources": [  
    { "id": "1001", "text": "Mumbai hotel eligibility for Grade E4 is INR 8,000." }  
  ]  
}  
  
• Errors: 400 for empty message, 500 for upstream failures.
```

Manager Endpoints (/manager, role = manager)

GET /manager/indents

- **Description:** Retrieve all indents visible to the manager, enriched with employee profile data.
- **Response 200:** Array of travel indent records with `status` and `status_code` fields.

GET /manager/pending

- **Description:** Pending indents waiting for manager decision.

GET /manager/approved

- **Description:** Indents approved by the manager.

GET /manager/employee-profile/{employee_id}

- **Description:** Fetch profile details for a specific employee.

POST /manager/approve/{indent_id}

- **Description:** Mark an indent as manager approved.
- **Response 200:** { "status": "approved", "indent_id": "IND-..." }

POST /manager/reject/{indent_id}

- **Description:** Mark an indent as rejected by the manager.
- **Response 200:** { "status": "rejected", "indent_id": "IND-..." }

Manager endpoints currently rely on the authenticated manager identity but do not re-check ownership in every query. Ensure RBAC is enforced at the database layer for production.

HR AI Booking Endpoints (/hr-mcp, role = hr)

POST /hr-mcp/chat

- **Description:** Conversational interface that uses LLM + MCP tools to plan and book travel.
- **Body (JSON):**

```
{  
  "message": "Book the approved trip for EMP001",  
  "session_id": null,  
  "indent_id": "IND-20251118120000-1A2B3C"  
}  
  
• Response 200:  
  
{  
  "response": "Flights booked on Indigo 6E-502 and hotel Tech Park Inn confirmed.",  
  "session_id": "0f5c4f88-95f6-4b57-8e16-4bb34a9f2d87",  
  "tools_used": ["book_flight", "book_hotel"],  
  "booking_complete": true  
}  
  
• Errors: 500 if tool invocation fails. Responses include tool telemetry for diagnostics.
```

GET /hr-mcp/travel-indents

- **Description:** Return indents eligible for HR action (pending, manager-approved, HR in-progress).

PATCH /hr-mcp/tickets/{ticket_id}/status

- **Description:** Manually update ticket status (e.g., hr_approved, completed_hr, booked). Manager approval is required before HR status transitions.
- **Body (JSON):** { "status": "completed_hr" }
- **Response 200:** { "message": "Ticket status updated to completed_hr" }

DELETE /hr-mcp/sessions/{session_id}

- **Description:** Delete an AI chat session and its history.
- **Response 200:** { "message": "Session deleted" }

GET /hr-mcp/sessions/{session_id}/history

- **Description:** Retrieve stored conversational history for auditing.
- **Response 200:**

```
{  
    "session_id": "0f5c4f88-95f6-4b57-8e16-4bb34a9f2d87",  
    "history": [  
        { "role": "user", "content": "Book the approved trip for EMP001" },  
        { "role": "assistant", "content": "Confirmed flights and hotel." }  
    ]  
}
```

GET /hr-mcp/health

- **Description:** Diagnostics for MCP availability and active chat sessions.
- **Response 200:**

```
{  
    "status": "healthy",  
    "mcp_available": true,  
    "tools_available": 2,  
    "active_sessions": 1  
}
```

Root and Shared Endpoints

GET /

- **Description:** Service banner with feature overview.

GET /health

- **Description:** Composite health check combining traditional routes and MCP diagnostics.

- **Response 200:**

```
{  
    "status": "healthy",  
    "traditional_routes": "active",  
    "mcp_routes": "active",  
    "mcp_details": {  
        "mcp_available": true,  
        "tools_count": 2,  
        "active_sessions": 1  
    }  
}
```

Error Handling Guide

- **400:** Validation or business rule violation (duplicate routes, missing fields, manager approval prerequisite).
- **401:** Invalid credentials or revoked token. Prompt user to log in again.
- **403:** Role mismatch; ensure the correct persona is logged in.
- **404:** Resource not found (indent, bookmark, session).
- **500:** Unexpected server or downstream failure (LLM/MCP errors, database issues).