https://gemini.google.com/u/1/app/42a77d08d7a45bcb

### OpenAPI 3.1 Spec (for Module 7: Curated Itinerary Module)

YAML

```

openapi: 3.1.0

info:

title: Curated Itinerary Module API

version: v2.1

description: API for managing and accessing curated pilgrimage itineraries and related master data.

servers:

- url: https://<YOUR\_SUPABASE\_PROJECT\_REF>.supabase.co/rest/v1

description: Supabase Edge (PostgREST)

components:

securitySchemes:

bearerAuth:

type: http

scheme: bearer

bearerFormat: JWT # Standard Supabase JWT

schemas:

Error:

type: object

properties:

code:

type: string

description: Error code string.

message:

type: string

description: Human-readable summary of the error.

detail:

type: string

nullable: true

description: More specific details or field errors.

hint:

type: string

nullable: true

description: Suggestion on how to resolve the error.

required:

- code

- message

Pagination:

type: object

properties:

current\_page:

type: integer

example: 1

per\_page:

type: integer

example: 20

total\_items:

type: integer

example: 150

total\_pages:

type: integer

example: 8

CuratedItineraryBase: # Base fields for POST/PATCH, excluding read-only and linked objects

type: object

properties:

name: # Stored as base language (e.g., English)

type: string

maxLength: 255

description: "The human-readable name of the itinerary (Base language). Translatable. Max 255 chars."

slug:

type: string

maxLength: 255

pattern: '^[a-z0-9]+(?:-[a-z0-9]+)\*$'

description: "URL-friendly identifier. Auto-generated from name recommended. Max 255 chars."

itinerary\_code:

type: string

nullable: true

maxLength: 50

pattern: '^[A-Z0-9\_-]+$'

description: "Optional short code for internal reference. Max 50 chars. Unique when present."

primary\_trail\_id:

type: integer # Should be bigint if trails.id is bigint

format: int64

nullable: true

description: "The main trail this itinerary is primarily associated with (FK to public.trails.id)."

based\_on\_primary\_route\_id:

type: integer # Should be bigint if routes.id is bigint

format: int64

nullable: true

description: "Optional: If this itinerary largely follows an existing official route (FK to public.routes.id)."

short\_description: # Base language

type: string

nullable: true

maxLength: 500

description: "A brief summary of the itinerary (Base language). Translatable. Max 500 chars."

long\_description: # Base language

type: string

nullable: true

description: "Detailed description of the itinerary (Base language). Translatable."

theme\_or\_focus: # Base language

type: string

nullable: true

maxLength: 255

description: "Primary theme or focus (Base language). Translatable. Max 255 chars."

suitability\_notes: # Base language

type: string

nullable: true

description: "Notes on who this itinerary is best suited for (Base language). Translatable."

total\_walking\_days\_approx:

type: integer

nullable: true

minimum: 1

total\_nights\_approx:

type: integer

nullable: true

minimum: 0

total\_distance\_km\_approx:

type: number

format: float

nullable: true

exclusiveMinimum: 0

overall\_difficulty\_level\_id: # FK to trail\_difficulty\_levels\_master

type: integer

nullable: true

primary\_start\_location\_text: # Base language

type: string

nullable: true

maxLength: 255

description: "Descriptive starting point (Base language). Translatable."

primary\_end\_location\_text: # Base language

type: string

nullable: true

maxLength: 255

description: "Descriptive ending point (Base language). Translatable."

start\_waypoint\_id\_approx:

type: integer # Should be bigint if waypoints.id is bigint

format: int64

nullable: true

end\_waypoint\_id\_approx:

type: integer # Should be bigint if waypoints.id is bigint

format: int64

nullable: true

banner\_image\_media\_id:

type: string

format: uuid

nullable: true

map\_overview\_image\_media\_id:

type: string

format: uuid

nullable: true

content\_status\_id: # FK to content\_statuses\_master

type: integer

description: "Publication status. Default for 'draft' handled by backend/app."

published\_at:

type: string

format: date-time

nullable: true

is\_featured\_itinerary:

type: boolean

default: false

version\_notes:

type: string

nullable: true

data\_last\_verified\_at:

type: string

format: date-time

nullable: true

# For POST/PATCH, categories and seasons are typically managed via separate endpoints

# e.g., POST /itineraries/{id}/categories with body { "category\_id": 123 }

CuratedItineraryResponseItem: # For list view

allOf:

- $ref: '#/components/schemas/CuratedItineraryBase' # Includes most fields

type: object

properties:

itinerary\_id: # Read-only in response

type: integer

format: int64

readOnly: true

language\_code:

type: string

example: "it"

name: # Localized

type: string

short\_description: # Localized

type: string

nullable: true

difficulty\_level\_code:

type: string

nullable: true

difficulty\_level\_name: # Localized

type: string

nullable: true

content\_status\_code:

type: string

content\_status\_name: # Localized

type: string

category\_codes:

type: array

items:

type: string

example: ["FRANCISCAN\_SITES\_FOCUS", "HISTORICAL\_CULTURAL\_FOCUS"]

season\_codes:

type: array

items:

type: string

example: ["SPRING", "AUTUMN"]

itinerary\_created\_at: # Renamed to avoid conflict with base created\_at

type: string

format: date-time

readOnly: true

itinerary\_updated\_at: # Renamed

type: string

format: date-time

readOnly: true

# Note: The prompt's 'name\_en' + 'translations' object pattern is handled by

# the 'lang' query param and the view providing the localized 'name'.

# If explicit base + translations array needed, schema & view must change.

CuratedItineraryDetail:

allOf:

- $ref: '#/components/schemas/CuratedItineraryBase'

type: object

properties:

itinerary\_id:

type: integer

format: int64

readOnly: true

language\_code:

type: string

name: # Localized

type: string

# All other translatable fields from CuratedItineraryBase would be localized here

# e.g., long\_description, theme\_or\_focus, etc.

difficulty\_details:

type: object

properties:

code:

type: string

name: # Localized

type: string

description: # Localized

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

status\_details:

type: object

properties:

code:

type: string

name: # Localized

type: string

description: # Localized

type: string

nullable: true

banner\_image\_details:

type: object

nullable: true

properties:

media\_id:

type: string

format: uuid

original\_url:

type: string

format: uri

variants: # Example, actual structure depends on media.image\_variants\_json

type: object

properties:

thumb\_s\_webp:

type: string

format: uri

display\_l\_jpeg:

type: string

format: uri

alt\_text: # Localized

type: string

nullable: true

map\_overview\_image\_details: # Similar to banner\_image\_details

type: object

nullable: true

categories:

type: array

items:

type: object

properties:

category\_code:

type: string

name: # Localized

type: string

description: # Localized

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

seasons: # Similar to categories

type: array

items:

type: object

# Read-only audit fields from base table

created\_at:

type: string

format: date-time

readOnly: true

updated\_at:

type: string

format: date-time

readOnly: true

created\_by\_display\_name: # Example from view

type: string

nullable: true

updated\_by\_display\_name: # Example from view

type: string

nullable: true

CuratedItinerarySegmentItem: # For listing segments of an itinerary

type: object

properties:

itinerary\_segment\_id: # PK of the curated\_itinerary\_segments table

type: integer

format: int64

segment\_id: # FK to segments table

type: integer

format: int64

day\_number\_in\_itinerary:

type: integer

order\_of\_segment\_within\_day:

type: integer

segment\_name: # Localized name of the segment

type: string

segment\_slug:

type: string

distance\_km:

type: number

format: float

estimated\_walking\_time\_minutes:

type: integer

nullable: true

notes\_for\_segment\_in\_itinerary: # Localized

type: string

nullable: true

is\_optional\_segment\_for\_day:

type: boolean

# Schemas for Master Data (abbreviated, assuming simple GET /<table> for them)

ItineraryCategoryMaster:

type: object

properties:

id:

type: integer

category\_code:

type: string

name: # Localized

type: string

description: # Localized

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

sort\_order:

type: integer

is\_active:

type: boolean

# ... (Similar schemas for SeasonsMaster, TrailDifficultyLevelsMaster, ContentStatusesMaster)

paths:

/itineraries:

get:

summary: List Curated Itineraries

tags:

- Curated Itineraries

parameters:

- name: lang

in: query

required: false

schema:

type: string

default: "en" # Default language

description: Language code for localized content.

- name: category\_code

in: query

required: false

schema:

type: string

description: Comma-separated itinerary category codes.

- name: season\_code

in: query

required: false

schema:

type: string

description: Comma-separated season codes.

- name: difficulty\_level\_code

in: query

required: false

schema:

type: string

description: Comma-separated difficulty level codes.

- name: is\_featured

in: query

required: false

schema:

type: boolean

- name: page

in: query

required: false

schema:

type: integer

default: 1

- name: per\_page # Note: Supabase uses 'limit' and 'offset'

in: query

required: false

schema:

type: integer

default: 20

- name: sort\_by

in: query

required: false

schema:

type: string

- name: order

in: query

required: false

schema:

type: string

enum: [asc, desc]

default: asc

responses:

'200':

description: A list of curated itineraries.

# Relies on public.v\_curated\_itineraries\_list\_localized view

content:

application/json:

schema:

type: object

properties:

data:

type: array

items:

$ref: '#/components/schemas/CuratedItineraryResponseItem'

meta:

$ref: '#/components/schemas/Pagination'

default:

description: Unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

security:

- bearerAuth: [] # Assumes authentication might be needed even for lists if non-public data is possible

- apiKey: [] # Supabase requires anon key

post:

summary: Create a Curated Itinerary

tags:

- Curated Itineraries

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/CuratedItineraryBase'

responses:

'201':

description: Curated Itinerary created successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/CuratedItineraryDetail' # Returns the full detail

default:

description: Error creating itinerary

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

security:

- bearerAuth: [] # Requires authentication

- apiKey: []

/itineraries/{itinerary\_id\_or\_slug}:

get:

summary: Get Curated Itinerary Details

tags:

- Curated Itineraries

parameters:

- name: itinerary\_id\_or\_slug

in: path

required: true

schema:

type: string # Can be integer ID or string slug

description: The ID or slug of the curated itinerary.

- name: lang

in: query

required: false

schema:

type: string

default: "en"

description: Language code for localized content.

responses:

'200':

description: Detailed information about the curated itinerary.

# Relies on public.v\_curated\_itinerary\_detail\_localized view

content:

application/json:

schema:

$ref: '#/components/schemas/CuratedItineraryDetail'

'404':

description: Itinerary not found.

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

default:

description: Unexpected error

content:

application/json:

schema:

$ref: '#/components/schemas/Error'

security:

- bearerAuth: []

- apiKey: []

patch:

summary: Update a Curated Itinerary

tags:

- Curated Itineraries

parameters:

- name: itinerary\_id\_or\_slug # Usually ID for PATCH

in: path

required: true

schema:

type: integer # Assuming ID for updates

format: int64

description: The ID of the curated itinerary to update.

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/CuratedItineraryBase' # Partial updates allowed

responses:

'200':

description: Curated Itinerary updated successfully.

content:

application/json:

schema:

$ref: '#/components/schemas/CuratedItineraryDetail'

'404':

description: Itinerary not found.

default:

description: Error updating itinerary

security:

- bearerAuth: [] # Requires authentication

- apiKey: []

/itineraries/{itinerary\_id\_or\_slug}/segments:

get:

summary: List Segments for a Curated Itinerary

tags:

- Curated Itineraries

parameters:

- name: itinerary\_id\_or\_slug

in: path

required: true

schema:

type: string

description: The ID or slug of the curated itinerary.

- name: lang

in: query

required: false

schema:

type: string

default: "en"

- name: day\_number

in: query

required: false

schema:

type: integer

description: Filter segments for a specific day number.

responses:

'200':

description: An ordered list of segments for the itinerary.

# Would benefit from a public.v\_curated\_itinerary\_segments\_localized view

content:

application/json:

schema:

type: object

properties:

data:

type: array

items:

$ref: '#/components/schemas/CuratedItinerarySegmentItem'

meta: # e.g., total\_segments\_for\_itinerary

type: object

'404':

description: Itinerary not found.

default:

description: Unexpected error

security:

- bearerAuth: []

- apiKey: []

# Placeholder endpoints for master data (categories, seasons, difficulties, statuses)

# Example: /itinerary\_categories

/itinerary\_categories:

get:

summary: List Itinerary Categories

tags:

- Master Data

parameters:

- name: lang

in: query

required: false

schema:

type: string

default: "en"

- name: is\_active

in: query

required: false

schema:

type: boolean

default: true

responses:

'200':

description: A list of itinerary categories.

# Queries itinerary\_categories\_master and joins translations

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/ItineraryCategoryMaster'

default:

description: Unexpected error

security:

- bearerAuth: []

- apiKey: []

# ... (Similar GET endpoints for /seasons, /trail\_difficulty\_levels, /content\_statuses)

```

\* \* \* \* \*

### Quick-Start Examples

1. List Curated Itineraries (filtered, Italian):

Bash

```

curl -X GET\

"https://<YOUR\_SUPABASE\_PROJECT\_REF>.supabase.co/rest/v1/itineraries?lang=it&category\_code=eq.FRANCISCAN\_SITES\_FOCUS&select=itinerary\_id,name,slug,difficulty\_level\_name,category\_codes&limit=1"\

-H "apikey: <YOUR\_SUPABASE\_ANON\_KEY>"\

-H "Authorization: Bearer <YOUR\_USER\_JWT>"

```

Sample Response (leveraging `v\_curated\_itineraries\_list\_localized` columns):

JSON

```

[

{

"itinerary\_id": 123,

"name": "Pellegrinaggio Francescano: Tappa Umbra",

"slug": "pellegrinaggio-francescano-tappa-umbra",

"difficulty\_level\_name": "Moderato",

"category\_codes": ["FRANCISCAN\_SITES\_FOCUS", "HISTORICAL\_CULTURAL\_FOCUS"]

}

]

```

2. Get Curated Itinerary Details (English):

Bash

```

curl -X GET\

"https://<YOUR\_SUPABASE\_PROJECT\_REF>.supabase.co/rest/v1/itineraries?itinerary\_id\_or\_slug=eq.pellegrinaggio-francescano-tappa-umbra&lang=en&select=itinerary\_id,name,long\_description,difficulty\_details,categories,seasons,banner\_image\_details"\

-H "apikey: <YOUR\_SUPABASE\_ANON\_KEY>"\

-H "Authorization: Bearer <YOUR\_USER\_JWT>"

```

Sample Response (leveraging `v\_curated\_itinerary\_detail\_localized` columns):

JSON

```

[

{

"itinerary\_id": 123,

"name": "Franciscan Pilgrimage: Umbrian Stage",

"long\_description": "Detailed description of the Umbrian itinerary in the footsteps of St. Francis...",

"difficulty\_details": {

"code": "MODERATE",

"name": "Moderate",

"description": "Requires a fair level of fitness, some challenging sections.",

"icon\_identifier": "difficulty\_moderate\_icon"

},

"categories": [

{ "category\_code": "FRANCISCAN\_SITES\_FOCUS", "name": "Franciscan Sites Focus", "icon\_identifier": "franciscan\_cross\_icon" }

],

"seasons": [

{ "season\_code": "SPRING", "name": "Spring", "icon\_identifier": "flower\_icon" }

],

"banner\_image\_details": {

"media\_id": "a1b2c3d4-e5f6-7890-1234-567890abcdef",

"alt\_text": "Banner of the Umbrian itinerary"

}

}

]

```

\*(Note: Supabase PostgREST with `eq.` on a view might require careful view definition or RPC for slug/id resolution. For simplicity, assuming the view handles this or the client uses the ID directly if slug resolution is complex for PostgREST on views.)\*

3. Create a Curated Itinerary:

Bash

```

curl -X POST\

"https://<YOUR\_SUPABASE\_PROJECT\_REF>.supabase.co/rest/v1/curated\_itineraries"\

-H "apikey: <YOUR\_SUPABASE\_ANON\_KEY>"\

-H "Authorization: Bearer <YOUR\_ADMIN\_JWT>"\

-H "Content-Type: application/json"\

-H "Prefer: return=representation"\

-d '{

"name": "New Test Itinerary (Base English)",

"slug": "new-test-itinerary",

"short\_description": "A short test description (Base English).",

"content\_status\_id": 1, # Assuming ID for DRAFT status

"overall\_difficulty\_level\_id": 2 # Assuming ID for MODERATE

}'

```

Sample Response (actual itinerary created, then re-fetched for detail view):

JSON

```

{

"itinerary\_id": 126,

"language\_code": "en", # Default or based on system

"name": "New Test Itinerary (Base English)",

"slug": "new-test-itinerary",

"short\_description": "A short test description (Base English).",

"content\_status\_id": 1,

"overall\_difficulty\_level\_id": 2,

"created\_at": "2025-05-18T21:15:00Z"

// ... other fields from CuratedItineraryDetail schema

}

```

\* \* \* \* \*

### Implementation Notes

- 🟢 Schema OK. The existing database schema for Module 7, including the master tables (`trail\_difficulty\_levels\_master`, `content\_statuses\_master`, `itinerary\_categories\_master`, `seasons\_master`) and the core tables (`curated\_itineraries`, `curated\_itinerary\_segments`, and junction tables), is generally well-suited to support these API endpoints.

- 🟢 SQL Views are Key for READ Endpoints:

- The `public.v\_curated\_itineraries\_list\_localized` view is essential for efficiently serving the `GET /itineraries` endpoint.

- The `public.v\_curated\_itinerary\_detail\_localized` view is crucial for the `GET /itineraries/{id\_or\_slug}` endpoint. Its complexity means performance testing is vital. Consider if all JSONB aggregations are best here or if some (like categories/seasons) might be split to separate calls by the API gateway/BFF if the view becomes too slow.

- A new view, e.g., `public.v\_curated\_itinerary\_segments\_localized`, would greatly benefit the `GET /itineraries/{id}/segments` endpoint by pre-joining `curated\_itinerary\_segments`, `segments`, and their translations.

- 🟢 Indexing:

- Ensure comprehensive indexing on all foreign keys and commonly filtered/sorted columns in the base tables, as well as on the `public.translations` table (especially the composite index: `(table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translation\_status)`). This is critical for the performance of the localized views.

- 🟢 Language Fallback: The `COALESCE` pattern in the views provides a good fallback to base language content if a translation is missing. This should be clearly documented for API consumers.

- 🟢 Write Operations (POST, PATCH, DELETE):

- These will operate directly on the base tables (`public.curated\_itineraries`, `public.curated\_itinerary\_to\_category`, etc.).

- RLS policies on these tables will enforce permissions.

- Audit triggers will populate `created\_by/updated\_by` fields.

- The application/API layer will need to handle creating/updating base language content in the main table columns and then managing corresponding entries in the `public.translations` table for various languages.

- 🟠 Pagination for Supabase: Supabase PostgREST uses `limit` and `offset` for pagination rather than `page` and `per\_page`. The API gateway or BFF would need to translate these if `page/per\_page` is the desired external interface. The `Pagination` object in the OpenAPI spec reflects a common pattern; actual response headers from PostgREST might differ (e.g., `Content-Range`).

- 🔴 No critical schema tweaks are required \*solely\* from this API conceptualization, assuming the views are implemented effectively and underlying table indexing is robust. The main focus should be on view performance and RLS policy correctness.