<https://gemini.google.com/u/1/app/5b5f3092ace62463>

\* \* \* \* \*

### OpenAPI 3.1 Spec (for Module 8)

YAML

```

openapi: 3.1.0

info:

title: Platform Content API (Module 8)

version: v1.0

description: |-

API for managing platform content, including articles, their associated media, and media role definitions.

This module enables creation, retrieval, and management of general content like news and informational posts.

Primary reference language for content is English. Other languages are provided via translation objects.

servers:

- url: https://<your-project-ref>.supabase.co/rest/v1

description: Supabase Project REST API

- url: http://localhost:54321/rest/v1 # Local Supabase docker instance

description: Local Supabase instance

security:

- BearerAuth: [] # Applies BearerAuth to all operations unless overridden

paths:

/articles:

get:

summary: List Published Articles

operationId: listPublishedArticles

tags:

- Articles

parameters:

- name: lang

in: query

description: ISO language code for localized content (e.g., 'it'). Defaults to English.

required: false

schema:

type: string

example: it

- name: tags

in: query

description: Comma-separated list of tags to filter articles by (e.g., 'history,nature').

required: false

schema:

type: string

- name: region\_id

in: query

description: Filter articles associated with a specific region ID.

required: false

schema:

type: integer

format: int64

- name: trail\_id

in: query

description: Filter articles associated with a specific trail ID.

required: false

schema:

type: integer

format: int64

- name: town\_id

in: query

description: Filter articles associated with a specific town ID.

required: false

schema:

type: integer

format: int64

- name: sort\_by

in: query

description: Field to sort articles by.

required: false

schema:

type: string

enum: [publication\_date, title, created\_at]

default: publication\_date

- name: order

in: query

description: Sort order.

required: false

schema:

type: string

enum: [asc, desc]

default: desc

- $ref: '#/components/parameters/Page'

- $ref: '#/components/parameters/PageSize'

responses:

'200':

description: A paginated list of published articles.

content:

application/json:

schema:

type: object

properties:

pagination:

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

data:

type: array

items:

$ref: '#/components/schemas/ArticleSummary' # Use a summary for lists

default:

$ref: '#/components/responses/ErrorResponse'

# DB Optimization: Leverages public.published\_articles\_view and GIN index on articles.tags.

# A database function could further optimize translation and complex filtering.

post:

summary: Create a New Article

operationId: createArticle

tags:

- Articles

security:

- BearerAuth: ['admin\_platform', 'regional\_content\_manager'] # Example scopes

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'201':

description: Article created successfully.

content:

application/json:

schema:

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

'400':

$ref: '#/components/responses/ErrorValidation'

'401':

$ref: '#/components/responses/ErrorUnauthorized'

'403':

$ref: '#/components/responses/ErrorForbidden'

default:

$ref: '#/components/responses/ErrorResponse'

/articles/{article\_slug}:

get:

summary: Get Article by Slug

operationId: getArticleBySlug

tags:

- Articles

parameters:

- name: article\_slug

in: path

required: true

description: The URL-friendly slug of the article.

schema:

type: string

- name: lang

in: query

description: ISO language code for localized content. Defaults to English.

required: false

schema:

type: string

responses:

'200':

description: Detailed information about the article.

content:

application/json:

schema:

$ref: '#/components/schemas/ArticleDetail' # A more detailed schema

'404':

$ref: '#/components/responses/ErrorNotFound'

default:

$ref: '#/components/responses/ErrorResponse'

# DB Optimization: Uses index on articles.slug.

# A database function get\_article\_by\_slug(slug, lang) is highly recommended for optimal data assembly,

# including translated content and associated media gallery (from article\_media\_details\_view).

/media-roles:

get:

summary: List Media Roles

operationId: listMediaRoles

tags:

- MediaRoles

parameters:

- name: lang

in: query

description: ISO language code for localized content. Defaults to English.

required: false

schema:

type: string

responses:

'200':

description: A list of active media roles.

content:

application/json:

schema:

type: array

items:

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

default:

$ref: '#/components/responses/ErrorResponse'

# DB Optimization: Queries public.media\_roles\_master with index on is\_active.

post:

summary: Create a Media Role

operationId: createMediaRole

tags:

- MediaRoles

security:

- BearerAuth: ['admin\_platform'] # Example scope

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'201':

description: Media role created successfully.

content:

application/json:

schema:

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

'400':

$ref: '#/components/responses/ErrorValidation'

default:

$ref: '#/components/responses/ErrorResponse'

/media-roles/{role\_code}:

get:

summary: Get Media Role by Code

operationId: getMediaRoleByCode

tags:

- MediaRoles

parameters:

- name: role\_code

in: path

required: true

description: The unique code of the media role.

schema:

type: string

- name: lang

in: query

description: ISO language code for localized content. Defaults to English.

required: false

schema:

type: string

responses:

'200':

description: Detailed information about the media role.

content:

application/json:

schema:

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

'404':

$ref: '#/components/responses/ErrorNotFound'

default:

$ref: '#/components/responses/ErrorResponse'

/articles/{article\_slug}/media:

get:

summary: List Media for an Article

operationId: listArticleMedia

tags:

- Articles

- Media

parameters:

- name: article\_slug

in: path

required: true

description: The slug of the article.

schema:

type: string

- name: lang

in: query

description: ISO language code for localized overrides. Defaults to English.

required: false

schema:

type: string

responses:

'200':

description: A list of media items linked to the article.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/ArticleMediaItem' # Schema representing an item from article\_media\_details\_view

'404':

$ref: '#/components/responses/ErrorNotFound' # If article not found

default:

$ref: '#/components/responses/ErrorResponse'

# DB Optimization: Uses public.article\_media\_details\_view, joining article\_media, media, and media\_roles\_master.

# Query needs to first get article ID from slug.

components:

securitySchemes:

BearerAuth:

type: http

scheme: bearer

bearerFormat: JWT

description: "Supabase JWT token obtained after login. Example: `Authorization: Bearer YOUR\_JWT\_TOKEN`"

parameters:

Page:

name: page

in: query

description: Page number for pagination.

required: false

schema:

type: integer

default: 1

minimum: 1

PageSize:

name: page\_size

in: query

description: Number of items per page.

required: false

schema:

type: integer

default: 10

minimum: 1

maximum: 100

schemas:

Pagination:

type: object

properties:

page:

type: integer

example: 1

page\_size:

type: integer

example: 10

total\_items:

type: integer

example: 153

total\_pages:

type: integer

example: 16

# --- Article Schemas ---

ArticleBase:

type: object

properties:

title: # This field holds the English (primary reference) version

type: string

maxLength: 255

description: "Title of the article (in primary reference language - English)."

title\_translations:

type: object

additionalProperties:

type: string

description: "Translations for the title, mapping language\_code to translated title text."

example: {"it": "Titolo Dell'Articolo", "fr": "Titre de l'article"}

slug:

type: string

maxLength: 255

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

description: "URL-friendly slug."

excerpt: # English version

type: string

nullable: true

maxLength: 500

description: "Short summary of the article (in primary reference language - English)."

excerpt\_translations:

type: object

nullable: true

additionalProperties:

type: string

description: "Translations for the excerpt."

author\_profile\_id:

type: string

format: uuid

nullable: true

publication\_date:

type: string

format: date-time

nullable: true

article\_status:

$ref: '#/components/schemas/ContentVisibilityStatusEnum' # Assuming this enum is globally defined

featured\_image\_media\_id:

type: string

format: uuid

nullable: true

associated\_trail\_id:

type: integer

format: int64

nullable: true

associated\_region\_id:

type: integer

format: int64

nullable: true

associated\_town\_id:

type: integer

format: int64

nullable: true

tags:

type: array

items:

type: string

nullable: true

ArticleSummary: # For list views

allOf:

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

- type: object

properties:

id:

type: integer

format: int64

readOnly: true

author\_public\_display\_name: # Denormalized from profiles

type: string

nullable: true

featured\_image\_thumbnail\_url: # Example constructed URL

type: string

format: url

nullable: true

# publication\_date is already in ArticleBase

ArticleCreate: # For POST /articles

type: object

required:

- title # English title is required

- body\_content # English body\_content is required

properties:

title:

type: string

maxLength: 255

description: "Title of the article (in primary reference language - English)."

title\_translations:

type: object

additionalProperties:

type: string

slug: # Optional on create, can be auto-generated

type: string

maxLength: 255

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

nullable: true

body\_content: # English version

type: string

description: "Main content of the article (in primary reference language - English)."

body\_content\_translations:

type: object

additionalProperties:

type: string

excerpt: # English version

type: string

nullable: true

maxLength: 500

excerpt\_translations:

type: object

nullable: true

additionalProperties:

type: string

author\_profile\_id: # Usually set to current user or by admin

type: string

format: uuid

nullable: true

publication\_date:

type: string

format: date-time

nullable: true

article\_status:

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

default: 'draft'

featured\_image\_media\_id:

type: string

format: uuid

nullable: true

associated\_trail\_id:

type: integer

format: int64

nullable: true

associated\_region\_id:

type: integer

format: int64

nullable: true

associated\_town\_id:

type: integer

format: int64

nullable: true

tags:

type: array

items:

type: string

nullable: true

Article: # Represents the DB table public.articles

allOf:

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

- type: object

properties:

id:

type: integer

format: int64

readOnly: true

body\_content: # English version

type: string

description: "Main content of the article (in primary reference language - English)."

body\_content\_translations:

type: object

additionalProperties:

type: string

description: "Translations for the body content."

created\_at:

type: string

format: date-time

readOnly: true

updated\_at:

type: string

format: date-time

readOnly: true

created\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true # Typically set by system

updated\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true # Typically set by system

deleted\_at:

type: string

format: date-time

nullable: true

readOnly: true

ArticleDetail: # For GET /articles/{slug} response

allOf:

- $ref: '#/components/schemas/Article' # Includes full body\_content

- type: object

properties:

author: # Denormalized author details

type: object

nullable: true

properties:

id:

type: string

format: uuid

public\_display\_name:

type: string

avatar\_url: # Example constructed URL

type: string

format: url

nullable: true

featured\_image\_details: # Denormalized featured image details

type: object

nullable: true

properties:

media\_id:

type: string

format: uuid

alt\_text: # English version

type: string

nullable: true

alt\_text\_translations:

type: object

additionalProperties:

type: string

variants: # From media.image\_variants\_json

type: object

additionalProperties: # e.g., "thumbnail", "large"

type: string # path to variant

example: {"thumbnail": "/media/uuid-featured/thumb.webp", "large": "/media/uuid-featured/large.jpg"}

gallery: # From article\_media\_details\_view

type: array

items:

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

ArticleMediaItem: # Represents an item in an article's media gallery

type: object

properties:

article\_media\_id:

type: integer

format: int64

media\_id:

type: string

format: uuid

media\_role\_code:

type: string

media\_role\_name: # English version

type: string

media\_role\_name\_translations:

type: object

additionalProperties:

type: string

display\_order:

type: integer

caption\_override: # English version

type: string

nullable: true

caption\_override\_translations:

type: object

additionalProperties:

type: string

alt\_text\_override: # English version

type: string

nullable: true

alt\_text\_override\_translations:

type: object

additionalProperties:

type: string

# Media file details from public.media

storage\_bucket\_name:

type: string

storage\_object\_path\_original:

type: string

image\_variants\_json:

type: object

additionalProperties:

type: string

file\_mime\_type:

type: string

# --- MediaRole Schemas ---

MediaRoleBase:

type: object

properties:

default\_display\_name: # English version

type: string

maxLength: 100

default\_display\_name\_translations:

type: object

additionalProperties:

type: string

default\_description: # English version

type: string

nullable: true

maxLength: 255

default\_description\_translations:

type: object

nullable: true

additionalProperties:

type: string

icon\_identifier:

type: string

nullable: true

maxLength: 50

is\_active:

type: boolean

default: true

MediaRoleCreate: # For POST /media-roles

type: object

required:

- role\_code

- default\_display\_name # English name is required

properties:

role\_code:

type: string

maxLength: 50

pattern: '^[a-z0-9\_]+$'

default\_display\_name:

type: string

maxLength: 100

default\_display\_name\_translations:

type: object

additionalProperties:

type: string

default\_description:

type: string

nullable: true

maxLength: 255

default\_description\_translations:

type: object

nullable: true

additionalProperties:

type: string

icon\_identifier:

type: string

nullable: true

maxLength: 50

is\_active:

type: boolean

default: true

MediaRole: # Represents the DB table public.media\_roles\_master

allOf:

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

- type: object

properties:

role\_code:

type: string

maxLength: 50

pattern: '^[a-z0-9\_]+$'

created\_at:

type: string

format: date-time

readOnly: true

updated\_at:

type: string

format: date-time

readOnly: true

created\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true

updated\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true

# --- Shared Enum (Assuming it's globally available or defined elsewhere) ---

ContentVisibilityStatusEnum:

type: string

enum: [draft, pending\_review, published, archived, private\_link]

description: "Status of content visibility."

# --- Standard Error Schemas ---

Error:

type: object

required:

- code

- message

properties:

code:

type: string

description: A slug-style error code.

example: "resource\_not\_found"

message:

type: string

description: A human-readable message explaining the error.

example: "The requested article could not be found."

detail: # Matches PostgREST error detail and our earlier Security spec

type: string # Can also be an object for validation errors

nullable: true

description: More specific details or field-specific errors.

example: "Article with slug 'non-existent-slug' not found."

hint:

type: string

nullable: true

description: An optional hint on how to resolve the error.

# PostgREST also includes "details" string which can be more technical.

responses:

ErrorResponse:

description: Generic error response.

content:

application/json:

schema:

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

ErrorNotFound:

description: The specified resource was not found.

content:

application/json:

schema:

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

ErrorValidation:

description: Validation failed for the request.

content:

application/json:

schema:

$ref: '#/components/schemas/Error' # Detail might contain an object with field errors

ErrorUnauthorized:

description: Authentication credentials missing or invalid.

content:

application/json:

schema:

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

ErrorForbidden:

description: Authenticated user does not have permission for the action.

content:

application/json:

schema:

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

# External Docs (Optional, for more detailed API guidelines)

# externalDocs:

# description: Find more info here

# url: https://example.com/api-docs

```

\* \* \* \* \*

### 2\. Quick-Start Examples

1. List Published Articles (filtered by tag, requesting Italian)

Bash

```

curl -X GET "https://<your-project-ref>.supabase.co/rest/v1/articles?tags=history&lang=it&page\_size=1"\

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

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

```

\*Sample Response (compacted):\*

JSON

```

{

"pagination": {"page": 1, "page\_size": 1, "total\_items": 5, "total\_pages": 5},

"data": [

{

"id": 101,

"title": "La Battaglia Storica",

"title\_translations": {"en": "The Historic Battle"},

"slug": "la-battaglia-storica",

"excerpt": "Un evento cruciale nella storia...",

"excerpt\_translations": {"en": "A crucial event in history..."},

"author\_public\_display\_name": "Mario Bianchi",

"featured\_image\_thumbnail\_url": "/media/uuid-img-battle/thumb.webp",

"publication\_date": "2024-10-15T00:00:00Z"

}

]

}

```

2. Get Single Article Detail (requesting Italian)

Bash

```

curl -X GET "https://<your-project-ref>.supabase.co/rest/v1/articles/la-battaglia-storica?lang=it"\

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

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

```

\*Sample Response (compacted):\*

JSON

```

{

"id": 101,

"title": "La Battaglia Storica",

"title\_translations": {"en": "The Historic Battle"},

"slug": "la-battaglia-storica",

"body\_content": "Testo completo della battaglia...",

"body\_content\_translations": {"en": "Full text of the battle..."},

"author": {"id": "uuid-author-2", "public\_display\_name": "Mario Bianchi"},

"featured\_image\_details": {"media\_id": "uuid-img-battle", "alt\_text": "Dipinto della battaglia", "alt\_text\_translations": {"en": "Painting of the battle"}, "variants": {"large": "/media/uuid-img-battle/large.jpg"}},

"gallery": [

{"media\_id": "uuid-map-img", "media\_role\_name": "Mappa", "media\_role\_name\_translations": {"en": "Map"}, "caption\_override": "Mappa delle posizioni", "caption\_override\_translations": {"en": "Map of positions"}}

]

}

```

3. Create a New Article (as an admin/editor)

Bash

```

curl -X POST "https://<your-project-ref>.supabase.co/rest/v1/articles"\

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

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

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

-H "Prefer: return=representation"\

-d '{

"title": "New Pilgrim Shelter Opens",

"title\_translations": { "it": "Apre Nuovo Rifugio Pellegrino" },

"body\_content": "Details about the new shelter...",

"body\_content\_translations": { "it": "Dettagli sul nuovo rifugio..." },

"article\_status": "published",

"publication\_date": "2025-05-20T00:00:00Z",

"tags": ["shelter", "news"]

}'

```

\*Sample Response (201 Created):\*

JSON

```

{

"id": 102,

"slug": "new-pilgrim-shelter-opens", // Auto-generated if not provided and logic exists

"title": "New Pilgrim Shelter Opens",

"title\_translations": { "it": "Apre Nuovo Rifugio Pellegrino" },

"body\_content": "Details about the new shelter...",

"body\_content\_translations": { "it": "Dettagli sul nuovo rifugio..." },

"article\_status": "published",

"publication\_date": "2025-05-20T00:00:00Z",

"tags": ["shelter", "news"],

"created\_at": "2025-05-18T21:30:00Z"

// ... other fields with defaults or nulls

}

```

\* \* \* \* \*

### 3\. Implementation Notes

- 🟢 Schema OK: The database schema for `public.articles`, `public.media\_roles\_master`, and `public.article\_media` generally supports these API endpoints.

- Views:

- 🟢 The `public.published\_articles\_view` is highly beneficial for `GET /articles` and the main content of `GET /articles/{article\_slug}`.

- 🟢 The `public.article\_media\_details\_view` is crucial for populating the `gallery` in `GET /articles/{article\_slug}`.

- Indexes:

- 🟢 Existing indexes on `articles.slug`, `articles.article\_status`, `articles.tags` (GIN), and FKs are important.

- 🟢 Indexes on `media\_roles\_master.is\_active` and `article\_media` FKs and composite keys support their respective endpoints.

- Database Functions (Highly Recommended):

- To efficiently implement the specified translation model (`title`, `title\_translations`, etc.) and to assemble complex responses (like the article detail with its gallery), PostgreSQL functions callable via RPC are strongly recommended.

- For example, `get\_article\_for\_api(p\_slug TEXT, p\_lang TEXT)` could query base tables/views, join `public.translations` intelligently for \*all\* translatable fields, and structure the nested JSON as required by the OpenAPI spec. This centralizes complex query logic and translation handling in the database, simplifying the API gateway.

- Translation Handling: If not using database functions that handle translation internally, the API backend will need to:

1. Fetch the main entity (e.g., article data where `title` is English).

2. Fetch corresponding entries from `public.translations` for each translatable field for all required languages (or at least for the languages needed in the `\*\_translations` object).

3. Construct the response object with `fieldName` (English) and `fieldName\_translations` (object of other languages).

- Slug Generation: For `POST /articles`, if `slug` is not provided, application-layer or a DB trigger logic will be needed to generate it from the title, ensuring uniqueness.

- Authorization: Write endpoints (`POST /articles`, `POST /media-roles`) need robust authorization logic based on user roles (`admin\_platform`, `regional\_content\_manager`, etc.) as defined in the security blueprint. RLS policies on the tables are the primary enforcement mechanism.