https://gemini.google.com/u/1/app/5c36af70115d07fa

Okay, all files have been received. I will now proceed with the analysis and generation of the API contract and supporting documentation.

### OpenAPI 3.1 Spec

YAML

```

openapi: 3.1.0

info:

title: Via di Francesco Pilgrimage Platform API

version: v1.0

description: |-

API for the Via di Francesco Pilgrimage Platform.

This V1 provides core read endpoints and essential write endpoints for managing platform data including user profiles, media, trails, waypoints, and user interactions.

All text fields that are translatable will be returned in the language specified by the `lang` query parameter, falling back to the primary reference language (English per [cite: 670, 776]) if the translation is not available or `lang` is not specified.

contact:

name: Via di Francesco Platform Team

url: https://example.com/contact # Placeholder

servers:

- url: https://api.viadifrancesco.example.com/v1 # Placeholder

description: Production Server

- url: http://localhost:8000/rest/v1 # Placeholder for Supabase local PostgREST

description: Development Server (PostgREST)

tags:

- name: User & Content Infrastructure

description: User profiles, roles, languages, media, and translations management. [cite: 703, 1715]

- name: Core Trail Hierarchy

description: Management of trails, routes, and segments. [cite: 716]

- name: Geographical Context

description: Management of regions, provinces, and towns. [cite: 727]

- name: Waypoints

description: Points of Interest (POIs), categories, tags, and related waypoint details. [cite: 735]

- name: Waypoint Master Data

description: Management of master data related to waypoints like categories, tags, and content statuses. [cite: 1, 1834]

- name: Dynamic Conditions

description: Management of trail warnings and dynamic conditions. [cite: 748]

- name: User Interaction

description: User-generated content like votes, tips, and reviews. [cite: 754]

- name: Curated Itineraries

description: Management of curated pilgrimage itineraries. [cite: 758]

- name: Platform Content

description: Management of general platform content like articles. [cite: 765]

components:

schemas:

# Common Schemas

ApiError:

type: object

description: Standard error object. [cite: 114]

properties:

code:

type: string

description: A machine-readable error code or PostgREST error code. [cite: 114]

example: "PGRST116" # Example from Supabase/PostgREST [cite: 114]

message:

type: string

description: A human-readable summary of the error. [cite: 114]

example: "Invalid input."

details:

type: string

description: More specific details about the error. [cite: 114]

example: "Field 'email' must be a valid email address."

hint:

type: string

description: Optional hint on how to resolve the error. [cite: 114]

example: "Ensure the email format is correct."

required:

- code

- message

Pagination:

type: object

properties:

total\_items:

type: integer

description: Total number of items available.

example: 150

items:

type: array

items:

type: object # This will be replaced by specific resource schemas

offset:

type: integer

description: The offset (number of items to skip) from the beginning of the collection.

example: 0

limit:

type: integer

description: The maximum number of items to return.

example: 20

# Module 1: User & Content Infrastructure

Profile:

description: User profile information. [cite: 917, 1044]

type: object

properties:

id:

type: string

format: uuid

description: User's ID from auth.users table. [cite: 932, 934]

readOnly: true

username:

type: string

nullable: true

description: Optional unique username. [cite: 940, 943]

full\_name:

type: string

nullable: true

description: User's full name. [cite: 944, 948]

public\_display\_name:

type: string

nullable: true

description: Name shown publicly. [cite: 949, 952]

public\_avatar\_media\_id:

type: string

format: uuid

nullable: true

description: ID of the media item for public avatar. [cite: 953, 956]

public\_bio:

type: string

nullable: true

description: Short public biography. [cite: 958, 961]

preferred\_language\_code:

type: string

description: User's preferred language code (e.g., 'en', 'it'). [cite: 962, 965]

default: "en"

preferred\_units\_of\_measure:

type: string

description: User's preference for units ('metric', 'imperial'). [cite: 967, 969]

enum: ["metric", "imperial"]

default: "metric"

pilgrim\_experience\_level:

type: string

nullable: true

description: Self-assessed pilgrim experience level. [cite: 973, 976]

enum: [null, "novice\_first\_pilgrimage", "intermediate\_few\_pilgrimages", "experienced\_many\_pilgrimages", "long\_distance\_veteran"]

pilgrimage\_interests\_tags:

type: array

items:

type: string

nullable: true

description: Array of interest tags. [cite: 977, 980]

is\_profile\_publicly\_visible:

type: boolean

description: Master switch for public visibility of profile fields. [cite: 1000, 1003]

default: false

account\_status:

type: string

description: Current status of the user's application account. [cite: 1008, 1010]

enum: ["active", "pending\_verification", "email\_unconfirmed", "suspended\_by\_admin", "deactivated\_by\_user"]

default: "active"

readOnly: true

roles:

type: array

items:

type: string

description: Array of role codes assigned to the user. These roles are synchronized from `public.profiles.roles` to the JWT. [cite: 935, 939, 1716]

example: ["pilgrim\_user", "content\_moderator"]

public\_avatar: # Populated if include=avatar\_details

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

nullable: true

role\_details: # Populated if include=role\_details

type: array

items:

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

nullable: true

last\_activity\_at:

type: string

format: date-time

nullable: true

readOnly: true

description: Timestamp of the last recorded user activity. [cite: 1018, 1021, 1047]

created\_at:

type: string

format: date-time

description: Timestamp of profile creation. [cite: 1022, 1025]

readOnly: true

updated\_at:

type: string

format: date-time

description: Timestamp of last profile update. [cite: 1026, 1029]

readOnly: true

updated\_by\_profile\_id:

type: string

format: uuid

nullable: true

description: Profile ID of admin who last updated this profile. [cite: 1030, 1033]

readOnly: true

required:

- id

- roles

- preferred\_language\_code

- preferred\_units\_of\_measure

- is\_profile\_publicly\_visible

- account\_status

ProfileUpdate:

description: Fields for updating a user profile.

type: object

properties:

username:

type: string

nullable: true

full\_name:

type: string

nullable: true

public\_display\_name:

type: string

nullable: true

public\_avatar\_media\_id: # User would select an existing media ID or trigger an upload then link

type: string

format: uuid

nullable: true

public\_bio:

type: string

nullable: true

preferred\_language\_code:

type: string

preferred\_units\_of\_measure:

type: string

enum: ["metric", "imperial"]

pilgrim\_experience\_level:

type: string

nullable: true

enum: [null, "novice\_first\_pilgrimage", "intermediate\_few\_pilgrimages", "experienced\_many\_pilgrimages", "long\_distance\_veteran"]

pilgrimage\_interests\_tags:

type: array

items:

type: string

nullable: true

is\_profile\_publicly\_visible:

type: boolean

UserRoleMaster:

description: Defines an available user role. [cite: 1133, 1218]

type: object

properties:

role\_code:

type: string

description: Short, unique, machine-readable role code. [cite: 1144, 1147, 1219]

readOnly: true

default\_display\_name:

type: string

description: Human-readable name in the primary reference language (English). Translatable. [cite: 1149, 1153, 1220]

default\_description:

type: string

nullable: true

description: Detailed description of the role in the primary reference language. Translatable. [cite: 1155, 1159, 1221]

icon\_identifier:

type: string

nullable: true

description: Optional identifier for a UI icon. [cite: 1161, 1165, 1222]

is\_role\_active:

type: boolean

description: If true, the role is active and assignable. [cite: 1183, 1187]

readOnly: true

# translations: # Example from conceptual API[cite: 834], actual API would return translated fields directly based on lang

# type: array

# items:

# type: object

# properties:

# lang:

# type: string

# default\_display\_name:

# type: string

# default\_description:

# type: string

# nullable: true

required:

- role\_code

- default\_display\_name

- is\_role\_active

LanguageMaster:

description: Defines a language supported by the platform. [cite: 1304, 1363]

type: object

properties:

language\_code:

type: string

description: ISO 639-1 code or IETF tag (e.g., 'en', 'it'). [cite: 1314, 1317, 1365]

readOnly: true

display\_name\_native:

type: string

description: Name of the language in its native script. [cite: 1319, 1322, 1366]

display\_name\_en:

type: string

description: Common name of the language in English. [cite: 1323, 1327, 1367]

icon\_identifier:

type: string

nullable: true

description: Optional identifier for a UI icon. [cite: 1328, 1331, 1368]

is\_active\_for\_platform:

type: boolean

description: If true, language is actively supported. [cite: 1333, 1335, 1369]

readOnly: true # Typically managed by admins

display\_order\_ui:

type: integer

nullable: true

description: Optional order for display in UI. [cite: 1340, 1342, 1372]

required:

- language\_code

- display\_name\_native

- display\_name\_en

- is\_active\_for\_platform

Media:

description: Metadata for a media item. [cite: 1550, 1662]

type: object

properties:

id:

type: string

format: uuid

description: Unique identifier for the media item. [cite: 1561, 1564]

readOnly: true

uploader\_profile\_id:

type: string

format: uuid

nullable: true

description: Profile ID of the user who uploaded this media. [cite: 1565, 1567]

readOnly: true

storage\_bucket\_name:

type: string

description: Supabase Storage bucket name. [cite: 1569, 1572]

readOnly: true

storage\_object\_path\_original:

type: string

description: Full, unique path to the original object in its bucket. [cite: 1573, 1575]

readOnly: true

file\_name\_original:

type: string

description: Original filename as provided by the uploader. [cite: 1576, 1579]

file\_mime\_type:

type: string

description: MIME type of the original file. [cite: 1580, 1583]

file\_size\_bytes\_original:

type: integer

format: int64

nullable: true

description: Size of the original uploaded file in bytes. [cite: 1584, 1586]

image\_width\_px\_original:

type: integer

nullable: true

description: For images, the original width in pixels. [cite: 1587, 1589]

image\_height\_px\_original:

type: integer

nullable: true

description: For images, the original height in pixels. [cite: 1590, 1592]

image\_variants\_json:

type: object # JSONB in DB [cite: 1593, 1665, 1681]

nullable: true

description: |-

JSON object storing relative paths for optimized image versions.

Example: `{"thumb\_s\_webp": "variants/uuid\_thumb\_s.webp", "display\_m\_jpeg": "variants/uuid\_display\_m.jpg"}` [cite: 1597, 1681]

additionalProperties: true

default\_alt\_text:

type: string

nullable: true

description: Default alternative text in the primary reference language (English). Translatable. [cite: 1598, 1601, 1666]

default\_caption:

type: string

nullable: true

description: Default caption in the primary reference language (English). Translatable. [cite: 1602, 1606, 1667]

attribution\_text:

type: string

nullable: true

description: Credit for the photographer/source. Potentially translatable. [cite: 1607, 1610]

licence:

type: string

nullable: true

description: Usage license associated with the media item. [cite: 1616, 1618]

enum: [null, "all\_rights\_reserved", "cc\_by", "cc\_by\_sa", "cc\_by\_nd", "cc\_by\_nc", "cc\_by\_nc\_sa", "cc\_by\_nc\_nd", "cc0\_public\_domain", "uploader\_owns\_contact\_for\_use", "official\_permission\_granted", "unknown\_licence"] # from media\_licence\_enum [cite: 1654]

media\_asset\_type:

type: string

description: General type of media asset. [cite: 1619, 1620]

enum: ["image", "document\_pdf", "audio\_clip", "video\_clip", "gpx\_file", "other\_file"] # from media\_asset\_type\_enum [cite: 1653]

media\_status:

type: string

description: Current workflow/visibility status of the media item. [cite: 1621, 1624]

enum: ["processing\_upload", "pending\_review", "published\_approved", "rejected\_hidden", "archived", "error\_uploading"] # from media\_status\_enum [cite: 1655]

readOnly: true # Managed by specific endpoints or admin actions

tags:

type: array

items:

type: string

nullable: true

description: Array of keywords/tags for categorizing or searching media. [cite: 1625, 1627]

uploader\_profile: # Populated if include\_uploader\_profile=true [cite: 832]

type: object

nullable: true

properties:

id:

type: string

format: uuid

public\_display\_name:

type: string

created\_at:

type: string

format: date-time

description: Timestamp of when this media record was created. [cite: 1639, 1640]

readOnly: true

updated\_at:

type: string

format: date-time

description: Timestamp of when this media record was last updated. [cite: 1641, 1643]

readOnly: true

required:

- id

- storage\_bucket\_name

- storage\_object\_path\_original

- file\_name\_original

- file\_mime\_type

- media\_asset\_type

- media\_status

# Module 4: Waypoint Master Data

WaypointCategoryMaster:

description: Defines a broad classification for waypoints. [cite: 350, 353, 403]

type: object

properties:

id:

type: integer

description: Unique identifier for the waypoint category. [cite: 357, 358]

readOnly: true

code:

type: string

description: Short, stable, machine-readable code (e.g., 'accommodation\_location'). [cite: 359, 360, 383]

readOnly: true

label:

type: string

description: Human-readable name for the category. Translatable. [cite: 361, 362, 384]

description:

type: string

nullable: true

description: Optional description of the category. Translatable. [cite: 363, 385]

icon\_identifier:

type: string

nullable: true

description: Name, class, or path for a UI icon. [cite: 364, 365, 386]

requires\_detail\_table:

type: string

nullable: true

description: Name of the specific detail table this category implies (e.g., 'accommodations'). [cite: 366, 367, 387]

sort\_order:

type: integer

description: Determines display order in UI. [cite: 368, 369, 388]

is\_active:

type: boolean

description: True if the category definition is active. [cite: 370, 390]

readOnly: true # Typically managed by admins

required:

- id

- code

- label

- sort\_order

- is\_active

TagMaster:

description: Defines a curated descriptive tag. [cite: 461, 465, 514]

type: object

properties:

id:

type: integer

description: Unique identifier for the tag. [cite: 469, 470]

readOnly: true

tag\_code:

type: string

description: Short, stable, machine-readable code for the tag (e.g., 'franciscan\_site'). [cite: 471, 472, 494]

readOnly: true

label:

type: string

description: Human-readable name for the tag. Translatable. [cite: 473, 474, 495]

description:

type: string

nullable: true

description: Optional description of the tag. Translatable. [cite: 475, 496]

tag\_type:

type: string

nullable: true

description: Optional type to group tags (e.g., 'amenity', 'poi\_feature'). [cite: 476, 477, 497]

icon\_identifier:

type: string

nullable: true

description: Optional UI icon identifier. [cite: 478, 479, 499]

sort\_order:

type: integer

description: Determines display order. [cite: 480, 481, 500]

is\_active:

type: boolean

description: True if the tag definition is active. [cite: 482, 501]

readOnly: true # Typically managed by admins

required:

- id

- tag\_code

- label

- sort\_order

- is\_active

ContentStatusMaster:

description: Defines a content lifecycle status. [cite: 571, 575, 600]

type: object

properties:

id:

type: integer

description: Unique identifier for the content status. [cite: 578, 579]

readOnly: true

code:

type: string

description: Short, stable, machine-readable code (e.g., 'draft', 'published'). [cite: 580, 581, 602]

readOnly: true

label:

type: string

description: Human-readable name for the status. Translatable. [cite: 582, 583, 603]

description:

type: string

nullable: true

description: Optional description of the status. Translatable. [cite: 584, 604]

is\_publicly\_visible:

type: boolean

description: Indicates if content with this status is generally visible to public users. [cite: 585, 605]

sort\_order:

type: integer

description: Determines display order. [cite: 586, 587, 606]

is\_active:

type: boolean

description: True if the status definition itself is active. [cite: 588, 589, 607]

readOnly: true # Typically managed by admins

required:

- id

- code

- label

- is\_publicly\_visible

- sort\_order

- is\_active

# Module 4: Waypoints

Waypoint:

description: A geographically located point of interest (POI). [cite: 140, 143, 234]

type: object

properties:

id:

type: integer # BIGINT in DB [cite: 153]

format: int64

description: Unique identifier for the waypoint. [cite: 153, 235]

readOnly: true

name:

type: string

description: Primary human-readable name of the waypoint. Translatable. [cite: 154, 155, 236]

slug:

type: string

nullable: true

description: URL-friendly identifier. [cite: 156, 157, 237]

alternate\_names\_primary\_lang:

type: array

items:

type: string

nullable: true

description: Array of other known names in the primary reference language (English). [cite: 158, 238]

waypoint\_primary\_category\_id:

type: integer

description: ID of the primary category from `waypoint\_categories\_master`. [cite: 160, 239]

waypoint\_subcategory\_tag\_ids:

type: array

items:

type: integer

nullable: true

description: Array of tag IDs from `tags\_master`. [cite: 162, 240]

description:

type: string

nullable: true

description: General, brief description of the waypoint. Translatable. [cite: 163, 242]

geom: # GeoJSON Point representation

type: object

description: GeoJSON Point geometry (PointZ, SRID 4326 WGS84). [cite: 164, 243, 282]

properties:

type:

type: string

enum: [Point]

coordinates:

type: array

items:

type: number

format: double

minItems: 2 # longitude, latitude

maxItems: 3 # longitude, latitude, elevation

required:

- type

- coordinates

latitude:

type: number

format: double

description: Latitude coordinate (WGS 84). Generated from geom. [cite: 166, 244]

readOnly: true

longitude:

type: number

format: double

description: Longitude coordinate (WGS 84). Generated from geom. [cite: 167, 245]

readOnly: true

elevation\_meters:

type: integer

nullable: true

description: Elevation in meters. Generated from geom's Z value. [cite: 168, 169, 246]

readOnly: true

town\_id:

type: integer

nullable: true

description: ID of the associated town from `towns` table. [cite: 170, 247]

parent\_waypoint\_id:

type: integer # BIGINT in DB [cite: 171]

format: int64

nullable: true

description: ID of the parent waypoint for hierarchical waypoints. [cite: 171, 172, 248]

address\_text:

type: string

nullable: true

description: Textual street address. Translatable. [cite: 173, 249]

primary\_image\_media\_id:

type: string

format: uuid

nullable: true

description: ID of the primary representative image from `media` table. [cite: 174, 175, 250]

primary\_thumbnail\_media\_id:

type: string

format: uuid

nullable: true

description: ID of the primary thumbnail image from `media` table. [cite: 176, 251]

content\_visibility\_status\_id:

type: integer

description: ID of the content status from `content\_statuses\_master`. [cite: 177, 178, 252]

is\_seasonal:

type: boolean

description: Indicates if availability is seasonal. [cite: 179, 253]

default: false

is\_trail\_access\_point:

type: boolean

description: Flags common trail access points. [cite: 180, 254]

default: false

is\_significant\_trail\_junction:

type: boolean

description: Flags major trail junctions. [cite: 181, 255]

default: false

is\_franciscan\_highlight\_site:

type: boolean

description: Flags key Franciscan sites. [cite: 182, 256]

default: false

is\_significant\_pilgrim\_poi:

type: boolean

description: Flags POIs of major significance to pilgrims. [cite: 183, 257]

default: false

short\_narrative\_for\_dynamic\_lists:

type: string

maxLength: 250

nullable: true

description: Brief descriptive snippet. Translatable. [cite: 184, 185, 258]

waypoint\_accessibility\_notes:

type: string

nullable: true

description: Specific accessibility notes. Translatable. [cite: 186, 259]

general\_tags\_text:

type: array

items:

type: string

nullable: true

description: Array of general, free-text descriptive tags. [cite: 187, 261]

primary\_data\_source\_waypoint:

type: string

nullable: true

description: Information on the source of this waypoint's data. [cite: 188, 262]

quality\_score:

type: integer

format: int32 # smallint in DB

minimum: 0

maximum: 100

nullable: true

description: Internal score indicating data confidence/completeness (0-100). [cite: 189, 190, 263]

# Audit fields

created\_at:

type: string

format: date-time

readOnly: true

description: Timestamp of record creation. [cite: 190, 264]

created\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true # Set on creation

description: Profile ID of the user who created the record. [cite: 191, 192]

updated\_at:

type: string

format: date-time

readOnly: true

description: Timestamp of last update. [cite: 193, 265]

updated\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true # Set on update

description: Profile ID of the user who last updated the record. [cite: 194, 195]

deleted\_at:

type: string

format: date-time

nullable: true

readOnly: true

description: Timestamp for soft deletion. [cite: 196, 266]

required:

- name

- waypoint\_primary\_category\_id

- geom

- content\_visibility\_status\_id

- is\_seasonal

- is\_trail\_access\_point

- is\_significant\_trail\_junction

- is\_franciscan\_highlight\_site

- is\_significant\_pilgrim\_poi

WaypointCreate:

description: Data for creating a new waypoint.

type: object

properties:

name:

type: string

maxLength: 255

slug:

type: string

nullable: true

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

maxLength: 100

alternate\_names\_primary\_lang:

type: array

items:

type: string

nullable: true

waypoint\_primary\_category\_id:

type: integer

waypoint\_subcategory\_tag\_ids:

type: array

items:

type: integer

nullable: true

description:

type: string

nullable: true

geom:

type: object

properties:

type:

type: string

enum: [Point]

coordinates:

type: array

items:

type: number

format: double

minItems: 2

maxItems: 3

required:

- type

- coordinates

town\_id:

type: integer

nullable: true

parent\_waypoint\_id:

type: integer

format: int64

nullable: true

address\_text:

type: string

nullable: true

primary\_image\_media\_id:

type: string

format: uuid

nullable: true

primary\_thumbnail\_media\_id:

type: string

format: uuid

nullable: true

content\_visibility\_status\_id: # Default is 'draft' [cite: 177]

type: integer

is\_seasonal:

type: boolean

default: false

is\_trail\_access\_point:

type: boolean

default: false

is\_significant\_trail\_junction:

type: boolean

default: false

is\_franciscan\_highlight\_site:

type: boolean

default: false

is\_significant\_pilgrim\_poi:

type: boolean

default: false

short\_narrative\_for\_dynamic\_lists:

type: string

maxLength: 250

nullable: true

waypoint\_accessibility\_notes:

type: string

nullable: true

general\_tags\_text:

type: array

items:

type: string

nullable: true

required:

- name

- waypoint\_primary\_category\_id

- geom

- content\_visibility\_status\_id

WaypointUpdate:

description: Data for updating an existing waypoint.

allOf: # Allows all fields from WaypointCreate to be optional

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

# Placeholder Schemas for other modules (conceptual)

Trail:

type: object

properties:

id: { type: integer, readOnly: true }

name: { type: string }

description: { type: string, nullable: true }

# ... other conceptual fields based on Module 2 overview [cite: 717]

created\_at: { type: string, format: date-time, readOnly: true }

updated\_at: { type: string, format: date-time, readOnly: true }

required: [name]

TrailCreate:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

required: [name]

Region:

type: object

properties:

id: { type: integer, readOnly: true }

name: { type: string }

# ... other conceptual fields based on Module 3 overview [cite: 728]

required: [name]

Town:

type: object

properties:

id: { type: integer, readOnly: true }

name: { type: string }

# ... other conceptual fields based on Module 3 overview [cite: 728]

required: [name]

SegmentWarning:

type: object

properties:

id: { type: integer, readOnly: true }

title: { type: string }

description: { type: string }

segment\_id: { type: integer } # Assuming linkage

warning\_type\_id: { type: integer }

severity\_id: { type: integer }

# ... other conceptual fields based on Module 5 overview [cite: 749]

created\_at: { type: string, format: date-time, readOnly: true }

updated\_at: { type: string, format: date-time, readOnly: true }

required: [title, description, segment\_id, warning\_type\_id, severity\_id]

SegmentWarningCreate:

type: object

properties:

title: { type: string }

description: { type: string }

segment\_id: { type: integer }

warning\_type\_id: { type: integer }

severity\_id: { type: integer }

required: [title, description, segment\_id, warning\_type\_id, severity\_id]

UserWaypointVote: # Conceptual based on Module 6 overview [cite: 755]

type: object

properties:

waypoint\_id: { type: integer }

profile\_id: { type: string, format: uuid }

vote\_type: { type: string, enum: ["up", "down"] } # from vote\_type\_enum [cite: 756]

created\_at: { type: string, format: date-time, readOnly: true }

required: [waypoint\_id, profile\_id, vote\_type]

UserWaypointShortTip: # Conceptual based on Module 6 overview [cite: 755]

type: object

properties:

id: { type: integer, readOnly: true }

waypoint\_id: { type: integer }

profile\_id: { type: string, format: uuid }

tip\_text: { type: string }

tip\_category\_id: { type: integer, nullable: true }

moderation\_status: { type: string, readOnly: true } # from content\_moderation\_status\_enum [cite: 756]

created\_at: { type: string, format: date-time, readOnly: true }

required: [waypoint\_id, profile\_id, tip\_text]

UserWaypointShortTipCreate:

type: object

properties:

tip\_text: { type: string }

tip\_category\_id: { type: integer, nullable: true }

required: [tip\_text]

AccommodationReview: # Conceptual based on Module 6 overview, new table [cite: 686, 755]

type: object

properties:

id: { type: integer, format: "int64", readOnly: true } # bigint in DB [cite: 686]

accommodation\_waypoint\_id: { type: integer } # Assuming this links to waypoint.id of an accommodation [cite: 686]

profile\_id: { type: string, format: "uuid" } [cite: 686]

rating: { type: integer, minimum: 1, maximum: 5 }

review\_title: { type: string, nullable: true }

review\_body: { type: string }

language\_code: { type: string } [cite: 686]

moderation\_status: { type: string, readOnly: true } # from content\_moderation\_status\_enum [cite: 756]

created\_at: { type: string, format: "date-time", readOnly: true }

required: [accommodation\_waypoint\_id, profile\_id, rating, review\_body, language\_code]

AccommodationReviewCreate:

type: object

properties:

rating: { type: integer, minimum: 1, maximum: 5 }

review\_title: { type: string, nullable: true }

review\_body: { type: string }

language\_code: { type: string }

required: [rating, review\_body, language\_code]

CuratedItinerary: # Conceptual based on Module 7 overview [cite: 759]

type: object

properties:

id: { type: integer, readOnly: true }

name: { type: string }

description: { type: string, nullable: true }

# ... other fields

created\_at: { type: string, format: date-time, readOnly: true }

required: [name]

CuratedItineraryCreate:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

required: [name]

Article: # Conceptual based on Module 8 overview, new table [cite: 688, 766]

type: object

properties:

id: { type: integer, format: "int64", readOnly: true } # bigint in DB [cite: 688]

title: { type: string }

body\_content: { type: string }

author\_profile\_id: { type: string, format: "uuid" } [cite: 688]

featured\_image\_media\_id: { type: string, format: "uuid", nullable: true } [cite: 688]

# ... other fields like associated\_trail\_id, region\_id, town\_id, status\_id [cite: 688, 771]

created\_at: { type: string, format: "date-time", readOnly: true }

required: [title, body\_content, author\_profile\_id]

ArticleCreate:

type: object

properties:

title: { type: string }

body\_content: { type: string }

featured\_image\_media\_id: { type: string, format: "uuid", nullable: true }

required: [title, body\_content]

parameters:

Lang:

name: lang

in: query

description: |-

The language code for translated content (e.g., `it`, `en`).

Defaults to primary reference language (English) if not provided or translation unavailable. [cite: 4, 824]

required: false

schema:

type: string

example: "it"

Offset:

name: offset

in: query

description: Number of items to skip for pagination.

required: false

schema:

type: integer

minimum: 0

default: 0

Limit:

name: limit

in: query

description: Maximum number of items to return for pagination.

required: false

schema:

type: integer

minimum: 1

maximum: 100 # Example maximum

default: 20

SortBy:

name: sort\_by

in: query

description: Field to sort by (e.g., `name`, `created\_at`). Specific to endpoint.

required: false

schema:

type: string

Order:

name: order

in: query

description: Sort order.

required: false

schema:

type: string

enum: [asc, desc]

default: asc

Include:

name: include

in: query

description: Comma-separated list of related resources to include (e.g., `role\_details`, `avatar\_details`). Specific to endpoint. [cite: 823]

required: false

schema:

type: string

securitySchemes:

bearerAuth: # JWT Authentication [cite: 95, 107, 868]

type: http

scheme: bearer

bearerFormat: JWT

description: "Supabase JWT token. Roles are embedded in the token (e.g., in `app\_metadata.roles`) and used by RLS policies. [cite: 99, 100, 101]"

security:

- bearerAuth: [] # Applies JWT bearer auth globally by default. Public endpoints will have security overrides.

paths:

# Module 1: User & Content Infrastructure

/profiles/me:

get:

tags: [User & Content Infrastructure]

summary: Get current user's profile

description: |-

Retrieve detailed profile information for the currently authenticated user. [cite: 822]

Requires `authenticated\_user` role. [cite: 103]

operationId: getCurrentUserProfile

parameters:

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

- $ref: '#/components/parameters/Include' # e.g. include=role\_details,avatar\_details [cite: 823]

responses:

'200':

description: Successful retrieval of user profile.

content:

application/json:

schema:

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

examples:

italianProfile:

value:

id: "123e4567-e89b-12d3-a456-426614174000"

username: "anna\_pilgrim"

full\_name: "Anna Rossi"

public\_display\_name: "Anna R."

preferred\_language\_code: "it"

roles: ["pilgrim\_user"]

# ... other fields as per schema example [cite: 833]

'401':

description: Unauthorized - Authentication token is missing or invalid. [cite: 118]

content:

application/json:

schema:

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

'403':

description: Forbidden - User does not have permission. [cite: 118]

content:

application/json:

schema:

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

put:

tags: [User & Content Infrastructure]

summary: Update current user's profile

description: |-

Update mutable fields of the currently authenticated user's profile.

Requires `authenticated\_user` role. [cite: 103]

operationId: updateCurrentUserProfile

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'200':

description: Profile updated successfully.

content:

application/json:

schema:

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

'400':

description: Bad Request - Invalid input data. [cite: 117]

content:

application/json:

schema:

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

'401':

description: Unauthorized. [cite: 118]

content:

application/json:

schema:

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

'403':

description: Forbidden. [cite: 118]

content:

application/json:

schema:

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

/user\_roles:

get:

tags: [User & Content Infrastructure]

summary: List available user roles

description: |-

Retrieve a list of all active and system-defined user roles. [cite: 1253]

Accessible by `anonymous` users. [cite: 103, 1253]

operationId: listUserRoles

security: [] # Public endpoint

parameters:

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

- name: is\_role\_active # Based on user\_roles\_master schema [cite: 1183]

in: query

required: false

schema:

type: boolean

default: true

responses:

'200':

description: A list of user roles.

content:

application/json:

schema:

type: array

items:

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

/languages:

get:

tags: [User & Content Infrastructure]

summary: List available platform languages

description: |-

Fetch a list of languages supported by the platform. [cite: 825, 1400]

Accessible by `anonymous` users. [cite: 103, 1400]

operationId: listLanguages

security: [] # Public endpoint

parameters:

- name: is\_active\_for\_platform # Based on languages\_master schema [cite: 1333]

in: query

required: false

schema:

type: boolean

default: true

- $ref: '#/components/parameters/SortBy' # e.g. display\_order\_ui [cite: 827, 1340]

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

responses:

'200':

description: A list of languages.

content:

application/json:

schema:

type: array

items:

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

examples:

languagesList:

value:

- language\_code: "en"

display\_name\_native: "English"

display\_name\_en: "English"

is\_active\_for\_platform: true

display\_order\_ui: 1

- language\_code: "it"

display\_name\_native: "Italiano"

display\_name\_en: "Italian"

is\_active\_for\_platform: true

display\_order\_ui: 2 # Based on example [cite: 838]

/media:

post:

tags: [User & Content Infrastructure]

summary: Upload a new media file

description: |-

Upload a new media file (e.g., image). The request should be multipart/form-data.

The API will process the file, store it, generate variants (backend process [cite: 779, 807, 1708]), and create a metadata record.

Requires `authenticated\_user` role. [cite: 103] (Specific roles like `accommodation\_host` might be implied for certain contexts).

operationId: uploadMedia

requestBody:

required: true

content:

multipart/form-data:

schema:

type: object

properties:

file:

type: string

format: binary

description: The media file to upload.

default\_alt\_text:

type: string

nullable: true

description: Default alt text for the media.

default\_caption:

type: string

nullable: true

description: Default caption for the media.

licence:

type: string

enum: ["all\_rights\_reserved", "cc\_by", "cc0\_public\_domain"] # Simplified

nullable: true

media\_asset\_type: # Usually inferred from file, but can be specified

type: string

enum: ["image", "document\_pdf", "gpx\_file"] # Simplified

default: "image"

tags: # Comma-separated string of tags

type: string

nullable: true

responses:

'201':

description: Media uploaded and metadata record created successfully.

content:

application/json:

schema:

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

'400':

description: Bad Request - Invalid file or metadata. [cite: 117]

'401':

description: Unauthorized. [cite: 118]

'403':

description: Forbidden. [cite: 118]

/media/{media\_id}:

get:

tags: [User & Content Infrastructure]

summary: Get media item details

description: |-

Retrieve comprehensive metadata for a specific media item. [cite: 829]

Publicly accessible if media\_status is 'published\_approved'. [cite: 1698] RLS handles this.

operationId: getMediaItemDetails

security: [] # Access controlled by RLS on media table

parameters:

- name: media\_id

in: path

required: true

description: UUID of the media item. [cite: 830]

schema:

type: string

format: uuid

- $ref: '#/components/parameters/Lang' # For translatable fields like alt text, caption [cite: 830]

- name: include\_uploader\_profile

in: query

description: If true, includes basic public details of the uploader profile. [cite: 832]

schema:

type: boolean

default: false

responses:

'200':

description: Successful retrieval of media item details.

content:

application/json:

schema:

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

examples:

mediaItemExample:

value:

id: "a1b2c3d4-e5f6-7890-1234-567890abcdef"

file\_name\_original: "SanDamiano.jpg"

file\_mime\_type: "image/jpeg"

default\_alt\_text: "Chiesa di San Damiano" # Italian, assuming lang=it

image\_variants\_json: { "thumb\_s\_webp": "variants/a1b2...thumb\_s.webp" }

# ... other fields [cite: 839]

'404':

description: Not Found - Media item does not exist or user lacks permission. [cite: 119]

# Module 4: Waypoint Master Data

/waypoint\_categories:

get:

tags: [Waypoint Master Data]

summary: List active waypoint categories

description: |-

Retrieve a list of all active waypoint categories. [cite: 2]

Accessible by `anonymous` users (RLS filters for is\_active=true). [cite: 60, 428]

operationId: listWaypointCategories

security: [] # Public endpoint, RLS filters by is\_active

parameters:

- $ref: '#/components/parameters/Lang' # For translated label and description [cite: 3]

- $ref: '#/components/parameters/SortBy' # e.g., sort\_order, label [cite: 4]

- $ref: '#/components/parameters/Order' # asc, desc [cite: 5]

responses:

'200':

description: A list of active waypoint categories.

content:

application/json:

schema:

type: array

items:

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

examples:

categoriesExample: # Based on [cite: 8]

value:

- id: 1

code: "accommodation\_location"

label: "Alloggio" # Italian, assuming lang=it

description: "Luoghi di soggiorno per pellegrini."

icon\_identifier: "icon-bed"

sort\_order: 10

is\_active: true

- id: 2

code: "attraction"

label: "Attrazione"

description: "Siti di interesse, inclusi religiosi, storici, naturali."

icon\_identifier: "icon-camera"

sort\_order: 20

is\_active: true

/tags:

get:

tags: [Waypoint Master Data]

summary: List active tags

description: |-

Retrieve a list of all active tags, optionally filtered by type. [cite: 5]

Accessible by `anonymous` users (RLS filters for is\_active=true). [cite: 87, 541]

operationId: listTags

security: [] # Public endpoint, RLS filters by is\_active

parameters:

- name: tag\_type

in: query

description: Filters tags by their specified type code. [cite: 6, 476]

required: false

schema:

type: string

- $ref: '#/components/parameters/Lang' # For translated label and description [cite: 6]

- $ref: '#/components/parameters/SortBy' # e.g., sort\_order, label [cite: 7]

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

responses:

'200':

description: A list of active tags.

content:

application/json:

schema:

type: array

items:

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

/content\_statuses:

get:

tags: [Waypoint Master Data]

summary: List active content statuses

description: |-

Retrieve a list of all active content status definitions.

Accessible by `anonymous` users (RLS filters for is\_active=true). [cite: 644]

operationId: listContentStatuses

security: [] # Public endpoint, RLS filters by is\_active

parameters:

- $ref: '#/components/parameters/Lang' # For translated label and description

- $ref: '#/components/parameters/SortBy' # e.g., sort\_order [cite: 586]

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

responses:

'200':

description: A list of active content statuses.

content:

application/json:

schema:

type: array

items:

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

# Module 4: Waypoints

/waypoints:

get:

tags: [Waypoints]

summary: List waypoints

description: |-

Retrieve a list of waypoints, with filtering and pagination.

Publicly accessible for published waypoints (RLS: content\_visibility\_status\_id links to 'published' and deleted\_at IS NULL). [cite: 1892]

operationId: listWaypoints

security: [] # Access controlled by RLS

parameters:

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

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

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

- $ref: '#/components/parameters/SortBy' # e.g. name, created\_at

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

- name: waypoint\_primary\_category\_id

in: query

description: Filter by primary category ID. [cite: 160]

schema: { type: integer }

- name: town\_id

in: query

description: Filter by town ID. [cite: 170]

schema: { type: integer }

- name: tags\_include\_any # Example for array field filtering

in: query

description: Comma-separated list of tag IDs to filter by (waypoint must have at least one).

schema: { type: string }

- name: is\_seasonal

in: query

description: Filter by seasonal status. [cite: 179]

schema: { type: boolean }

# Add other relevant filter parameters based on Waypoint schema (e.g. is\_franciscan\_highlight\_site [cite: 182])

responses:

'200':

description: A paginated list of waypoints.

content:

application/json:

schema:

allOf:

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

- type: object

properties:

items:

type: array

items:

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

'400':

description: Bad Request - Invalid filter parameters.

post:

tags: [Waypoints]

summary: Create a new waypoint

description: |-

Create a new waypoint.

Requires role with content creation privileges (e.g., `regional\_content\_manager`, `admin`). [cite: 1894]

The `created\_by\_profile\_id` will be set to the authenticated user's ID.

operationId: createWaypoint

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'201':

description: Waypoint created successfully.

content:

application/json:

schema:

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

'400':

description: Bad Request - Invalid input data.

'401':

description: Unauthorized.

'403':

description: Forbidden.

/waypoints/{waypoint\_id}:

get:

tags: [Waypoints]

summary: Get waypoint by ID

description: |-

Retrieve a specific waypoint by its ID.

Publicly accessible if published (RLS). [cite: 1892]

operationId: getWaypointById

security: [] # Access controlled by RLS

parameters:

- name: waypoint\_id

in: path

required: true

description: ID of the waypoint to retrieve.

schema:

type: integer # BIGINT in DB [cite: 153]

format: int64

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

# - $ref: '#/components/parameters/Include' # e.g. include=category\_details,tags\_details,primary\_image\_details

responses:

'200':

description: Successful retrieval of waypoint details.

content:

application/json:

schema:

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

'404':

description: Not Found - Waypoint does not exist or user lacks permission.

put:

tags: [Waypoints]

summary: Update a waypoint

description: |-

Update an existing waypoint.

Requires role with content editing privileges (e.g., owner, `regional\_content\_manager`, `admin`). [cite: 1894]

The `updated\_by\_profile\_id` will be set to the authenticated user's ID.

operationId: updateWaypoint

parameters:

- name: waypoint\_id

in: path

required: true

schema:

type: integer

format: int64

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'200':

description: Waypoint updated successfully.

content:

application/json:

schema:

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

'400':

description: Bad Request.

'401':

description: Unauthorized.

'403':

description: Forbidden.

'404':

description: Not Found.

delete:

tags: [Waypoints]

summary: Delete a waypoint (soft delete)

description: |-

Soft delete a waypoint by setting its `deleted\_at` timestamp.

Requires role with content deletion privileges (e.g., owner, `regional\_content\_manager`, `admin`). [cite: 1894]

operationId: deleteWaypoint

parameters:

- name: waypoint\_id

in: path

required: true

schema:

type: integer

format: int64

responses:

'204':

description: Waypoint soft deleted successfully. [cite: 116]

'401':

description: Unauthorized.

'403':

description: Forbidden.

'404':

description: Not Found.

# Conceptual endpoints for other modules will follow a similar pattern

# Module 2: Core Trail Hierarchy

/trails:

get:

tags: [Core Trail Hierarchy]

summary: List trails (Conceptual)

description: Retrieve a list of trails. Publicly accessible for published trails.

operationId: listTrails

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of trails.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/Trail' }}}}}

post:

tags: [Core Trail Hierarchy]

summary: Create a trail (Conceptual)

description: Create a new trail. Requires admin/manager role.

operationId: createTrail

requestBody: { required: true, content: { application/json: { schema: { $ref: '#/components/schemas/TrailCreate' }}}}

responses:

'201': { description: "Trail created.", content: { application/json: { schema: { $ref: '#/components/schemas/Trail' }}}}

# Module 3: Geographical Context

/regions:

get:

tags: [Geographical Context]

summary: List regions (Conceptual)

description: Retrieve a list of geographical regions. Publicly accessible.

operationId: listRegions

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of regions.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/Region' }}}}}

/towns:

get:

tags: [Geographical Context]

summary: List towns (Conceptual)

description: Retrieve a list of towns. Publicly accessible.

operationId: listTowns

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of towns.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/Town' }}}}}

# Module 5: Dynamic Conditions

/segment\_warnings:

get:

tags: [Dynamic Conditions]

summary: List segment warnings (Conceptual)

description: Retrieve a list of segment warnings. Publicly accessible for active warnings.

operationId: listSegmentWarnings

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of segment warnings.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/SegmentWarning' }}}}}

post:

tags: [Dynamic Conditions]

summary: Create a segment warning (Conceptual)

description: Create a new segment warning. Requires regional\_content\_manager or admin role. [cite: 103]

operationId: createSegmentWarning

requestBody: { required: true, content: { application/json: { schema: { $ref: '#/components/schemas/SegmentWarningCreate' }}}}

responses:

'201': { description: "Segment warning created.", content: { application/json: { schema: { $ref: '#/components/schemas/SegmentWarning' }}}}

# Module 6: User Interaction

/waypoints/{waypoint\_id}/votes:

post:

tags: [User Interaction]

summary: Cast a vote for a waypoint (Conceptual)

description: Allows an authenticated user to cast an up or down vote for a waypoint. Requires `authenticated\_user` role. [cite: 103]

operationId: castWaypointVote

parameters:

- name: waypoint\_id

in: path

required: true

schema: { type: integer, format: int64 }

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

vote\_type: { type: string, enum: ["up", "down"] }

required: [vote\_type]

responses:

'201': { description: "Vote cast successfully.", content: { application/json: { schema: { $ref: '#/components/schemas/UserWaypointVote' }}}}

'409': { description: "User has already voted or other conflict." } # Based on [cite: 119]

/waypoints/{waypoint\_id}/tips:

get:

tags: [User Interaction]

summary: List short tips for a waypoint (Conceptual)

description: Retrieve short tips for a specific waypoint. Publicly accessible for moderated tips.

operationId: listWaypointShortTips

security: []

parameters:

- name: waypoint\_id

in: path

required: true

schema: { type: integer, format: int64 }

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

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

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

responses:

'200': { description: "A list of short tips.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/UserWaypointShortTip' }}}}}

post:

tags: [User Interaction]

summary: Add a short tip for a waypoint (Conceptual)

description: Allows an authenticated user to add a short tip for a waypoint. Requires `authenticated\_user` role. [cite: 103]

operationId: createWaypointShortTip

parameters:

- name: waypoint\_id

in: path

required: true

schema: { type: integer, format: int64 }

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'201': { description: "Tip added successfully, pending moderation.", content: { application/json: { schema: { $ref: '#/components/schemas/UserWaypointShortTip' }}}}

/accommodations/{accommodation\_waypoint\_id}/reviews:

get:

tags: [User Interaction]

summary: List reviews for an accommodation (Conceptual)

description: Retrieve reviews for a specific accommodation. Publicly accessible for moderated reviews.

operationId: listAccommodationReviews

security: []

parameters:

- name: accommodation\_waypoint\_id # Assuming accommodation is a type of waypoint [cite: 682]

in: path

required: true

schema: { type: integer, format: int64 }

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

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

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

responses:

'200': { description: "A list of reviews.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/AccommodationReview' }}}}}

post:

tags: [User Interaction]

summary: Add a review for an accommodation (Conceptual)

description: Allows an authenticated user to add a review for an accommodation. Requires `authenticated\_user` role. [cite: 103]

operationId: createAccommodationReview

parameters:

- name: accommodation\_waypoint\_id

in: path

required: true

schema: { type: integer, format: int64 }

requestBody:

required: true

content:

application/json:

schema:

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

responses:

'201': { description: "Review added successfully, pending moderation.", content: { application/json: { schema: { $ref: '#/components/schemas/AccommodationReview' }}}}

# Module 7: Curated Itineraries

/curated\_itineraries:

get:

tags: [Curated Itineraries]

summary: List curated itineraries (Conceptual)

description: Retrieve a list of curated itineraries. Publicly accessible for published ones.

operationId: listCuratedItineraries

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of curated itineraries.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/CuratedItinerary' }}}}}

post:

tags: [Curated Itineraries]

summary: Create a curated itinerary (Conceptual)

description: Create a new curated itinerary. Requires admin role.

operationId: createCuratedItinerary

requestBody: { required: true, content: { application/json: { schema: { $ref: '#/components/schemas/CuratedItineraryCreate' }}}}

responses:

'201': { description: "Itinerary created.", content: { application/json: { schema: { $ref: '#/components/schemas/CuratedItinerary' }}}}

# Module 8: Platform Content

/articles:

get:

tags: [Platform Content]

summary: List articles (Conceptual)

description: Retrieve a list of articles. Publicly accessible for published ones.

operationId: listArticles

security: []

parameters: [$ref: '#/components/parameters/Lang', $ref: '#/components/parameters/Offset', $ref: '#/components/parameters/Limit']

responses:

'200': { description: "A list of articles.", content: { application/json: { schema: { type: array, items: { $ref: '#/components/schemas/Article' }}}}}

post:

tags: [Platform Content]

summary: Create an article (Conceptual)

description: Create a new article. Requires admin/manager role. [cite: 103]

operationId: createArticle

requestBody: { required: true, content: { application/json: { schema: { $ref: '#/components/schemas/ArticleCreate' }}}}

responses:

'201': { description: "Article created.", content: { application/json: { schema: { $ref: '#/components/schemas/Article' }}}}

```

### Quick-Start Examples

1. Get Current User's Profile (with Italian translation for role details)

- `curl -X GET "https://api.viadifrancesco.example.com/v1/profiles/me?include=role\_details&lang=it" -H "Authorization: Bearer <YOUR\_JWT\_TOKEN>"`

- Sample JSON Response:

JSON

```

{

"id": "123e4567-e89b-12d3-a456-426614174000",

"username": "pellegrino\_verdi",

"full\_name": "Giuseppe Verdi",

"public\_display\_name": "G. Verdi",

"public\_avatar\_media\_id": "a1b2c3d4-e5f6-7890-1234-567890abcdef",

"public\_bio": "Appassionato della Via di Francesco, in cammino verso la pace interiore.",

"preferred\_language\_code": "it",

"preferred\_units\_of\_measure": "metric",

"pilgrim\_experience\_level": "experienced\_many\_pilgrimages",

"pilgrimage\_interests\_tags": ["spiritualità", "natura", "storia\_francescana"],

"is\_profile\_publicly\_visible": true,

"account\_status": "active",

"roles": ["pilgrim\_user", "regional\_content\_manager"],

"public\_avatar": null, // Assuming not requested or no avatar set

"role\_details": [

{

"role\_code": "pilgrim\_user",

"default\_display\_name": "Utente Pellegrino", // Translated [cite: 834]

"icon\_identifier": "person",

"is\_role\_active": true

},

{

"role\_code": "regional\_content\_manager",

"default\_display\_name": "Manager Contenuti Regionali", // Translated

"icon\_identifier": "edit\_location",

"is\_role\_active": true

}

],

"last\_activity\_at": "2025-05-18T10:30:00Z",

"created\_at": "2023-03-15T09:00:00Z",

"updated\_at": "2025-05-17T14:20:00Z",

"updated\_by\_profile\_id": null

}

```

2. List Active Waypoint Categories (default language - English)

- `curl -X GET "https://api.viadifrancesco.example.com/v1/waypoint\_categories?sort\_by=label"`

- Sample JSON Response:

JSON

```

[

{

"id": 1,

"code": "accommodation\_location",

"label": "Accommodation",

"description": "Places to stay for pilgrims.",

"icon\_identifier": "icon-bed",

"requires\_detail\_table": "accommodations",

"sort\_order": 10,

"is\_active": true

},

{

"id": 2,

"code": "attraction",

"label": "Attraction", // Default (English) [cite: 8]

"description": "Sites of interest, including religious, historical, natural.",

"icon\_identifier": "icon-camera",

"requires\_detail\_table": "attractions\_details",

"sort\_order": 20,

"is\_active": true

},

{

"id": 3,

"code": "food\_water\_source",

"label": "Food & Water Source",

"description": "Locations to find food and drinkable water.",

"icon\_identifier": "icon-restaurant",

"requires\_detail\_table": "food\_water\_sources\_details",

"sort\_order": 15,

"is\_active": true

}

]

```

3. Create a New Waypoint (as a Content Manager)

- curl -X POST "https://api.viadifrancesco.example.com/v1/waypoints?lang=it" -H "Authorization: Bearer <YOUR\_JWT\_TOKEN>" -H "Content-Type: application/json" -d '{ "name": "Fonte Sacra di San Francesco", "waypoint\_primary\_category\_id": 4, "geom": { "type": "Point", "coordinates": [12.6789, 43.1234, 550] }, "content\_visibility\_status\_id": 1, "description": "Antica fonte dove la tradizione vuole che San Francesco si sia fermato a bere.", "is\_seasonal": false, "is\_trail\_access\_point": true, "is\_significant\_trail\_junction": false, "is\_franciscan\_highlight\_site": true, "is\_significant\_pilgrim\_poi": true, "town\_id": 101 }'

(Assuming waypoint\_primary\_category\_id: 4 is "Water Source" and content\_visibility\_status\_id: 1 is "Draft", town\_id: 101 is "Assisi")

- Sample JSON Response (201 Created):

JSON

```

{

"id": 578,

"name": "Fonte Sacra di San Francesco", // Returned in Italian as per lang=it on request for translatable fields

"slug": "fonte-sacra-di-san-francesco", // System generated slug

"alternate\_names\_primary\_lang": null,

"waypoint\_primary\_category\_id": 4,

"waypoint\_subcategory\_tag\_ids": null,

"description": "Antica fonte dove la tradizione vuole che San Francesco si sia fermato a bere.",

"geom": {

"type": "Point",

"coordinates": [12.6789, 43.1234, 550]

},

"latitude": 43.1234,

"longitude": 12.6789,

"elevation\_meters": 550,

"town\_id": 101,

"parent\_waypoint\_id": null,

"address\_text": null,

"primary\_image\_media\_id": null,

"primary\_thumbnail\_media\_id": null,

"content\_visibility\_status\_id": 1,

"is\_seasonal": false,

"is\_trail\_access\_point": true,

"is\_significant\_trail\_junction": false,

"is\_franciscan\_highlight\_site": true,

"is\_significant\_pilgrim\_poi": true,

"short\_narrative\_for\_dynamic\_lists": null,

"waypoint\_accessibility\_notes": null,

"general\_tags\_text": null,

"primary\_data\_source\_waypoint": null,

"quality\_score": null,

"created\_at": "2025-05-18T18:15:00Z",

"created\_by\_profile\_id": "regional\_manager\_profile\_uuid", // Set by system

"updated\_at": "2025-05-18T18:15:00Z",

"updated\_by\_profile\_id": "regional\_manager\_profile\_uuid", // Set by system

"deleted\_at": null

}

```

### Implementation Notes

- SQL Views / Indexes:

- Localized Views: The provided `v\_waypoint\_categories\_localized` and `v\_tags\_localized` views are excellent for API performance when fetching translated master data labels and descriptions. The API should leverage these by querying the view and extracting the specific language from the `all\_translations` JSONB field based on the `lang` parameter, or use the primary language fields directly.

- Indexes on Translatable Fields in Master Tables: For `waypoint\_categories\_master` and `tags\_master`, indexes on `label` (e.g., `idx\_wc\_master\_label`, `idx\_tags\_master\_label`) are recommended if sorting by translated label (fetched from the `translations` table or the view's primary label column) becomes frequent and needs to be performed at the DB level before language-specific extraction.

- Indexes for `translations` table: The `idx\_translations\_lookup` on `(table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translation\_status)` is crucial for efficient lookups.

- Indexes for `waypoints` table:

- `idx\_waypoints\_geom` (GIST) for spatial queries.

- `idx\_waypoints\_content\_visibility\_status\_id` for filtering by publication status.

- `idx\_waypoints\_waypoint\_primary\_category\_id` for filtering by category.

- `idx\_waypoints\_town\_id` for filtering by town.

- `idx\_waypoints\_subcategory\_tag\_ids` (GIN) for filtering by tags.

- `idx\_waypoints\_name\_trgm` (GIN with `pg\_trgm`) for text search on name.

- Indexes for `profiles` table: `idx\_profiles\_roles` (GIN), `idx\_profiles\_account\_status`, `idx\_profiles\_last\_activity\_at`.

- Indexes for Audit FKs: All `created\_by\_profile\_id` and `updated\_by\_profile\_id` foreign key columns should be indexed.

- Endpoints Likely to Need Pagination, Caching, or Materialized Views:

- Pagination:

- `GET /waypoints`

- `GET /trails`, `/routes`, `/segments` (Conceptual)

- `GET /articles` (Conceptual)

- `GET /curated\_itineraries` (Conceptual)

- `GET /segment\_warnings` (Conceptual)

- `GET /waypoints/{waypoint\_id}/tips` (Conceptual)

- `GET /accommodations/{accommodation\_waypoint\_id}/reviews` (Conceptual)

- All list endpoints for master data (`/languages`, `/user\_roles`, `/waypoint\_categories`, `/tags`, `/content\_statuses`, `/regions`, `/towns`) should support pagination, although the total number of items might be small for some.

- Caching (Application-level or CDN):

- `GET /languages`

- `GET /user\_roles`

- `GET /waypoint\_categories`

- `GET /tags`

- `GET /content\_statuses`

- Responses for specific media items (`GET /media/{media\_id}`) especially if they don't change often.

- Published `GET /articles/{article\_id}` (Conceptual)

- Materialized Views (Potential DB-level optimization if views are complex and data doesn't change too frequently):

- The existing `v\_waypoint\_categories\_localized` and `v\_tags\_localized` are standard views. If performance becomes an issue due to joins with a very large `translations` table, they could be candidates for materialization with a defined refresh strategy.

- Any complex aggregation views (e.g., `accommodations\_capacity\_summary\_view` mentioned in ) might also be candidates.

- Schema Feedback:

- 🟢 Schema OK. The provided database specifications for Module 1 (User & Content Infrastructure) and Module 4 master tables (`waypoint\_categories\_master`, `tags\_master`, `content\_statuses\_master`, and `waypoints`) appear robust, well-normalized, and production-ready for V1 API needs.

- The extensive use of foreign keys, clear data types, audit columns, `is\_active` flags for master data lifecycle, and soft delete (`deleted\_at`) patterns are good practices.

- The strategy for translations using a central `translations` table and localized views is solid.

- The definition of array foreign key integrity to be enforced by database triggers (e.g., for `waypoints.waypoint\_subcategory\_tag\_ids` referencing `tags\_master` ) is crucial and must be implemented.

- RLS helper functions like `public.has\_role(TEXT)` are essential for the security model and assumed to be implemented correctly.

- The synchronization of `profiles.roles` to `auth.users.raw\_app\_meta\_data.roles` via triggers for JWT inclusion is a key aspect of the authentication architecture.