<https://gemini.google.com/u/1/app/1881f72a50cc0fbc>

### 1\. OpenAPI 3.1 Spec (for User & Content Infrastructure - Re-Clarified)

YAML

```

openapi: 3.1.0

info:

title: Via di Francesco - User & Content Infrastructure API

version: "v1.0-module1-reclarified"

description: |

API endpoints for managing users (profiles), roles, languages, media assets, and translations.

This module forms the foundational layer for user identity, permissions, internationalization (i18n), and media management.

Primary reference language (English) content is served in base fields (e.g., `default\_display\_name`).

Other language translations are provided in corresponding `[fieldname]\_translations` objects.

The field `display\_name\_en` in the LanguageMaster schema reflects its actual database column name.

servers:

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

description: Production Server

- url: http://localhost:3000/v1 # Placeholder

description: Development Server

tags:

- name: User Profiles

description: Managing user-specific data and authentication-related profile information.

- name: User Roles

description: Defining and managing user roles for role-based access control (RBAC). (Primarily Admin)

- name: Languages

description: Managing supported languages for platform internationalization. (Primarily Admin for CUD, Public Read)

- name: Media

description: Managing metadata for media assets (images, documents, etc.).

- name: Translations

description: Managing translated text content for various entities across the platform. (Primarily Admin/Translator)

components:

schemas:

# ENUMERATIONS (from DB specs for Module 1)

UnitsPreferenceEnum:

type: string

enum: [metric, imperial]

description: User's preference for units of measure.

PilgrimExperienceEnum:

type: string

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

nullable: true

description: Self-assessed pilgrim experience level.

UserAccountStatusEnum:

type: string

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

description: Current status of the user's application account.

MediaAssetTypeEnum:

type: string

enum: [image, document\_pdf, audio\_clip, video\_clip, gpx\_file, other\_file]

description: General type of media asset.

MediaLicenceEnum:

type: string

enum: [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, null]

nullable: true

description: Usage license associated with the media item.

MediaStatusEnum:

type: string

enum: [processing\_upload, pending\_review, published\_approved, rejected\_hidden, archived, error\_uploading]

description: Current workflow/visibility status of the media item.

TranslationDBStatusEnum: # From translations.translation\_status CHECK constraint

type: string

enum: [draft, needs\_review, published\_live, needs\_update, null]

nullable: true

description: Workflow status of a translation entry in the database.

# CORE SCHEMAS for Module 1

Profile:

description: Application-specific user data, extending Supabase's auth.users.

type: object

properties:

id:

type: string

format: uuid

description: User's ID from auth.users table.

readOnly: true

roles:

type: array

items:

type: string

description: Array of role codes (e.g., "pilgrim\_user"). Synchronized to auth.users.raw\_app\_meta\_data.roles.

username:

type: string

nullable: true

description: Optional unique username for community features.

full\_name:

type: string

nullable: true

description: User's full name. Stored in the primary reference language or language of submission.

public\_display\_name:

type: string

nullable: true

description: Name shown publicly. Stored in the primary reference language or language of submission. User-set or derived. Unique if provided.

public\_avatar\_media\_id:

type: string

format: uuid

nullable: true

description: ID of the media item (from media table) for the public avatar.

public\_avatar\_details: # For include=avatar\_details

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

nullable: true

readOnly: true

public\_bio:

type: string

nullable: true

description: "Short public biography, stored in the primary reference language (English) or the language of submission, as per database design[cite: 554]."

public\_bio\_translations:

type: object

additionalProperties: # language\_code

type: string # translated text for public\_bio

nullable: true

description: "Translations for the public biography, if applicable and translated via the translations table."

example:

it: "Pellegrino che esplora la Via di Francesco."

preferred\_language\_code:

type: string

default: 'en'

pattern: '^[a-z]{2}(-[A-Z]{2})?$'

description: User's preferred language (IETF tag). FK to languages\_master.

preferred\_units\_of\_measure:

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

preferred\_timezone:

type: string

nullable: true

default: 'Europe/Rome'

description: User's preferred IANA timezone string.

pilgrim\_experience\_level:

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

pilgrimage\_interests\_tags:

type: array

items:

type: string

nullable: true

description: Array of interest tags for pilgrim users.

contributor\_organization\_name:

type: string

nullable: true

contributor\_organization\_role:

type: string

nullable: true

contact\_public\_email:

type: string

format: email

nullable: true

website\_url\_profile:

type: string

format: url

nullable: true

notification\_preferences\_json:

type: object

nullable: true

additionalProperties: true

is\_profile\_publicly\_visible:

type: boolean

default: false

contribution\_score:

type: integer

default: 0

minimum: 0

readOnly: true # Typically managed by system

account\_status:

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

terms\_accepted\_at:

type: string

format: date-time

nullable: true

last\_login\_at:

type: string

format: date-time

nullable: true

readOnly: true

last\_activity\_at:

type: string

format: date-time

nullable: true

readOnly: true

created\_at:

type: string

format: date-time

readOnly: true

updated\_at:

type: string

format: date-time

readOnly: true

updated\_by\_profile\_id: # Admin use

type: string

format: uuid

nullable: true

readOnly: true

role\_details: # For include=role\_details

type: array

items:

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

readOnly: true

required:

- id

- roles

- preferred\_language\_code

- preferred\_units\_of\_measure

- is\_profile\_publicly\_visible

- account\_status

- created\_at

- updated\_at

ProfileInput:

description: Data for updating a user's own 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: { type: string, format: uuid, nullable: true }

public\_bio: { type: string, nullable: true, description: "Public biography in the primary reference language (English) or language of submission. For other specific language versions, use the main /translations endpoint." }

preferred\_language\_code: { type: string, pattern: '^[a-z]{2}(-[A-Z]{2})?$' }

preferred\_units\_of\_measure: { $ref: '#/components/schemas/UnitsPreferenceEnum' }

preferred\_timezone: { type: string, nullable: true }

pilgrim\_experience\_level: { $ref: '#/components/schemas/PilgrimExperienceEnum' }

pilgrimage\_interests\_tags: { type: array, items: { type: string }, nullable: true }

contributor\_organization\_name: { type: string, nullable: true }

contributor\_organization\_role: { type: string, nullable: true }

contact\_public\_email: { type: string, format: email, nullable: true }

website\_url\_profile: { type: string, format: url, nullable: true }

notification\_preferences\_json: { type: object, nullable: true, additionalProperties: true }

is\_profile\_publicly\_visible: { type: boolean }

terms\_accepted\_at: { type: string, format: "date-time", nullable: true }

UserRoleMaster:

description: Defines an available user role within the platform.

type: object

properties:

role\_code:

type: string

description: Short, unique, machine-readable, lowercase role code (e.g., "pilgrim\_user").

default\_display\_name: # DB column name, stores English

type: string

description: "Human-readable name in the primary reference language (English)[cite: 601]."

default\_display\_name\_translations:

type: object

additionalProperties: { type: string } # lang\_code: translated\_text

nullable: true

description: "Translations for the role's display name, from the translations table[cite: 602]."

default\_description: # DB column name, stores English

type: string

nullable: true

description: "Detailed description in the primary reference language (English)[cite: 607]."

default\_description\_translations:

type: object

additionalProperties: { type: string } # lang\_code: translated\_text

nullable: true

description: "Translations for the role's description, from the translations table[cite: 608]."

icon\_identifier:

type: string

nullable: true

description: Optional identifier for a UI icon representing the role[cite: 613].

permissions\_summary\_json:

type: object

additionalProperties: true

nullable: true

description: JSONB storing a human-readable summary of key permissions[cite: 617].

role\_hierarchy\_level:

type: integer

nullable: true

is\_system\_role:

type: boolean

default: false

is\_role\_active:

type: boolean

default: true

default\_for\_new\_pilgrim\_users:

type: boolean

default: false

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

deleted\_at:

type: string

format: date-time

nullable: true

readOnly: true

required:

- role\_code

- default\_display\_name

- is\_system\_role

- is\_role\_active

- default\_for\_new\_pilgrim\_users

- created\_at

- updated\_at

UserRoleMasterInput: # For Admin POST/PUT

description: Data for creating or updating a user role.

type: object

properties:

role\_code: # Required on POST, not on PUT (path param)

type: string

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

default\_display\_name: # This will be the English version to store in the main column

type: string

description: "Display name in the primary reference language (English)."

default\_description: # This will be the English version to store in the main column

type: string

nullable: true

description: "Description in the primary reference language (English)."

icon\_identifier:

type: string

nullable: true

permissions\_summary\_json:

type: object

additionalProperties: true

nullable: true

role\_hierarchy\_level:

type: integer

nullable: true

is\_system\_role:

type: boolean

is\_role\_active:

type: boolean

default\_for\_new\_pilgrim\_users:

type: boolean

required:

- default\_display\_name # English version

- is\_system\_role

- is\_role\_active

- default\_for\_new\_pilgrim\_users

UserRoleMasterSummary: # For embedding in Profile.role\_details

type: object

properties:

role\_code:

type: string

display\_name: # This would be translated based on the main request's 'lang' parameter

type: string

description: "Display name of the role, translated if 'lang' parameter was provided."

icon\_identifier:

type: string

nullable: true

required: [role\_code, display\_name]

LanguageMaster:

description: Defines a language supported by the platform.

type: object

properties:

language\_code:

type: string

pattern: '^[a-z]{2}(-[A-Z]{2})?$'

minLength: 2

maxLength: 5

display\_name\_native:

type: string

description: "The name of the language in its own native script and language[cite: 770]."

display\_name\_en: # This is the actual DB column name [cite: 772]

type: string

description: "The common name of the language in English[cite: 775]."

icon\_identifier:

type: string

nullable: true

description: "Optional UI icon identifier[cite: 779]."

is\_active\_for\_platform:

type: boolean

default: false

is\_primary\_content\_language:

type: boolean

default: false

display\_order\_ui:

type: integer

nullable: true

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

required:

- language\_code

- display\_name\_native

- display\_name\_en

- is\_active\_for\_platform

- is\_primary\_content\_language

- created\_at

- updated\_at

LanguageMasterInput: # For Admin POST/PUT

description: Data for creating or updating a language.

type: object

properties:

language\_code: # Required on POST

type: string

pattern: '^[a-z]{2}(-[A-Z]{2})?$'

minLength: 2

maxLength: 5

display\_name\_native:

type: string

display\_name\_en: # Actual column name

type: string

icon\_identifier:

type: string

nullable: true

is\_active\_for\_platform:

type: boolean

is\_primary\_content\_language:

type: boolean

display\_order\_ui:

type: integer

nullable: true

required:

- display\_name\_native

- display\_name\_en

- is\_active\_for\_platform

- is\_primary\_content\_language

MediaItem:

description: Metadata for a media asset.

type: object

properties:

id:

type: string

format: uuid

readOnly: true

uploader\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true # Set on creation

uploader\_profile\_summary: # For include=uploader\_summary

type: object

readOnly: true

nullable: true

properties:

id: { type: string, format: uuid }

public\_display\_name: { type: string, nullable: true }

storage\_bucket\_name:

type: string

storage\_object\_path\_original:

type: string

description: "Unique path to original object. Example: uploads/images/2025/05/{uuid}\_original.jpg"

file\_name\_original:

type: string

file\_mime\_type:

type: string

file\_size\_bytes\_original:

type: integer

format: int64

nullable: true

image\_width\_px\_original:

type: integer

nullable: true

image\_height\_px\_original:

type: integer

nullable: true

image\_variants\_json:

type: object

additionalProperties: { type: string } # variant\_name: path\_to\_variant

nullable: true

description: "Relative paths for optimized versions. Example: {\"thumb\_s\": \"variants/uuid\_thumb\_s.webp\"} [cite: 1045]"

default\_alt\_text: # DB column name, stores English

type: string

nullable: true

description: "Default alt text in the primary reference language (English)[cite: 1049]."

default\_alt\_text\_translations:

type: object

additionalProperties: { type: string }

nullable: true

description: "Translations for alt text, from the translations table."

default\_caption: # DB column name, stores English

type: string

nullable: true

description: "Default caption in the primary reference language (English)[cite: 1054]."

default\_caption\_translations:

type: object

additionalProperties: { type: string }

nullable: true

description: "Translations for caption, from the translations table."

attribution\_text: # DB column name, primary reference language

type: string

nullable: true

description: "Attribution text in the primary reference language. Potentially translatable via translations table[cite: 1058]."

attribution\_text\_translations:

type: object

additionalProperties: {type: string }

nullable: true

description: "Translations for attribution text, from the translations table."

attribution\_url:

type: string

format: url

nullable: true

licence:

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

media\_asset\_type:

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

media\_status:

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

tags:

type: array

items: { type: string }

nullable: true

checksum\_sha256\_original:

type: string

nullable: true

readOnly: true # Set by backend

dominant\_color\_hex:

type: string

pattern: '^#([0-9a-fA-F]{3}|[0-9a-fA-F]{6})$'

nullable: true

readOnly: true # Set by backend

last\_linked\_or\_used\_at:

type: string

format: date-time

nullable: true

readOnly: true

created\_at:

type: string

format: date-time

readOnly: true

updated\_at:

type: string

format: date-time

readOnly: true

updated\_by\_profile\_id:

type: string

format: uuid

nullable: true

readOnly: true

deleted\_at:

type: string

format: date-time

nullable: true

readOnly: true

required:

- id

- storage\_bucket\_name

- storage\_object\_path\_original

- file\_name\_original

- file\_mime\_type

- media\_asset\_type

- media\_status

- created\_at

- updated\_at

MediaItemMinimal: # For embedding (e.g. Profile.public\_avatar\_details)

description: Minimal media item details, typically for an image.

type: object

properties:

id: { type: string, format: uuid }

media\_asset\_type: { $ref: '#/components/schemas/MediaAssetTypeEnum' }

file\_mime\_type: { type: string }

default\_alt\_text: { type: string, nullable: true, description: "Alt text in requested language, or primary reference language (English)." } # Resolved by API

image\_variants\_json: { type: object, additionalProperties: { type: string }, nullable: true }

attribution\_text: { type: string, nullable: true, description: "Attribution text in requested language, or primary reference language (English)." } # Resolved by API

required: [id, media\_asset\_type, file\_mime\_type]

MediaItemInput: # For POST (metadata registration)

description: Data for registering new media metadata. File upload is separate.

type: object

properties:

storage\_bucket\_name: { type: string }

storage\_object\_path\_original: { type: string }

file\_name\_original: { type: string }

file\_mime\_type: { type: string }

file\_size\_bytes\_original: { type: integer, format: int64, nullable: true }

default\_alt\_text: { type: string, nullable: true, description: "Alt text in primary reference language (English)." }

default\_caption: { type: string, nullable: true, description: "Caption in primary reference language (English)." }

attribution\_text: { type: string, nullable: true, description: "Attribution text in primary reference language (English)." }

attribution\_url: { type: string, format: url, nullable: true }

licence: { $ref: '#/components/schemas/MediaLicenceEnum' }

media\_asset\_type: { $ref: '#/components/schemas/MediaAssetTypeEnum' }

tags: { type: array, items: { type: string }, nullable: true }

required:

- storage\_bucket\_name

- storage\_object\_path\_original

- file\_name\_original

- file\_mime\_type

- media\_asset\_type

MediaItemUpdateInput: # For PUT (updating existing media metadata)

description: Data for updating existing media metadata.

type: object

properties:

default\_alt\_text: { type: string, nullable: true, description: "Alt text in primary reference language (English)." }

default\_caption: { type: string, nullable: true, description: "Caption in primary reference language (English)." }

attribution\_text: { type: string, nullable: true, description: "Attribution text in primary reference language (English)." }

attribution\_url: { type: string, format: url, nullable: true }

licence: { $ref: '#/components/schemas/MediaLicenceEnum' }

media\_status: { $ref: '#/components/schemas/MediaStatusEnum' } # Moderator/Admin

tags: { type: array, items: { type: string }, nullable: true }

TranslationEntry:

description: Stores a single piece of translated text.

type: object

properties:

id:

type: integer

format: int64

readOnly: true

table\_identifier:

type: string

description: "Name of the source table (e.g., 'user\_roles\_master')[cite: 914]."

column\_identifier:

type: string

description: "Name of the column being translated (e.g., 'default\_display\_name')[cite: 918]."

row\_foreign\_key:

type: string

description: "PK value of the source record as text[cite: 922]."

language\_code:

type: string

description: "Language of this translated text. FK to languages\_master[cite: 927]."

translated\_text:

type: string

description: "The actual translated content[cite: 930]."

translation\_status:

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

translated\_by\_profile\_id:

type: string

format: uuid

nullable: true

translation\_last\_updated\_at:

type: string

format: date-time

nullable: true

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

required:

- id

- table\_identifier

- column\_identifier

- row\_foreign\_key

- language\_code

- translated\_text

- created\_at

- updated\_at

TranslationEntryInput: # For Admin/Translator POST/PUT

description: Data for creating or updating a translation entry.

type: object

properties:

table\_identifier: { type: string }

column\_identifier: { type: string }

row\_foreign\_key: { type: string }

language\_code: { type: string }

translated\_text: { type: string }

translation\_status: { $ref: '#/components/schemas/TranslationDBStatusEnum' }

required:

- table\_identifier

- column\_identifier

- row\_foreign\_key

- language\_code

- translated\_text

# UTILITY SCHEMAS

ErrorResponse: # Standard Error Object (from Auth blueprint)

type: object

properties:

code:

type: string

description: A machine-readable error code slug.

example: "resource\_not\_found"

message:

type: string

description: A human-readable summary of the error.

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

detail:

type: string

nullable: true

description: More specific details or field-specific errors.

hint:

type: string

nullable: true

description: Optional hint on how to resolve the error.

required:

- code

- message

Pagination:

type: object

properties:

total\_items:

type: integer

description: Total number of items available across all pages.

total\_pages:

type: integer

description: Total number of pages.

current\_page:

type: integer

description: The current page number (1-indexed).

page\_size:

type: integer

description: The number of items per page.

required:

- total\_items

- total\_pages

- current\_page

- page\_size

# PAGINATED RESPONSE WRAPPERS

PaginatedProfiles:

type: object

properties:

data:

type: array

items:

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

pagination:

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

required: [data, pagination]

PaginatedUserRoleMasters:

type: object

properties:

data:

type: array

items:

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

pagination:

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

required: [data, pagination]

PaginatedLanguageMasters:

type: object

properties:

data:

type: array

items:

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

pagination:

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

required: [data, pagination]

PaginatedMediaItems:

type: object

properties:

data:

type: array

items:

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

pagination:

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

required: [data, pagination]

PaginatedTranslationEntries:

type: object

properties:

data:

type: array

items:

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

pagination:

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

required: [data, pagination]

parameters:

Page:

name: page

in: query

description: Page number for pagination (1-indexed).

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: 20

minimum: 1

maximum: 100 # Max page size

Lang:

name: lang

in: query

description: |

ISO language code for requesting translated content.

If provided, the main field (e.g. `public\_bio`, `default\_display\_name`) will be returned in the requested language if a translation exists, otherwise it defaults to the primary reference language (English).

The corresponding `[fieldname]\_translations` object will always list all available translations for that field.

required: false

schema:

type: string

example: "it"

Include:

name: include

in: query

description: Comma-separated list of related resources to include (e.g., `role\_details`, `avatar\_details`, `uploader\_summary`).

required: false

schema:

type: string

SortBy:

name: sort\_by

in: query

description: Field to sort by (e.g., `created\_at`, `default\_display\_name`). Prepend '-' for descending order (e.g. `-created\_at`).

required: false

schema:

type: string

# Specific filters

IsActiveForPlatformFilter:

name: is\_active\_for\_platform

in: query

description: Filter languages by their active status.

schema:

type: boolean

MediaStatusFilter:

name: media\_status

in: query

description: Filter media items by status.

schema:

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

MediaAssetTypeFilter:

name: media\_asset\_type

in: query

description: Filter media items by asset type.

schema:

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

TranslationTableIdentifierFilter:

name: table\_identifier

in: query

description: Filter translations by source table name.

schema:

type: string

TranslationLanguageCodeFilter:

name: language\_code

in: query

description: Filter translations by language code.

schema:

type: string

responses: # Standard error responses

UnauthorizedError:

description: Authentication information is missing or invalid.

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

ForbiddenError:

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

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

NotFoundError:

description: The requested resource was not found.

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

BadRequestError:

description: The request was malformed or contained invalid parameters.

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

UnprocessableEntityError:

description: The request was well-formed but contains semantic errors (e.g., validation failure, unique constraint).

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

TooManyRequestsError:

description: Rate limit exceeded.

headers:

Retry-After: { schema: { type: integer }, description: "Seconds to wait before retrying." }

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

InternalServerError:

description: An unexpected error occurred on the server.

content:

application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } }

securitySchemes:

bearerAuth: # From global Auth & Security Blueprint

type: http

scheme: bearer

bearerFormat: JWT

description: "Uses JWT Bearer token. Roles from `public.profiles.roles` are synchronized to `auth.users.raw\_app\_meta\_data.roles` for JWT inclusion."

security: # Default security for all paths, can be overridden

- bearerAuth: []

paths:

# PROFILES

/profiles/me:

get:

tags: [User Profiles]

summary: Get current user's profile

description: Retrieve detailed profile information for the currently authenticated user.

operationId: getCurrentUserProfile

parameters:

- $ref: '#/components/parameters/Include' # e.g. role\_details,avatar\_details

- $ref: '#/components/parameters/Lang' # For role\_details display\_name, avatar alt\_text, and profile's own translatable fields

responses:

'200':

description: Successful retrieval of user profile.

content:

application/json: { schema: { $ref: '#/components/schemas/Profile' } }

'401': { $ref: '#/components/responses/UnauthorizedError' }

put: # Exemplar WRITE for Profile

tags: [User Profiles]

summary: Update current user's profile

description: Allows an authenticated user to update their own profile information.

operationId: updateCurrentUserProfile

requestBody:

required: true

content:

application/json: { schema: { $ref: '#/components/schemas/ProfileInput' } }

responses:

'200':

description: Profile updated successfully.

content:

application/json: { schema: { $ref: '#/components/schemas/Profile' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

# LANGUAGES

/languages:

get:

tags: [Languages]

summary: List available platform languages

description: Fetches a list of languages. Publicly accessible for active languages.

operationId: listLanguages

security: [] # Public read for active languages

parameters:

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

- $ref: '#/components/parameters/SortBy' # e.g. display\_order\_ui, display\_name\_en

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

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

responses:

'200':

description: A list of languages.

content:

application/json: { schema: { $ref: '#/components/schemas/PaginatedLanguageMasters' } }

post: # Exemplar WRITE for LanguageMaster (Admin)

tags: [Languages]

summary: Create a new language (Admin)

description: Adds a new language to the platform. Requires admin privileges.

operationId: createLanguage

requestBody:

required: true

content:

application/json:

schema:

allOf:

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

- type: object # Ensure language\_code is required for POST

required: [language\_code]

responses:

'201':

description: Language created successfully.

content:

application/json: { schema: { $ref: '#/components/schemas/LanguageMaster' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

/languages/{language\_code}:

get:

tags: [Languages]

summary: Get a specific language

description: Retrieves details for a specific language code. Publicly accessible.

operationId: getLanguageByCode

security: [] # Public read

parameters:

- name: language\_code

in: path

required: true

description: The language code (e.g., 'en', 'it').

schema: { type: string }

responses:

'200':

description: Language details.

content:

application/json: { schema: { $ref: '#/components/schemas/LanguageMaster' } }

'404': { $ref: '#/components/responses/NotFoundError' }

# USER ROLES (Admin)

/user\_roles:

get:

tags: [User Roles]

summary: List user roles (Admin)

description: Retrieves a list of all defined user roles. Requires admin privileges.

operationId: listUserRoles

parameters:

- $ref: '#/components/parameters/SortBy' # e.g. role\_code, default\_display\_name

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

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

responses:

'200':

description: A list of user roles.

content:

application/json: { schema: { $ref: '#/components/schemas/PaginatedUserRoleMasters' } }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

post: # Exemplar WRITE for UserRoleMaster (Admin)

tags: [User Roles]

summary: Create a new user role (Admin)

description: Adds a new user role definition. Requires admin privileges.

operationId: createUserRole

requestBody:

required: true

content:

application/json:

schema:

allOf:

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

- type: object # Ensure role\_code is required for POST

required: [role\_code]

responses:

'201':

description: User role created successfully.

content:

application/json: { schema: { $ref: '#/components/schemas/UserRoleMaster' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

/user\_roles/{role\_code}:

get:

tags: [User Roles]

summary: Get a specific user role (Admin)

description: Retrieves details for a specific user role. Requires admin privileges for most roles, some system roles might be public.

operationId: getUserRoleByCode

parameters:

- name: role\_code

in: path

required: true

description: The role code.

schema: { type: string }

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

responses:

'200':

description: User role details.

content:

application/json: { schema: { $ref: '#/components/schemas/UserRoleMaster' } }

'401': { $ref: '#/components/responses/UnauthorizedError' } # If not admin and role isn't public

'403': { $ref: '#/components/responses/ForbiddenError' }

'404': { $ref: '#/components/responses/NotFoundError' }

# MEDIA

/media:

get:

tags: [Media]

summary: List media items

description: Retrieve a list of media items. Access depends on media\_status and RLS. Public typically sees 'published\_approved'.

operationId: listMediaItems

# Security: Default bearerAuth applies for non-public. Can be security: [] if RLS handles public access to published\_approved.

parameters:

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

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

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

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

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

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

- $ref: '#/components/parameters/Include' # e.g. uploader\_summary

responses:

'200':

description: A list of media items.

content:

application/json: { schema: { $ref: '#/components/schemas/PaginatedMediaItems' } }

'400': { $ref: '#/components/responses/BadRequestError' }

post: # Exemplar WRITE for MediaItem (Authenticated User)

tags: [Media]

summary: Register new media metadata

description: Registers metadata for a new media item. Actual file upload to Storage is a separate step. `uploader\_profile\_id` is set from authenticated user.

operationId: createMediaItemMetadata

requestBody:

required: true

content:

application/json: { schema: { $ref: '#/components/schemas/MediaItemInput' } }

responses:

'201':

description: Media item metadata registered successfully.

content:

application/json: { schema: { $ref: '#/components/schemas/MediaItem' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

/media/{media\_id}:

get:

tags: [Media]

summary: Get media item details

description: Retrieve metadata for a specific media item.

operationId: getMediaItemById

parameters:

- name: media\_id

in: path

required: true

description: UUID of the media item.

schema: { type: string, format: uuid }

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

- $ref: '#/components/parameters/Include' # e.g. uploader\_summary

responses:

'200':

description: Media item details.

content:

application/json: { schema: { $ref: '#/components/schemas/MediaItem' } }

'404': { $ref: '#/components/responses/NotFoundError' }

put: # Owner, Moderator, Admin

tags: [Media]

summary: Update media item metadata

description: Allows an owner or authorized role to update media metadata.

operationId: updateMediaItemMetadata

parameters:

- name: media\_id

in: path

required: true

schema: { type: string, format: uuid }

requestBody:

required: true

content:

application/json: { schema: { $ref: '#/components/schemas/MediaItemUpdateInput' } }

responses:

'200':

description: Media item metadata updated.

content:

application/json: { schema: { $ref: '#/components/schemas/MediaItem' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

'404': { $ref: '#/components/responses/NotFoundError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

# TRANSLATIONS (Admin/Translator)

/translations:

get:

tags: [Translations]

summary: List translation entries (Admin/Translator)

description: Retrieves a list of translation entries. Requires admin/translator privileges.

operationId: listTranslationEntries

parameters:

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

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

- name: row\_foreign\_key

in: query

schema: { type: string }

- name: column\_identifier

in: query

schema: { type: string }

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

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

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

responses:

'200':

description: A list of translation entries.

content:

application/json: { schema: { $ref: '#/components/schemas/PaginatedTranslationEntries' } }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

post: # Exemplar WRITE for TranslationEntry (Admin/Translator)

tags: [Translations]

summary: Create a new translation entry (Admin/Translator)

description: Adds a new translation entry. Requires admin/translator privileges.

operationId: createTranslationEntry

requestBody:

required: true

content:

application/json: { schema: { $ref: '#/components/schemas/TranslationEntryInput' } }

responses:

'201':

description: Translation entry created.

content:

application/json: { schema: { $ref: '#/components/schemas/TranslationEntry' } }

'400': { $ref: '#/components/responses/BadRequestError' }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

'422': { $ref: '#/components/responses/UnprocessableEntityError' }

/translations/{translation\_id}:

get:

tags: [Translations]

summary: Get a specific translation entry (Admin/Translator)

description: Retrieves details for a specific translation entry. Requires admin/translator privileges.

operationId: getTranslationEntryById

parameters:

- name: translation\_id

in: path

required: true

description: ID of the translation entry.

schema: { type: integer, format: int64 }

responses:

'200':

description: Translation entry details.

content:

application/json: { schema: { $ref: '#/components/schemas/TranslationEntry' } }

'401': { $ref: '#/components/responses/UnauthorizedError' }

'403': { $ref: '#/components/responses/ForbiddenError' }

'404': { $ref: '#/components/responses/NotFoundError' }

```

### 2\. Quick-Start Examples

1. Get Current User's Profile (with Italian preference for resolved fields and avatar details):

Bash

```

curl -X GET 'https://api.viadifrancesco.example.com/v1/profiles/me?include=role\_details,avatar\_details&lang=it'\

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

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

```

\*Sample JSON Response:\*

JSON

```

{

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

"roles": ["pilgrim\_user"],

"username": "anna\_pilgrim",

"public\_display\_name": "Anna R.",

"public\_bio": "Pellegrino che esplora la Via di Francesco.", // Resolved to Italian due to lang=it

"public\_bio\_translations": { // All available translations for public\_bio

"en": "Pilgrim exploring the Via di Francesco.",

"it": "Pellegrino che esplora la Via di Francesco."

},

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

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

"account\_status": "active",

"role\_details": [

{

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

"display\_name": "Utente Pellegrino", // Resolved from user\_roles\_master default\_display\_name and its translations

"icon\_identifier": "person"

}

],

"public\_avatar\_details": {

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

"media\_asset\_type": "image",

"file\_mime\_type": "image/jpeg",

"default\_alt\_text": "Avatar di Anna R.", // Resolved from media default\_alt\_text and its translations

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

},

"created\_at": "2024-01-15T10:00:00Z",

"updated\_at": "2025-05-10T14:30:00Z"

}

```

2. List Media Items (Images, published, with Italian alt text/caption preference):

Bash

```

curl -X GET 'https://api.viadifrancesco.example.com/v1/media?media\_asset\_type=image&media\_status=published\_approved&page\_size=1&lang=it'\

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

```

\*Sample JSON Response:\*

JSON

```

{

"data": [

{

"id": "mediauuid001",

"storage\_bucket\_name": "platform-media",

"storage\_object\_path\_original": "images/trailA/image1.jpg",

"file\_name\_original": "Beautiful scenery.jpg",

"file\_mime\_type": "image/jpeg",

"image\_variants\_json": {"thumb\_s\_webp": "variants/mediauuid001\_thumb\_s.webp"},

"default\_alt\_text": "Una splendida vista delle montagne sul Sentiero A.", // Resolved to Italian

"default\_alt\_text\_translations": {

"en": "A stunning view of the mountains on Trail A.",

"it": "Una splendida vista delle montagne sul Sentiero A."

},

"default\_caption": "Alba sugli Appennini.", // Resolved to Italian

"default\_caption\_translations": {

"en": "Sunrise over the Apennines.",

"it": "Alba sugli Appennini."

},

"media\_asset\_type": "image",

"media\_status": "published\_approved",

"created\_at": "2025-03-10T08:00:00Z",

"updated\_at": "2025-03-10T09:30:00Z"

}

],

"pagination": {

"total\_items": 150,

"total\_pages": 150,

"current\_page": 1,

"page\_size": 1

}

}

```

3. Admin Updates a User Role's English Description (Primary Language Content):

This example shows updating the main default\_description (English). Translations would be managed via the /translations endpoint.

Bash

```

curl -X PUT 'https://api.viadifrancesco.example.com/v1/user\_roles/community\_guide'\

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

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

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

-d '{

"default\_display\_name": "Community Guide",

"default\_description": "A trusted pilgrim who provides local insights, helps other users, and can suggest content updates.",

"icon\_identifier": "local\_library",

"role\_hierarchy\_level": 50,

"is\_system\_role": false,

"is\_role\_active": true,

"default\_for\_new\_pilgrim\_users": false

}'

```

\*Sample JSON Response (200 OK):\*

JSON

```

{

"role\_code": "community\_guide",

"default\_display\_name": "Community Guide",

"default\_display\_name\_translations": null, // Assuming no other translations exist yet

"default\_description": "A trusted pilgrim who provides local insights, helps other users, and can suggest content updates.",

"default\_description\_translations": null,

"icon\_identifier": "local\_library",

"permissions\_summary\_json": null,

"role\_hierarchy\_level": 50,

"is\_system\_role": false,

"is\_role\_active": true,

"default\_for\_new\_pilgrim\_users": false,

"created\_at": "2025-05-18T20:15:00Z", // Original creation time

"updated\_at": "2025-05-18T21:05:00Z", // New update time

"created\_by\_profile\_id": "admin-user-uuid",

"updated\_by\_profile\_id": "admin-user-uuid-who-updated",

"deleted\_at": null

}

```

### 3\. Implementation Notes

- SQL Views/Indexes:

- The database schemas for Module 1 (versions 2.1, 2.2, 2.3) are well-indexed.

- Database Functions for Complex GETs and Translation Resolution: This remains a critical recommendation. For endpoints like `GET /profiles/me` or `GET /media/{media\_id}` that involve `include` parameters and resolving the main field content based on the `lang` parameter (while also providing the `\*\_translations` object), PostgreSQL functions are ideal.

- These functions would fetch the base entity, identify translatable fields, query `public.translations` for those fields and the specific `row\_foreign\_key`, and then construct the response. If `lang=it` is passed, the function would place the Italian text into the main field (e.g., `public\_bio`) if available, otherwise, the English content from the base table column remains. The `public\_bio\_translations` object would contain all key-value pairs of available translations.

- Schema Status:

- 🟢 Schema OK. The V2.1/V2.2/V2.3 database specifications for Module 1 align with this API model where base fields (e.g., `default\_display\_name` in `user\_roles\_master`) store the primary reference language (English), and the `\*\_translations` objects in the API response are populated from the `public.translations` table. The one exception, `languages\_master.display\_name\_en`, is correctly reflected in the API as `display\_name\_en` because that is its actual database column name.

- Translation Handling in API Backend:

- When a `lang` query parameter is provided, the API backend (or the database functions) should:

1. Retrieve the main entity. The direct columns for translatable fields (e.g., `media.default\_alt\_text`) contain the English version.

2. For each such translatable field, query the `public.translations` table to get all language versions and populate the corresponding `[fieldname]\_translations` object in the API response.

3. For the main field in the API response (e.g., `default\_alt\_text` in the `MediaItem` schema), if `lang=it` was requested and an Italian translation exists in `default\_alt\_text\_translations`, then the value of `default\_alt\_text` should be the Italian text. Otherwise, it should remain the English text from the base table.

- Write Operations (Primary Language vs. Other Translations):

- When creating/updating an entity (e.g., via `POST /user\_roles` or `PUT /media/{media\_id}`), the input schema fields like `default\_display\_name` or `default\_alt\_text` will populate the main columns in the respective tables (which store the English/primary reference language content).

- To add or update translations for \*other\* languages, clients must use the dedicated `/translations` endpoint (e.g., `POST /translations` or `PUT /translations/{id}`).