<https://gemini.google.com/u/1/app/bc262419c46fd657>

### OpenAPI 3.1 Spec (for Module 3 Geographical Context)

YAML

```

openapi: 3.1.0

info:

title: Via di Francesco - Module 3: Geographical Context API

version: v1.0.0

description: |-

API endpoints for managing and accessing geographical context data,

including regions, provinces, towns, and related taxonomies.

For READ operations (GET):

- Translatable fields in responses are resolved based on the `lang` query parameter.

- An `other\_available\_translations` object may be provided as a hint.

For WRITE operations (POST, PATCH):

- Main textual fields (e.g., `name`, `description`) should contain the content in the primary reference language (English, or as specified by a `lang` query parameter or `X-Content-Language` header on the request).

- Additional translations for other languages can be provided in an `additional\_translations` array. Each object in this array specifies the `language\_code`, `column\_identifier` (field name), and `translated\_text`.

servers:

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

description: Main API Server

tags:

- name: Regions

description: Operations related to geographical regions.

- name: Provinces

description: Operations related to administrative provinces or states.

- name: Towns

description: Operations related to towns, cities, and villages.

- name: Geographical Taxonomies

description: Access to master classification data like town types, service tags, and characteristic tags.

components:

schemas:

# Shared Schemas (Conceptually from a global definition)

ErrorResponse:

type: object

properties:

code:

type: string

description: An application-specific error code or a PostgREST error code.

example: "PGRST116"

message:

type: string

description: A human-readable summary of the error.

example: "Failed to parse query parameter 'limit'. Expected a positive integer."

details:

type: string

nullable: true

description: More specific details about the error (optional).

hint:

type: string

nullable: true

description: Suggestions for how to resolve the error (optional).

required:

- code

- message

Pagination:

type: object

properties:

current\_page:

type: integer

example: 1

page\_size:

type: integer

example: 20

total\_items:

type: integer

example: 153

total\_pages:

type: integer

example: 8

next\_page\_url:

type: string

format: uri-relative

nullable: true

example: "/towns?page=2&page\_size=20"

prev\_page\_url:

type: string

format: uri-relative

nullable: true

example: null

required:

- current\_page

- page\_size

- total\_items

- total\_pages

MediaItemSummary:

type: object

description: Summarized media item information.

properties:

id:

type: string

format: uuid

description: Unique identifier for the media item.

alt\_text:

type: string

nullable: true

description: Resolved alternative text in the requested language.

image\_variants:

type: object

properties:

thumbnail\_s:

type: string

format: uri-relative

display\_l:

type: string

format: uri-relative

additionalProperties:

type: string

format: uri-relative

example:

thumbnail\_s: "media/variants/image\_id\_thumb\_s.webp"

display\_l: "media/variants/image\_id\_display\_l.jpg"

required:

- id

# Module 3 Specific Schemas

# --- Master Data Schemas (for GET responses) ---

CharacteristicTagMaster:

type: object

description: A characteristic tag for regions.

properties:

code:

type: string

example: "mountainous"

name:

type: string

example: "Mountainous"

description:

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

example: "icon-mountain"

display\_order:

type: integer

example: 10

is\_active:

type: boolean

example: true

other\_available\_translations:

type: object

additionalProperties:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

example:

it: { name: "Montuoso" }

required:

- code

- name

- is\_active

ServiceTagMaster:

type: object

description: A service tag for towns.

properties:

code:

type: string

example: "pharmacy"

name:

type: string

example: "Pharmacy"

description:

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

example: "icon-rx"

category:

type: string

nullable: true

example: "health"

display\_order:

type: integer

example: 1

is\_active:

type: boolean

example: true

other\_available\_translations:

type: object

additionalProperties:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

example:

it: { name: "Farmacia" }

required:

- code

- name

- is\_active

TownTypeMaster:

type: object

description: A type or classification for towns.

properties:

code:

type: string

example: "village\_paese"

name:

type: string

example: "Village (Paese)"

description:

type: string

nullable: true

icon\_identifier:

type: string

nullable: true

example: "icon-village"

display\_order:

type: integer

example: 5

is\_active:

type: boolean

example: true

other\_available\_translations:

type: object

additionalProperties:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

example:

it: { name: "Villaggio (Paese)" }

required:

- code

- name

- is\_active

# --- Entity Schemas (for GET responses) ---

Region:

type: object

description: Detailed information about a geographical region.

properties:

id: { type: integer, readOnly: true, example: 1 }

slug: { type: string, example: "tuscany" }

name: { type: string, example: "Tuscany" } # Resolved

description: { type: string, nullable: true, example: "A region in central Italy..." } # Resolved

cultural\_highlights: { type: string, nullable: true } # Resolved

regional\_contact\_info: { type: string, nullable: true } # Resolved

best\_time\_to\_visit: { type: string, nullable: true } # Resolved

iso\_3166\_2\_code: { type: string, nullable: true, example: "IT-52" }

characteristics:

type: array

items: { $ref: '#/components/schemas/CharacteristicTagMaster' }

primary\_image: { $ref: '#/components/schemas/MediaItemSummary', nullable: true }

official\_tourism\_url: { type: string, format: url, nullable: true, example: "https://www.visittuscany.com" }

map\_default\_latitude: { type: number, format: double, nullable: true, example: 43.7714 }

map\_default\_longitude: { type: number, format: double, nullable: true, example: 11.2546 }

map\_default\_zoom: { type: integer, nullable: true, example: 7 }

geo\_boundary: { type: object, nullable: true, description: "GeoJSON object" }

is\_featured: { type: boolean, example: true }

content\_visibility\_status: { type: string, enum: ["draft", "pending\_review", "published", "archived", "rejected"], example: "published" }

other\_available\_translations:

type: object

additionalProperties:

type: object

properties:

name: { type: string }

description: { type: string, nullable: true }

# ... other translatable fields

example: { it: { name: "Toscana", description: "Una regione dell'Italia centrale..." } }

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

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

required: [id, slug, name, is\_featured, content\_visibility\_status]

RegionListItem:

type: object

properties:

id: { type: integer, example: 1 }

slug: { type: string, example: "tuscany" }

name: { type: string, example: "Tuscany" } # Resolved

description: { type: string, nullable: true, example: "A region in central Italy..." } # Resolved (short)

is\_featured: { type: boolean, example: true }

primary\_image\_thumbnail\_url: { type: string, format: uri-relative, nullable: true, example: "media/variants/region\_1\_thumb\_s.webp" }

required: [id, slug, name, is\_featured]

RegionListResponse:

type: object

properties:

data: { type: array, items: { $ref: '#/components/schemas/RegionListItem' } }

pagination: { $ref: '#/components/schemas/Pagination' }

required: [data, pagination]

Province:

type: object

properties:

id: { type: integer, readOnly: true, example: 25 }

region\_id: { type: integer, example: 1 }

code: { type: string, nullable: true, example: "FI" }

name: { type: string, example: "Province of Florence" } # Resolved

description: { type: string, nullable: true } # Resolved

wikidata\_id: { type: string, nullable: true, example: "Q3820" }

geonames\_id: { type: integer, nullable: true, example: 3176959 }

is\_active: { type: boolean, example: true }

other\_available\_translations:

type: object

additionalProperties: { type: object, properties: { name: { type: string }, description: { type: string, nullable: true } } }

example: { it: { name: "Provincia di Firenze" } }

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

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

required: [id, region\_id, name, is\_active]

ProvinceListItem:

type: object

properties:

id: { type: integer, example: 25 }

code: { type: string, nullable: true, example: "FI" }

name: { type: string, example: "Province of Florence" } # Resolved

region\_id: { type: integer, example: 1 }

is\_active: { type: boolean, example: true }

required: [id, name, region\_id, is\_active]

ProvinceListResponse:

type: object

properties:

data: { type: array, items: { $ref: '#/components/schemas/ProvinceListItem' } }

pagination: { $ref: '#/components/schemas/Pagination' }

required: [data, pagination]

Town:

type: object

properties:

id: { type: integer, readOnly: true, example: 123 }

slug: { type: string, example: "assisi" }

name: { type: string, example: "Assisi" } # Resolved

alternate\_names: { type: array, items: { type: string }, nullable: true, example: ["Asisium"] } # Resolved

short\_description: { type: string, nullable: true, example: "A historic and spiritual town..." } # Resolved

full\_description: { type: string, nullable: true } # Resolved

latitude\_centroid: { type: number, format: double, example: 43.0700 }

longitude\_centroid: { type: number, format: double, example: 12.6170 }

geom\_centroid: { type: object, nullable: true, description: "GeoJSON Point" }

geom\_boundary: { type: object, nullable: true, description: "GeoJSON MultiPolygon" }

elevation\_meters: { type: integer, nullable: true, example: 405 }

population: { type: integer, nullable: true, example: 28000 }

town\_type: { $ref: '#/components/schemas/TownTypeMaster', nullable: true } # Resolved

region: # Embedded summary

type: object

properties: { id: {type: integer}, slug: {type: string}, name: {type: string} }

required: [id, slug, name]

nullable: true

province: # Embedded summary

type: object

properties: { id: {type: integer}, code: {type: string, nullable: true}, name: {type: string} }

required: [id, name]

nullable: true

key\_services: { type: array, items: { $ref: '#/components/schemas/ServiceTagMaster' }, nullable: true } # Resolved

town\_transport\_information\_urls:

type: array

items:

type: object

properties:

operator\_name: { type: string, nullable: true } # Resolved

url: { type: string, format: "url" }

notes: { type: string, nullable: true } # Resolved

required: [url]

nullable: true

has\_train\_station: { type: boolean, example: true }

has\_bus\_services: { type: boolean, example: true }

primary\_image: { $ref: '#/components/schemas/MediaItemSummary', nullable: true }

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

is\_major\_stage\_town: { type: boolean, example: true }

content\_visibility\_status: { type: string, enum: ["draft", "pending\_review", "published", "archived", "rejected"], example: "published" }

data\_last\_verified\_at: { type: string, format: "date-time", nullable: true }

other\_available\_translations:

type: object

additionalProperties: { type: object, properties: { name: { type: string }, short\_description: { type: string, nullable: true } } }

example: { it: { name: "Assisi", short\_description: "Una città storica e spirituale..." } }

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

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

required: [id, slug, name, latitude\_centroid, longitude\_centroid, is\_major\_stage\_town, content\_visibility\_status]

TownListItem:

type: object

properties:

id: { type: integer, example: 123 }

slug: { type: string, example: "assisi" }

name: { type: string, example: "Assisi" } # Resolved

short\_description: { type: string, nullable: true, example: "A historic town..." } # Resolved

latitude\_centroid: { type: number, format: double, example: 43.0700 }

longitude\_centroid: { type: number, format: double, example: 12.6170 }

is\_major\_stage\_town: { type: boolean, example: true }

region\_name: { type: string, nullable: true, example: "Umbria" } # Resolved

primary\_image\_thumbnail\_url: { type: string, format: uri-relative, nullable: true, example: "media/variants/town\_123\_thumb\_s.webp" }

required: [id, slug, name, latitude\_centroid, longitude\_centroid, is\_major\_stage\_town]

TownListResponse:

type: object

properties:

data: { type: array, items: { $ref: '#/components/schemas/TownListItem' } }

pagination: { $ref: '#/components/schemas/Pagination' }

required: [data, pagination]

# --- Schemas for WRITE operations (POST/PATCH bodies) ---

AdditionalTranslationInput:

type: object

properties:

language\_code:

type: string

description: Target language code (e.g., 'it', 'de').

example: "it"

column\_identifier:

type: string

description: The field name being translated (e.g., 'name', 'description').

example: "name"

translated\_text:

type: string

description: The translated text.

example: "Testo Tradotto"

required:

- language\_code

- column\_identifier

- translated\_text

RegionCreate:

type: object

description: Data for creating a new region. Main textual fields (name, description, etc.) are for the primary reference language (e.g., English, or language specified by 'lang' param/header of the request).

properties:

slug: { type: string, description: "URL-friendly identifier. Must be unique.", example: "new-umbria-sud" }

name: { type: string, description: "Name in primary reference language.", example: "New South Umbria" }

description: { type: string, nullable: true, description: "Description in primary reference language." }

cultural\_highlights: { type: string, nullable: true, description: "Cultural highlights in primary reference language." }

regional\_contact\_info: { type: string, nullable: true, description: "Contact info in primary reference language." }

best\_time\_to\_visit: { type: string, nullable: true, description: "Best time to visit in primary reference language." }

country\_code: { type: string, default: "IT", example: "IT" }

iso\_3166\_2\_code: { type: string, nullable: true, example: "IT-XX" }

characteristics\_tags: { type: array, items: { type: string }, nullable: true, example: ["hilly\_terrain"] } # Codes

primary\_media\_id: { type: string, format: uuid, nullable: true }

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

map\_default\_latitude: { type: number, format: double, nullable: true }

map\_default\_longitude: { type: number, format: double, nullable: true }

map\_default\_zoom: { type: integer, nullable: true }

geo\_boundary: { type: object, nullable: true, description: "GeoJSON object" }

is\_featured: { type: boolean, default: false }

content\_visibility\_status: { type: string, enum: ["draft", "pending\_review", "published"], default: "draft" }

additional\_translations:

type: array

nullable: true

items: { $ref: '#/components/schemas/AdditionalTranslationInput' }

description: Provides translations for specified fields in other languages.

required: [slug, name, country\_code]

TownPatch:

type: object

description: Data for partially updating a town. Main textual fields are for the primary reference language. All fields are optional.

properties:

slug: { type: string, nullable: true }

name: { type: string, nullable: true }

description: { type: string, nullable: true }

short\_description: { type: string, nullable: true }

# ... other translatable fields like 'full\_description', 'history\_notes'

latitude\_centroid: { type: number, format: double, nullable: true }

longitude\_centroid: { type: number, format: double, nullable: true }

elevation\_meters: { type: integer, nullable: true }

population: { type: integer, nullable: true }

town\_type\_code: { type: string, nullable: true }

key\_services\_summary\_tags: { type: array, items: { type: string }, nullable: true } # Codes

primary\_media\_id: { type: string, format: uuid, nullable: true }

is\_major\_stage\_town: { type: boolean, nullable: true }

content\_visibility\_status: { type: string, enum: ["draft", "pending\_review", "published", "archived", "rejected"], nullable: true }

additional\_translations:

type: array

nullable: true

items: { $ref: '#/components/schemas/AdditionalTranslationInput' }

description: Provides translations for specified fields in other languages.

parameters:

langQueryParam:

name: lang

in: query

description: |-

The desired language code for translatable fields in responses (e.g., `it`, `en`).

For WRITE operations, this can indicate the language of the main textual fields if not English. Defaults to English.

required: false

schema: { type: string, example: "it" }

idOrSlugPath:

name: id\_or\_slug

in: path

required: true

description: The unique ID (integer) or URL-friendly slug (string) of the resource.

schema: { type: string, example: "assisi" }

pageQueryParam:

name: page

in: query

description: Page number for pagination.

required: false

schema: { type: integer, minimum: 1, default: 1 }

pageSizeQueryParam:

name: page\_size

in: query

description: Number of items per page.

required: false

schema: { type: integer, minimum: 1, maximum: 100, default: 20 }

sortByQueryParam:

name: sort\_by

in: query

description: Field to sort results by.

required: false

schema: { type: string }

sortOrderQueryParam:

name: sort\_order

in: query

description: Sort order.

required: false

schema: { type: string, enum: [asc, desc], default: asc }

securitySchemes:

bearerAuth:

type: http

scheme: bearer

bearerFormat: JWT

description: "Supabase JWT token. Example: `Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...`"

security: [{ bearerAuth: [] }] # Default security

paths:

# --- Regions ---

/regions:

get: # Needs v\_regions\_list\_localized view

tags: [Regions]

summary: List regions

operationId: listRegions

parameters:

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

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

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

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

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

- name: is\_featured

in: query

schema: { type: boolean }

required: false

responses:

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

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

security: []

post:

tags: [Regions]

summary: Create a new region

operationId: createRegion

parameters:

- $ref: '#/components/parameters/langQueryParam' # To indicate language of main fields if not 'en'

requestBody:

required: true

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

responses:

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

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

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

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

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

security: [{ bearerAuth: [] }]

/regions/{id\_or\_slug}:

get:

tags: [Regions]

summary: Get region details

operationId: getRegionDetails

parameters:

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

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

- name: include\_towns\_limit

in: query

description: Max number of towns to embed.

schema: { type: integer, minimum: 0, default: 0 }

required: false

responses:

'200': { description: "Region details.", content: { application/json: { schema: { $ref: '#/components/schemas/Region' } } } }

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

security: []

# --- Provinces ---

/provinces:

get: # May need v\_provinces\_list\_localized view for translated name sorting

tags: [Provinces]

summary: List provinces

operationId: listProvinces

parameters:

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

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

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

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

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

- name: region\_id

in: query

schema: { type: integer }

required: false

- name: is\_active

in: query

schema: { type: boolean, default: true }

required: false

responses:

'200': { description: "A list of provinces.", content: { application/json: { schema: { $ref: '#/components/schemas/ProvinceListResponse' } } } }

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

security: []

/provinces/{id}:

get:

tags: [Provinces]

summary: Get province details

operationId: getProvinceDetails

parameters:

- name: id

in: path

required: true

schema: { type: integer }

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

responses:

'200': { description: "Province details.", content: { application/json: { schema: { $ref: '#/components/schemas/Province' } } } }

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

security: []

# --- Towns ---

/towns:

get: # Needs v\_towns\_list\_localized view and FTS index on translations.translated\_text for search\_term

tags: [Towns]

summary: List towns

operationId: listTowns

parameters:

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

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

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

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

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

- name: region\_id

in: query

schema: { type: integer }

required: false

- name: province\_id

in: query

schema: { type: integer }

required: false

- name: services

in: query

schema: { type: array, items: { type: string } }

style: form

explode: true

required: false

- name: town\_type\_code

in: query

schema: { type: string }

required: false

- name: is\_major\_stage\_town

in: query

schema: { type: boolean }

required: false

- name: search\_term

in: query

schema: { type: string }

required: false

responses:

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

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

security: []

/towns/{id\_or\_slug}:

get:

tags: [Towns]

summary: Get town details

operationId: getTownDetails

parameters:

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

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

responses:

'200': { description: "Town details.", content: { application/json: { schema: { $ref: '#/components/schemas/Town' } } } }

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

security: []

patch:

tags: [Towns]

summary: Partially update a town

operationId: patchTown

parameters:

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

- $ref: '#/components/parameters/langQueryParam' # To indicate language of main fields if not 'en'

requestBody:

required: true

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

responses:

'200': { description: "Town updated.", content: { application/json: { schema: { $ref: '#/components/schemas/Town' } } } }

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

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

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

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

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

security: [{ bearerAuth: [] }]

# --- Taxonomies ---

/characteristic-tags:

get:

tags: [Geographical Taxonomies]

summary: List characteristic tags

operationId: listCharacteristicTags

parameters:

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

- name: is\_active {in: query, schema: {type: boolean, default: true}, required: false}

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

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

responses:

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

security: []

/service-tags:

get:

tags: [Geographical Taxonomies]

summary: List service tags

operationId: listServiceTags

parameters:

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

- name: is\_active {in: query, schema: {type: boolean, default: true}, required: false}

- name: category {in: query, schema: {type: string}, required: false}

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

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

responses:

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

security: []

/town-types:

get:

tags: [Geographical Taxonomies]

summary: List town types

operationId: listTownTypes

parameters:

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

- name: is\_active {in: query, schema: {type: boolean, default: true}, required: false}

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

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

responses:

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

security: []

# Shared responses

responses:

BadRequest: { description: "Bad Request.", content: { application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } } } }

Unauthorized: { description: "Unauthorized.", content: { application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } } } }

Forbidden: { description: "Forbidden.", content: { application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } } } }

NotFound: { description: "Not Found.", content: { application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } } } }

UnprocessableEntity: { description: "Unprocessable Entity.", content: { application/json: { schema: { $ref: '#/components/schemas/ErrorResponse' } } } }

```

### Quick-Start Examples

1\. List the first page of towns in Italian, filtered by `is\_major\_stage\_town=true`:

Bash

```

# Replace YOUR\_API\_BASE\_URL and YOUR\_SUPABASE\_ANON\_KEY

curl -X GET "YOUR\_API\_BASE\_URL/towns?lang=it&is\_major\_stage\_town=true&page\_size=5"

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

```

Sample JSON Response:

JSON

```

{

"data": [

{

"id": 123,

"slug": "assisi",

"name": "Assisi",

"short\_description": "Una città storica e spirituale in Umbria, città natale di San Francesco.",

"latitude\_centroid": 43.0700,

"longitude\_centroid": 12.6170,

"is\_major\_stage\_town": true,

"region\_name": "Umbria",

"primary\_image\_thumbnail\_url": "media/variants/town\_123\_thumb\_s.webp"

}

],

"pagination": {

"current\_page": 1,

"page\_size": 5,

"total\_items": 12,

"total\_pages": 3,

"next\_page\_url": "/towns?lang=it&is\_major\_stage\_town=true&page\_size=5&page=2",

"prev\_page\_url": null

}

}

```

2\. Get details for the region with slug "umbria" in English:

Bash

```

curl -X GET "YOUR\_API\_BASE\_URL/regions/umbria?lang=en"

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

```

Sample JSON Response:

JSON

```

{

"id": 5,

"slug": "umbria",

"name": "Umbria",

"description": "The green heart of Italy, rich in history and nature.",

"iso\_3166\_2\_code": "IT-UM",

"characteristics": [

{

"code": "hilly\_terrain",

"name": "Hilly Terrain",

"icon\_identifier": "icon-hills",

"is\_active": true,

"display\_order": 1

}

],

"primary\_image": {

"id": "media-uuid-for-umbria-primary",

"alt\_text": "Rolling hills of Umbria",

"image\_variants": {"thumbnail\_s": "media/variants/umbria\_thumb.webp"}

},

"official\_tourism\_url": "https://www.umbriatourism.it/",

"is\_featured": true,

"content\_visibility\_status": "published",

"created\_at": "2023-01-10T10:00:00Z",

"updated\_at": "2023-01-10T10:00:00Z"

}

```

3\. Create a new region (requires authentication and admin role), specifying English as the primary language for main fields and providing an Italian translation:

Bash

```

# Replace YOUR\_JWT\_TOKEN with a valid JWT for an admin user

# Main fields are in English (default or via ?lang=en)

curl -X POST "YOUR\_API\_BASE\_URL/regions"

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

-H "Authorization: Bearer YOUR\_JWT\_TOKEN"

-H "Content-Type: application/json"

-d '{

"slug": "lazio-nord-new",

"name": "Northern Lazio New",

"description": "The northern part of the Lazio region, newly defined.",

"country\_code": "IT",

"is\_featured": false,

"content\_visibility\_status": "draft",

"additional\_translations": [

{

"language\_code": "it",

"column\_identifier": "name",

"translated\_text": "Lazio Nord Nuovo"

},

{

"language\_code": "it",

"column\_identifier": "description",

"translated\_text": "La parte settentrionale della regione Lazio, recentemente definita."

}

]

}'

```

Sample JSON Response (201 Created, assuming `lang=en` for the response):

JSON

```

{

"id": 16,

"slug": "lazio-nord-new",

"name": "Northern Lazio New",

"description": "The northern part of the Lazio region, newly defined.",

"iso\_3166\_2\_code": null,

"characteristics": null,

"primary\_image": null,

"official\_tourism\_url": null,

"is\_featured": false,

"content\_visibility\_status": "draft",

"other\_available\_translations": {

"it": {"name": "Lazio Nord Nuovo", "description": "La parte settentrionale della regione Lazio, recentemente definita."}

},

"created\_at": "2025-05-18T20:00:00Z",

"updated\_at": "2025-05-18T20:00:00Z"

}

```

### Implementation Notes

- SQL Views:

- `public.v\_towns\_list\_localized`: Essential for `GET /towns`.

- `public.v\_regions\_list\_localized`: Essential for `GET /regions`.

- Consider `public.v\_provinces\_list\_localized` if complex localized filtering/sorting is needed for `GET /provinces`.

- SQL Indexes:

- `public.translations` Table: Critically important. Composite index on `(table\_identifier, column\_identifier, row\_foreign\_key, language\_code, translation\_status)` for lookups. Full-Text Search (FTS) GIN index on a `tsvector` column from `translated\_text` for `search\_term` functionality.

- Array/JSONB Columns: GIN indexes on `regions.characteristics\_tags` and `towns.key\_services\_summary\_tags`. Also, `towns.town\_transport\_information\_urls` if queried.

- Geospatial Columns: GIST indexes on `regions.geo\_boundary`, `towns.geom\_centroid`, `towns.geom\_boundary`.

- Standard indexes on Foreign Keys, `slug`, status flags (e.g., `is\_featured`, `content\_visibility\_status` ), and `is\_active` on master tables are necessary.

- Materialized Views:

- If performance of `v\_towns\_list\_localized` or `v\_regions\_list\_localized` becomes an issue, consider converting them to materialized views.

- Schema Assessment:

- 🟢 Schema OK: The Module 3 database schemas (V2.1+ specs for `regions`, `provinces`, `towns`, `characteristic\_tags\_master`, `service\_tags\_master`, `town\_types\_master`) are well-suited.

- Other Considerations:

- Database Functions for Detail Endpoints: For complex `GET` detail endpoints, consider PostgreSQL functions returning JSON to encapsulate data aggregation and translation logic.

- Write Operations & Translations: The backend for `POST`/`PATCH` operations must correctly interpret the main fields as the primary reference language text (e.g., English, or language from `lang` param/header) and process the `additional\_translations` array to create corresponding entries in `public.translations`.

- Transaction Management: Write operations that affect both an entity table and the `translations` table should be performed within a single transaction to ensure data consistency.