<https://gemini.google.com/u/1/app/440838222ac514d7>

API Specification Conceptualization - User & Content Infrastructure Module

Version: 1.0

Date: 2025-05-17

This document outlines key conceptual API endpoints, example responses, database support analysis, and potential schema tweaks for the User & Content Infrastructure Module. This module includes the `profiles`, `user\_roles\_master`, `languages\_master`, `media`, and `translations` tables.

### 1. Key Conceptual API Endpoints

1. \*\*Endpoint: Get Current User's Profile\*\*

\* \*\*Purpose\*\*: Retrieve detailed profile information for the currently authenticated user to personalize their experience and manage settings.

\* \*\*Path Pattern\*\*: `GET /profiles/me`

\* \*\*Query Params / Filters / Options\*\*:

\* `include` (optional, string, comma-separated): e.g., `role\_details` (to expand `role\_codes` with their names/descriptions from `user\_roles\_master`, including translations if a `lang` param is also present), `avatar\_details` (to get the full media object for `public\_avatar\_media\_id`).

\* `lang` (optional, string, e.g., `it`): ISO language code. If `include=role\_details` is used, role names/descriptions would be returned in this language if available, otherwise in the primary reference language.

2. \*\*Endpoint: List Available Platform Languages\*\*

\* \*\*Purpose\*\*: Fetch a list of languages actively supported by the platform, typically for populating UI language selectors.

\* \*\*Path Pattern\*\*: `GET /languages`

\* \*\*Query Params / Filters / Options\*\*:

\* `is\_active\_for\_platform` (optional, boolean, default `true`): Filters languages based on their active status for platform use.

\* `sort\_by` (optional, string, default `display\_order\_ui`): Field to sort by (e.g., `display\_order\_ui`, `display\_name\_en`).

\* `order` (optional, string, default `asc`): Sort order (`asc` or `desc`).

3. \*\*Endpoint: Get Media Item Details\*\*

\* \*\*Purpose\*\*: Retrieve comprehensive metadata for a specific media item, including its variants and translated text for fields like alt text and caption.

\* \*\*Path Pattern\*\*: `GET /media/{media\_id}` (where `{media\_id}` is the UUID of the media item)

\* \*\*Query Params / Filters / Options\*\*:

\* `lang` (optional, string, e.g., `it`): The desired language code for translatable fields (`default\_alt\_text`, `default\_caption`). If not provided, or if a translation doesn't exist for the requested language, the API should return the text from the primary reference language.

\* `include\_uploader\_profile` (optional, boolean, default `false`): If true, includes basic public details of the uploader profile.

---

### 2. Example JSON Responses

1. \*\*`GET /profiles/me?include=role\_details,avatar\_details&lang=it`\*\*

```json

{

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

"username": "anna\_pilgrim",

"full\_name": "Anna Rossi",

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

"public\_bio": "Pilgrim exploring the Via di Francesco. Love history and nature!",

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

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

"pilgrim\_experience\_level": "intermediate\_few\_pilgrimages",

"pilgrimage\_interests\_tags": ["history", "nature", "spirituality"],

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

"account\_status": "active",

"roles": [

{

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

"default\_display\_name": "Utente Pellegrino",

"icon\_identifier": "person",

"translations": [

{"lang": "en", "default\_display\_name": "Pilgrim User"}

]

},

{

"role\_code": "content\_moderator",

"default\_display\_name": "Moderatore Contenuti",

"icon\_identifier": "gavel",

"translations": [

{"lang": "en", "default\_display\_name": "Content Moderator"}

]

}

],

"public\_avatar": {

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

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

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

"default\_alt\_text": "Avatar di Anna R.",

"translations": [

{"lang": "en", "default\_alt\_text": "Anna R. avatar"}

],

"image\_variants\_json": {

"thumb\_s\_webp": "variants/a1b2c3d4\_thumb\_s.webp",

"display\_m\_jpeg": "variants/a1b2c3d4\_display\_m.jpg"

},

"attribution\_text": null

},

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

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

"updated\_by\_profile\_id": null

}

```

2. \*\*`GET /languages?is\_active\_for\_platform=true&sort\_by=display\_order\_ui`\*\*

```json

[

{

"language\_code": "en",

"display\_name\_native": "English",

"display\_name\_en": "English",

"icon\_identifier": "flag-icon-gb",

"is\_primary\_content\_language": true,

"display\_order\_ui": 1

},

{

"language\_code": "it",

"display\_name\_native": "Italiano",

"display\_name\_en": "Italian",

"icon\_identifier": "flag-icon-it",

"is\_primary\_content\_language": false,

"display\_order\_ui": 2

}

]

```

3. \*\*`GET /media/a1b2c3d4-e5f6-7890-1234-567890abcdef?lang=it&include\_uploader\_profile=true`\*\*

```json

{

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

"uploader\_profile": {

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

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

},

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

"storage\_object\_path\_original": "originals/2024/avatars/anna\_r\_original.jpg",

"file\_name\_original": "anna\_avatar.jpg",

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

"file\_size\_bytes\_original": 102400,

"image\_width\_px\_original": 800,

"image\_height\_px\_original": 800,

"image\_variants\_json": {

"thumb\_s\_webp": "variants/a1b2c3d4\_thumb\_s.webp",

"display\_m\_jpeg": "variants/a1b2c3d4\_display\_m.jpg"

},

"default\_alt\_text": "Avatar di Anna R.",

"default\_caption": "Il mio nuovo avatar!",

"translations": [

{"lang": "en", "default\_alt\_text": "Anna R. avatar", "default\_caption": "My new avatar!"}

],

"attribution\_text": "Foto scattata da Marco",

"attribution\_url": null,

"licence": "cc\_by",

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

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

"tags": ["avatar", "profile"],

"created\_at": "2024-01-15T10:05:00Z",

"updated\_at": "2024-01-15T11:00:00Z",

"updated\_by\_profile\_id": "system\_moderator\_uuid"

}

```

---

### 3. Database-Support Analysis

1. \*\*Endpoint: `GET /profiles/me`\*\*

\* \*\*Indexes\*\*: Primary key on `profiles.id` is sufficient for the main lookup. Indexes on `user\_roles\_master.role\_code` (PK) and `media.id` (PK) support includes. `idx\_translations\_lookup` aids role translations.

\* \*\*Join Complexity\*\*: Basic profile data is simple. Including `role\_details` with translations requires multiple lookups (profile -> roles array -> user\_roles\_master -> translations).

\* \*\*Recommendation\*\*: A \*\*database function\*\* (e.g., `get\_user\_profile\_details(profile\_id uuid, request\_lang\_code text)`) is highly recommended to encapsulate this logic and return a complete JSON object, optimizing data retrieval.

\* \*\*Performance Gotchas\*\*: RLS policies on `profiles`, `user\_roles\_master`, and `translations` are generally efficient. The main complexity is efficiently assembling the nested `role\_details` with translations.

\* \*\*Missing Data?\*\*: No obvious missing data for core profile needs.

2. \*\*Endpoint: `GET /languages`\*\*

\* \*\*Indexes\*\*: `idx\_languages\_master\_active\_order` (on `is\_active\_for\_platform, display\_order\_ui`) is ideal for common filtering and sorting.

\* \*\*Join Complexity\*\*: Single table query; no joins required.

\* \*\*Performance Gotchas\*\*: Minimal; table is small and RLS is simple.

\* \*\*Missing Data?\*\*: No.

3. \*\*Endpoint: `GET /media/{media\_id}`\*\*

\* \*\*Indexes\*\*: `media.id` (PK) for the main lookup. `idx\_translations\_lookup` for alt text/caption translations. `idx\_media\_uploader\_profile\_id` for optional uploader profile join.

\* \*\*Join Complexity\*\*: Fetching the base media item is simple. Handling translations for `default\_alt\_text` and `default\_caption` involves a join or lookup to `translations`. Optional join to `profiles` for uploader info.

\* \*\*Recommendation\*\*: Similar to profiles, a \*\*database function\*\* (e.g., `get\_media\_details(target\_media\_id uuid, request\_lang\_code text, include\_uploader boolean)`) could streamline fetching the media item with its translations and optional uploader data.

\* \*\*Performance Gotchas\*\*: RLS on `media` (checking status and `deleted\_at`) is efficient with existing indexes.

\* \*\*Missing Data?\*\*: No obvious missing fields.

---

### 4. Immediate Schema Tweaks (if any)

The User & Content Infrastructure module's tables, after re-evaluation against Checklist V2-05-17-B, are largely robust and well-prepared to support these conceptual API endpoints. The schema tweaks identified during those re-evaluations (addition of audit columns like `updated\_by\_profile\_id`, `icon\_identifier` fields, and ensuring translation cleanup mechanisms are in place) have already addressed many potential API needs.

No \*new critical (🔴 Must-fix)\* schema adjustments are identified solely from this API conceptualization exercise that were not already covered in the table spec updates (Versions 2.1/2.2).

\* 🟢 \*\*Optional future/Optimization\*\*:

\* \*\*Database Functions for Complex Objects\*\*: Strongly consider creating PostgreSQL functions callable via RPC (e.g., for Supabase) to assemble complex JSON responses like the user profile with full role details (including translations) or media items with their translations. This shifts complexity to the database, potentially improving performance and simplifying the API gateway/backend-for-frontend logic.

\* 🟠 \*\*Full-Text Search (FTS)\*\*: For enhanced search capabilities over fields like `media.default\_alt\_text`, `media.default\_caption`, `media.tags`, `profiles.public\_bio`, etc., implementing FTS (`tsvector` columns and GIN indexes) would be a valuable V2+ improvement. This was noted as an improvement during individual table reviews.

The current schema provides a solid foundation. The primary focus for efficient API delivery will be on crafting performant queries or database functions for complex data aggregation, especially involving translations.