Module Data Schema Reference

Overview

This document defines the Firestore data schema for module-related content.
It is the source of truth for anyone reading from or writing to module collections.

Collections covered:

- `subjects`
- `modules`
- `questions`
- `dialogues`

Cloud Storage (GCS) folder structure is also described for the upload pipeline (Sprint 4+).

Schema change policy:
Any schema change should be reviewed with the module/data owners before implementation, because it affects retrieval behavior and quiz scoring.

Subjects

Collection: `/subjects/{subject_id}`

One document per supported subject.

Field	Type	Description
`subject_id`	`string`	Unique identifier. One of: `math`, `english`, `science`, `custom`
`name`	`string`	Display name (for example, "Mathematics")
`description`	`string`	Short description of the subject
`icon_url`	`string`	GCS path to the subject icon for app UI
`created_at`	`timestamp`	Document creation timestamp

Example document (`subjects/math`):

```
{
  "subject_id": "math",
  "name": "Mathematics",
  "description": "Elementary math concepts for K-6 students.",
  "icon_url": "gs://your-project-bucket/icons/math.png",
  "created_at": "2026-02-23T00:00:00Z"
}
```

Supported subject IDs:

subject_id	Name	Notes
math	Mathematics	Sourced from grade-school-math, math_elem
english	English / Reading	Sourced from english_words
science	Science	Hand-authored for now
custom	Custom	Generated via supervisor upload pipeline (Sprint 4+)

Modules

Collection: /modules/{module_id}

The core unit of content. One module represents one topic at one grade level.

Field	Type	Description
module_id	string	Unique ID. Format: {subject}_grade{N}_{topic} for manual, custom_{uuid} for pipeline-generated
subject_id	string	Reference to /subjects/{subject_id}
title	string	Display title shown in app
description	string	Short summary of what the module covers
grade_level	integer	Grade level. 0 = Kindergarten, 1-12 = Grades 1-12
standard_tags	array<string>	Curriculum standard codes (for example, Common Core tags)
session_mode	string	One of: teach_then_quiz, dialogue, both
instructional_content	map	Teaching content used during lesson phase
prerequisites	array<string>	List of module IDs that should be completed first
source	string	manual or pipeline
created_at	timestamp	Document creation timestamp

instructional_content sub-fields:

Sub-field	Type	Description
text	string	Main explanatory text Reachy reads/uses
example_walkthrough	array<map>	Worked examples. Each item has problem (string), steps (array)

Example document (modules/math_grade2_addition):

```
{
  "module_id": "math_grade2_addition",
  "subject_id": "math",
  "title": "Addition with Carrying",
  "description": "Learn how to add two-digit numbers with carrying.",
  "grade_level": 2,
  "standard_tags": ["CCSS.MATH.CONTENT.2.NBT.B.5"],
  "session_mode": "teach_then_quiz",
  "instructional_content": {
    "text": "When the sum of digits in a column is 10 or more, we carry over to
the next column.",
    "example_walkthrough": [
      {
        "problem": "27 + 35",
        "steps": [
          "Add the ones: 7 + 5 = 12. Write 2, carry the 1.",
          "Add the tens: 2 + 3 + 1 = 6. Write 6.",
          "Answer: 62"
        ]
      }
    ]
  },
  "prerequisites": ["math_grade1_addition"],
  "source": "manual",
  "created_at": "2026-02-23T00:00:00Z"
}
```

Questions

Collection: `/questions/{question_id}`

Questions are decoupled from modules so they can be queried and reused independently.
Each question references its parent module by `module_id`.

Field	Type	Description
`question_id`	`string`	Unique ID. Format: `q_{subject}_grade{N}_{topic}_{NNN}`
`module_id`	`string`	Reference to `/modules/{module_id}`
`subject_id`	`string`	Denormalized for query efficiency
`grade_level`	`integer`	Denormalized from parent module
`type`	`string`	One of: `example`, `instructional`, `quiz`
`prompt`	`string`	Question text shown/spoken to student
`answer`	`string`	Correct answer
`explanation`	`string`	Full explanation of answer

Field	Type	Description
`hints`	`array<string>`	Optional hints Reachy can provide
`media_url`	`string or null`	Optional GCS path to image/audio
`source`	`string`	Dataset or source origin (for example, `grade-school-math`, `manual`)
`created_at`	`timestamp`	Document creation timestamp

Question type behavior:

`type`	Used During	Explanation Shown	Score Recorded
`example`	Teaching phase	Yes, full walkthrough	No
`instructional`	Mid-lesson check	Yes, after student answers	No
`quiz`	End-of-session quiz	After quiz completion	Yes

Quiz note:
Store quiz performance against `question_id` and `module_id`.
Use `type == "quiz"` when fetching scored quiz items.

Example document (`questions/q_math_grade2_add_001`):

```
{
  "question_id": "q_math_grade2_add_001",
  "module_id": "math_grade2_addition",
  "subject_id": "math",
  "grade_level": 2,
  "type": "quiz",
  "prompt": "What is 14 + 28?",
  "answer": "42",
  "explanation": "Add the ones: 4 + 8 = 12, write 2 carry 1. Add the tens: 1 +
2 + 1 = 4. Answer: 42.",
  "hints": ["Start with the ones column.", "Do you need to carry anything?"],
  "media_url": null,
  "source": "grade-school-math",
  "created_at": "2026-02-23T00:00:00Z"
}
```

Dialogues

Collection: `/dialogues/{dialogue_id}`

Pre-scripted teaching exchanges sourced from `Education-Dialogue-Dataset` and `babi_qa`.
Used during `dialogue` or `both` session modes.

Field	Type	Description
dialogue_id	string	Unique ID. Format: dlg_{subject}_grade{N}_{topic}_{NNN}
module_id	string	Reference to /modules/{module_id}
subject_id	string	Denormalized for query efficiency
grade_level	integer	Denormalized from parent module
turns	array<map>	Ordered list of conversation turns
source	string	Dataset origin (for example, Education-Dialogue-Dataset, babi_qa)
created_at	timestamp	Document creation timestamp

turns sub-fields:

Sub-field	Type	Description
speaker	string	reachy or student
text	string or null	Spoken text (typically present for reachy)
expected_intent	string or null	Expected student intent (for example, provide_answer, ask_for_hint)

Example document (dialogues/dlg_math_grade2_add_001):

```
{
  "dialogue_id": "dlg_math_grade2_add_001",
  "module_id": "math_grade2_addition",
  "subject_id": "math",
  "grade_level": 2,
  "turns": [
    {
      "speaker": "reachy",
      "text": "Let's talk about adding big numbers. If you have 13 apples and I give you 19 more, how many do you have?",
      "expected_intent": null
    },
    {
      "speaker": "student",
      "text": null,
      "expected_intent": "provide_answer"
    },
    {
      "speaker": "reachy",
      "text": "That's right. We add the ones first: 3 + 9 = 12. We write 2 and carry 1. Then the tens: 1 + 1 + 1 = 3. So the answer is 32.",
      "expected_intent": null
    }
  ],
  "source": "Education-Dialogue-Dataset",
  "created_at": "2026-02-23T00:00:00Z"
}
```

Common Firestore Queries

```
# All modules for a given subject and grade
db.collection("modules") \
  .where("subject_id", "==", "math") \
  .where("grade_level", "==", 2)

# All quiz questions for a module
db.collection("questions") \
  .where("module_id", "==", "math_grade2_addition") \
  .where("type", "==", "quiz")

# All example questions for a module
db.collection("questions") \
  .where("module_id", "==", "math_grade2_addition") \
  .where("type", "==", "example")

# All dialogues for a module
db.collection("dialogues") \
  .where("module_id", "==", "math_grade2_addition")

# All custom modules (pipeline-generated)
db.collection("modules") \
  .where("source", "==", "pipeline")
```

Important:
Compound queries with multiple `.where(...)` clauses require composite indexes in Firestore. Create them in Firebase console (or `firestore.indexes.json`) before running these queries in production.

Cloud Storage (GCS) Structure

```
gs://your-project-bucket/
  raw-uploads/              # Supervisor-uploaded files (Sprint 4 trigger)
    custom_{uuid}/
      original_file.pdf
      metadata.json
  processed/                # Parsed output after pipeline runs
    custom_{uuid}/
```

```
        chunks.json
        embeddings.json
    module-media/            # Images/audio tied to modules
      math/
      english/
      science/
      custom/
    icons/                   # Subject icons for app UI
      math.png
      english.png
      science.png
      custom.png
```

The `custom_{uuid}` prefix is shared between GCS and Firestore (`modules/custom_{uuid}`), so the pipeline can map processed files back to the correct module document.

Dataset to Collection Mapping

Dataset	**Target**	**Notes**
`grade-school-math`	`questions`	Word problems, usually `example` or `quiz`
`math_elem`	`questions`	TBD - pending final dataset mapping review for `type` assignment
`english_words`	`questions`	Vocabulary-style items, usually `example` or `instructional`
`Education-Dialogue-Dataset`	`dialogues`	Teaching exchanges to `turns`
`babi_qa`	`dialogues`	Reasoning QA converted to `turns`
`standards-data`	`modules`	Not a standalone collection target; populates `standard_tags` on module documents