

API Quick Reference Card

Base URL: <http://localhost:5200/api>

Page 1: Dashboard

Get Statistics

```
// Get all exams
GET /api/hierarchy/exams

// Get question count
GET /api/hierarchy/{level}/{id}/question-count
// level: exam, subject, chapter, topic

// Get detailed stats
GET /api/hierarchy/{level}/{id}/stats
```

Page 2: Hierarchical Creation

Create Hierarchy

```
// Create Exam
POST /api/exams
{ "name": "JEE Main", "exam_type": "competitive" }

// Create Subject
POST /api/subjects
{ "exam_id": "...", "name": "Physics" }

// Create Chapter
POST /api/chapters
{ "subject_id": "...", "name": "Mechanics" }

// Create Topic
POST /api/topics
{ "chapter_id": "...", "name": "Newton's Laws" }

// Create Concept (optional)
POST /api/concepts
{ "topic_id": "...", "name": "First Law" }
```

Get Hierarchy Tree

```
GET /api/hierarchy/tree
GET /api/hierarchy/subjects?exam_id={id}
GET /api/hierarchy/chapters?subject_id={id}
GET /api/hierarchy/topics?chapter_id={id}
GET /api/hierarchy/concepts?topic_id={id}
```

Page 3: Question Upload

Single Question with Attributes

```
POST /api/questions/create-with-attributes
{
  "question": {
    "content": "Question text",
    "options": { "A": "...", "B": "...", "C": "...", "D": "..." },
    "correct_answer": "A",
    "topic_id": "...",
    "difficulty": 0.5,
    "discrimination": 1.2,
    "guessing": 0.25
  },
  "selected_attributes": [
    { "attribute_id": "...", "value": true }
  ]
}
```

Batch Upload

```
POST /api/questions/batch
{
  "questions": [
    { "content": "...", "options": {...}, "correct_answer": "A", ... },
    { "content": "...", "options": {...}, "correct_answer": "B", ... }
  ]
}
```

Image Upload

```
// Question image
POST /api/questions/{id}/image
FormData: { image: File }

// Option images
POST /api/questions/{id}/options/images
FormData: { option_A: File, option_B: File, ... }
```

Question CRUD

```
GET    /api/questions/{id}          // Get
PUT    /api/questions/{id}          // Update
DELETE /api/questions/{id}          // Soft delete
DELETE /api/questions/{id}/permanent // Permanent delete
```

Attributes

```
// Get topic attributes
GET /api/topic/{topic_id}/attributes

// Create attributes
POST /api/topic/{topic_id}/attributes
[
  { "name": "Understanding Laws", "description": "..." },
  { "name": "Applying Forces", "description": "..." }
]
```

Page 4: Resource Upload

Add Resources to Topic

```
// Single resource
POST /api/topics/{id}/resources
{
  "resource_type": "video", // video, pdf, 3d_model, virtual_lab, etc.
  "title": "Introduction Video",
  "url": "https://youtube.com/...",
  "duration": 600,
  "metadata": { "platform": "YouTube", "quality": "1080p" }
}

// Bulk resources
POST /api/topics/{id}/resources/bulk
{
  "resources": [
    { "resource_type": "video", "title": "...", "url": "..." },
    { "resource_type": "pdf", "title": "...", "url": "..." }
  ]
}
```

Resource Types

- **video** - Videos (YouTube, Vimeo)

- **image** - Images/diagrams
- **3d_model** - 3D models
- **animation** - Animations
- **virtual_lab** - Virtual labs
- **pdf** - PDF documents
- **interactive** - Interactive widgets
- **article** - Articles
- **simulation** - Simulations

Get Resources

```
// Get topic resources
GET /api/topics/{id}/resources
GET /api/topics/{id}/resources?resource_type=video

// Get chapter resources (grouped by topics)
GET /api/chapters/{id}/resources
```

Resource CRUD

```
GET    /api/resources/{id}           // Get
PUT    /api/resources/{id}           // Update
DELETE /api/resources/{id}           // Soft delete
DELETE /api/resources/{id}/permanent // Permanent delete
```

Page 5: Enhanced Question Fetching

Get Questions with Q-Matrix

```
// Enhanced format with Q-vector
GET /api/hierarchy/{level}/{id}/questions/enhanced?page=1&page_size=20
// level: exam, subject, chapter, topic

// Response includes:
{
  "attributes": [...],
  "questions": [
    {
      "id": "...",
      "content": "...",
      "q_vector": [1, 1, 0, 0], // Q-matrix as binary vector
      "attribute_count": 2,
      "question_image_url": "...",
      "option_images": { "A": "...", "B": "..." }
    }
  ],
}
```

```
"pagination": { "page": 1, "total": 150, "has_more": true }
}
```

Standard Format

```
// Standard format with attributes array
GET /api/hierarchy/{level}/{id}/questions?page=1&page_size=20

// Response includes:
{
  "data": [
    {
      "id": "...",
      "content": "...",
      "attributes": [
        { "id": "...", "name": "Understanding Laws", "value": true }
      ]
    }
  ],
  "pagination": {...}
}
```

Search Questions

```
GET /api/search/questions?topic_id=
{id}&difficulty_min=0.5&difficulty_max=0.8&page=1
```

Get Item Bank

```
GET /api/item-bank/{level}/{id}?page=1&page_size=20&difficulty_min=0.5
```

Common Patterns

Complete Question Upload Workflow

```
// 1. Get topic attributes
const attrs = await fetch(`/api/topic/${topicId}/attributes`).then(r =>
r.json());

// 2. Create question with attributes
const response = await fetch('/api/questions/create-with-attributes', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
```

```

    body: JSON.stringify({
      question: { content, options, correct_answer, topic_id, ... },
      selected_attributes: selectedAttrIds.map(id => ({ attribute_id: id,
value: true })))
    })
  });

const { question } = await response.json();

// 3. Upload question image (if exists)
if (questionImage) {
  const formData = new FormData();
  formData.append('image', questionImage);
  await fetch(`/api/questions/${question.id}/image`, {
    method: 'POST',
    body: formData
  });
}

// 4. Upload option images (if exist)
if (optionImages) {
  const formData = new FormData();
  Object.entries(optionImages).forEach(([key, file]) => {
    formData.append(`option_${key}`, file);
  });
  await fetch(`/api/questions/${question.id}/options/images`, {
    method: 'POST',
    body: formData
  });
}

```

Cascading Dropdown Navigation

```

// Load subjects when exam selected
useEffect(() => {
  if (selectedExam) {
    fetch(`/api/hierarchy/subjects?exam_id=${selectedExam.id}`)
      .then(r => r.json())
      .then(data => setSubjects(data));
  }
}, [selectedExam]);

// Load chapters when subject selected
useEffect(() => {
  if (selectedSubject) {
    fetch(`/api/hierarchy/chapters?subject_id=${selectedSubject.id}`)
      .then(r => r.json())
      .then(data => setChapters(data));
  }
}, [selectedSubject]);

```

```
// Continue pattern for topics and concepts
```

Display Q-Matrix

```
// Q-vector is a binary array [1, 1, 0, 0]
// Index maps to attributes array
const QMatrix = ({ qVector, attributes }) => (
  <div className="q-matrix">
    {attributes.map((attr, idx) => (
      <div key={attr.id} className={qVector[idx] ? 'active' : 'inactive'}>
        {qVector[idx] ? '✓' : '○'} {attr.name}
      </div>
    ))}
  </div>
);
```

HTTP Status Codes

- 200 OK - Success
- 201 Created - Resource created
- 400 Bad Request - Validation error
- 404 Not Found - Resource doesn't exist
- 500 Server Error - Internal error

TypeScript Types

```
interface Exam {
  id: string;
  name: string;
  description: string;
  exam_type: 'competitive' | 'school';
  created_at: string;
}

interface Question {
  id: string;
  content: string;
  options: { [key: string]: string };
  correct_answer: string;
  difficulty: number;
  discrimination: number;
  guessing: number;
  topic_id: string;
  metadata?: any;
}
```

```
interface EnhancedQuestion extends Question {
  q_vector: number[];
  attribute_count: number;
  question_image_url?: string;
  option_images?: { [key: string]: string };
}

interface Attribute {
  id: string;
  name: string;
  description: string;
  topic_id: string;
}

interface Resource {
  id: string;
  topic_id: string;
  resource_type: string;
  title: string;
  description: string;
  url: string;
  thumbnail_url?: string;
  duration?: number;
  file_size?: number;
  metadata?: any;
  order_index: number;
  is_active: boolean;
}

interface PaginationInfo {
  page: number;
  page_size: number;
  total: number;
  total_pages: number;
  has_more: boolean;
}
```

Environment Setup

```
# Start server
python -m app.main

# Server runs on
http://localhost:5200

# Base API URL
http://localhost:5200/api
```


Testing

Use test notebooks:

- [topic_resources_test.ipynb](#)
- [endpoint_coverage_test.ipynb](#)

Or cURL:

```
curl http://localhost:5200/api/hierarchy/exams
```

File Upload Notes

- **Question Images:** Max size ~10MB
- **Option Images:** Max size ~5MB each
- **Supported formats:** PNG, JPG, JPEG, GIF
- **Storage:** Supabase Storage
- **Access:** Public URLs returned

3PL Parameters Guide

- **Difficulty** (-3 to 3)
 - [-3 to -1](#): Very easy
 - [-1 to 0](#): Easy
 - [0 to 1](#): Medium
 - [1 to 2](#): Hard
 - [2 to 3](#): Very hard
- **Discrimination** (0 to 3)
 - Higher = better differentiates students
- **Guessing** (0 to 1)
 - Usually [0.25](#) for 4-option MCQ
 - [0.33](#) for 3-option MCQ

For Help

- Full Guide: [/docs/FRONTEND_ADMIN_APP_GUIDE.md](#)
- API Docs: [/docs/TOPIC_RESOURCES_API.md](#)
- Setup: [/TOPIC_RESOURCES_SETUP.md](#)