Document

September 4, 2026

Document

Kind: Database Model

Source: `atloria-monorepo/libs/database/prisma/schema.prisma`

`Document` stores structured document content and its rendered HTML representation. It associates a document with a URL identifier and tracks classification, origin, lifecycle state, and page type for use across content ingestion, rendering, and publishing workflows.

Fields

Field	Type	Required	Key
`id`	`String`	✓	PK
`urlId`	`String`	-	unique
`title`	`String`	✓	
`slug`	`String`	✓	
`content`	`String`	✓	
`contentHtml`	`String`	-	
`type`	`DocumentType`	✓	
`source`	`DocumentSource`	✓	
`state`	`DocumentState`	✓	
`pageType`	`String`	-	
`version`	`Int`	✓	
`parentVersionId`	`String`	-	
`canRegenerate`	`Boolean`	✓	
`kgEntityId`	`String`	-	

Field	Type	Required	Key
`kgEntityType`	`String`	-	
`isPublic`	`Boolean`	✓	
`publishedAt`	`DateTime`	-	
`scheduledPublishAt`	`DateTime`	-	
`parentId`	`String`	-	
`collectionId`	`String`	-	
`toc`	`Json`	-	

Diagram

```
erDiagram
  DOCUMENT {
    String id
    String urlId
    String title
    String slug
    String content
    String contentHtml
    DocumentType type
    DocumentSource source
    DocumentState state
    String pageType
  }
```

Usage

```
import { PrismaClient, type DocumentSource, type DocumentState, type DocumentType } from "@pris

const prisma = new PrismaClient();

async function createDocument(
  type: DocumentType,
  source: DocumentSource,
  state: DocumentState,
) {
  const document = await prisma.document.create({
    data: {
      id: crypto.randomUUID(),
      urlId: "url_123",
      title: "Getting Started",
      slug: "getting-started",
```

```
      content: "# Getting Started\n\nDocument source content.",
      contentHtml: "<h1>Getting Started</h1><p>Document source content.</p>",
      type,
      source,
      state,
      pageType: "documentation",
    },
  });

  return document;
}
```

AI Coding Instructions

- Keep `content` and `contentHtml` synchronized: store source content in `content` and the rendered output in `contentHtml`.
- Use the generated Prisma client (`prisma.document`) for reads and writes rather than constructing database queries manually.
- Provide valid `DocumentType`, `DocumentSource`, and `DocumentState` enum values defined in the Prisma schema; do not persist arbitrary strings for these fields.
- Treat `slug` and `urlId` as important lookup and routing identifiers; validate their values before creating or updating documents.
- Update document state deliberately as part of ingestion and publishing workflows so consumers can distinguish draft, processed, and published content.

Relationships

- HAS_ONE → `Document`
- HAS_ONE → `Document`
- HAS_ONE → `Collection`

Referenced By

- `User` (HAS_ONE)
- `User` (HAS_ONE)
- `Document` (HAS_ONE)
- `Document` (HAS_ONE)
- `Comment` (HAS_ONE)
- `Suggestion` (HAS_ONE)
- `CodeSymbol` (HAS_ONE)

- `CollaborationSession` (HAS_ONE)
- `DocumentVersion` (HAS_ONE)
- `DocumentRevision` (HAS_ONE)