

Organization

September 4, 2026

Organization

Kind: Database Model

Source: [atloria-monorepo/libs/database/prisma/schema.prisma](https://github.com/atloria-monorepo/libs/database/prisma/schema.prisma)

The `Organization` model represents a tenant or customer workspace within the application. It stores organization identity, configurable settings, subscription and Stripe billing metadata, and the effective plan configuration used to control product access.

Fields

Field	Type	Required	Key
<code>id</code>	<code>String</code>	✓	PK
<code>name</code>	<code>String</code>	✓	
<code>slug</code>	<code>String</code>	✓	unique
<code>settings</code>	<code>Json</code>	-	
<code>createdAt</code>	<code>DateTime</code>	✓	
<code>updatedAt</code>	<code>DateTime</code>	✓	
<code>plan</code>	<code>OrganizationPlan</code>	✓	
<code>stripeCustomerId</code>	<code>String</code>	-	unique
<code>stripeSubscriptionId</code>	<code>String</code>	-	
<code>planOverride</code>	<code>OrganizationPlan</code>	-	
<code>planOverrideExpiresAt</code>	<code>DateTime</code>	-	
<code>planOverrideReason</code>	<code>String</code>	-	
<code>planOverrideBy</code>	<code>String</code>	-	
<code>invitations</code>	<code>Invitation[]</code>	✓	

Field	Type	Required	Key
projects	Project[]	✓	
themes	Theme[]	✓	
users	User[]	✓	
templates	DocumentTemplate[]	✓	
collections	Collection[]	✓	
screenshotJobs	ScreenshotJob[]	✓	
screenshots	Screenshot[]	✓	
supportAgentConfigs	SupportAgentConfig[]	✓	
sourceConnectors	SourceConnector[]	✓	
customDomains	CustomDomain[]	✓	
slackIntegrations	SlackIntegration[]	✓	
samlConfig	SamlConfig	-	
whiteLabelConfig	WhiteLabelConfig	-	
complianceConfig	ComplianceConfig	-	
scimTokens	ScimToken[]	✓	
scimGroupMappings	ScimGroupMapping[]	✓	
docVersions	DocVersion[]	✓	

Diagram

```

erDiagram
    ORGANIZATION {
        String id PK
        String name
        String slug UK
        Json settings
        DateTime createdAt
        DateTime updatedAt
        OrganizationPlan plan
        String stripeCustomerId
        String stripeSubscriptionId
        OrganizationPlan planOverride
    }

```

Usage

```
import { PrismaClient, OrganizationPlan } from '@prisma/client';

const prisma = new PrismaClient();

// Create an organization when provisioning a new customer workspace.
const organization = await prisma.organization.create({
  data: {
    name: 'Acme Inc.',
    slug: 'acme-inc',
    settings: {
      timezone: 'America/New_York',
      allowInvitations: true,
    },
    plan: OrganizationPlan.FREE,
  },
});

// Look up an organization by its URL-safe slug.
const foundOrganization = await prisma.organization.findUnique({
  where: {
    slug: 'acme-inc',
  },
});

// Update billing data after receiving a Stripe subscription event.
await prisma.organization.update({
  where: { id: organization.id },
  data: {
    stripeCustomerId: 'cus_123',
    stripeSubscriptionId: 'sub_123',
    plan: OrganizationPlan.PRO,
  },
});
```

AI Coding Instructions

- Use `slug` as the human-readable, URL-safe organization identifier; validate and normalize it before creation.
- Treat `settings` as flexible JSON and provide safe defaults when reading optional configuration values.
- Update `stripeCustomerId`, `stripeSubscriptionId`, and `plan` from verified Stripe webhook events rather than client-provided input.
- Apply `planOverride` consistently when evaluating feature access, typically preferring it over the subscribed `plan` when present.

- Use Prisma `createdAt` and `updatedAt` timestamps for auditing; do not manually overwrite them unless performing a deliberate migration.

Referenced By

- `User` (HAS_ONE)
- `Invitation` (HAS_ONE)
- `Project` (HAS_ONE)
- `Theme` (HAS_ONE)
- `DocumentTemplate` (HAS_ONE)
- `ScreenshotJob` (HAS_ONE)
- `Screenshot` (HAS_ONE)
- `SupportAgentConfig` (HAS_ONE)
- `SourceConnector` (HAS_ONE)
- `CustomDomain` (HAS_ONE)
- `SlackIntegration` (HAS_ONE)
- `ScimToken` (HAS_ONE)
- `ScimGroupMapping` (HAS_ONE)
- `WhitelabelConfig` (HAS_ONE)
- `ComplianceConfig` (HAS_ONE)