

Storyblok content model — essentials

- Blocks are the building unit. Three types: **Content type** (a full story, e.g., `page`, `post`), **Nestable** (layout pieces like `hero`, `text`, `image`), and **Universal** (can be either). Use a **Blocks** field on a content type to compose pages from nestable blocks. (storyblok.com)
- Model with **fields**: Text, Rich text (JSON, TipTap-based), Asset (single/multi), Option (single/multi), Date/Time, Boolean, **References**, Link, Table, Group, Plugin, etc. Configure validations, conditional fields, and translatability per field. (storyblok.com)
- **References** (and Option fields with story sources) let you relate entries (e.g., Post → Author, Post → Categories). At delivery, resolve linked entries via `resolve_relations`. (storyblok.com)
- **Datasources** are great for enums/lookups (e.g., tag lists, constants) and can use dimensions (e.g., locales). (storyblok.com)

WP → Storyblok migration — practical steps

1. **Audit & design**: Inventory WP types (Posts, Pages, Categories, Tags, Media, Authors, Menus). Draft Storyblok components: `post` (content type) + nestables (`rich_text`, `image`, `quote`, `cta`, etc.), `author`, `category`. (storyblok.com)
2. **Create schema in Storyblok**: Build your content types & fields (e.g., Post: `title` (Text), `slug` (Text), `featured_image` (Asset), `excerpt` (Rich text), `content` (Rich text/Blocks), `categories` (References/Options), `author` (Reference)). (storyblok.com)

follow Storyblok's **tutorial** that shows schema design + mapping examples and importer usage. (storyblok.com)

3. **Map fields** (typical):
 - WP `title` → Storyblok `name`; `slug` → `slug`; `date` → `first_published_at`; `content` → `content` (Rich text or Blocks); `featured_media` → `featured_image` (Asset); `categories/tags` → References/Options; `author`

→ Reference. (storyblok.com)

4. **Map Gutenberg blocks:** Define `blocks_mapping` so `core/paragraph`, `core/image`, etc., become your nestable components (e.g., `rich_text`, `image`). ([GitHub](https://github.com/storyblok/storyblok))
5. **Run the importer** against your WP REST endpoint; verify stories, assets, and relations in your space. (storyblok.com)
6. **Front-end hookup:** In your app, request stories and **resolve relations** (`featured_articles.articles`, `post.author`, etc.) so linked entries render with a single request. (storyblok.com)
7. **Operational tips:** Use the **Storyblok CLI** to pull/push component schemas, sync components/data between spaces, and generate migration files/types as your model evolves. ([GitHub](https://github.com/storyblok/storyblok))

Quick mapping template (example)

- **Content types:** `post`, `page`, `author`, `category`.
- **Nestable blocks:** `rich_text`, `image`, `gallery`, `quote`, `cta`, `video`, `accordion`, `hero`.
- **Post fields:** `title`(Text), `slug`, `featured_image`(Asset), `excerpt`(Rich text), `body`(Blocks), `categories`(References/Options), `author`(Reference), `seo`(Group).
- **Author fields:** `name`(Text), `bio`(Text/Rich), `photo`(Asset), `links`(Link/Options).
- **Category fields:** `name`(Text), `slug`(Text), optional `description`(Text).