Getting Started with BlockNote AI

This guide walks you through the steps to add AI functionality to your BlockNote rich text editor.

First, install the `@blocknote/xl-ai` package:

```
npm install @blocknote/xl-ai
```

Creating a Model

BlockNote AI uses the the AI SDK to standardize integrating artificial intelligence (AI) models across supported providers.

As a first step, you'll need to register a new model with the AI SDK. For example, for Llama hosted on Groq:

```
npm install @ai-sdk/groq
```

```
import { createGroq } from "@ai-sdk/groq";

const provider = createGroq({
  apiKey: "YOUR_GROQ_API_KEY",
});

const model = provider("llama-3.3-70b-versatile");
```

⚠ Note that this setup directly calls the provider from the client, and exposes your API keys on the client. For Production scenarios, a more common approach is to handle authentication on your own server and proxy requests to a provider. See our Demo AI Server for a Node.js example or check the AI SDK best practices.

Setting up the editor

```
import { createBlockNoteEditor } from "@blocknote/core";
import { BlockNoteAIExtension } from "@blocknote/xl-ai";
import { en } from "@blocknote/core/locales";
import { en as aiEn } from "@blocknote/xl-ai/locales";
import { createAIExtension } from "@blocknote/xl-ai";
import "@blocknote/xl-ai/style.css"; // add the AI stylesheet

const editor = createBlockNoteEditor({
  dictionary: {
    ...en,
    ai: aiEn, // add default translations for the AI extension
  },
  extensions: [
    createAIExtension({
      model,
    }),
  ],
  // ... other editor options
});
```

See the API Reference for more information on the `createAIExtension` method.

Adding AI UI elements

Now, the only thing left to do is to customize the UI elements of your editor. We want to:

- add an AI button to the formatting toolbar (shown when selecting text)
- add an AI option to the slash menu (shown when typing a `/`)

We do this by disabling the default FormattingToolbar and SuggestionMenu and adding our own. See Changing UI Elements for more information.

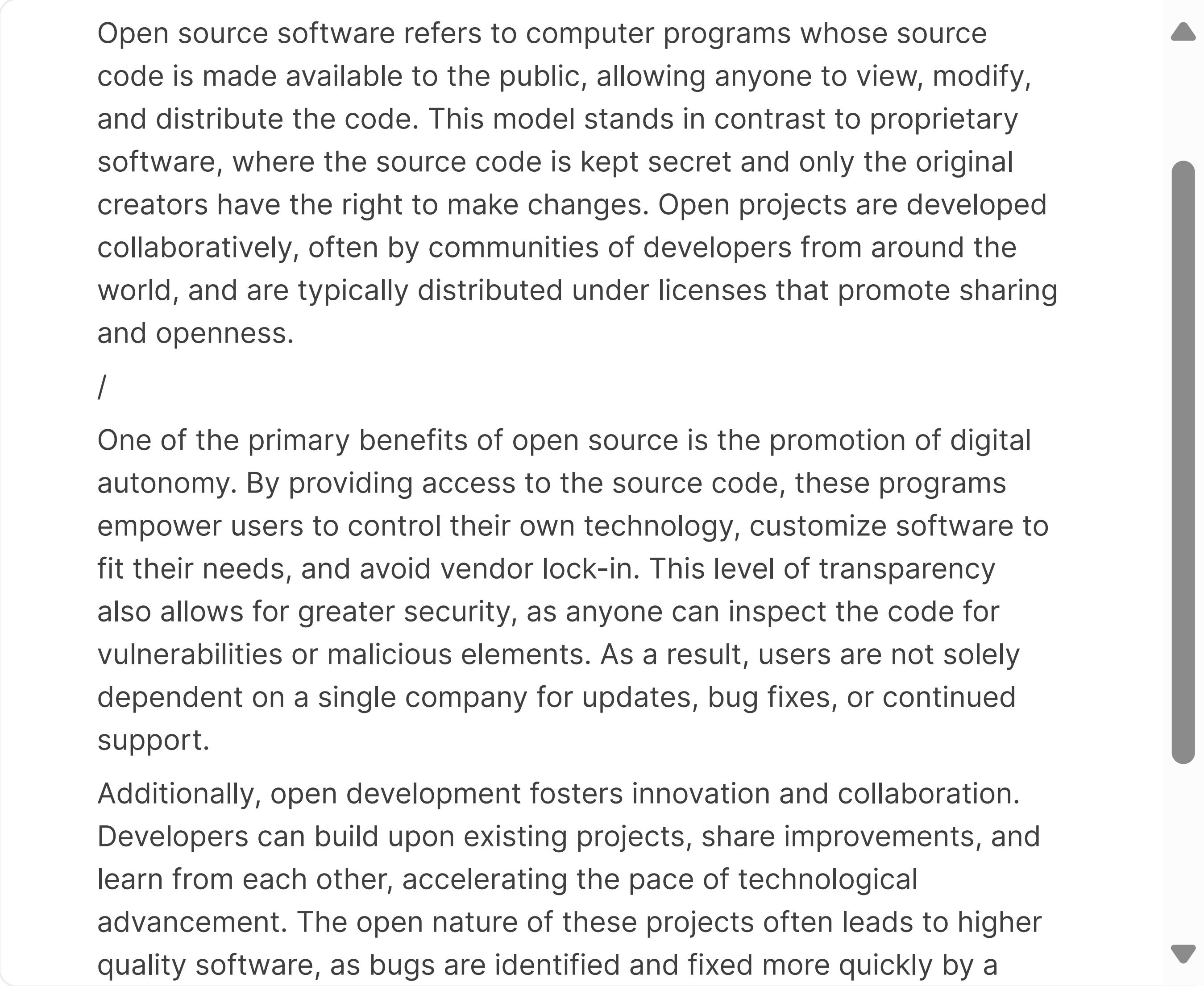
```
<BlockNoteView
  editor={editor}
  // We're disabling some default UI elements
  formattingToolbar={false}
  slashMenu={false}
>
  {/* Add the AI Command menu to the editor */}
  <AIMenuController />
  {/* Create you own Formatting Toolbar with an AI button,
```

```
  {/* Create you own SlashMenu with an AI option,
    (see the full example code below) */}
  <SuggestionMenuWithAI editor={editor} />
</BlockNoteView>
```

Full Example

See the full example code and live demo. Select some text and click the AI (stars) button, or type `/ai` anywhere in the editor to access AI functionality.

GitHub StackBlitz



Open source software refers to computer programs whose source code is made available to the public, allowing anyone to view, modify, and distribute the code. This model stands in contrast to proprietary software, where the source code is kept secret and only the original creators have the right to make changes. Open projects are developed collaboratively, often by communities of developers from around the world, and are typically distributed under licenses that promote sharing and openness.

/

One of the primary benefits of open source is the promotion of digital autonomy. By providing access to the source code, these programs empower users to control their own technology, customize software to fit their needs, and avoid vendor lock-in. This level of transparency also allows for greater security, as anyone can inspect the code for vulnerabilities or malicious elements. As a result, users are not solely dependent on a single company for updates, bug fixes, or continued support.

Additionally, open development fosters innovation and collaboration. Developers can build upon existing projects, share improvements, and learn from each other, accelerating the pace of technological advancement. The open nature of these projects often leads to higher quality software, as bugs are identified and fixed more quickly by a

App.tsx getEnv.ts

```
import { createGroq } from "@ai-sdk/groq";
import { BlockNoteEditor, filterSuggestionItems } from "@blocknote/core";
import "@blocknote/core/fonts/inter.css";
```

```
import {
  FormattingToolbar,
  FormattingToolbarController,
  SuggestionMenuController,
  getDefaultReactSlashMenuItems,
  getFormattingToolbarItems,
  useCreateBlockNote,
} from "@blocknote/react";
import {
  AIMenuController,
  AIToolbarButton,
  createAIExtension,
  createBlockNoteAIClient,
  getAISlashMenuItems,
} from "@blocknote/xl-ai";
import { en as aiEn } from "@blocknote/xl-ai/locales";
import "@blocknote/xl-ai/style.css";

import { getEnv } from "./getEnv";

// Optional: proxy requests through the `@blocknote/xl-ai-server` proxy server
// so that we don't have to expose our API keys to the client
const client = createBlockNoteAIClient({
  apiKey: getEnv("BLOCKNOTE_AI_SERVER_API_KEY") || "PLACEHOLDER",
```

‹ AI Rich Text Editing
Add AI functionality to your BlockNote rich tex...

Custom AI Commands ›
Customize the AI menu items (commands) in y...

🚀 BlockNote AI is here! *Access the early preview.*

BlockNote

Docs AI Examples Pricing About

Terms & Conditions

Privacy Policy

☀ ☾ 🖥