# Unlocking the Power of AI: Build Locally with oLama and Dy.ai

Welcome back to the blog! If you’re looking to dive into the world of artificial intelligence without getting lost in the complex jargon or heavy coding, you’ve landed in the right place. Today, we’re exploring how to use a no-code platform known as **Dy.ai** to run the open-source large language model **oLama** right on your own machine.

## Why Go Local with AI?

When you think of AI models like GPT-4 or Gemini, you usually think of cloud services that require an API key. While these options are robust, they may not offer the data confidentiality many users desire. Here’s where running a local model like oLama becomes beneficial. By self-hosting your model, you have total control over your data and can tweak the setup to meet your exact needs.

## Getting Started with oLama

### Step 1: Install oLama

First things first, you’ll need to download oLama from its official website. Whether you’re using **Windows**, **MacOS**, or **Linux**, the installation process is straightforward. Just download the installer and follow the instructions.



Once installed, you can check if it’s running by looking for the oLama icon on your screen or running a simple command in your terminal.

### Step 2: Check Available Models

To see what models you have at your disposal, open your terminal and type:

oLama list

This command will show you a list of models available for use, including popular ones like **oLama 3.2 3B** and various embedding models.

### Step 3: Running a Model

If you’re ready to test things out, you can simply run a model by typing:

oLama run <model\_name>

Once it’s running, you can start asking your AI queries right from your terminal. For instance, you might type:

Tell me a joke.

And voilà! The AI responds with something like: “What do you call a fake noodle? An impasta!”

## Integrating with Dy.ai

Now that you have oLama up and running, let’s explore Dy.ai, a no-code platform that makes building AI applications a breeze.

### Step 1: Clone the Dy.ai Repository

You’ll want to get the Dy.ai code by cloning its GitHub repository. In your terminal, type:

git clone <repository\_link>

Once that’s done, navigate to the cloned folder and open it in your code editor, like Visual Studio Code.

### Step 2: Setting Up the Docker Environment

Dy.ai relies on Docker, a container technology that simplifies software deployment. Initialize Docker by running:

docker-compose up --detach

This command will download the necessary images and set everything up for you.

### Step 3: Adding oLama to Dy.ai

In Dy.ai, navigate to **Settings > Model Provider**, and select **oLama**. Be careful when entering the base URL; it should look something like this:

http://host.docker.internal:11434

The model name must match exactly how oLama presents it, ensuring seamless connectivity.

### Step 4: Building Your Application

You can now create a new project within Dy.ai. Click on **Create from Blank** and choose from various options like chatbots or text generators.

For example, if you choose to build a text generator, you might set a prefix prompt such as asking what AI is and run the application.

## Conclusion: The Empowering World of Local AI

Incorporating oLama with Dy.ai opens up a plethora of possibilities for developers and enthusiasts alike without the complications of API keys and cloud services. You can create your own AI-powered applications faster while keeping your data secure and private.

If you have any questions, thoughts, or experiences with setting up your own local AI environment, please share in the comments below! If you found this blog useful, don’t forget to hit that like button and subscribe for more engaging content. Until next time, happy coding!