

[Menu](#)[On this page](#)

Welcome! First, if you haven't, read the [Setting Up Locally](#) guide to get Opsloom running end-to-end.

What We'll Be Creating

We'll be taking advantage of Opsloom's capabilities for Retrieval-Augmented Generation (RAG). RAG allows you to supply your AI assistant with context for its answers, giving it a base of knowledge to pull from.

In our particular case, we will be creating an AI assistant for the travel ministry of a fictional country called Raglandia. This assistant will pull from a guide with instructions on the online visa application process. It will then use the information in this document to answer customers' questions with appropriate responses.

Creating the Knowledge Base

After the initial setup, you should have your API and database running on your local machine. You can use [Bruno](#) to invoke these API endpoints for account and assistant setup. To do this, download Bruno and load in the `/docs/api_docs` folder.

Configure your local environment on Bruno to point to the API by setting the `server` variable's value to `http://localhost:8080/opsloom-api/v1` . If you haven't already, also set the `short_code` variable to `default` .

Open up the `kbase/create` endpoint. We'll call this in order to create the knowledge base. First, though, we'll want to modify the request to reflect the kind of knowledge

base we are interested in. Change the body to:

JSON

```
{  
  "name": "visa_kbase",  
  "description": "Raglandia Visa Process Knowledge Base"  
}
```

Now invoke the endpoint. As a response, you should get back the newly created entry in JSON format. We can verify that everything worked as expected by calling the [kbase/list](#) endpoint. In addition to any knowledge bases that were created when you initialized your account, you should now also see the Raglandia visa process knowledge base.

Adding Files to the Knowledge Base

The process of uploading files to a knowledge base is known as *indexing*. You can index files by invoking the [index/index_document](#) endpoint.

First, download the [Raglandia visa application process PDF](#). This is a simple document we have created for testing purposes. It includes some basic information about application steps, prices, turnaround times, and requirements.

On the [index/index_document](#) endpoint, open up the body tab and upload the PDF you just downloaded. In the [kbase_name](#) field, add the short name of the knowledge base you just created, "visa_kbase". Invoke the API—if it's successful, you'll get the S3 URI for the uploaded PDF back.

At this point our knowledge base is completely set up, so we are ready to create the RAG AI assistant.

Creating the Assistant

Open up the `assistant/create_assistant` endpoint. In the body tab, you'll see several values that will need to be customized. Make the following changes:

- `name` is used internally for identifying your assistant. You'll want to set it to something like *"openai_visa."*
- `config/type` is used to determine whether this is a RAG application or not. Set it to *"rag."*
- `metadata/title` is the header users will see when given the list of assistants. Name it *"Raglandia Visa Assistant."*
- `metadata/description` is the subtitle users will see when given the list of assistants. You'll want to set it to something like *"A guide through the Raglandia visa application process."*
- `system_prompts/system` is the prompt passed to the LLM to define its behavior in this assistant. We suggest setting it to the following:

You are a customer representative for the travel ministry of the country of Raglandia. You are tasked with responding to all questions about the visa application process to Raglandia. You are to respond only using the data from the provided "Raglandia Visa Application" document. Be concise and grammatically correct.

- `system_prompts/prompts` is an array of prompts to be shown to the user when they first open the assistant. These are good for guiding users to the most commonly asked questions. Here are some good prompts to start with:
 - *"What is the cost for a Business & Investor visa?"*
 - *"How long will it take to get my visa?"*
 - *"What are the requirements for the candidate picture?"*

Once you have made all of the updates above, invoke the API to create your assistant.

Testing the Assistant

To bring it all together, run the frontend code locally or refresh the page if it's already up. You should now see the Raglandia Visa Assistant on your dashboard. If you open

it, you'll be greeted with the three prompts you wrote. Click on any of the three, and observe as the assistant answers in its own words, pulling from the provided guide.

You're now done! Check out our other tutorials if you're interested in creating other kinds of assistants, or check out the [Styling](#) and [Embedding](#) guides to learn how to further customize this one.

Previous page

[Setting Up Locally](#)

Next page

[Styling](#)