

Documentación técnica de Denodo para HackUDC

Este documento técnico explica los pasos necesarios para obtener el entorno necesario para poder elaborar el reto propuesto por Denodo en el HackUDC 2025.

Jan 9, 2025

Copyright © 2024

Denodo Technologies Proprietary and Confidential

denodo.com

Índice de contenido

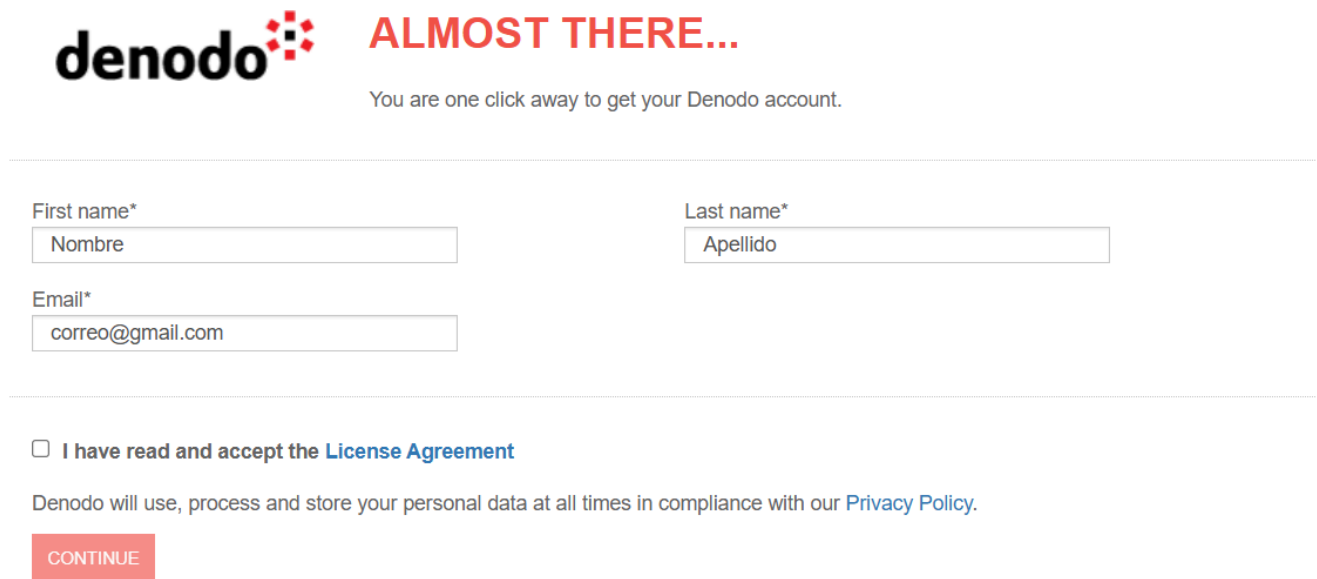
Índice de contenido	2
Cuenta de Denodo	3
Configuración de Docker	3
Clonación de Denodo Community Lab Environment Docker Compose project	4
Cuenta de Google AI Studio	6
Configuración completada	7
Mover los datos que se van a usar en DS al docker	8
Regenerar BBDD del AI SDK	10
Endpoints del AI SDK que necesito	11


Cuenta de Denodo

Lo primero que vamos a necesitar es crear una cuenta de Denodo. Esto nos permitirá descargar las imágenes de docker que necesitamos. Para ello, nos dirigimos a la siguiente dirección:

<https://auth.denodo.com/user-management/express-previous-register>

Una vez ahí, introducimos nuestro nombre y apellido junto con el email que vamos a querer usar para la cuenta de Denodo:



denodo  **ALMOST THERE...**

You are one click away to get your Denodo account.

First name*

Last name*

Email*

☐ I have read and accept the [License Agreement](#)

Denodo will use, process and store your personal data at all times in compliance with our [Privacy Policy](#).

CONTINUE

Posteriormente, se nos enviará un correo con un código de verificación, y dos enlaces muy importantes: uno para acceder a las instrucciones para descargar Denodo Express, otro para descargar nuestra nueva licencia de Denodo Express.

Una vez verificada nuestra cuenta de denodo, es hora de instalar Denodo Express y Denodo AI SDK.

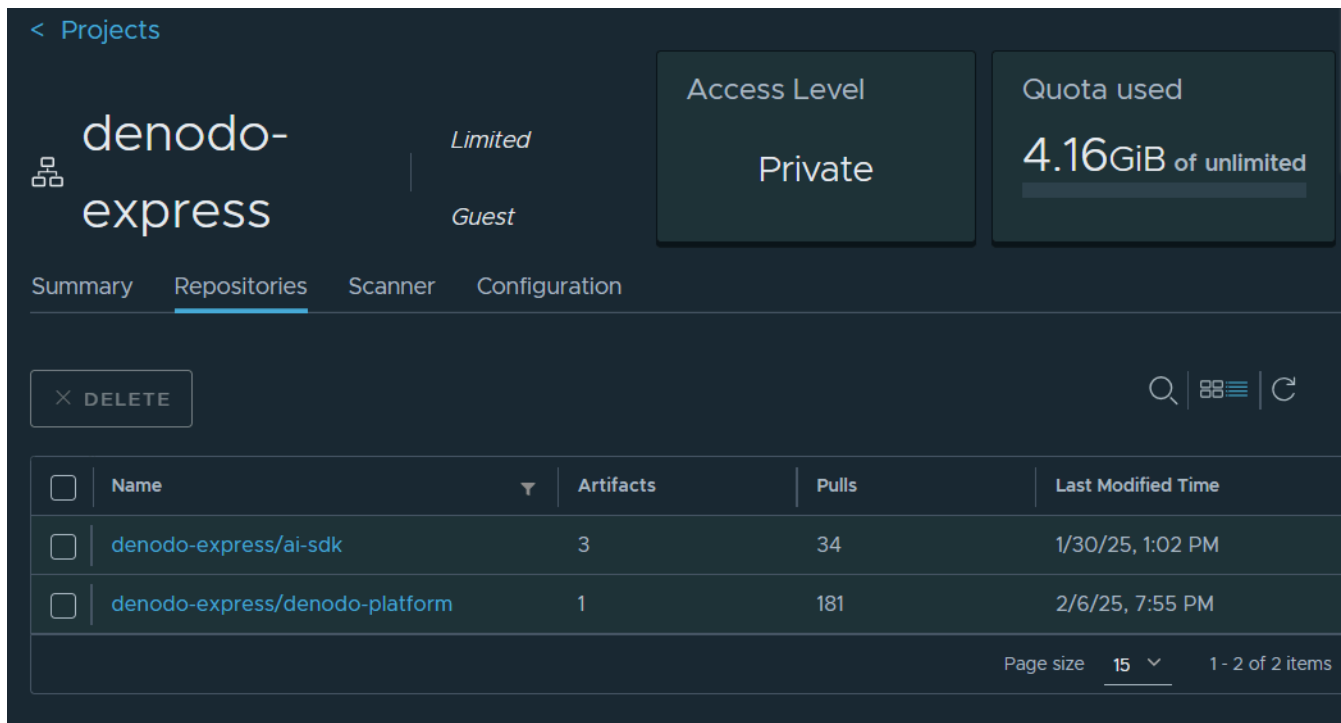
Configuración de Docker

Para la configuración del entorno que necesitamos es imprescindible tener instalado Docker en nuestro ordenador. Se puede obtener mediante [Docker Desktop](#) o también [Docker Engine](#) (si se opta por esta opción, a mayores del Docker Engine hay que instalar el plugin [Docker Compose](#)).

Para probar que la cuenta de Denodo está correctamente creada y obtener el nombre de usuario y el CLI secret que necesitaremos para docker, accederemos al Harbor de Denodo a través de este enlace:

<https://harbor.open.denodo.com/harbor/projects>

Si todo está correcto nos encontraremos con el proyecto **denodo-express**. Accedemos a él y comprobamos que nos aparecen tanto el repositorio **denodo-platform** como el **ai-sdk**:



< Projects

denodo-express

Limited Guest

Access Level Private

Quota used 4.16GiB of unlimited

Summary Repositories Scanner Configuration

X DELETE

Q 88 C

<input type="checkbox"/>	Name	Artifacts	Pulls	Last Modified Time
<input type="checkbox"/>	denodo-express/ai-sdk	3	34	1/30/25, 1:02 PM
<input type="checkbox"/>	denodo-express/denodo-platform	1	181	2/6/25, 7:55 PM

Page size 15 1 - 2 of 2 items

Ahora, pinchamos en la esquina superior derecha en el icono de usuario e iremos a User Profile. De aquí nos interesan dos parámetros para más adelante: **username** y **CLI Secret**. Estos parámetros serán utilizados en la [siguiente sección](#), así que nos los guardamos para más adelante.

Una vez comprobado el acceso al proyecto en Harbor, nos autenticamos en docker al Harbor de Denodo. Para ello, escribiremos el siguiente comando a docker:

Unset

```
docker login harbor.open.denodo.com
```

A continuación se nos pedirá el **username** y el **CLI secret** que hemos recabado en el paso anterior. Ahora, ya podremos ejecutar los comandos necesarios.

Clonación de Denodo Community Lab Environment

Docker Compose project

Vamos ahora a clonar un repositorio que nos permitirá rápidamente obtener todo el software que necesitamos para completar el reto de este HackUDC. El proyecto nos ayudará a instalar de manera desatendida el software que se menciona en [el documento en donde se describe el reto](#). Para comenzar, simplemente ejecutamos el siguiente comando, el cual clonará el repositorio de nuestro GitHub en el equipo:

Unset

```
git clone https://github.com/denodo/denodocommunity-lab-environment.git
```

Y una vez termine, ejecutaremos estos dos nuevos comandos:

Unset

```
# Accedemos a la carpeta en donde se encuentra el fichero de configuración
cd denodocommunity-lab-environment/lab-environment-containers/build/

# Procedemos a crearlo quitándole la terminación template
cp .env.template .env
```

Este fichero .env que acabamos de crear necesita ser configurado para funcionar, puesto que es una plantilla. Sin embargo, gran parte de los parámetros ya son válidos. Ahora mismo solo necesitamos modificar estos tres parámetros:

DENODO_VERSION=harbor.open.denodo.com/denodo-express/denodo-platform:latest

DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-express/ai-sdk:latest

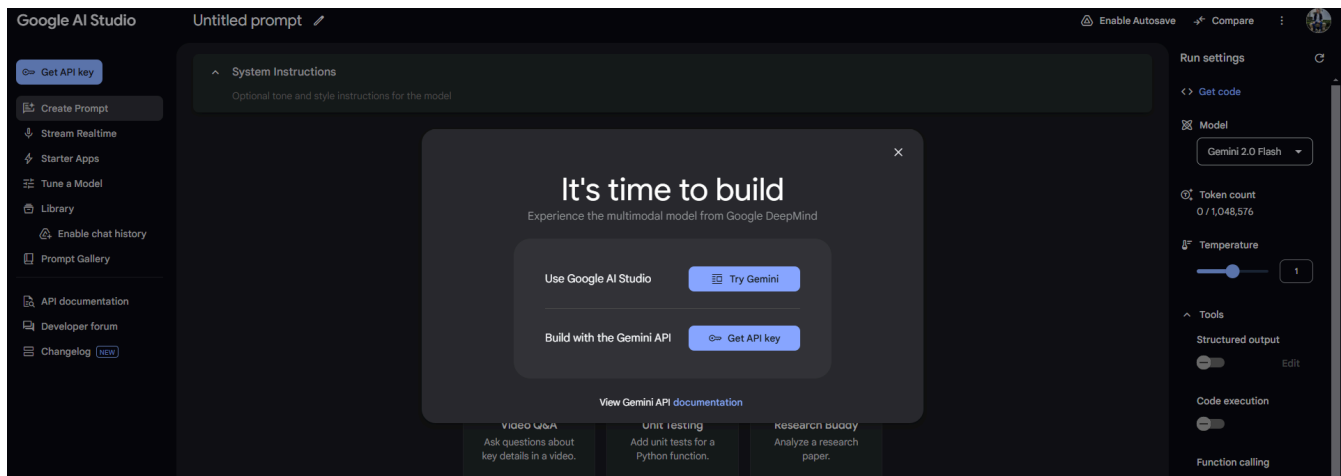
DENODO_SA_LIC= <path que apunte a donde tenemos almacenada nuestra licencia de denodo express que se nos concedió al crear una cuenta> (paso 2 de [express-previous-register](#)):

```
#####
# Denodo #
#####
# For Denodo Customers:    DENODO_VERSION=harbor.open.denodo.com/denodo-9/images/denodo-platform:latest
# For Denodo Express Users: DENODO_VERSION=harbor.open.denodo.com/denodo-express/denodo-platform:latest
DENODO_VERSION=harbor.open.denodo.com/denodo-express/denodo-platform:latest
# For Denodo Customers:    DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-connects-9/images/ai-sdk:latest
# For Denodo Express Users: DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-express/ai-sdk:latest
DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-express/ai-sdk:latest
# Full Path to your Denodo Standalone license file
DENODO_SA_LIC=../res/denodo/license/denodo-express-lic-9-202410.lic
```

Cuenta de Google AI Studio

Para este reto vamos a utilizar Google Gemini, en concreto usaremos el LLM **gemini-1.5-flash**. Usaremos Google Gemini porque nos permite unas cuotas de uso decentes y una buena potencia de manera totalmente gratuita. Para conseguir acceso, tenemos que registrarnos en la página de Google AI Studio, a la que podréis acceder desde este enlace:

<https://aistudio.google.com/>



Una vez creamos nuestra cuenta, ya podremos indicarle a Denodo AI SDK que utilice Google Gemini para resolver nuestras preguntas. Para ello, volvemos al fichero de configuración `.env` y añadimos los siguientes valores a estos parámetros:

PROVIDER: googleaistudio

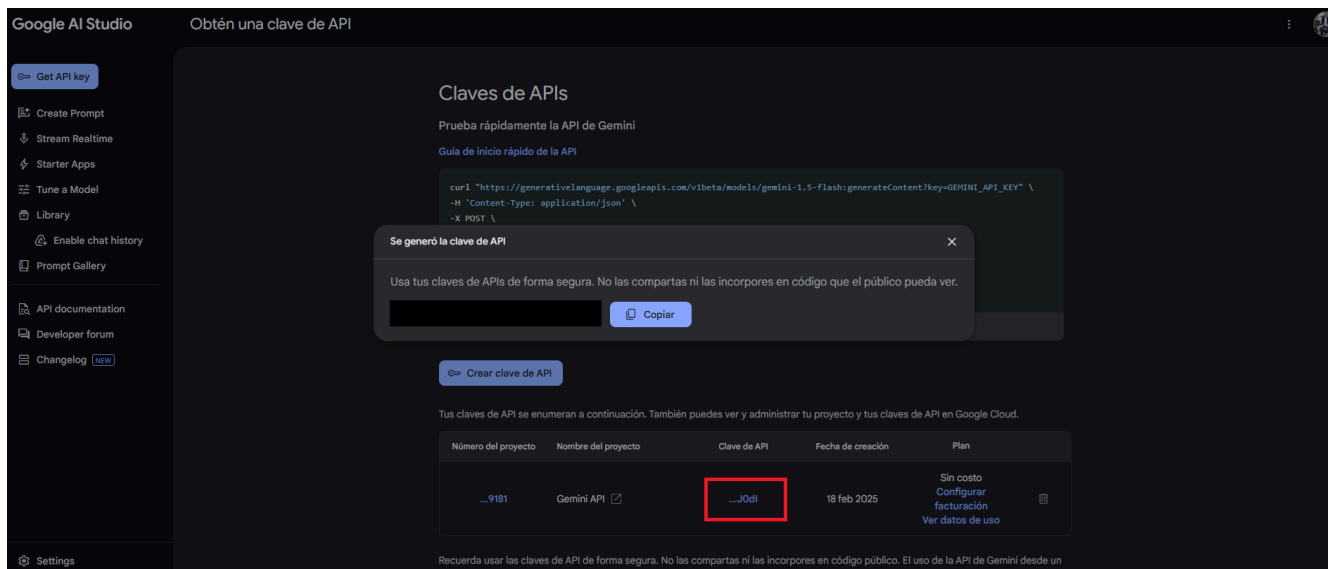
LLM_MODEL: gemini-1.5-flash (si tenéis problemas, probad gemini-2.0-flash)

EMB_MODEL: models/text-embedding-004

Y por último nos faltaría indicarle nuestras credenciales, para ello tendremos que añadir en el fichero de configuración `.env` la siguiente propiedad

`GOOGLE_AI_STUDIO_API_KEY=`

con el valor de la key generada previamente desde la consola de AI Studio:



Y posteriormente referenciamos esta nueva variable desde el docker compose modificando el fichero `docker-compose-sample-chatbot.yml` (que está en `denodocommunity-lab-environment/lab-environment-containers/build/docker-compose-sample-chatbot.yml`)

Unset

```
docker-ai-sdk-chatbot > environment:
```

```
...
```

```
GOOGLE_AI_STUDIO_API_KEY: ${GOOGLE_AI_STUDIO_API_KEY}
```

Configuración completada

Tras seguir los pasos anteriores, ya estamos listos para arrancar todo. Lanzamos los siguientes comandos (importante lanzarlos estando dentro del directorio `denodocommunity-lab-environment/lab-environment-containers/build`):

Unset

```
docker compose -f docker-compose-sample-chatbot.yml pull
```

```
docker compose -f docker-compose-sample-chatbot.yml up -d
```

Luego de descargar e iniciar, deberíamos de ver lo siguiente:

```
denodo@DESKTOP-8LKKV0V:~/lab-environment-containers/build$ docker compose -f docker-compose-sample-chatbot.yml up -d
[+] Running 3/3
✔Network build_default          Created                                0.1s
✔Container denodo-platform-chatbot Healthy                          182.5s
✔Container denodo-ai-sdk-chatbot Started                             182.7s
```

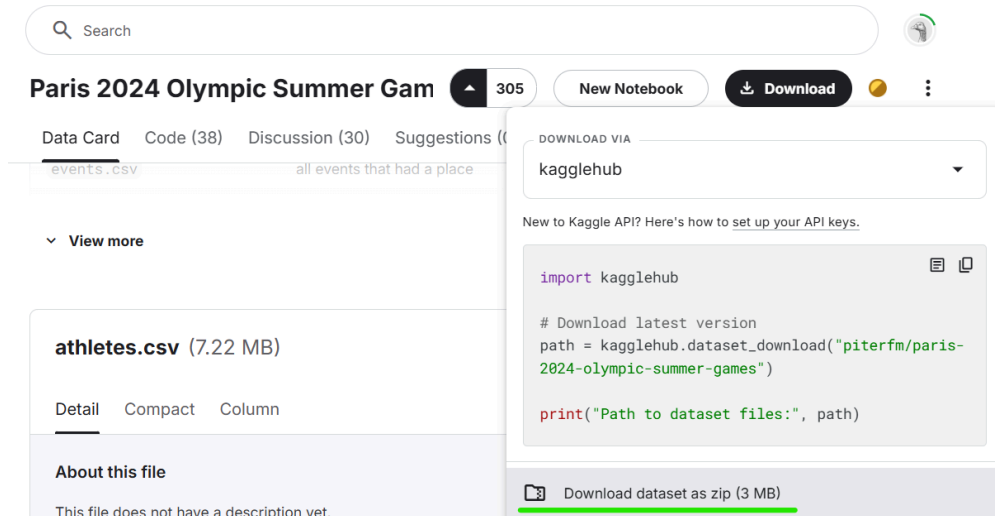
En este punto has terminado de configurar el entorno. Ahora ya puedes visitar los siguientes productos:

1. [Design Studio](#)
2. [Data Catalog](#)
3. [AI SDK \(APIs\)](#)
4. [Chatbot](#).

Usuario: admin, Contraseña: admin

Mover los datos que se van a usar en DS al docker

Lo primero que hay que hacer es tener disponibles los datos que se van a utilizar. Si se sigue el ejemplo de los JJOO 2024, estos se pueden obtener desde la descarga de Kaggle como zip (acordarse de descomprimirlo después):



Para poder encontrar desde Design Studio los datos que queremos usar tendremos que moverlos primero al contenedor de docker en donde está lanzado el Design Studio. Para ello se usará el siguiente comando de docker:

Unset

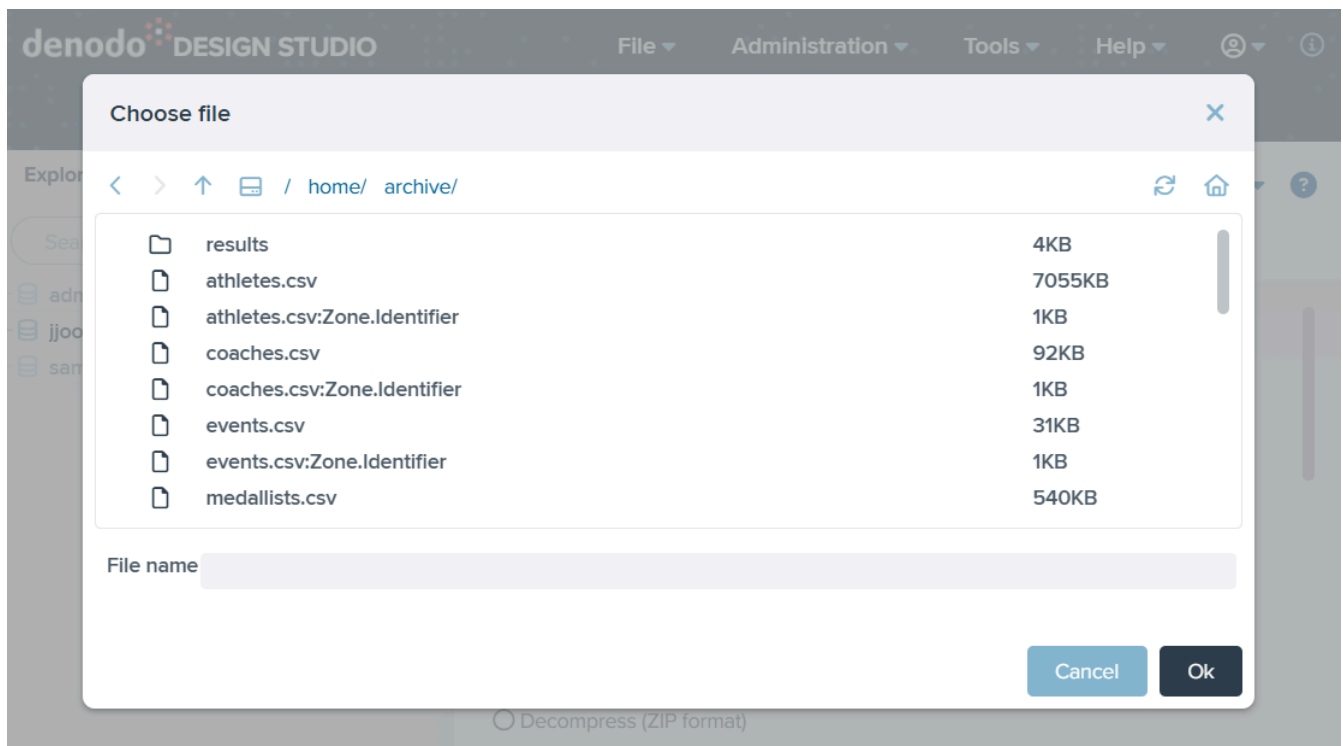
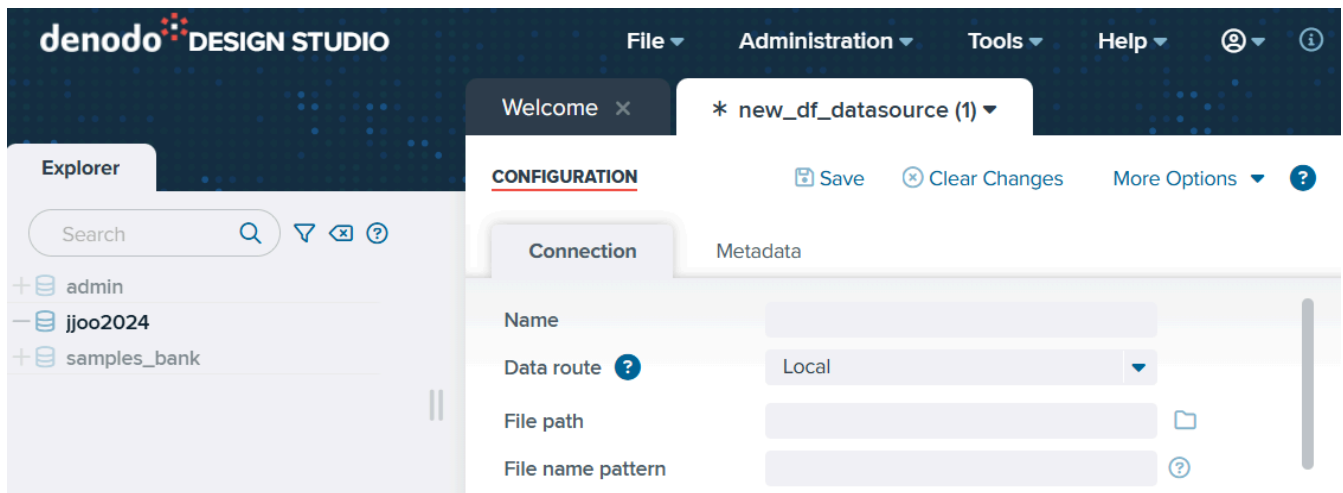
```
docker cp <nombre de la carpeta> <contenedor>:<ruta dentro del contenedor>
```

Por ejemplo, si el contenedor de docker en donde está lanzado denodo-platform-chatbot es el 9241d8c73ce6, para mover la carpeta archive que contiene todos los datos descargados de la BBDD de los juegos olímpicos de 2024 haríamos:

Unset

```
docker cp archive 9241d8c73ce6:/home
```

Ahora desde Design Studio al crear los datasources para los ficheros, podremos seleccionar Local en el campo Data Route y luego al buscar el File Path veremos que la carpeta archive está en /home:



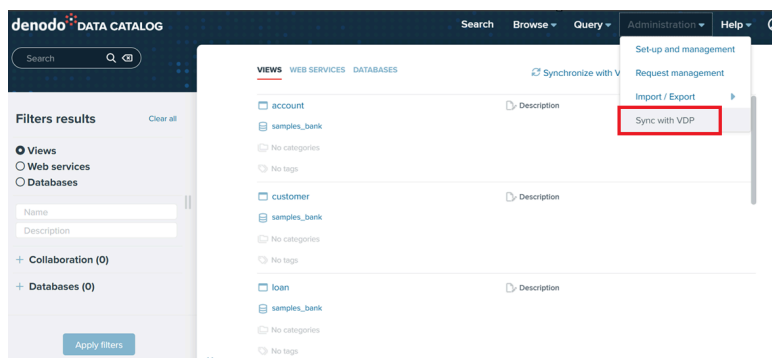
Con esto ya podremos diseñar el modelo de datos que luego servirá para contestar preguntas junto con el AI SDK.

Si no vas a hacer el ejemplo de los JJOO 2024, puedes tomar este proceso como inspiración para tus datos.

Regenerar BBDD del AI SDK

Luego de crear nuestros datos, tendremos que hacer dos cosas: sincronizar el Data Catalog y regenerar la BBDD del AI SDK para que pueda usarla para contestar preguntas.

Primero, lo más fácil. Abre el [Data Catalog](#) para sincronizar.



Segundo, nos vamos al fichero .env que estaba en `denodocommunity-lab-environment/lab-environment-containers/build/.env` y ahí tendremos que poner el nombre de la BBDD que hemos creado en los parámetros `VDB_NAMES` y `DENODO_CHATBOT_SAMPLE_VDB_NAMES`.

Después de eso, lanzad los siguientes comandos:

Unset

```
docker rm -f denodo-ai-sdk-chatbot
```

```
docker compose -f docker-compose-sample-chatbot.yml up -d denodo-ai-sdk-chatbot
```

Endpoints del AI SDK que necesito

Para posteriormente elaborar el chatbot propio o lo que queráis hacer, es importante conocer los endpoints que ofrece el AI SDK. Estos los podréis ver desde la siguiente URL: [AI SDK \(APIs\)](#)

El endpoint que os servirá para hacer un chatbot es el siguiente: **/answerQuestion**

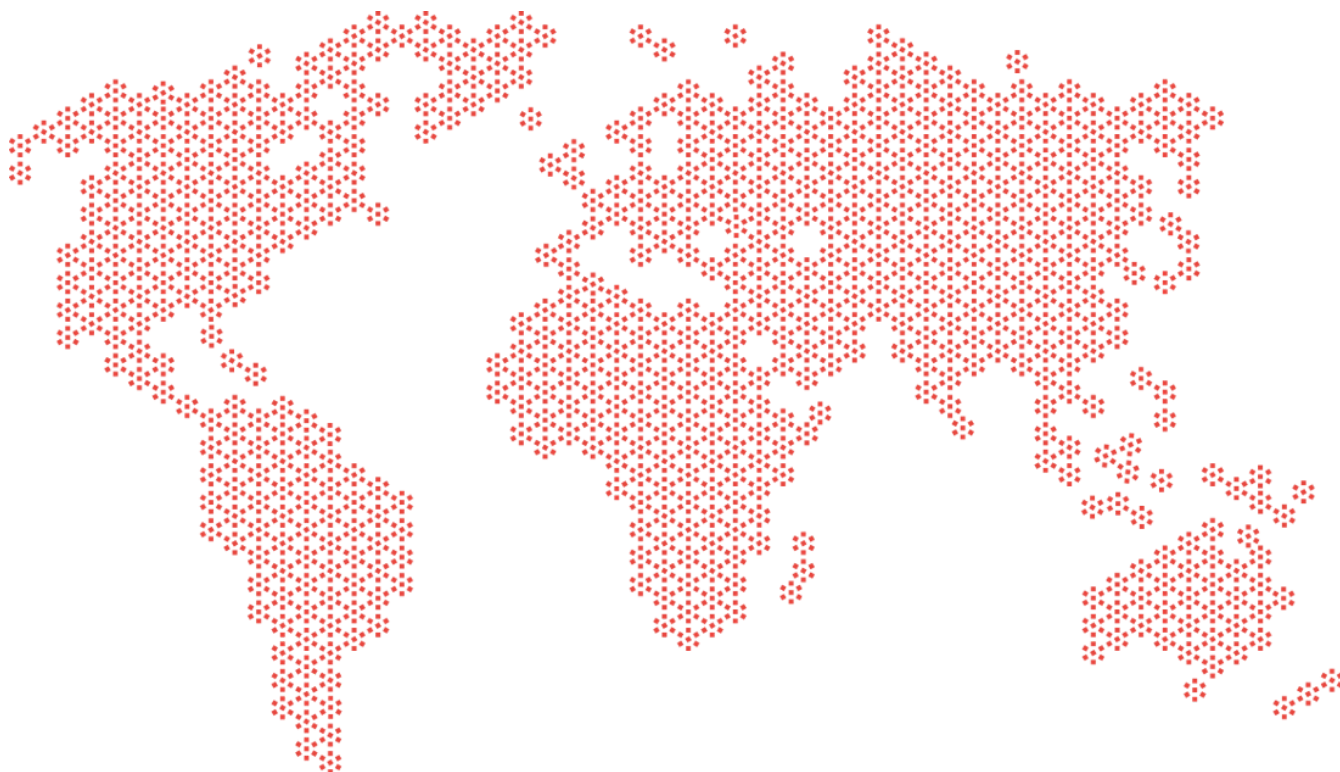
Este endpoint ya se encarga de prácticamente todo lo necesario para responder a la pregunta que vosotros le paséis en el cuerpo, de ahí que estemos usando el AI SDK para ayudaros. Fijaros bien en los parámetros que os pide y modificarlos acorde:

Request body required

Example Value | Schema

```
{
  "question": "string",
  "plot": false,
  "plot_details": "",
  "embeddings_provider": "googleaistudio",
  "embeddings_model": "models/text-embedding-004",
  "vector_store_provider": "Chroma",
  "sql_gen_provider": "googleaistudio",
  "sql_gen_model": "gemini-1.5-flash",
  "chat_provider": "googleaistudio",
  "chat_model": "gemini-1.5-flash",
  "vdp_database_names": "samples_bank",
  "use_views": "",
  "expand_set_views": true,
  "custom_instructions": "",
  "markdown_response": true,
  "vector_search_k": 5,
  "mode": "default",
  "disclaimer": true,
  "verbose": true
}
```

Todos os sirven excepto el **vdp_database_names** en donde tendréis que poner el nombre de la BBDD que hayáis creado anteriormente y el **question** en donde irá la pregunta de los usuarios. Además desde la URL tendréis este mismo endpoint pero en versión GET, para que podáis probar de manera interactiva el endpoint.



NOTE: *This document is confidential and proprietary of Denodo. No part of this document may be reproduced in any form by any means without prior written authorization of Denodo.*



Visit www.denodo.com Email info@denodo.com Discover community.denodo.com



Control name of the document