Docs

Hugging Face

Hugging Face is a leading platform in the field of natural language processing (NLP) that provides a comprehensive collection of pre-trained language models. Hugging Face facilitates easy access to a wide range of state-of-the-art models for various NLP tasks. Its focus on democratizing access to cutting-edge NLP capabilities has made Hugging Face a pivotal player in the advancement of language technology.

Using Hugging Face models

To employ Hugging Face LLMs, integrate the following dependency into your project:

```
<dependency>
    <groupId>io.quarkiverse.langchain4j</groupId>
    <artifactId>quarkus-langchain4j-hugging-face</artifactId>
    <version>0.14.1</version>
</dependency>
```

If no other LLM extension is installed, AI Services will automatically utilize the configured Hugging Face model.

IMPORTANT

Hugging Face provides multiple kind of models. We only support **text-to-text** models, which are models that take a text as input and return a text as output.

By default, the extension uses:

- tiiuae/falcon-7b-instruct as chat model (inference endpoint: *https://api-inference.huggingface.co/models/tiiuae/falcon-7b-instruct*)
- sentence-transformers/all-MiniLM-L6-v2 as embedding model (inference endpoint: *https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-MiniLM-L6-v2*)

Configuration

Configuring Hugging Face models mandates an API key, obtainable by creating an account on the Hugging Face platform.

The API key can be set in the `application.properties` file:

```
quarkus.langchain4j.huggingface.api-key=hf-...
```

TIP

Alternatively, leverage the `QUARKUS_LANGCHAIN4J_HUGGINGFACE_API_KEY` environment variable.

Several configuration properties are available:

🔒 Configuration property fixed at build time - All other configuration properties are overridable at runtime

Configuration property	Type	Default
🔒 `quarkus.langchain4j.huggingface.chat-model.enabled` Whether the model should be enabled Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_ENABLED`	boolean	`true`
🔒 `quarkus.langchain4j.huggingface.embedding-model.enabled` Whether the model should be enabled Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_EMBEDDING_MODEL_ENABLED`	boolean	`true`
🔒 `quarkus.langchain4j.huggingface.moderation-model.enabled` Whether the model should be enabled Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_MODERATION_MODEL_ENABLED`	boolean	`true`
`quarkus.langchain4j.huggingface.api-key` HuggingFace API key Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_API_KEY`	string	`dummy`
`quarkus.langchain4j.huggingface.timeout` Timeout for HuggingFace calls Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_TIMEOUT`	Duration	`10s`

`quarkus.langchain4j.huggingface.chat-model.inference-endpoint-url` The URL of the inference endpoint for the chat model. When using Hugging Face with the inference API, the URL is `https://api-inference.huggingface.co/models/<model-id>;` , for example `https://api-inference.huggingface.co/models/google/flan-t5-small` . When using a deployed inference endpoint, the URL is the URL of the endpoint. When using a local hugging face model, the URL is the URL of the local model. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_INFERENCE_ENDPOINT_URL`	URL	`https://api-inference.huggingface.co/models/tiiuae/falcon-7b-instruct`
`quarkus.langchain4j.huggingface.chat-model.temperature` Float (0.0-100.0). The temperature of the sampling operation. 1 means regular sampling, 0 means always take the highest score, 100.0 is getting closer to uniform probability Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_TEMPERATURE`	double	`1.0`
`quarkus.langchain4j.huggingface.chat-model.max-new-tokens` Int (0-250). The amount of new tokens to be generated, this does not include the input length it is a estimate of the size of generated text you want. Each new tokens slows down the request, so look for balance between response times and length of text generated Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_MAX_NEW_TOKENS`	int	
`quarkus.langchain4j.huggingface.chat-model.return-full-text` If set to `false` , the return results will not contain the original query making it easier for prompting Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_RETURN_FULL_TEXT`	boolean	
`quarkus.langchain4j.huggingface.chat-model.wait-for-model` If the model is not ready, wait for it instead of receiving 503. It limits the number of requests required to get your inference done. It is advised to only set this flag to true after receiving a 503 error as it will limit hanging in your application to known places Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_WAIT_FOR_MODEL`	boolean	`true`

`quarkus.langchain4j.huggingface.chat-model.do-sample` Whether or not to use sampling ; use greedy decoding otherwise. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_DO_SAMPLE`	boolean	
`quarkus.langchain4j.huggingface.chat-model.top-k` The number of highest probability vocabulary tokens to keep for top-k-filtering. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_TOP_K`	int	
`quarkus.langchain4j.huggingface.chat-model.top-p` If set to less than `1` , only the most probable tokens with probabilities that add up to `top_p` or higher are kept for generation. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_TOP_P`	double	
`quarkus.langchain4j.huggingface.chat-model.repetition-penalty` The parameter for repetition penalty. 1.0 means no penalty. See this paper for more details. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_REPETITION_PENALTY`	double	
`quarkus.langchain4j.huggingface.chat-model.log-requests` Whether chat model requests should be logged Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_LOG_REQUESTS`	boolean	`false`
`quarkus.langchain4j.huggingface.chat-model.log-responses` Whether chat model responses should be logged Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_CHAT_MODEL_LOG_RESPONSES`	boolean	`false`

`quarkus.langchain4j.huggingface.embedding-model.inference-endpoint-url` The URL of the inference endpoint for the embedding. When using Hugging Face with the inference API, the URL is `https://api-inference.huggingface.co/pipeline/feature-extraction/<model-id>;` , for example `https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-mpnet-base-v2` . When using a deployed inference endpoint, the URL is the URL of the endpoint. When using a local hugging face model, the URL is the URL of the local model. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_EMBEDDING_MODEL_INFERENCE_ENDPOINT_URL`	URL	`https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-MiniLM-L6-v2`
`quarkus.langchain4j.huggingface.embedding-model.wait-for-model` If the model is not ready, wait for it instead of receiving 503. It limits the number of requests required to get your inference done. It is advised to only set this flag to true after receiving a 503 error as it will limit hanging in your application to known places Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_EMBEDDING_MODEL_WAIT_FOR_MODEL`	boolean	`true`
`quarkus.langchain4j.huggingface.log-requests` Whether the HuggingFace client should log requests Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_LOG_REQUESTS`	boolean	`false`
`quarkus.langchain4j.huggingface.log-responses` Whether the HuggingFace client should log responses Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_LOG_RESPONSES`	boolean	`false`
`quarkus.langchain4j.huggingface.enable-integration` Whether or not to enable the integration. Defaults to `true` , which means requests are made to the OpenAI provider. Set to `false` to disable all requests. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE_ENABLE_INTEGRATION`	boolean	`true`
Named model config	Type	Default

`quarkus.langchain4j.huggingface."model-name".api-key` HuggingFace API key Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__API_KEY`	string	`dummy`
`quarkus.langchain4j.huggingface."model-name".timeout` Timeout for HuggingFace calls Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__TIMEOUT`	Duration	`10s`
`quarkus.langchain4j.huggingface."model-name".chat-model.inference-endpoint-url` The URL of the inference endpoint for the chat model. When using Hugging Face with the inference API, the URL is `https://api-inference.huggingface.co/models/<model-id>;` , for example `https://api-inference.huggingface.co/models/google/flan-t5-small` . When using a deployed inference endpoint, the URL is the URL of the endpoint. When using a local hugging face model, the URL is the URL of the local model. Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_INFERENCE_ENDPOINT_URL`	URL	`https://api-inference.huggingface.co/models/tiiuae/falcon-7b-instruct`
`quarkus.langchain4j.huggingface."model-name".chat-model.temperature` Float (0.0-100.0). The temperature of the sampling operation. 1 means regular sampling, 0 means always take the highest score, 100.0 is getting closer to uniform probability Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_TEMPERATURE`	double	`1.0`
`quarkus.langchain4j.huggingface."model-name".chat-model.max-new-tokens` Int (0-250). The amount of new tokens to be generated, this does not include the input length it is a estimate of the size of generated text you want. Each new tokens slows down the request, so look for balance between response times and length of text generated Environment variable: `QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_MAX_NEW_TOKENS`	int	

`quarkus.langchain4j.huggingface."model-name".chat-model.return-full-text`
 	
If set to `false`, the return results will not contain the original query making it easier for prompting
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_RETURN_FULL_TEXT`
	boolean	

`quarkus.langchain4j.huggingface."model-name".chat-model.wait-for-model`
 	
If the model is not ready, wait for it instead of receiving 503. It limits the number of requests required to get your inference done. It is advised to only set this flag to true after receiving a 503 error as it will limit hanging in your application to known places
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_WAIT_FOR_MODEL`
	boolean	`true`

`quarkus.langchain4j.huggingface."model-name".chat-model.do-sample`
 	
Whether or not to use sampling ; use greedy decoding otherwise.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_DO_SAMPLE`
	boolean	

`quarkus.langchain4j.huggingface."model-name".chat-model.top-k`
 	
The number of highest probability vocabulary tokens to keep for top-k-filtering.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_TOP_K`
	int	

`quarkus.langchain4j.huggingface."model-name".chat-model.top-p`
 	
If set to less than `1`, only the most probable tokens with probabilities that add up to `top_p` or higher are kept for generation.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_TOP_P`
	double	

`quarkus.langchain4j.huggingface."model-name".chat-model.repetition-penalty`
 	
The parameter for repetition penalty. 1.0 means no penalty. See this paper for more details.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_REPETITION_PENALTY`
	double	

`quarkus.langchain4j.huggingface."model-name".chat-model.log-requests`
 	
Whether chat model requests should be logged
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_LOG_REQUESTS`
	boolean	`false`

`quarkus.langchain4j.huggingface."model-name".chat-model.log-responses`
 	
Whether chat model responses should be logged
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__CHAT_MODEL_LOG_RESPONSES`
	boolean	`false`

`quarkus.langchain4j.huggingface."model-name".embedding-model.inference-endpoint-url`
 	
The URL of the inference endpoint for the embedding.
 	
When using Hugging Face with the inference API, the URL is `https://api-inference.huggingface.co/pipeline/feature-extraction/<model-id>;` , for example `https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-mpnet-base-v2` .
 	
When using a deployed inference endpoint, the URL is the URL of the endpoint. When using a local hugging face model, the URL is the URL of the local model.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__EMBEDDING_MODEL_INFERENCE_ENDPOINT_URL`
	URL	`https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-MiniLM-L6-v2`

`quarkus.langchain4j.huggingface."model-name".embedding-model.wait-for-model`
 	
If the model is not ready, wait for it instead of receiving 503. It limits the number of requests required to get your inference done. It is advised to only set this flag to true after receiving a 503 error as it will limit hanging in your application to known places
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__EMBEDDING_MODEL_WAIT_FOR_MODEL`
	boolean	`true`

`quarkus.langchain4j.huggingface."model-name".log-requests`
 	
Whether the HuggingFace client should log requests
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__LOG_REQUESTS`
	boolean	`false`

`quarkus.langchain4j.huggingface."model-name".log-responses`
 	
Whether the HuggingFace client should log responses
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__LOG_RESPONSES`
	boolean	`false`

`quarkus.langchain4j.huggingface."model-name".enable-integration`
 	
Whether or not to enable the integration. Defaults to `true`, which means requests are made to the OpenAI provider. Set to `false` to disable all requests.
 	
Environment variable:
`QUARKUS_LANGCHAIN4J_HUGGINGFACE__MODEL_NAME__ENABLE_INTEGRATION`
	boolean	`true`

NOTE

About the Duration format

To write duration values, use the standard `java.time.Duration` format. See the Duration#parse() Java API documentation for more information.

You can also use a simplified format, starting with a number:

- If the value is only a number, it represents time in seconds.
- If the value is a number followed by `ms`, it represents time in milliseconds.

In other cases, the simplified format is translated to the `java.time.Duration` format for parsing:

- If the value is a number followed by `h`, `m`, or `s`, it is prefixed with `PT`.
- If the value is a number followed by `d`, it is prefixed with `P`.

Configuring the chat model

You can change the chat model by setting the `quarkus.langchain4j.huggingface.chat-model.inference-endpoint-url` property. When using a model hosted on Hugging Face, the property should be set to: `https://api-inference.huggingface.co/models/<model-id>;`.

For example, to use the `google/flan-t5-small` model, set:

```
quarkus.langchain4j.huggingface.chat-model.inference-endpoint-url=https://api-inference.huggingface.co/models/google/flan-t5-small
```

Remember that only text to text models are supported.

Using inference endpoints and local models

Hugging Face models can be deployed to provide inference endpoints. In this case, configure the `quarkus.langchain4j.huggingface.inference-endpoint-url` property to point to the endpoint URL:

```
quarkus.langchain4j.huggingface.chat-model.inference-endpoint-url=https://j9dkyuliy170f3ia.us-east-1.aws.endpoints.huggingface.cloud
```

If you run a model locally, adapt the URL accordingly:

```
quarkus.langchain4j.huggingface.chat-model.inference-endpoint-url=http://localhost:8085
```

Document Retriever and Embedding

When utilizing Hugging Face models, the recommended practice involves leveraging the `EmbeddingModel` provided by Hugging Face.

1. If no other LLM extension is installed, retrieve the embedding model as follows:

```
@Inject EmbeddingModel model; // Injects the embedding model
```

You can configure the model using:

```
quarkus.langchain4j.huggingface.embedding-model.inference-endpoint-url=https://api-inference.huggingface.co/pipeline/feature-extraction/sentence-transformers/all-MiniLM-L6-v2
```

WARNING

Not every sentence transformers are supported by the embedding model. If you want to use a custom sentence transformers, you need to create your own embedding model.

Tools

The Hugging Face LLMs do not support tools.