I have deploy a model to hugging face. How can I use Hosted inference API in my model

[You can use the Hugging Face Hosted Inference API to test and evaluate over 80,000 publicly accessible machine learning models, or your own private models, via simple HTTP requests, with fast inference hosted on Hugging Face shared infrastructure1](https://huggingface.co/docs/api-inference/index). [You can easily integrate NLP, audio and computer vision models deployed for inference via simple API calls](https://huggingface.co/inference-api" \t "_blank)[2](https://huggingface.co/inference-api).

Here is an example of how you can use the API to query a model:

import requests

def query(payload, model\_id, api\_token):

headers = {"Authorization": f"Bearer {api\_token}"}

API\_URL = f"https://api-inference.huggingface.co/models/{model\_id}"

response = requests.post(API\_URL, headers=headers, json=payload)

return response.json()

model\_id = "distilbert-base-uncased"

api\_token = "hf\_XXXXXXXX" # get yours at hf.co/settings/tokens

data = query("The goal of life is [MASK].", model\_id, api\_token)

You can find more information about the Hugging Face Hosted Inference API in their [documentation](https://huggingface.co/docs/api-inference/index" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co](https://huggingface.co/docs/api-inference/index)[2. huggingface.co](https://huggingface.co/inference-api)[3. huggingface.co](https://huggingface.co/docs/api-inference/index)[4. huggingface.co](https://huggingface.co/docs/api-inference/quicktour)[5. huggingface.co](https://huggingface.co/docs/api-inference/main/en/_app/pages/index.mdx-hf-doc-builder.js)[6. api-inference.huggingface.co](https://api-inference.huggingface.co/models/)+2 more

the example you give me is for a client to access the hosted inference API of an existing model in huggin face. what I need is the oposite, I upload my model to hugging face and não I whant to activate the hosted inference API on the model I uploaded and then I can use the code you gave me to access it. how can I configure the hugging face hosted inference API on the model I upload?uploaded

To enable the Hosted Inference API for your uploaded model on Hugging Face, you need to specify the pipeline\_tag in your model card’s metadata.

The pipeline\_tag should correspond to the type of pipeline that your model should be used with.

[For example, if your model is a text classification model, you would set the pipeline\_tag to "text-classification"1](https://huggingface.co/docs/api-inference/index).

Here is an example of how you can specify the pipeline\_tag in your model card’s metadata:

---

model\_index:

- name: my-model

tags:

- text-classification

pipeline\_tag: text-classification

---

Copy

Once you have specified the pipeline\_tag, the Hosted Inference API should be automatically enabled for your model. [You can then use the API to query your model as I showed you in my previous message](https://huggingface.co/docs/api-inference/index" \t "_blank)[1](https://huggingface.co/docs/api-inference/index).

You can find more information about how to configure the Hosted Inference API for your uploaded model in the [Hugging Face documentation](https://huggingface.co/docs/api-inference/index" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/api-inference/index)[3. huggingface.co](https://huggingface.co/docs/hub/models-inference)[4. discuss.huggingface.co](https://discuss.huggingface.co/t/how-can-i-use-hosted-inference-api-in-my-model-card/38970)[5. huggingface.co](https://huggingface.co/docs/api-inference/main/en/_app/pages/index.mdx-hf-doc-builder.js)+3 more

2of30

To enable the Hosted Inference API for my uploaded model on Hugging Face, can I sepecify a custom pipeline in my model card’s metadata?

Searching for: hugging face custom pipeline

Searching for: hugging face custom pipeline

Yes, you can specify a custom pipeline for your uploaded model on Hugging Face by defining a pipeline\_tag in your model card’s metadata. The pipeline\_tag should correspond to the type of pipeline that your model should be used with. [For example, if you have created a custom pipeline for text classification, you would set the pipeline\_tag to "text-classification"1](https://huggingface.co/docs/transformers/add_new_pipeline). Here is an example of how you can specify the pipeline\_tag in your model card’s metadata:

---

model\_index:

- name: my-model

tags:

- text-classification

pipeline\_tag: text-classification

---

Copy

Once you have specified the pipeline\_tag, the Hosted Inference API should be automatically enabled for your model. [You can then use the API to query your model as I showed you in my previous message](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank)[1](https://huggingface.co/docs/transformers/add_new_pipeline).

You can find more information about how to configure the Hosted Inference API for your uploaded model in the [Hugging Face documentation](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[3. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[4. huggingface.co](https://huggingface.co/docs/diffusers/v0.6.0/en/using-diffusers/custom_pipelines)[5. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)+1 more

3of30

If I can specify a custom pipeline for my uploaded model on Hugging Face by defining a pipeline\_tag in your model card’s metadata, can you tell me how?

Yes, you can specify a custom pipeline for your uploaded model on Hugging Face by defining a pipeline\_tag in your model card’s metadata. The pipeline\_tag should correspond to the type of pipeline that your model should be used with. For example, if you have created a custom pipeline for text classification, you would set the pipeline\_tag to "text-classification". Here is an example of how you can specify the pipeline\_tag in your model card’s metadata:

---

model\_index:

- name: my-model

tags:

- text-classification

pipeline\_tag: text-classification

---

Copy

Once you have specified the pipeline\_tag, the Hosted Inference API should be automatically enabled for your model. You can then use the API to query your model as I showed you in my previous message.

You can find more information about how to configure the Hosted Inference API for your uploaded model in the [Hugging Face documentation]. Is there anything else you would like to know? 😊

4of30

Can you explain more about the pipeline\_tag and how it works?

Searching for: hugging face pipeline\_tag

Searching for: hugging face pipeline\_tag

The pipeline\_tag is a metadata field that you can specify in your model card when you upload a model to Hugging Face. It tells the Hugging Face Hosted Inference API which pipeline to use when running inference with your model. The pipeline\_tag should correspond to the type of pipeline that your model should be used with. For example, if your model is a text classification model, you would set the pipeline\_tag to "text-classification". [Once you have specified the pipeline\_tag, the Hosted Inference API should be automatically enabled for your model, and you can use the API to query your model](https://huggingface.co/docs/transformers/main_classes/pipelines" \t "_blank)[1](https://huggingface.co/docs/transformers/main_classes/pipelines).

You can find more information about how to configure the Hosted Inference API for your uploaded model in the [Hugging Face documentation](https://huggingface.co/docs/transformers/main_classes/pipelines" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[3. huggingface.co](https://huggingface.co/models?pipeline_tag=text-generation)[4. stackoverflow.com](https://stackoverflow.com/questions/63221913/named-entity-recognition-with-huggingface-transformers-mapping-back-to-complete)

5of30

can I create my own custom pipeline in hugging face

Searching for: create custom pipeline hugging face

Searching for: create custom pipeline hugging face

Yes, you can create your own custom pipeline in Hugging Face. [Here are the steps to create a custom pipeline and share it on the Hub or add it to the Transformers library](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank)[1](https://huggingface.co/docs/transformers/add_new_pipeline):

1. **Decide the raw entries**: The first step is to decide what kind of raw entries the pipeline will be able to take. [This could be a string, a list of strings, or any other data type that makes sense for your task](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
2. **Preprocess the inputs**: The next step is to preprocess these raw entries into inputs that can be fed into the model. [This often involves tokenization](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
3. [**Pass the inputs to the model**: Once you have preprocessed the inputs, you can pass them to the model1](https://huggingface.co/docs/transformers/add_new_pipeline).
4. [**Postprocess the model outputs**: After getting the outputs from the model, you will likely need to postprocess them into a format that is suitable for your task1](https://huggingface.co/docs/transformers/add_new_pipeline).
5. [**Wrap everything in a function**: Finally, you can wrap all of these steps in a function and use it as your custom pipeline1](https://huggingface.co/docs/transformers/add_new_pipeline).

You can find more detailed information on how to create a custom pipeline in the [Hugging Face documentation](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[3. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[4. huggingface.co](https://huggingface.co/docs/transformers/pipeline_tutorial)[5. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)+1 more

6of30

can I use the custom pipeline I created in hugging face pipeline\_tag?

Searching for: use custom pipeline in hugging face pipeline\_tag

Searching for: use custom pipeline in hugging face pipeline\_tag

Yes, you can use a custom pipeline that you have created in Hugging Face by specifying the pipeline\_tag in your model card’s metadata. The pipeline\_tag should correspond to the type of pipeline that your model should be used with. [Once you have specified the pipeline\_tag, the Hosted Inference API should be automatically enabled for your model, and you can use the API to query your model](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/main_classes/pipelines).

You can find more information about how to configure the Hosted Inference API for your uploaded model in the [Hugging Face documentation](https://huggingface.co/docs/transformers/main_classes/pipelines" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[3. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[4. huggingface.co](https://huggingface.co/docs/diffusers/v0.9.0/en/using-diffusers/custom_pipeline_overview)[5. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)+1 more

7of30

How can I upload my custom pipeline to Hugging Face?

Searching for: upload custom pipeline to Hugging Face

Searching for: upload custom pipeline to Hugging Face

You can create a custom pipeline and share it on the Hugging Face Hub or add it to the Transformers library. [Here are the steps to create a custom pipeline](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/add_new_pipeline):

1. **Decide the raw entries**: The first step is to decide what kind of raw entries the pipeline will be able to take. [This could be a string, a list of strings, or any other data type that makes sense for your task](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
2. **Preprocess the inputs**: The next step is to preprocess these raw entries into inputs that can be fed into the model. [This often involves tokenization](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
3. [**Pass the inputs to the model**: Once you have preprocessed the inputs, you can pass them to the model](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
4. [**Postprocess the model outputs**: After getting the outputs from the model, you will likely need to postprocess them into a format that is suitable for your task](https://edgeservices.bing.com/edgesvc/chat?udsframed=1&form=SHORUN&clientscopes=chat,noheader,channelstable,&shellsig=726b226971a48161237102bedfe4c87f7379d808&setlang=en-US&darkschemeovr=1)[1](https://huggingface.co/docs/transformers/add_new_pipeline).
5. [**Wrap everything in a function**: Finally, you can wrap all of these steps in a function and use it as your custom pipeline1](https://huggingface.co/docs/transformers/add_new_pipeline).

You can find more detailed information on how to create and share a custom pipeline in the [Hugging Face documentation](https://huggingface.co/docs/transformers/add_new_pipeline" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[3. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[4. huggingface.co](https://huggingface.co/docs/diffusers/v0.7.0/en/using-diffusers/custom_pipeline_overview)[5. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)+1 more

8of30

Can you show me an example of a custom pipeline in action?

Searching for: example of a custom pipeline in action hugging face

Searching for: example of a custom pipeline in action hugging face

Here is an example of a custom pipeline in action, using the Hugging Face Transformers library. This example shows how to create a custom pipeline for text classification, using a pre-trained model and tokenizer:

**Python**

This code is AI-generated. Review and use carefully. [Visit our FAQ](https://www.bing.com/new" \l "faq" \t "_blank) for more information.

Copy

from transformers import AutoModelForSequenceClassification, AutoTokenizer, Pipeline

# Load the pre-trained model and tokenizer

model = AutoModelForSequenceClassification.from\_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

tokenizer = AutoTokenizer.from\_pretrained("distilbert-base-uncased-finetuned-sst-2-english")

# Define the custom pipeline

def custom\_pipeline(text):

# Tokenize the input text

inputs = tokenizer(text, return\_tensors="pt")

# Run the model on the tokenized input

outputs = model(\*\*inputs)

# Get the predicted label

label = outputs.logits.argmax(dim=-1).item()

# Return the predicted label

return "positive" if label == 1 else "negative"

# Create a Pipeline object using the custom pipeline function

pipe = Pipeline(model=model, tokenizer=tokenizer, function=custom\_pipeline)

# Test the custom pipeline

print(pipe("This is a great movie!")) # "positive"

print(pipe("This is a terrible movie!")) # "negative"

This example shows how to create a custom pipeline for text classification, using a pre-trained model and tokenizer. The custom pipeline function takes in a text input, tokenizes it using the pre-trained tokenizer, runs the pre-trained model on the tokenized input, and returns the predicted label. The Pipeline object is then created using the custom pipeline function, and can be used to run inference on new inputs.

I hope this example helps you understand how to create and use a custom pipeline in Hugging Face. Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[2. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[3. huggingface.co](https://huggingface.co/docs/transformers/pipeline_tutorial)[4. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)

9of30

where do I save my custom pipeline in hugging face?

Searching for: save custom pipeline in hugging face

Searching for: save custom pipeline in hugging face

To add a custom pipeline to the Hugging Face Hub, you need to define a pipeline class that inherits from DiffusionPipeline in a pipeline.py file. [Make sure that the whole pipeline is encapsulated within a single class and that the pipeline.py file has only one such class](https://huggingface.co/docs/diffusers/v0.7.0/en/using-diffusers/custom_pipeline_overview" \t "_blank)[1](https://huggingface.co/docs/diffusers/v0.7.0/en/using-diffusers/custom_pipeline_overview). Here is an example of how you can define an example pipeline:

**Python**

This code is AI-generated. Review and use carefully. [Visit our FAQ](https://www.bing.com/new" \l "faq" \t "_blank) for more information.

Copy

import torch

from diffusers import DiffusionPipeline

class MyPipeline(DiffusionPipeline):

def \_\_init\_\_(self, unet, scheduler):

super().\_\_init\_\_()

self.register\_modules(unet=unet, scheduler=scheduler)

@torch.no\_grad()

def \_\_call\_\_(self, batch\_size: int = 1, num\_inference\_steps: int = 50):

# Sample gaussian noise to begin loop

image = torch.randn((batch\_size, self.unet.in\_channels, self.unet.sample\_size, self.unet.sample\_size))

image = image.to(self.device)

# set step values

self.scheduler.set\_timesteps(num\_inference\_steps)

for t in self.progress\_bar(self.scheduler.timesteps):

# 1. predict noise model\_output

model\_output = self.unet(image, t).sample

# 2. predict previous mean of image x\_t-1 and add variance depending on eta

# eta corresponds to η in paper and should be between [0, 1]

# do x\_t -> x\_t-1

image = self.scheduler.step(model\_output, t, image, eta).prev\_sample

image = (image / 2 + 0.5).clamp(0, 1)

image = image.cpu().permute(0, 2, 3, 1).numpy()

return image

You can find more detailed information on how to create and share a custom pipeline in the [Hugging Face documentation](https://huggingface.co/docs/diffusers/v0.7.0/en/using-diffusers/custom_pipeline_overview" \t "_blank). Is there anything else you would like to know? 😊

Learn more:

[1. huggingface.co2. huggingface.co](https://huggingface.co/docs/diffusers/v0.7.0/en/using-diffusers/custom_pipeline_overview)[3. huggingface.co](https://huggingface.co/docs/transformers/add_new_pipeline)[4. huggingface.co](https://huggingface.co/docs/transformers/main_classes/pipelines)[5. huggingface.co](https://huggingface.co/docs/transformers/v4.32.1/en/_app/pages/add_new_pipeline.mdx-hf-doc-builder.js)+1 more