# BentoML Quickstart: Text Summarization with Hugging Face

This guide walks you through building a text summarization application with BentoML and a pre-trained Transformer model from the Hugging Face Model Hub.

## 1. Prerequisites

• Python 3.9 or later with pip installed. (Download Python: https://www.python.org/downloads/)  
• (Optional) A virtual environment for dependency isolation. Refer to Conda documentation (https://docs.conda.io/docs/using/pkgs.html) or Python documentation (https://docs.python.org/3/tutorial/venv.html) for details.

## 2. Setting Up the Environment

Clone the Repository:

git clone https://github.com/bentoml/quickstart.git  
cd quickstart

Install Dependencies:

pip install -r requirements.txt

## 3. Creating a BentoML Service

Understanding service.py:  
  
The service.py file defines a BentoML Service class named Summarization. This file resides in the project directory you cloned. Let's break down the code:

Imports:  
• bentoml library for building BentoML services.  
• pipeline function from transformers for text summarization.  
• EXAMPLE\_INPUT is a sample text for demonstration.

Summarization Class:  
• Decorated with @bentoml.service to define it as a service.  
• resources argument specifies CPU allocation ('2' in this case).  
• traffic argument defines timeout duration (10 seconds).  
• The constructor (\_\_init\_\_) retrieves the pre-trained model and initializes a pipeline for summarization.  
• @bentoml.api decorator marks the summarize method as an API endpoint. It accepts a string input (text) with a default value set to EXAMPLE\_INPUT, processes it with the pipeline, and returns the summarized text.

## 4. Running the Service Locally

Start the BentoML Server:

bentoml serve service:Summarization

This command starts a local BentoML server at http://localhost:3000.

## 5. Interacting with the Service

There are multiple ways to interact with your service:  
  
Using CURL:

curl -X 'POST' \  
 'http://localhost:3000/summarize' \  
 -H 'accept: text/plain' \  
 -H 'Content-Type: application/json' \  
 -d '{ "text": "Your text to summarize here..." }'

Using Python Client:  
Refer to BentoML documentation for details on creating a Python client for your service.  
  
Using Swagger UI:  
Access the Swagger UI at http://localhost:3000/openapi.json to explore the API documentation and interact with the service visually.

## 6. Expected Output

The service should return a summarized version of the provided text.

## 7. Next Steps

Once you've successfully run the service locally, you can explore further options:  
  
• Deploying to BentoCloud: The BentoML platform allows you to deploy your service to a cloud environment for wider access. Refer to BentoML documentation for details.  
• Creating a Docker Image: Package your service as a Docker image for deployment on your preferred infrastructure.  
  
This guide provides a basic understanding of building and running a text summarization application with BentoML. Explore the BentoML documentation for further information on advanced features and functionalities.