### **Documentation: Text-to-3D and Image-to-3D Generation Scripts**

#### **Overview**

The provided Python scripts utilize the **Meshy API** to generate 3D models based on text descriptions or images. These scripts automate the process of creating tasks, polling for task status, and downloading the resulting 3D models. Each script includes clear steps for inputting data, interacting with the API, and retrieving outputs in a professional and user-friendly manner.

### **Text-to-3D Generation Script**

#### **Purpose**

This script converts textual descriptions into 3D models using the Meshy API. It supports two modes:

1. **Preview Mode**: Generates a basic version of the 3D model for review without any textures and details.
2. **Refine Mode**: Enhances the preview model with adding textures and finer details.

#### **Key Functions**

1. **generate\_task**
   * Creates a task in either preview or refine mode based on user input.
   * Inputs: Prompt, API key, mode (preview or refine), and additional settings (e.g., remesh, texture richness, etc.).
   * Output: Task ID of the created task.
2. **fetch\_task\_status**
   * Polls the API to check the status of the generated task.
   * Inputs: Task ID, API key, retry interval, and maximum retries.
   * Output: Links to the generated 3D model or an error message if the task fails.
3. **save\_model\_file**
   * Downloads the resulting 3D model and saves it as a .glb file.
   * Inputs: Model file URL and the desired output file path.
4. **Main Workflow**
   * The user provides a textual description (prompt) and API key.
   * A preview task is created, and its status is monitored until completion.
   * If successful, the preview model is downloaded.
   * The refine mode is triggered to generate a high-quality model, which is then downloaded.

### 

### **Image-to-3D Generation Script**

#### **Purpose**

This script transforms an input image into a 3D model using the Meshy API. It supports options like enabling PBR, remeshing, and applying textures.

#### **Key Functions**

1. **generate\_image\_to\_3d\_task**
   * Creates a task to generate a 3D model from an image.
   * Inputs: Image path, API key, and optional settings (e.g., PBR, remesh, texture application).
   * Output: Task ID for the generated model.
2. **fetch\_task\_status\_image\_to\_3d**
   * Monitors the task status until completion or failure.
   * Inputs: Task ID, API key, retry interval, and maximum retries.
   * Output: Links to the generated 3D model or an error message if the task fails.
3. **save\_model\_file**
   * Downloads the generated 3D model from a provided URL.
   * Inputs: Model file URL and the desired output file path.
4. **Main Workflow**
   * The user provides the path to an input image and an API key.
   * The script encodes the image into a base64 format and sends it to the Meshy API.
   * It polls for task status until completion.
   * Upon success, the resulting 3D model is downloaded and saved locally.

### **Features**

1. **Ease of Use**
   * Both scripts prompt the user for inputs like API keys, text prompts, or image paths.
   * Automated polling ensures the user is updated on task progress.
2. **Customization**
   * Options like enabling PBR, remeshing, and texture richness allow for tailored 3D model generation.
3. **Error Handling**
   * Comprehensive error handling ensures users are informed of issues, such as API failures or task errors.
4. **Output**
   * Models are saved locally in .glb format, ready for use in 3D applications.

### 

### 

### **Instructions for Use**

First Install python then Install the required Python libraries using below command:  
pip install requests

1. For Text-to-3D:
   * Run the script and provide a descriptive text prompt and API key.
   * Monitor the preview and refine task progress.
   * Locate the downloaded model in the current directory.
2. For Image-to-3D:
   * Run the script and provide the path to an input image and API key.
   * Wait for the task to complete and download the model.

### **Notes**

* Ensure your API key is valid and has sufficient permissions to use the Meshy API.
* Input images for the Image-to-3D script should be in .png format for proper encoding.
* The scripts handle both basic and advanced 3D modeling needs with customizable parameters.