

```
import requests

from loguru import logger

import os

from typing import Optional

from urllib.parse import urlparse

import time
```

```
def download_image(

    url: str, save_dir: str, retries: int = 3, timeout: int = 10

) -> Optional[str]:

    """

    Downloads an image from a given URL and saves it to the specified directory.

    :param url: URL of the image to download.

    :param save_dir: Directory where the image will be saved.

    :param retries: Number of retries in case of failure (default: 3).

    :param timeout: Timeout for the request in seconds (default: 10).

    :return: Path to the saved image or None if the download fails.

    """

    logger.info(f"Starting download of image from {url}")

    # Parse the image file name from the URL

    parsed_url = urlparse(url)

    image_name = os.path.basename(parsed_url.path)
```

```
if not image_name:

    logger.error(f"Could not parse image name from URL: {url}")

    return None


# Ensure the save directory exists

os.makedirs(save_dir, exist_ok=True)

save_path = os.path.join(save_dir, image_name)


attempt = 0

while attempt < retries:

    try:

        # Attempt to download the image

        logger.info(

            f"Attempt {attempt + 1} to download {image_name} from {url}"

        )

        response = requests.get(url, timeout=timeout)


        # Check if the request was successful

        if response.status_code == 200:

            # Save the image to the specified path

            with open(save_path, "wb") as f:

                f.write(response.content)

            logger.info(

                f"Successfully downloaded and saved image to {save_path}"

            )

            return save_path
```

else:

```
    logger.error(  
        f"Failed to download image. Status code: {response.status_code}"  
    )
```

except requests.exceptions.RequestException as e:

Handle request errors and log them

```
    logger.error(f"Error downloading image from {url}: {e}")
```

Wait before retrying if not successful

attempt += 1

if attempt < retries:

```
    logger.warning(  
        f"Retrying download in 3 seconds... ({attempt}/{retries})"  
    )  
  
    time.sleep(3)
```

```
logger.error(  
    f"Failed to download image from {url} after {retries} attempts"  
)  
  
return None
```

Example usage

if __name__ == "__main__":

image_url = "https://example.com/path/to/image.jpg"

```
# save_directory = "downloaded_images"

# download_image(image_url, save_directory)
```