This project provides a File Integrity Check Service, a simple API that calculates cryptographic hash values for uploaded files. The service is containerized using Docker for easy deployment.

Table of Contents

Overview

The File Integrity Check Service within the Sony-Systems-Project allows users to upload files and receive their corresponding hash values (e.g., MD5, SHA-1, SHA-256). This functionality is useful for verifying file integrity, ensuring that files have not been tampered with or corrupted during transmission.

Deployment Instructions

Prerequisites

- **Docker**: Ensure that Docker is installed on your system. You can download Docker from https://www.docker.com/products/docker-desktop/.

Building the Docker Image

1. **Clone the Repository**:

   ```
   git clone https://github.com/Olisaemeka111/sony-systems-project.git
   cd sony-systems-project
   ```

2. **Build the Docker Image**:

   Execute the following command in the root directory of the project (where the `Dockerfile` is located):

   ```
   Bash


   docker build -f Dockerfile -t myimage .   #to build the docker image of the application
   ```

Running the Docker Container

1. **Start the Container**:

```
bash


docker images                          ##to list all the images already built

docker run -p 8010:5000 <IMAGE ID>     ##to start and run the container with the image
Id
```

- `-d`: Runs the container in detached mode (in the background).
- `-p 8010:5000`: Maps port 5000 of the host machine to port 5000 of the container.

2. **Verify the Service is Running**:

   Open a web browser and navigate to `http://localhost:8010`. You should see the homepage or API documentation of the service.

User Interaction Guide

Users can interact with the service either through command-line tools like cURL or via a simple HTML frontend.

Using cURL

1. **Upload a File for Integrity Check**:

   Replace `path_to_your_file` with the actual path of the file you want to check.

   ```
   bash
   Copy code
   curl -X POST -F 'file=@path_to_your_file' http://localhost:5000/check
   ```

   Response:

   ```
   json

   {
     "filename": "example.txt",
     "md5": "1a79a4d60de6718e8e5b326e338ae533",

   }
   ```

2. **Interpret the Response**:

   The response provides the filename and its corresponding hash values. Users can use these hashes to verify the integrity of the file or compare it against known values.

Using the HTML Frontend

1. **Access the Frontend**:

   Navigate to `http://localhost:8010` in your web browser.

2. **Upload a File**:

   Use the provided interface to select and upload a file.

3. **View Results**:

   After uploading, the page will display the hash values of the uploaded file.

Additional Notes

Assumptions

- The service assumes that uploaded files are not malicious. Users should ensure they upload safe files.
- The service is designed for small to medium-sized files. Extremely large files might lead to performance issues.

Limitations

- **Load Handling**: The service is intended for light load and simple use cases. It may not perform optimally under heavy traffic or with concurrent large file uploads.
- **Security**: While basic security measures are in place, for production environments, additional security layers (like authentication, HTTPS, etc.) should be considered.
- **File Storage**: Uploaded files are processed in-memory and not stored persistently. If persistent storage is required, modifications to the service will be necessary.