Project_A1 documentation

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

Main Module Documentation

File Synchronization Project

This script provides functionality for synchronizing files between different storage locations, including FTP servers, zip archives, and local folders. It supports automatic synchronization of added, deleted, and modified files, as well as periodic monitoring of FTP and folder changes.

Modules: - FTPLocation: Interacts with an FTP server for file operations. - ZipLocation: Interacts with zip archives for file operations. - FolderLocation: Interacts with local file systems for file operations. - sync_files: Synchronizes files between two locations. - monitor_ftp_changes: Periodically monitors an FTP server for changes and syncs them. - monitor_folder: Monitors local folders or FTP servers for changes. - create_location: Creates the appropriate location object based on a given string. - calculate_checksum: Calculates the MD5 checksum of a file to detect modifications. - MyHandler: A custom handler for monitoring file system events.

Dependencies: - ftplib: For FTP interaction. - zipfile: For zip file handling. - os: For file and directory manipulation. - hashlib: For checksum calculation. - watchdog: For monitoring file system events.

class main.**FTPLocation**(*ftp_url*, *username*, *password*, *folder_path=None*)

Bases: **Location**

Represents an FTP server location for file synchronization.

Parameters:
- **ftp_url** (*str*) – The FTP server URL.
- **username** (*str*) – The username for FTP login.
- **password** (*str*) – The password for FTP login.
- **folder_path** (*str, optional*) – The path to a specific folder on the FTP server.

list_files()

Lists files on the FTP server.

read_file()

Reads a file from the FTP server.

write_file()

Writes a file to the FTP server.

delete_file()

Deletes a file from the FTP server.

get_file_mod_time()

Gets the modification time of a file on the FTP server.

delete_file(*file_name*)

Deletes a file from the FTP server.

Parameters: **file_name** (*str*) – The name of the file to delete.

get_file_mod_time(*file_name*)

Gets the modification time of a file on the FTP server.

Parameters: **file_name** (*str*) – The name of the file.
Returns: The modification time as a Unix timestamp.
Return type: float

`list_files()`
Lists all files in the current directory on the FTP server.

Returns: A list of filenames in the current directory on the FTP server.
Return type: list

`read_file(file_name)`
Reads the content of a file from the FTP server.

Parameters: **file_name** (*str*) – The name of the file to read.
Returns: The content of the file.
Return type: bytes

`write_file(file_name, content)`
Writes content to a file on the FTP server.

Parameters:
- **file_name** (*str*) – The name of the file to write to.
- **content** (*bytes*) – The content to write to the file.

class main.**FolderLocation**(*folder_path*)
Bases: **Location**

Represents a local folder location for file synchronization.

Parameters: **folder_path** (*str*) – The path to the folder.

`list_files()`
Lists files in the local folder.

`read_file()`
Reads a file from the local folder.

`write_file()`
Writes content to a file in the local folder.

`delete_file()`
Deletes a file from the local folder.

`get_file_mod_time()`
Gets the modification time of a file in the local folder.

`delete_file(file_name)`
Deletes a file from the local folder.

Parameters: **file_name** (*str*) – The name of the file to delete.

`get_file_mod_time(file_name)`
Gets the modification time of a file in the local folder.

Parameters: **file_name** (*str*) – The name of the file.

Returns: The modification time as a Unix timestamp.
Return type: float

`list_files()`
Lists all files in the local folder.

Returns: A list of filenames in the local folder.
Return type: list

`read_file(file_name)`
Reads the content of a file from the local folder.

Parameters: **file_name** (*str*) – The name of the file to read.
Returns: The content of the file.
Return type: bytes

`write_file(file_name, content)`
Writes content to a file in the local folder.

Parameters:
- **file_name** (*str*) – The name of the file to write to.
- **content** (*bytes*) – The content to write to the file.

class main.**Location**
Bases: **ABC**

Abstract base class for different storage locations (FTP, Zip, Folder).

`list_files()`
Lists all files in the location.

`read_file()`
Reads a file's content.

`write_file()`
Writes content to a file.

`delete_file()`
Deletes a file.

`get_file_mod_time()`
Returns the modification time of a file.

abstract `delete_file(file_name)`
Deletes a file.

Parameters: **file_name** (*str*) – The name of the file to delete.

abstract `get_file_mod_time(file_name)`
Gets the modification time of a file.

Parameters: **file_name** (*str*) – The name of the file.
Returns: The modification time as a Unix timestamp.
Return type: float

abstract `list_files()`
Lists all files in the storage location.

Returns: List of filenames in the location.

abstract **read_file**(*file_name*)

Reads the content of a file.

Parameters: **file_name** (*str*) – The name of the file to read.
Returns: The content of the file.
Return type: bytes

abstract **write_file**(*file_name, content*)

Writes content to a file.

Parameters:
- **file_name** (*str*) – The name of the file to write to.
- **content** (*bytes*) – The content to write to the file.

class main.**MyHandler**(*loc1, loc2*)

Bases: **FileSystemEventHandler**

Custom event handler for monitoring file system changes.

Parameters:
- **loc1** (*Location*) – The first location to synchronize.
- **loc2** (*Location*) – The second location to synchronize.

on_created()

Handles file creation events.

on_deleted()

Handles file deletion events.

on_modified()

Handles file modification events.

on_created(*event*)

Handles file creation events.

Parameters: **event** (*FileSystemEvent*) – The event object containing event details.

on_deleted(*event*)

Handles file deletion events.

Parameters: **event** (*FileSystemEvent*) – The event object containing event details.

on_modified(*event*)

Handles file modification events.

Parameters: **event** (*FileSystemEvent*) – The event object containing event details.

class main.**ZipLocation**(*zip_path*)

Bases: **Location**

Represents a zip archive location for file synchronization.

Parameters: **zip_path** (*str*) – The path to the zip archive.

list_files()

Lists files in the zip archive.

`read_file()`
Reads a file from the zip archive.

`write_file()`
Writes content to a file in the zip archive.

`delete_file()`
Deletes a file from the zip archive.

`get_file_mod_time()`
Gets the modification time of a file in the zip archive.

`delete_file(file_name)`
Deletes a file from the zip archive.

Parameters: **file_name** (*str*) – The name of the file to delete.

`get_file_mod_time(file_name)`
Gets the modification time of a file in the zip archive.

Parameters: **file_name** (*str*) – The name of the file.
Returns: The modification time as a Unix timestamp.
Return type: float

`list_files()`
Lists all files in the zip archive.

Returns: A list of filenames in the zip archive.
Return type: list

`read_file(file_name)`
Reads the content of a file from the zip archive.

Parameters: **file_name** (*str*) – The name of the file to read.
Returns: The content of the file.
Return type: bytes

`write_file(file_name, content)`
Writes content to a file in the zip archive.

Parameters:
- **file_name** (*str*) – The name of the file to write to.
- **content** (*bytes*) – The content to write to the file.

`main.calculate_checksum(file_content)`
Calculates the MD5 checksum of a file content.

Parameters: **file_content** (*bytes*) – The content of the file.
Returns: The MD5 checksum of the file content.
Return type: str

`main.create_location(location_str)`
Creates a location object based on a string representation.

Parameters: **location_str** (*str*) – The location string (e.g., "ftp://...").
Returns: The corresponding location object (FTPLocation, ZipLocation, or FolderLocation).
Return type: Location

Raises: **ValueError** – If the location string is not valid.

main.**monitor_folder**(*loc1, loc2*)

Monitors changes in local folders or FTP servers and synchronizes them.

Parameters:
- **loc1** (*Location*) – The first location to monitor.
- **loc2** (*Location*) – The second location to monitor.

main.**monitor_ftp_changes**(*ftp_loc, other_loc, poll_interval=10*)

Monitors FTP for changes and synchronizes with the other location.

Parameters:
- **ftp_loc** (*FTPLocation*) – The FTP location to monitor.
- **other_loc** (*Location*) – The location to synchronize with.
- **poll_interval** (*int*) – The polling interval in seconds.

main.**sync_files**(*loc1, loc2*)

Synchronizes files between two locations.

Parameters:
- **loc1** (*Location*) – The first storage location.
- **loc2** (*Location*) – The second storage location.