Social Media Downloader Documentation

Introduction

The Social Media Downloader is a Python-based tool designed to download media content from popular social media platforms like YouTube, TikTok, Instagram, and Facebook. It supports features such as batch downloads, format selection, logging, and update checks, ensuring a smooth and user-friendly experience.

This document provides a detailed explanation of the source code, tools required for the project, and setup instructions. The goal is to help you understand how the project works and how you can modify or extend it.

Tools Required

1. **Python 3.8+**: The programming language used for this project.
2. **pip**: Python package manager to install dependencies.
3. **VS Code**: A lightweight editor used for writing and testing code.
    - Extensions: Python extension for debugging and IntelliSense.
4. **PyCharm**: An alternative IDE offering advanced features like code inspection and profiling.
5. **Libraries**:
    - `yt_dlp` : Handles YouTube and TikTok video downloads.
    - `instaloader` : For downloading Instagram posts.
    - `requests` : For handling HTTP requests.
    - `BeautifulSoup` : For parsing HTML content (used for Facebook video downloads).
    - `tqdm` : For progress bars.
    - `logging` : For recording events and debugging information.

Project Structure and Line-by-Line Explanation

Version and Update Variables

```
CURRENT_VERSION = "1.0.0"
UPDATE_URL = "https://api.github.com/repos/nayandas69/Social-Media-Downloader/releases/latest"
WHATS_NEW_FILE = "whats_new.txt"
```

- **Purpose**: Defines the current version and URLs for updates. This ensures users can easily check for and download the latest version of the software.
- **Why Needed**: Keeps the software up-to-date with bug fixes and new features.

Logging Setup

```
logging.basicConfig(
    filename='downloader.log',
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s'
)
```

- **Purpose**: Configures the logging system to store logs in a file with a specific format.
- **Why Needed**: Helps in debugging by recording all major events and errors.

Configuration File Setup

```
config_file = 'config.json'
default_config = {
    "default_format": "show_all",
    "download_directory": "media",
    "history_file": "download_history.csv"
}
```

- **Purpose**: Defines the default configuration. If the `config.json` file does not exist, it will be created with these default values.
- **Why Needed**: Provides flexibility to users to customize settings without modifying the source code.

Loading Configuration

```
def load_config():
    if not os.path.exists(config_file):
        with open(config_file, 'w') as f:
            json.dump(default_config, f, indent=4)
    with open(config_file, 'r') as f:
        return json.load(f)

config = load_config()
download_directory = config['download_directory']
history_file = config['history_file']
```

- **Purpose**: Loads the configuration from `config.json`. If the file does not exist, it creates one.
- **Why Needed**: Centralizes settings for better maintainability.

Author Details

```
def display_author_details():
    print("=" * 50)
    print("Social Media Downloader")
    print("Author: Nayan Das")
    print("Email: nayanchandradas@hotmail.com")
    print("Website: https://socialportal.nayanchandradas.com")
    print("Version: " + CURRENT_VERSION)
    print("=" * 50)
    time.sleep(1)

display_author_details()
```


- **Purpose**: Displays the author's details and project version.
- **Why Needed**: Adds professionalism and provides contact information for support.

Helper Functions

Operating System Detection

```
def is_windows():
    return sys.platform.startswith('win')

def is_linux():
    return sys.platform.startswith('linux')
```

- **Purpose**: Determines the operating system.
- **Why Needed**: Helps in OS-specific operations, such as file handling or update downloads.

Logging Downloads

```
def log_download(url, status, timestamp=None):
    if not timestamp:
        timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    with open(history_file, 'a', newline='') as f:
        writer = csv.writer(f)
        writer.writerow([url, status, timestamp])
    logging.info(f"Download status for {url}: {status}")
```

- **Purpose**: Records the download status in a history file and logs it.
- **Why Needed**: Keeps track of download history for auditing or debugging purposes.

Progress Bar

```
def progress_bar(iterable, description="Processing"):
    return tqdm(iterable, desc=description, ncols=100, leave=False)
```

- **Purpose**: Wraps an iterable with a progress bar.
- **Why Needed**: Improves user experience by showing the download progress.

Check for Updates

```
def check_for_updates():
    print(f"Current version: {CURRENT_VERSION}")
    print("Checking for updates...")
    try:
        response = requests.get(UPDATE_URL)
        response.raise_for_status()
        data = response.json()
        ...
```

- **Purpose**: Checks GitHub for newer versions and prompts the user to update.
- **Why Needed**: Keeps the software current and secure.

Media Download Functions

1. **YouTube/TikTok**

```
def download_youtube_or_tiktok_video(url):
    ydl_opts = {'listformats': True}
    with yt_dlp.YoutubeDL(ydl_opts) as ydl:
        ...
```

- **Purpose**: Downloads videos with format selection or as MP3.
- **Why Needed**: Supports multiple formats to cater to user preferences.

2. **Instagram**

```
def download_instagram_post(url):
    L = instaloader.Instaloader()
    shortcode = url.split("/")[-2]
    post = instaloader.Post.from_shortcode(L.context, shortcode)
    L.download_post(post, target=download_directory)
    ...
```

- **Purpose**: Downloads Instagram posts using `instaloader`.
- **Why Needed**: Automates the tedious process of manual downloads.

3. **Facebook**

```
def download_facebook_video(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')
    video_url = soup.find('meta', property="og:video")['content']
    ...
```

- **Purpose**: Extracts and downloads videos from Facebook.
- **Why Needed**: Handles platforms that don't provide direct download options.

Main CLI Menu

```
def main():
    print("Welcome to Social Media Downloader!")
    while True:
        print("\nAvailable Options:")
        print("1. Download YouTube/TikTok Video")
        ...
        choice = input("Enter your choice: ").strip().lower()
        if choice == "1":
            ...
```

- **Purpose**: Provides an interactive interface for users to access all features.
- **Why Needed**: Simplifies usage by offering a guided menu system.

How to Use

1. Install the required Python libraries using `pip install -r requirements.txt`.
2. Run the script using `python downloader.py`.
3. Follow the CLI menu to download media or check for updates.

Welcome to Contribute

The **Social Media Downloader** project is open-source, and contributions are welcome! Whether you are a seasoned developer or a beginner, you can help improve the tool by adding new features, fixing bugs, or improving the documentation. Here's how you can contribute:

Contribution Guidelines

1. **Fork the Repository**
   Visit the GitHub repository:
   https://github.com/nayandas69/Social-Media-Downloader.git
   - Click the **Fork** button in the top-right corner to create a copy of the repository in your GitHub account.
2. **Clone Your Fork**
   Open a terminal or command prompt and run the following command to clone the repository to your local machine:

   ```
   git clone https://github.com/your-username/Social-Media-Downloader.git
   cd Social-Media-Downloader
   ```

3. **Create a New Branch**
   Create a new branch for your contribution:

   ```
   git checkout -b feature-or-bugfix-name
   ```

4. **Make Your Changes**
   - Add or improve features.
   - Fix bugs or issues.
   - Enhance the documentation or add comments to the code.
   - Follow the project's coding style and structure.
5. **Test Your Changes**
   Ensure your changes work as expected and do not break existing functionality.
6. **Commit Your Changes**
   Commit your changes with a meaningful commit message:

   ```
   git add .
   git commit -m "Describe your changes here"
   ```

7. **Push Your Changes**
   Push the changes to your forked repository:

   ```
   git push origin feature-or-bugfix-name
   ```

8. **Create a Pull Request**
   - Go to the original repository: https://github.com/nayandas69/Social-Media-Downloader.git.
   - Click the **Pull Requests** tab and then the **New Pull Request** button.
   - Select your branch and submit a detailed description of your changes.

Areas Where You Can Contribute

1. **Bug Fixes**
   Identify and resolve issues in the code. Check the repository's **Issues** section for known problems.
2. **Feature Enhancements**
   - Add support for more social media platforms.
   - Improve the user interface of the CLI.
   - Implement a GUI version for users who prefer visual interfaces.
3. **Documentation**
   - Expand this guide for better clarity.
   - Add more examples or FAQs to help users.
4. **Testing**

- Write unit tests to ensure code reliability.
- Perform cross-platform testing on different operating systems.

Thank You for Contributing!

Your contributions make this project better. Feel free to ask questions or suggest ideas via the **Issues** section in the GitHub repository or by contacting the author.

Conclusion

This project showcases how Python can automate media downloads efficiently. With clear code structure and modular design, it is easy to extend or adapt for other platforms. Use this documentation to understand and enhance the functionality further.

Contact ME

- Author: Nayan Das
- Email: mailto:nayanchandradas@hotmail.com
- Website: https://socialportal.nayanchandradas.com