Python Sample Code Overview

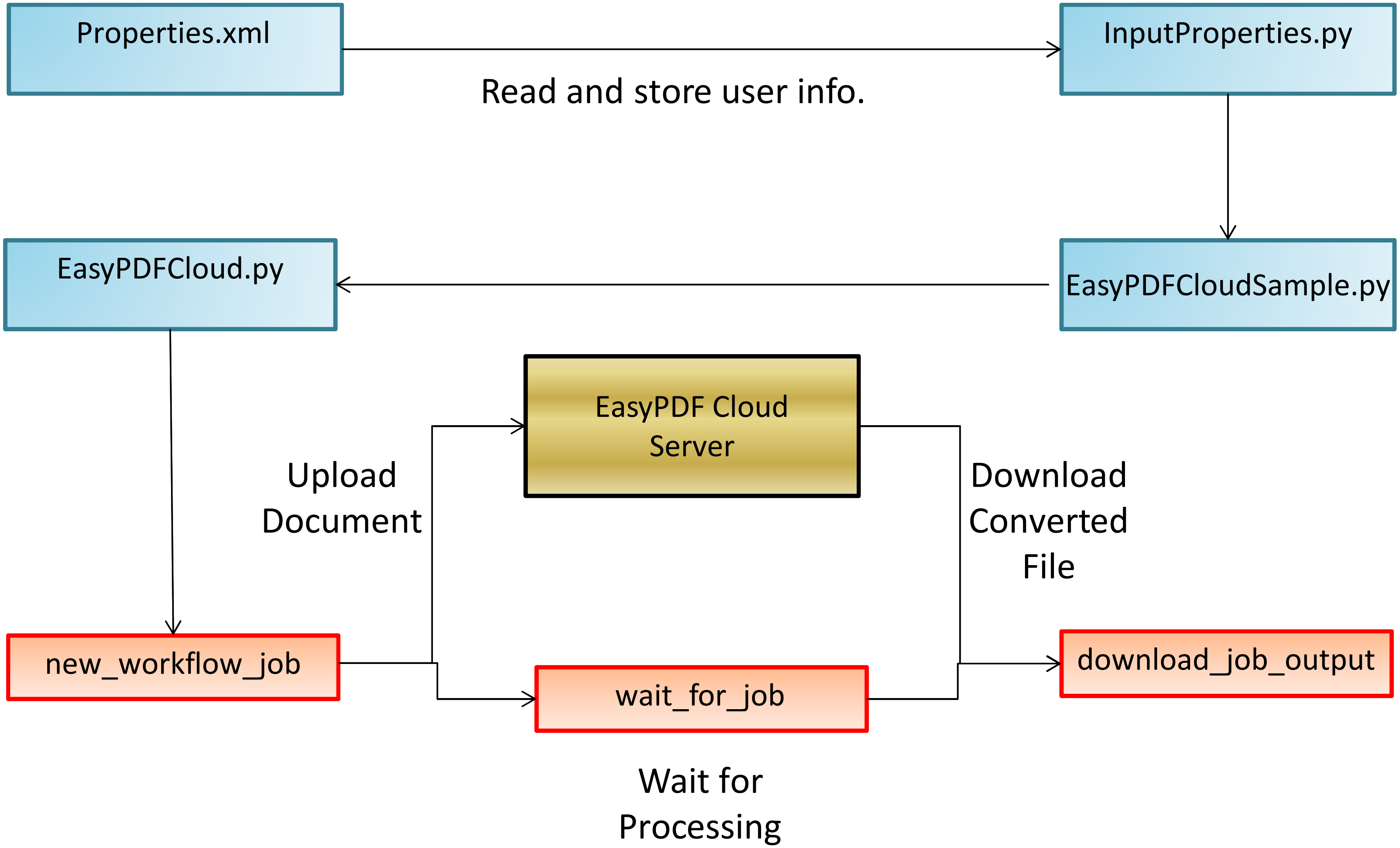
EasyPDF Cloud REST API

BCL Technologies

Requirements

- Python 3.3 or above.
- Pip (if you do not have Python 3.4)
- Requests (external library)
- See README.txt for more information.

Conversion Flow



Properties.xml

Read and store user info.

InputProperties.py

EasyPDFCloudSample.py

EasyPDFCloud.py

new_workflow_job

Upload Document

EasyPDF Cloud Server

wait_for_job

Wait for Processing

Download Converted File

download_job_output

Configuration

- Properties.xml
  - User configuration for the EasyPDF Cloud REST API.
  - The following parameters must be configured:
    - OAuth 2 Client ID
    - OAuth 2 Client Secret
    - Workflow Name
    - Input Directory
    - Output Directory
    - Input File Name
    - Workflow ID (Optional)

File Descriptions

InputProperties.py

- Uses an XML parser to read user information from Properties.xml.
- Provides accessor functions so the information can be used to create instances of other classes and call other functions.

AppInfo.py

- Stores client information from Properties.xml so it can be accessed later by other functions.
  - Contains OAuth2 Client ID, OAuth2 Client Secret and a redirect URI (if applicable).

TokenInfo.py

- Stores an OAuth2 access token so it can be accessed later by other functions.
  - Contains the access token, as well as a refresh token, expiration information, and the scope of the token (if applicable).

TokenManager.py

- Stores an instance of the current TokenInfo class.
- Primary use is to ensure that the API does not call for unnecessary access tokens from the server.

EasyPDFCloudAPIExceptions.py

- Instantiated in most situations when the API runs into a situation that results in an error.
  - Handles HTTP request errors as well as incomplete, inaccurate, or unnecessary parameters passed to functions.
- Gives more descriptive error messages than modules in the standard library.

EasyPDFCloud.py

- Wraps the features of EasyPDF Cloud REST API into a single Python class.
  - Contains functions to retrieve workflow information, get access tokens, create jobs, get job information, etc.
  - Contains a function, check_response, which looks at the HTTP response object and raises an exception if an issue occurs.

EasyPDFCloudSample.py

- Contains a sample testing function that shows how the API can be used. It does the following:
  - Retrieves workflow information.
  - Uploads a file to be converted.
  - Waits for the job to finish.
  - Downloads the output.
  - Deletes the job from the server.

Running the Code

Running the Code

1. Ensure that you are running Python 3.3 or above, and have the Requests library.
2. Create an account at https://www.easypdfcloud.com and obtain developer status. See README.txt for details.
3. Create a workflow.
4. Open and configure Properties.xml with the appropriate information.
5. Run EasyPDFCloudSample.py