

Homework 2 Phase 1: Operations for the Secure File Uploader/Ingest, Text NLP Analysis, and
Newsfeed Ingest

Secure File Uploader/Ingest

- API Type: Entity-based
- Operations:
 - UploadFiles(userID, files[], filetype)
 - Will upload file to account determined by user's ID, method of upload will change based on type of being uploaded
 - For multiple files, number of files in file list will determine how many times this function is ran in a loop
 - This operation needs to be secure
 - DisplayUploadStatus(percent)
 - Will return progress of which files have been successfully uploaded
 - RenderProgressBar(switch)
 - Renders and un-renders progress bar based on upload status
 - UploadError(file)
 - Shows which file caused an error while uploading
 - CancelUpload()
 - Will cancel current upload process
 - FileDelete(userID, file)
 - Will delete file from current list displayed (in list files[])
 - FileEditName(userID, file, new_name)
 - Will allow user to change name of selected file to a new name
 - OrganizeFileList(files[], organize_type)
 - Will organize files[] array with several different options
 - Alphabetical, Latest Uploaded, Earliest Uploaded, etc.

- SendLogReport(error)
 - Logs the current process of the user to be used for debugging purposes, saves .log file or specific error triggered to database statistics file
 - Uses error types to organize types of errors, takes in an error as input
- Diagnostics()
 - Profiling discussed in class such as determining CPU usage, memory usage; show traces, warnings, etc.
- Data:
 - Text data: TXT, PPT, PPTX, RTF, PDF, DOC, DOCX, CSV, XLS, XLSM, XLSX, etc.
 - Potentially incorporate zipped data: ZIP, RAR, TAR, 7Z, etc.
 - Will handle large files, multiple files at once, etc.
 - Only articles, no multimedia (?)
- Status:
 - Before upload
 - API gives option to upload one file or many files to a user's account
 - During upload
 - API shows the current status of the upload, shows how many uploads have been uploaded so far
 - Option to cancel uploading mid-upload
 - After successful upload
 - API displays message to user that upload of files has been completed successfully
 - Error during upload process
 - API displays any errors, lets user know that an error occurred, discusses details of specific errors to the user regarding usage
 - Error information relayed in development logs
 - Files uploaded display
 - Will display files that have been uploaded, currently on user's account
 - Will be able to organize, delete, edit name of files

Text NLP Analysis

- API Type: Procedure-based
- Operations:
 - ConvertFilesToText(userID, files[])
 - Convert file data into TXT/JSON data to be used in text analysis such as Google NLP
 - CreateKeywords(text)
 - Takes in input from user and analyzes it using web-based data such as API data from many different news websites, generate keywords
 - ObtainArticles(text)
 - Keywords used from AssessData() will be used to obtain articles that discuss the topic that the text is discussing, obtaining relevant writing
 - AssessData(article_data, keywords, text)
 - This is where the program will take all aspects, article data from ObtainArticles(), keywords from CreateKeywords(), and the original text or file/s to create a sentiment
 - SaveSentiment(userID, sentiment)
 - Save the sentiment that the program generates to a user's account
 - EditSentiment(userID, sentiment, new_sentiment)
 - This is where a user can edit a sentiment that the program generated
 - DeleteSentiment(sentiment)
 - A user can also delete a sentiment and try again with different parameters
 - Translate(text or sentiment, language)
 - Translates text or sentiment into different languages
 - Will start with popular languages: English, Spanish, Chinese, French, Hindi, Portuguese, Russian, Japanese, German, Danish, etc.
 - SendLogReport()

- Logs the current process of the user to be used for debugging purposes, saves a file to the user's system (?)
 - Diagnostics()
 - Profiling discussed in class such as determining CPU usage, memory usage; show traces, warnings, etc.
- Data:
 - Text data turned into TXT or JSON: TXT, PPT, PPTX, RTF, PDF, DOC, DOCX, CSV, XLS, XLSM, XLSX, ZIP, RAR, TAR, 7Z, etc.
 - Extract data from other sources: Tweets, emails, reviews, social media posts, etc. -> turn into TXT or JSON
 - Input is converted into TXT or JSON to be analyzed by the system, outputted as TXT sentiment
- Status:
 - The API will be used in a text-input interface:
 - User can enter text or use uploaded files
 - API will show progress/status of text analysis
 - API will display any errors that occurred while analyzing text or files
 - API will show user sentiment that the system calculates based on the text inputted, user is then able to translate, save, edit, or delete report generated

Newsfeed Ingest

- API Type: Entity-based
- Operations:
 - DiscoverContent(search_text)
 - Takes in text that user enters to search for relevant information on that topic, uses keywords in the text entered
 - Uses NLP analysis or Google API to return specific articles
 - Returns with relevant searches
 - DisplayContent(searches[])
 -

- After articles are found from the Internet, this command will display the specific articles
- OrganizeContent(organize_type)
 - Able to organize content that the program returns in different ways
 - Alphabetical, Most Recent, Most Relevant (in terms of the result of the NLP output)
- ReadLater(userID, articleID)
 - Allows a user to save a specific article to their user account to read later
- SendLogReport()
 - Logs the current process of the user to be used for debugging purposes, saves a file to the user's system (?)
- Diagnostics()
 - Profiling discussed in class such as determining CPU usage, memory usage; show traces, warnings, etc.
- Data:
 - API takes in text based data, entered in a text field for example
 - API returns a list of articles that a user can browse through or save to read later
- Status:
 - API shows progress of current search, then displays relevant articles
 - API grants user ability to organize articles in a graphical user interface, API grants user ability to save articles to read later with indicator that the article has been saved