­­­­­

|  |  |
| --- | --- |
|  |  |



Solution Design

Document

Table of Contents

[I. Purpose 3](#_Toc5787525)

[II. Automated process details 4](#_Toc5787528)

[3 Runtime guide 5](#_Toc5787529)

[3.1 Architectural structure of the Master Project 5](#_Toc5787530)

[3.2 Master Project Runtime Details 5](#_Toc5787531)

[3.3 Project name 1 6](#_Toc5787532)

[3.4 Project(s) workflows 6](#_Toc5787533)

[3.5 Packages 7](#_Toc5787534)

[3.6 Architectural structure of the Master Project 7](#_Toc5787535)

[4 Other Details 8](#_Toc5787536)

[Future Improvements 8](#_Toc5787537)

[Other Remarks 8](#_Toc5787538)

[5 Glossary 9](#_Toc5787539)

# Purpose



YouTube is a vast online video library that offers virtually unlimited free content.

However, keeping track of the videos you want to watch can be challenging, and using playlists to track videos is somewhat limited with YouTube's current playlist management options.

The solution is the YouTube Automation Suite (YAS).

YAS will allow you to manage your YouTube playlists automatically using the automation power of UiPath combined with Azure SQL Server to store user data and lower the costs of API calls to YouTube.

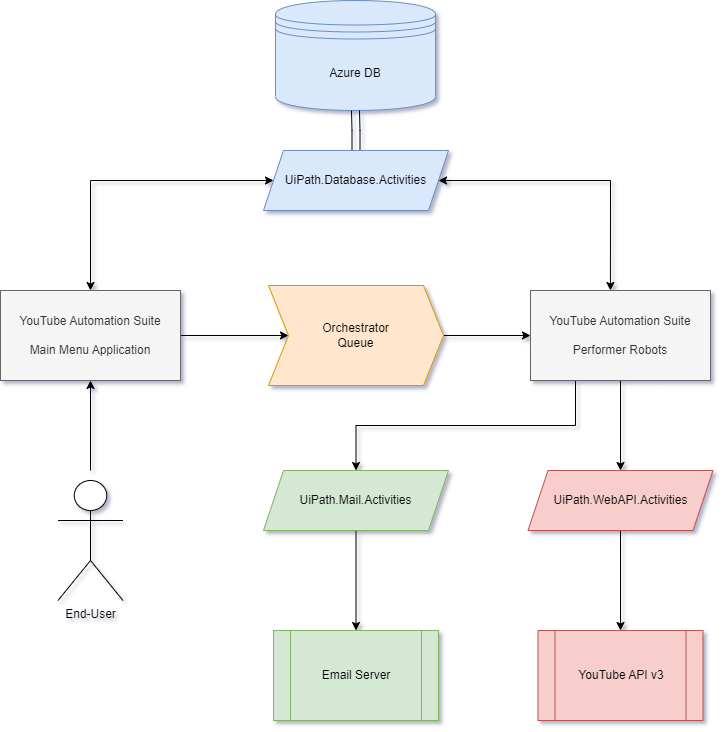
YAS can also provide email updates for each video using Google integration.

# Automated process details

|  |  |
| --- | --- |
| Item | Description |
| Master Project Name | YouTube Automation Suite |
| Robot Type | N/A |
| Orchestrator used? | Yes |
| Scalable | Yes |
| UiPath version used | Studio 2023.4.0 |

# Runtime guide

## Architectural structure of the Master Project



## Master Project Runtime Details

|  |  |
| --- | --- |
| ITEM NAME | DESCRIPTION |
| Production environment details | *Dispatcher runs at user request. Performer currently runs at user request (would normally be a bot that runs on items being added to the orchestrator queue).* |
| Prerequisites to run | *YouTube Account*  *User adds tasks to Orchestrator queue through Dispatcher* |
| Input Data | *User Input* |
| Expected output | *Requested changes reflected in YouTube* |
| How to start the automated process | *The process will be started from UiPath Studio.* |
| Reporting  (queues reporting, Kibana or another platform) | *N/A(?)* |
| How is Orchestrator used? | *Orchestrator is used ot store credentials and manage a transaction queue.* |
| Password policies  (mention any specific compliance requests) | *N/A* |
| Stored credentials  (Never use hardcoded credentials in the workflow!) | *Azure DB Login*  *YouTube API Keys* |
| List of queues names  (Naming convention: ProcessName\_QueueName) | *YAS\_Transactions* |
| Schedule Details | *N/A* |
| Multiple Resolutions Supported?  (in case of image automation / Citrix and VDI) | *N/A* |
| Recommended Resolution | *N/A* |

## YouTube Automation Suite

|  |  |
| --- | --- |
| ITEM NAME | DESCRIPTION |
| Environment used for development  (name, location, configuration details etc) | *UiPath Studio(?)* |
| Environment prerequisites  (OS details, libraries, required apps) | *Studio License*  *Orchestrator*  *Browser of your choice* |
| Repository for project  (where is the developed project stored) | [*https://github.com/230206-UiPath-BNYM/YouTubeAutomationSuite*](https://github.com/230206-UiPath-BNYM/YouTubeAutomationSuite) |
| Configuration method  (assets, excel file, Json file) | *Excel File* |
| List of reused components | *N/A* |
|
| List of new reusable components | *N/A* |

## Project(s) workflows

Workflows specific to: YAS\_Dispatcher

|  |  |
| --- | --- |
| Workflow Name | Description |
| Main | *Invokes all the other workflows* |
| GetTransactionData | *Logic workflow, see workflows below* |
| Process | *Takes user input from workflows below and pushes queue transaction containing the data to Orchestrator* |
| Run\_Database\_Query | *Runs SQL Queries on the DB* |
| Run\_Database\_Command | *Runs SQL Commands to modify the DB* |
| Selector\_CreatePlaylist | *Prompts user for relevant data* |
| Selector\_DeletePlaylist | *Prompts user for relevant data* |
| Selector\_LikeVideo | *Prompts user for relevant data* |
| Selector\_ListVideos | *Prompts user to choose playlist and shows the videos within* |
| Selector\_MergePlaylists | *Prompts user for relevant data* |
| Selector\_ModifyPlaylist | *Logic workflow, see workflows below* |
| Selector\_AddVideo | *Prompts user for relevant data* |
| Selector\_RemoveVIdeo | *Prompts user for relevant data* |
| Selector\_RemoveDuplicates | *Prompts user for relevant data* |
| Selector\_EditPlaylistName | *Prompts user for relevant data* |
| Selector\_EditPlaylistDescription | *Prompts user for relevant data* |
| Selector\_EditPlaylistPrivacy | *Prompts user for relevant data* |

Workflows specific to: YAS\_Dispatcher

|  |  |
| --- | --- |
| Workflow Name | Description |
| Main | *Invokes all the other workflows* |
| Create\_Playlist | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Delete\_Playlist | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Copy\_Playlist | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Merge\_Playlistss | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Edit\_Playlist\_Name | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Edit\_Playlist\_Description | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Edit\_Playlist\_Privacy | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Like\_Video | *Takes Transaction from Queue and requests changes through the YouTube API v3* |
| Get\_Videos | *Retrieves a list of YouTube videos contained within a specified YouTube playlist* |
| Add\_Video | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Remove\_Video | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| Subscribe\_To\_Channel | *Takes Transaction from Queue, requests changes through the YouTube API v3, and updates the Database accordingly* |
| OAuthGen | *Retrieves the YouTube account’s fresh OAuth token to allow for interaction with the YouTube API* |
| Run\_Database\_Command | *Runs a specified SQL Command to update the Database* |
| Run\_Database\_Query | *Runs a specified SQL Query to pull data from the Database* |
| Send\_Emails | *Sends an email with specified contents to a specified user* |

## Packages

|  |  |
| --- | --- |
| Package Name | Description |
| *UiPath.WebAPI.Activities* | *Used to send HTTP Requests to YouTube API v3 to manage playlists* |
| *UiPath.Database.Activities* | *Used to manage connection to send SQL Queries/Commands to Azure DB* |
| *UiPath.Mail.Activities* | *Used to send acknowledgement emails for completion of tasks* |

# Other Details

### Future Improvements

* *Convert “Input Dialog” menu system to UiPath App Studio Application*
* *Upload Performer Process to Orchestrator to be performed by bots*
* *Enable bots to act automatically when a new transaction is uploaded to the queue*

# Glossary

The main terms used in the Solution Architecture Document are defined below:

**Master project** - the overall output of the development, containing one or multiple projects that together cover the scope of the robotic process automation. There is a 1 to 1 connection between the Master Project and the Process to be automated (As presented in the PDD).

**Project** - an UiPath Studio project containing one or multiple workflow files. A project can be converted to a package and run independently, covering a particular scope within the master project. Or multiple projects can be converted into one package depending on the aims and restrictions of the automation. The project is used when defining the development and support phase of the automation.

**Package** - the output of compiling one or multiple projects. A package can be deployed on the robot machine and be executed by the robot service. Only one package can be executed at a given time by a robot. The package is used when defining the running phase of the automation.

Workflow - a component of the package, the workflow encapsulates a part of the project logic. The workflow can be of type: sequence, flowchart or state machine. A workflow is saved as an .xaml file inside the project folder. A workflow file can be invoked from another workflow and by default there is an initial workflow file that will run when executing the package.



**Activity** - an action that the robot executes.

**Sequence** - a workflow where activities are executed one after another, in a sequential order

**Flowchart** - a workflow where activities are connected by arrows and the logic of the workflow can be easily followed in a visual manner. The flowchart can also be exported as an image from UiPath studio.

**State machine** - a more advanced way of organizing a workflow, similar to a flowchart.

**BOR** - Back office robot

**FOR** – Front office robot

**Orchestrator** – Enterprise architecture server platform supporting: release management, centralized logging, reporting, auditing and monitoring tools, remote control, centralized scheduling, queue/robot workload management, assets management.