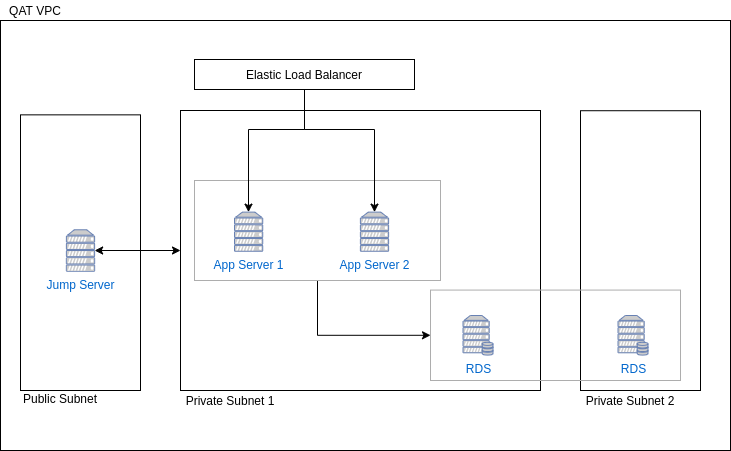
# **Technical Documentation for QAT Server side application**

## Server specifications

The below diagram explains the setup that is the suggested setup for the QAT application. 

This is the suggested setup the user can choose to downgrade this as per requirements. On a minimum level, the following components are required

* 1. Load balancer or WebServer
  2. Application Server
  3. Database Server

## Operating system requirements

* 1. Application Server

The Application server requires a Linux based server preferably Ubuntu with at least version 18.04LTS or higher.

* 1. Database Server

The Database server needs MySql v 5.7

## Other Softwares

* 1. Application Server
     1. Java JDK ver 11

sudo apt-get install openjdk-11-jdk-headless

* + 1. Apache WebServer

sudo apt-get install apache2

* + 1. Node JS
    2. PM2

## Setup process

* 1. Application Server
     1. Apache server

To setup Apache server use the following commands

sudo apt-get install apache2

Copy the SSL Certificates into a folder called /etc/apache2/ssl/

Create a file called www.quantificationanalytics.org.conf in the following folder /etc/apache2/sites-available/ with the following contents

<VirtualHost \*:80>

ServerName quantificationanalytics.org  
 ServerAlias www.quantificationanalytics.org

ServerAdmin info@altius.cc

ErrorLog /var/log/apache2/qat\_error.log

# Possible values include: debug, info, notice, warn, error, crit,

# alert, emerg.

LogLevel warn

RewriteEngine On

RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}

CustomLog /var/log/apache2/qat\_access.log combined

</VirtualHost>

<VirtualHost \*:443>

ServerName quantificationanalytics.org

ServerAlias www.quantificationanalytics.org

ServerAdmin info@altius.cc

ErrorLog /var/log/apache2/qat\_error.log

# Possible values include: debug, info, notice, warn, error, crit,

# alert, emerg.

LogLevel warn

CustomLog /var/log/apache2/qat\_access.log combined

ProxyRequests Off

ProxyPreserveHost On

ProxyPass / http://localhost:4202/

SSLEngine On

SSLCertificateFile /etc/apache2/ssl/WWW.QUANTIFICATIONANALYTICS.ORG.crt

SSLCertificateKeyFile /etc/apache2/ssl/www.quantificationanalytics.org.txt

SetEnvIf User-Agent ".\*MSIE.\*" nokeepalive ssl-unclean-shutdown

Header always set Strict-Transport-Security "max-age=63072000;"

Header always unset X-Frame-Options

Header set X-Frame-Options "SAMEORIGIN"

Header always set X-Content-Type-Options nosniff

Header set Feature-Policy: "fullscreen 'self'"

Header set Referrer-Policy: "no-referrer"

</VirtualHost>

Create a file called api.quantificationanalytics.org.conf in the following folder /etc/apache2/sites-available/ with the following contents

<VirtualHost \*:80>

ServerAlias api.quantificationanalytics.org

ServerAdmin info@altius.cc

ErrorLog /var/log/apache2/qatapi\_error.log

# Possible values include: debug, info, notice, warn, error, crit,

# alert, emerg.

LogLevel warn

RewriteEngine On

RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI}

CustomLog /var/log/apache2/qatapi\_access.log combined

</VirtualHost>

<VirtualHost \*:443>

ServerAlias api.quantificationanalytics.org

ServerAdmin info@altius.cc

ErrorLog /var/log/apache2/qatapi\_error.log

# Possible values include: debug, info, notice, warn, error, crit,

# alert, emerg.

LogLevel warn

CustomLog /var/log/apache2/qatapi\_access.log combined

ProxyRequests Off

ProxyPreserveHost On

ProxyPass / ajp://localhost:8109/

SSLEngine On

SSLCertificateFile /etc/apache2/ssl/API.QUANTIFICATIONANALYTICS.ORG.crt

SSLCertificateKeyFile /etc/apache2/ssl/api.quantificationanalytics.org.txt

SSLCertificateChainFile /etc/apache2/ssl/DV\_USERTrustRSACertificationAuthority.crt

SetEnvIf User-Agent ".\*MSIE.\*" nokeepalive ssl-unclean-shutdown

</VirtualHost>

To enable the modules required for Apache to work execute the following

sudo a2enmod proxy proxy\_ajp proxy\_http rewrite ssl

To enabled the sites you need to run the following command

sudo a2ensite [www.quantificationanalytics.org.conf](http://www.quantificationanalytics.org.conf)

sudo a2ensite api.quantificationanalytics.org.conf

To restart the apache2 service and activate your new settings run the following command.

sudo systemctl restart apache2.service

* 1. Node JS and PM2
  2. API Application

To Install the application and run it on the Server download the compiled QAT.war file on to the Server /home/ubuntu/qatApi folder

Make sure the QAT.war file is executable

Create a service file in the /etc/systemd/system/QATAPI.service with the following content

[Unit]

Description=QAT API

[Service]

User=nobody

# The configuration file application.properties should be here:

WorkingDirectory=/home/ubuntu/qatApi

ExecStart=/usr/bin/java -Xmx256m -jar QATAPI.jar

SuccessExitStatus=143

# TimeoutStopSec=10

# Restart=on-failure

# RestartSec=5

[Install]

WantedBy=multi-user.target

Make sure that service file has 755 rights

Then you need to let run sudo systemctl daemon-reload to reload the daemon services

You can now use the following commands to start, stop or restart the application

sudo systemctl start QATAPI.service

sudo systemctl stop QATAPI.service

sudo systemctl status QATAPI.service

sudo systemctl restart QATAPI.service

## List of API’s

## How to call an API

## Error codes

## React Specifications

1. React JS Version : 16.8.6
2. NPM Version : 6.14.4
3. Nodejs Version : 12.16.1
4. Html-webpack-plugin Version : 4.2.0
5. Workbox-webpack-plugin Version : 5.1.2
6. Babel-loader Version : 8.1.0
7. @babel/core Version : 7.9.0
8. I18next Version : 19.3.3
9. React-dom Version : 16.8.6
10. Jexcel Version : 4.4.1
11. **React JS Overview :**

React is a library for building composable user interfaces, it supports and encourages the creation of reusable UI components, which presents data that changes over time by using declarative syntax.

It uses a concept called Virtual Dom, that selectively renders subtrees of nodes upon state changes. It does the least amount of DOM manipulation possible in order to keep your components up to date.React finds out what changes have been made, and changes only what needs to be changed.

1. **Indexed DB:**

IndexedDB is a large-scale, NoSQL storage system. It lets you store just about anything in the user's browser. In addition to the usual search, get, and put actions, IndexedDB also supports transactions. Here is the definition of IndexedDB on MDN:

"IndexedDB is a low-level API for client-side storage of significant amounts of structured data, including files/blobs. This API uses indexes to enable high performance searches of this data. While DOM Storage is useful for storing smaller amounts of data, it is less useful for storing larger amounts of structured data. IndexedDB provides a solution."

Each IndexedDB database is unique to an origin (typically, this is the site domain or subdomain), meaning it cannot access or be accessed by any other origin. Data storage limits are usually quite large, if they exist at all, but different browsers handle limits and data eviction differently

Browsers supported :Chrome,Firefox,Opera,Safari

1. **Local Storage:**

The Local Storage is an internal database created into the browser, which you can use to save data in a key-value format. Most popular and commonly used browsers like Chrome, Firefox and Safari all support Local Storage. The data stored in local storage has no expiration date, so it will persist over the closed browser window and session.

Local storage has a very simple set, retrieve and remove API, and the key-value pair is always of string type.

1. **Webpack and Workbox:**

Webpack is a popular module bundling system built on top of Node. js. It can handle not only combination and minification of JavaScript and CSS files, but also other assets such as image files (spriting) through the use of plugins.

**Workbox** is the successor to sw-precache and sw-toolbox . It is a collection of libraries and tools used for generating a service worker, and for precaching, routing, and runtime-caching. **Workbox** also includes modules for easily integrating background sync and Google Analytics into your service worker.

1. **Service Worker:**

A service worker is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction.It's a JavaScript Worker, so it can't access the DOM directly. Instead, a service worker can communicate with the pages it controls by responding to messages sent via the postMessage interface, and those pages can manipulate the DOM if needed.

Service worker is a programmable network proxy, allowing you to control how network requests from your page are handled.

It's terminated when not in use, and restarted when it's next needed, so you cannot rely on global state within a service worker's onfetch and onmessage handlers. If there is information that you need to persist and reuse across restarts, service workers do have access to the IndexedDB API.

The reason this is such an exciting API is that it allows you to support offline experiences.

Service workers are supported by Chrome, **Firefox** and Opera and Safari

1. **I18next**

i18next is an i18n framework written in and for JavaScript. It provides the standard i18n features of interpolation, formatting, and handling plurals and context.

1. **App Installation Feature :**

Browsers Supported : Chrome

# **Technical Documentation for** **QAT PipeLine DB to JSON Conversion Tool**

QAT

**(PipeLine Access DB to JSON Conversion Tool)**

# **OVERVIEW**

This software tool has been developed to convert the PipeLine MS Access database (.accdb) to a JSON format data file to use it in an easy way. It is a standalone application and it does not need internet access. So, end users can easily install this tool on their machines and use it in offline mode. Also, this tool has a feature that allows the user to choose preferable language options.

# **GOALS**

The main goal of this tool is to convert the PipeLine MS Access database into JSON data.

# **SYSTEM CONFIGURATION REQUIREMENTS**

As this is the standalone application, users must have the below system configuration to use this tool on their system.

Operating System: *Windows / Linux / Ubuntu*

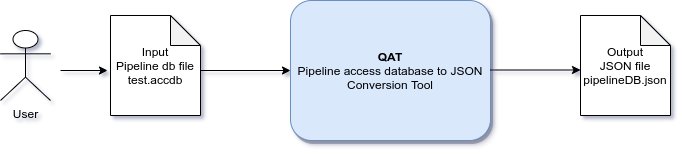
Other Software : *Java JDK 1.8 and above.*

If the system has a Windows operating system then the user has to download and use the .exe file of the tool; if the user has Linux operating system, they can use an executable file.

# **FUNCTIONALITY REQUIREMENTS**

Users should provide the correct access database file to the tool to generate the correct JSON output file.

# **FUNCTIONAL DIAGRAM**



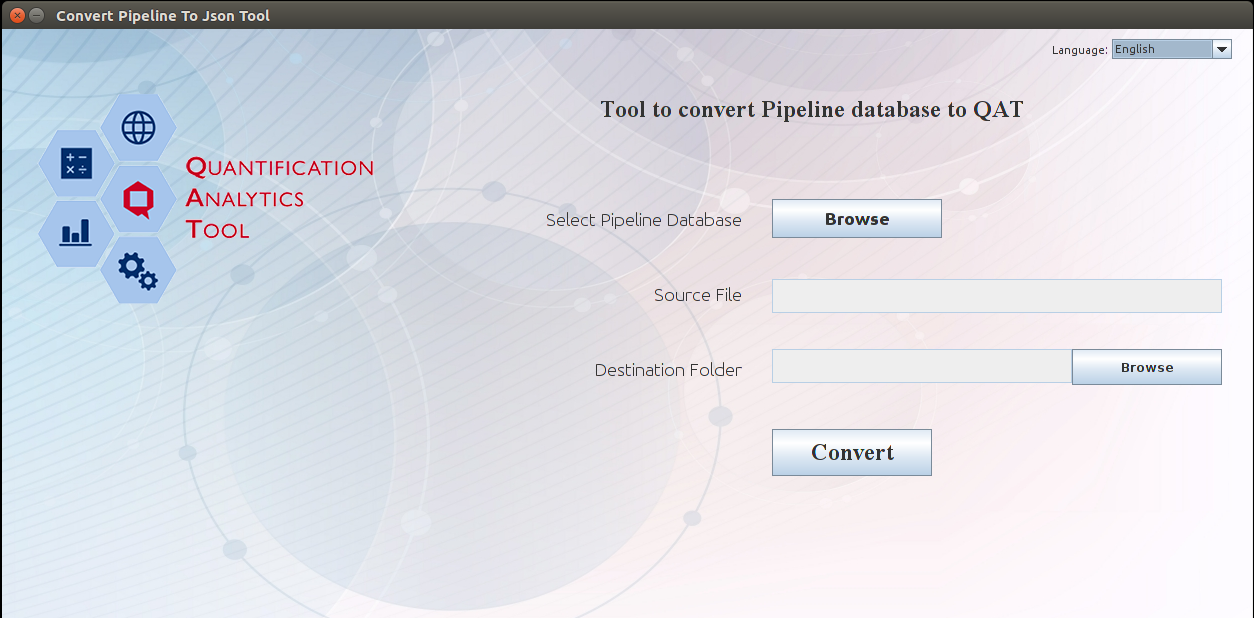
# **HOW TO USE**

Follow the below steps to use this tool to convert the PipeLine MS Access database to JSON:

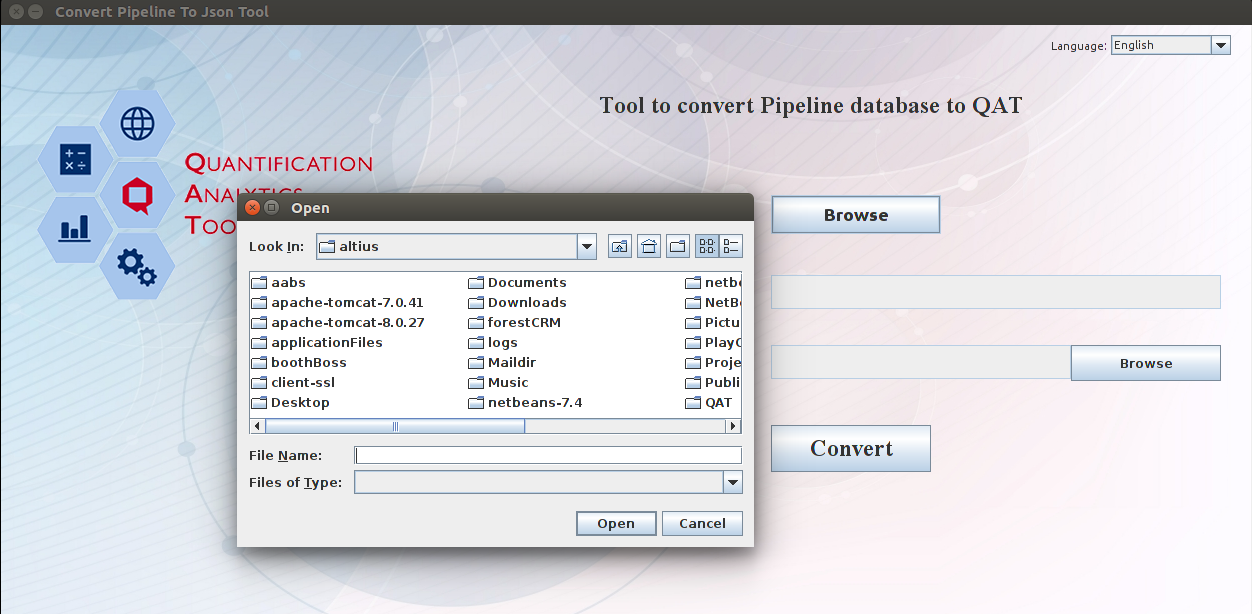
1. Run the .exe file / .java executable file
2. Click on the first browse button to choose the PipeLine MS Access database file. The file’s location will show underneath.
3. Click on the second browse button to choose a folder where you want to save the output JSON file.
4. Click on the convert button to start the conversion functionality.
5. After conversion is done, you will get the output file details. If an error occurres while converting the file, the error will be shown on the screen.

# **SOME SCREENSHOTS**

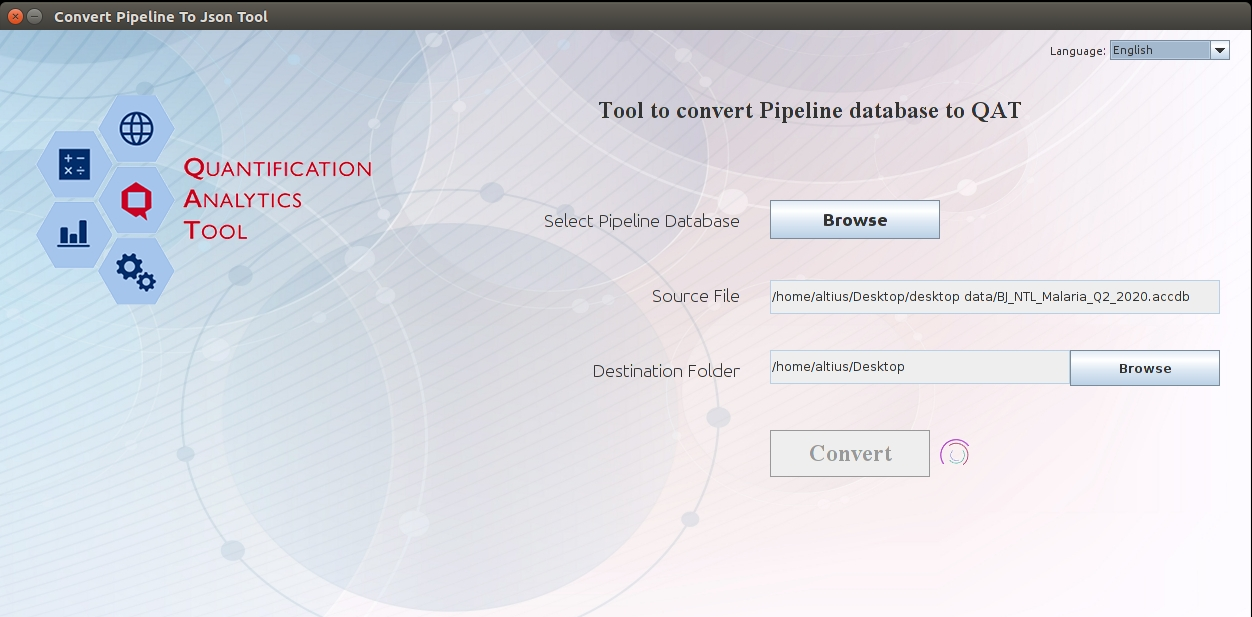
QAT PipeLine DB to JSON Conversion Tool \_1



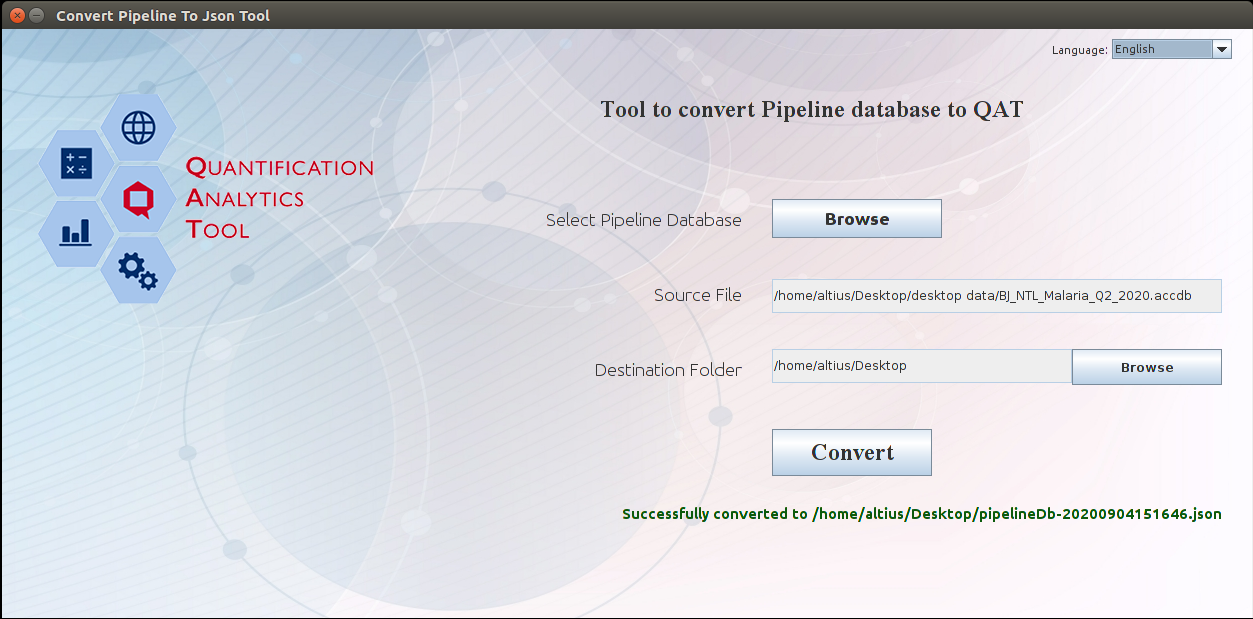
QAT PipeLine DB to JSON Conversion Tool \_2



QAT PipeLine DB to JSON Conversion Tool \_3



QAT PipeLine DB to JSON Conversion Tool \_4



**React App Installation steps**

open our Terminal and paste the below command.

curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash -

sudo apt-get install -y nodejs

Update npm to latest version using below command

sudo npm install npm@latest -g

**Running QAT Application**

To run a QAT application on your local machine download it from git repository. Enter the project directory and execute npm install and npm run dev

cd fasp-core-ui

npm install

npm run dev

It starts the React application on port 4204. Open new tab on browser with below address

http://localhost:4202

.