8/24/2020

USER DOCUMENT

TEST DATA MANAGEMENT

Amritha Surgi Kumar
Sarin Preman Sindhu
Vishnu Venugopal
Dharanya Ayyappan
Jaqueline Benedict

EasyClub
ALLIANZ TECHNOLOGY

Table of Contents

Change History

Revision	Date	Author	Section/Pages affected	Remarks
EasyClub_UserDocument_V1.0	August 22,2020	Amritha S Kumar , Sarin P S , Vishnu Venugopal, Jacquline Benedict, Dharanya A	All sections – Initial draft	NA
EasyClub_UserDocument_V1.1	August 23,2020	Amritha S Kumar , Sarin P S , Vishnu Venugopal, Jacquline Benedict, Dharanya A	All sections- Reference Added	NA

Preface

This document is prepared as part of the requirement for a reusable **Test Data Management** utility which does data setup automatically for any automation tool. The TDM utility is scalable to accept data from different data sources and create test data sheet for automation scripts.

1. Introduction

1.1 Purpose

Many automation test suites expect detailed Test Data input for its execution. But the actual data often changes due to other activities in the test environment. Thus tests fail due to incorrect data.

To counter this, many projects deploy a manual test data setup stage which is cumbersome and error prone.

A reusable utility is proposed to do test data setup automatically. The TDM utility is designed to be scalable to accept data from different data sources and create test data sheet for automation scripts.

1.2 Scope

The prototype designed here is used to convert a CSV input file to the result file containing the required columns and rows. User can input the columns to be present in the test data file and the result file will be generated based on the same.

1.3 Reference and resources

https://pandas.pydata.org/docs/getting_started/intro_tutorials/02_read_write.html#min-tut-02-read-write

2. Prototype features:

1. Completely open source
2. User friendly commands
3. Zero expense
4. Highly robust

3. Future Work

1. We can convert the prototype to an online platform – Provide GUI for smoother user experience
2. The utility can be made 100% automated by following ways:
   - Automatically running it on regular intervals so that any change in the input instance will reflect in the test result sheet.
   - Any change in the input instance should trigger automatic execution such that test result sheet is always in sync with input sheet.
3. The code can be customised to remove any redundant / erroneous data

4. User manual

Pre Requisite – System installed with Python v3 , pip3, VS code , pandas (used in data science)
As per current implementation a file of csv format is taken as the input (input.csv) and the resulting data is saved as a csv file (result.csv).
On execution , user is prompted to choose number of columns required in the result file
On entering the column count, user is prompted to choose the column headers – Provide exact same column headers as in the input CSV file
Once columns are chosen, user is prompted to enter number of rows.
On submitting number of rows, the result will be generated and stored in result.csv