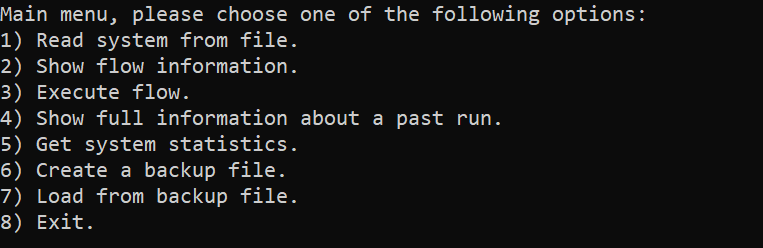
**Stepper**

1. מגיש: עומר ארליכמן – 322281833

מייל: omerec@mta.ac.il

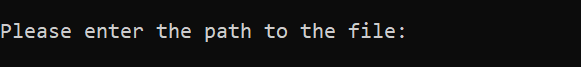
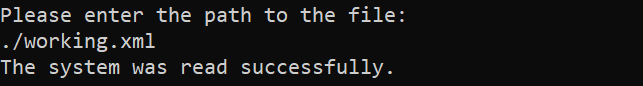
1. בונוסים שמומשו: שמירה וטעינה של המערכת (אפשרויות 6,7 בתפריט)
2. ממשק משתמש:

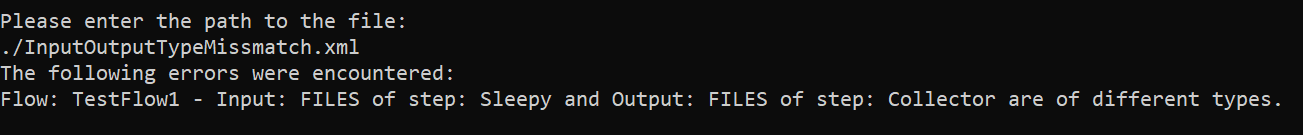
מיד עם פתיחת המערכת יוצג מסך התפריט:

דרך מסך התפריט ניתן להגיע לכל אחת מהאפשרויות לעיל ע"י הכנסת מספר האפשרות ולחיצה על 'Enter':

1. [קריאת המערכת מקובץ (xml): (Read system from file)](#ReadSystemFromFile)
2. [הצגת מידע על flow: (Show flow information)](#ShowFlowInformation)
3. [הרצת flow: (Execute flow)](#ExecuteFlow)
4. [הצגת מידע על ריצת עבר של flow: (Show full information about a past run)](#ShowFullInformationAboutAPastRun)
5. [הצגת סטטיסטיקות של המערכת: (Get system statistics)](#GetSystemStatistics)
6. [יצירת קובץ גיבוי למצב הנוכחי: (Create a backup file)](#CreateABackupFile)
7. [טעינת קובץ גיבוי (שנוצר על ידי פקודה 6): (Load from backup file)](#LoadFromBackupFile)
8. [יציאה וסגירה: (Exit)](#Exit)
9. מחלקות מרכזיות:
10. [Engine](#EngineClass)
11. [ConsoleApplication](#ConsoleApplicationClass)
12. [Flow](#FlowClass)
13. [FlowExecutor](#FlowExecutor)
14. [StepRegistry](#StepRegistryClass)
15. ['Parsers'](#ParsersClass)
16. ['Validators'](#ValidatorsClass)
17. ['Contexts'](#ContextsClass)

קריאת המערכת מקובץ (xml): (Read system from file)

1. לאחר בחירת אפשרות 1 בתפריט יופיע הטקסט הבא:
2. בשלב הזה צריך להכניס את הנתיב המלא אל קובץ ה-xml שנרצה לטעון ולהקיש 'Enter'.
3. המערכת תנסה לטעון את הקובץ ותחזיר פלט בהתאם, למשל אם הקובץ תקין יוחזר:

בעוד שעבור קובץ לא תקין תוחזר הסיבה לכישלון, למשל:

הצגת מידע על flow: (Show flow information)

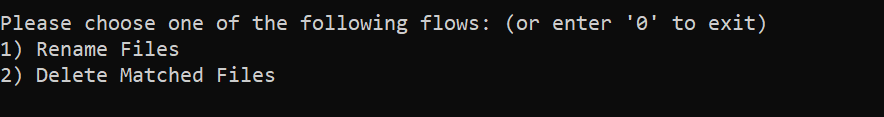
1. Text

   Description automatically generatedלאחר בחירת אפשרות 2 בתפריט יופיע תפריט בחירה בין כל ה-flows במערכת כרגע:

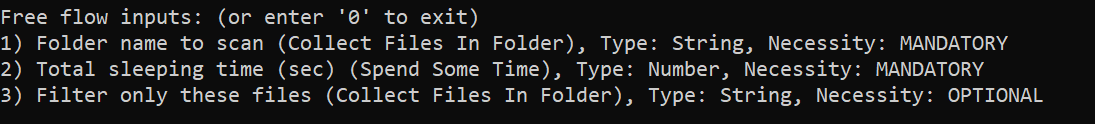
על מנת לבחור flow צריך להקיש את המספר שמופיע לידו ואחריו 'Enter'.

1. לאחר בחירת flow תופיע רשימת המידעים עליו, היא כוללת:
   1. שם ה-flow.
   2. תיאור ה-flow.
   3. פלטים מוצהרים של ה-flow.
   4. האם ה-flow מוגדר read-only או לא.
   5. ה-steps ב-flow.
   6. קלטים שפתוחים למשתמש. (חובה ורשות בסדר הזה)
   7. פלטים פנימיים. (של שלבי ביניים)

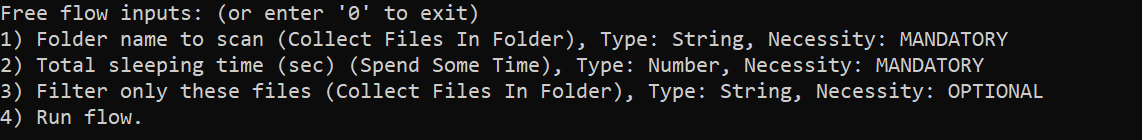
הרצת flow: (Execute flow)

1. לאחר בחירת אפשרות 3 בתפריט יופיע תפריט בחירה בין כל ה-flows במערכת כרגע:

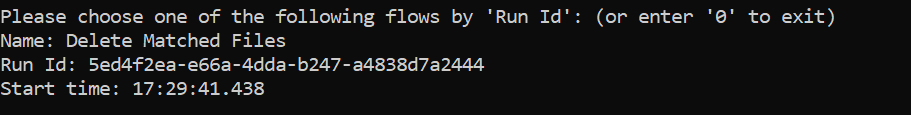
על מנת לבחור flow צריך להקיש את המספר שמופיע לידו ואחריו 'Enter'.

1. לאחר בחירת flow יופיע תפריט חוזר נוסף של כל הקלטים שה-flow מבקש, על מנת לקבל את האפשרות להריץ את ה-flow יש להכניס קלט לכל ה-MANDATORY לפחות.

כמו כן, כדאי לציין שבמידה והקלט לא מתאים לסוג המשתנה נקבל על כך הודעה במידה ואפשר לזהות זאת לפני הריצה. (הכנסת קלט String למשתנה מספרי למשל)

1. לאחר הכנסת כל משתני החובה נפתחת אפשרות נוספת (בסוף הרשימה) להריץ את ה-flow
2. לאחר ריצת ה-flow מתקבלות כמה שורות סיכום הריצה מהפורמט הבא:
   1. מחרוזת הזהות של הריצה.
   2. שם ה-flow שהורץ.
   3. תוצאת הריצה המקוצרת. (Success, Warning, Failure)
   4. הפלטים הרשמיים של ה-flow.

הצגת מידע על ריצת עבר של flow: (Show full information about a past run)

1. לאחר בחירת אפשרות 4 בתפריט יופיע תפריט בחירה בין כל הריצות שקרו:

על מנת לבחור ריצה, יש לכתוב את ה-'Run Id' שלה ולהקיש 'Enter'.

1. לאחר בחירת ריצה יופיע המידע המלא עליה כפי שהוא שמור במערכת, הכולל:
   1. מחרוזת הזהות של הריצה.
   2. שם ה-flow שהורץ.
   3. תוצאת הריצה המקוצרת. (Success, Warning, Failure)
   4. זמן הריצה במילי-שניות.
   5. הקלטים שמולאו על ידי המשתמש וערכם.
   6. הפלטים הרשמיים של ה-flow.
   7. רשימת הצעדים ב-flow שהורץ, לכל אחד:
2. שם ה-step.
3. זמן הריצה במילי-שניות.
4. תוצאת הריצה המקוצרת.
5. שורת הסיכום.
6. לוגים של מהלך הריצה.

הצגת סטטיסטיקות של המערכת: (Get system statistics)

1. לאחר בחירת אפשרות 5 בתפריט תופיע רשימה שבה תחילה יופיעו ה-flows במערכת ולאחריהן יופיע ה-steps, עבור כל רשומה של flow/step יופיעו המידעים הבאים:
   1. שם.
   2. מספר הפעמים שהם הורצו. (0 אם עוד לא הורצו)
   3. זמן הריצה הממוצע במילי-שניות.
2. חשוב לציין שבמידה ו-flow/step נגמר ב-Failure הוא לא ייספר לטובת הסטטיסטיקות.

יצירת קובץ גיבוי למצב הנוכחי: (Create a backup file)

1. Text

   Description automatically generated with medium confidenceלאחר בחירת אפשרות 6 בתפריט יופיע הטקסט הבא:
2. בשלב הזה צריך להכניס נתיב מלא לקובץ שבו רוצים ליצור את השמירה של המערכת.
3. Text

   Description automatically generatedבמידה והקובץ נוצר בהצלחה תתקבל ההודעה הבאה:
4. חשוב לציין שהמערכת תיצור את הנתיב המלא (כולל תיקיות אל הקובץ שהתקבל בקלט), בנוסף אם הקובץ קיים היא תדרוס אותו ותכתוב מחדש.

טעינת קובץ גיבוי (שנוצר על ידי פקודה 6): (Load from backup file)

1. לאחר בחירת אפשרות 7 בתפריט יופיע הטקסט הבא:
2. בשלב הזה צריך להכניס נתיב מלא לקובץ שבו נשמרה המערכת (על ידי פקודה 6) כדי לטעון אותו.
3. Text

   Description automatically generated with low confidenceבמידה והקובץ קיים ותקין המערכת תיקרא ממנו ותתקבל ההודעה הבאה:

Graphical user interface, text

Description automatically generatedבמידה והקובץ לא קיים:

Text

Description automatically generatedבמידה והקובץ לא מכיל גיבוי של המערכת:

יציאה וסגירה: (Exit)

1. לאחר בחירת אפשרות 8 בתפריט המערכת תיסגר ותצא.

Engine

תפקיד המחלקה הוא להיות הממשק בין ה-UI למימוש המערכת.

ה-Engine מכיל בתוכו את כל ה-state של המערכת, ומעבר לכך הוא הממשק החשוף ל-UI כדי לבצע את הפעולות שהמשתמש מבקש.

כל אחת מהפעולות שה-UI מסוגל לבצע קורות על ידי קריאה לאחת הפונקציות של Engine.

ConsoleApplication

תפקיד המחלקה הוא לנהל את זרימת התכנית מצד המשתמש.

מחלקת ה-ConsoleApplication היא מימוש ה-UI והיא הגשר בין המשתמש למנוע המערכת, דרכה נקלטות הבקשות של המשתמש והיא קוראת לפונקציות המתאימות במנוע.

המחלקה מכילה מופע של אובייקט Engine שהוא אלמנט ה-state היחיד בה.

Flow

המחלקה מכילה את כל המידעים שמגדירים flow ברמת המערכת (לא כולל ריצה).

דרכה המנוע מגדיר flows ובמחלקה הזאת נמצאת חלק גדול מיכולת האימות של flow כתקין מערכתית.

FlowExecutor

זוהי מחלקה חסרת state שמגדירה את סדר הפעולות בעת הרצת flow בצורה גנרית.

הפונקציה היחידה והחשובה במחלקה זו היא executeFlow שמקבלת את ה-flow ו-context להרצה, מבצעת את הריצה ומחזירה אובייקט של תוצאה (ומתעדת אותו).

StepRegistry

זוהי מחלקת enum שמכילה מופע אחד מכל סוג של step אשר מוגדר במערכת.

מכיוון שבחרתי לנסות להפריד בין האלמנטים של ההגדרה והריצה במערכת דרוש לי רק מופע אחד מכל סוג step, על ידי שימוש ב-registry אני יכול לחסוך את ההקצאה ולהשתמש בזה שיש לי רשימה של כל הצעדים האפשריים גם במקומות אחרים שבהם זה שימושי (למשל אימות תקינות).

Parsers

תפקיד ה-parser'ים במערכת הוא לקחת מידע גולמי ולהעביר אותו לפורמט שאני יכול להשתמש בו ביתר קלות (בנוסף לאימות תקינות כל אחד בתחומו).

הגדרתי כמה סוגי parser'ים והם:

1. Step Parser – אחראי על המרה של step מקובץ xml ל-step כפי שמוגדר במערכת.
2. Mapping Parser – אחראי על המרה של מיפויים (flow level ו-custom) מקובץ xml למיפויים באובייקט ה-flow במערכת שלי.
3. Flow Parser – אחראי על המרה של flow מקובץ xml ל-flow כפי שמוגדר במערכת.

בצורה הזאת פירקתי את הבעיה הגדולה של המרות המידעים, מיפויים ואימות התקינות לכמה משימות קטנות שיותר קל לנהל.

Validators

ישנן הרבה מחלקות Validator במערכת שלי, כל אחת מהן היא בדיקה קטנה שצריך לוודא על הקובץ xml/flow/mapping לפני שניתן להכריז על קובץ/flow כתקין מבחינת המערכת.

זוהי הדרך שבחרתי לחלק בה את המשימה הגדולה של בדיקת האימות למטלות קטנות יותר ובה בחרתי לבצע את הרוב המוחלט של בדיקות האימות.

Contexts

ישנן 2 מחלקות שמספקות context לריצה, אחת של Step ואחת של Flow, אלה מחלקות שמכילות רק את ה-state שנדרש להפעלה, ביניהם המזהה של הריצה, יכולות של logging ותיעוד לסטטיסטיקה, המשתנים של הריצה הנוכחית ובמקרה של ה-flow גם תוצאות הביניים של ה-step'ים הקודמים.