**Project Guidelines**

* The solution contains 3 projects:
  + WebAPI as server endpoint façade
    - MainController - for handling client requests
    - ClientController – simple client html/jquery based.
  + Business as the server side logic, using HtmlAgilityPack open source lib for web scrapping
  + UnitTest with 2 test methods (of course many are missing), using Moq open-source lib
* How to run the project:
  + Create a folder c:\temp\celeb The app is configured to this path for log and the json
  + Run the WebAPI from within the solution.
  + In ConfigUtil.cs, you can set the number of “celeb” items you wish to pull. Currently is set to 10.
* Points on interest:
  + Using dependency injection as ASP NET Core best practice
  + Using OOP Interfaces for the IRepository and IScrapEngine (as at some point in the future we’ll want to replace any of them)
  + Log4Net embedded for logging errors and information
  + Exception handling at major failure points.
  + Notice use of Parallel.ForEach in the GoScrap method. This improve performance however celeb are aggregated not in the same order as in origin site
* Future improvements:
  + Using async/await in all external calls (e.g. web calls and json calls) for better handling thread pooling and multi-threading patterns
  + Leverage the use of Threading.Task library for multi threading
  + Scaling up – using Queueing pattern for balancing pressure and able to scale between several services
  + Using Socket/SignalR Technology so client will get rendered as celeb items are being processed and before all items are finished to be scrapped.
  + More.. let’s discuss 😊

**תרגיל בית**

מועמד/ת יקר/ה, אנו מודים לך על הירתמותך והקדשת זמנך לתהליך, מטרת התרגיל היא לבחון רמה טכנולוגית ויכולות בסיסיות.

במידה ונתקלת בשאלה או הבהרה אנא פנה/י אלינו למייל זה [or@marketman.com](mailto:or@marketman.com) ונשמח להבהיר.

בסיום אנא שלח/י את הקוד בזיפ/DropBox לאימייל עם הנחיות (אם נדרש)

**הנחיות:**

בתרגיל זה יהיה עליך להקים מערכת"ידוענים" בדרישות הבאות:

-המערכת תציג טבלה של ידוענים, לכל ידוען: שם, תאריך לידה (אם קיים), מין (זכר/נקבה) ,תפקיד (שחקן/שחקנית/מפיק וכו') ,תמונה

לדוגמא

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image | Role | Gender | Date of birth | Name |
| Johnny Depp Picture | Actor | Male | June 9, 1963 | Johnny Depp |

-המשתמש יוכל למחוק או ל"אפס" את כל המידע.

**בהיבט הטכני:**

Client Side

-יפותח בכל טכנולוגיה שתבחר/י בה. אין צריך להשקיע בחלק הזה.

Server Side

-.Net Framework/.Net Core + Restful Web API

-את המידע יש לאסוף מאתר <https://www.imdb.com/list/ls052283250/> באמצעות קריאה סקרייפינג.

-יש לשמור את הנתונים בקובץ Json (שימו לב איסוף המידע מהאתר הינו חלק מהתרגיל)

-פעולת המחיקה תמחק את הרשומה שהשתמש ביקש למחוק מה Json ותעדכן תצוגה.

-פעולת האיפוס תשחזר את הנתונים לנקודת ההתחלה

**דגשים:**

-יש לשמור על פשטות בצד הפיתוח

-אין צורך לתמוך בScale

-אין צורך לתמוך בקריאות א-סינכרוניות

-ניתן להשתמש ברכיבים חינמיים חיצוניים

-הקוד בשרת יהיה מחולק למודלים ושכבות (אין צורך ברבים)

**בהצלחה!**

**צוות Marketman**