

## **הוספת ספק**

המערכת בונה כך שלכל ספק **שאינו** ספק פרטי יש כמה פונקציות וקבצים הנפרדים משאר הספקים זאת על מנת לאפשר גמישות בשינויים בניהול המידע של אותו ספק, כמו למשל שינוי סוג או מבנה קובץ המתאר את החיבורים של הרשות מול אותו ספק או שינוי המתבקש על ידי רשות לצורך הצגה של המידע ועוד.

מצב זה אכן יוצר שכפול קוד, אך אפשר למערכת להיות גמישה ונוטה לשינויים במידת הצורך.

### **חשיבות לצין:**

ספקים שאינם פרטיים מתארים ספקים שהינם נתונים שירות לכל הרשותי בארץ, לרוב מדובר בשירותי תשתיות. הספק שעלה בסיסו נוצרה המערכת הינה 'בקץ', המשקף בצורה מעולה את הרעיון הנ"ל

### צד-שרת:

#### **קריאה קובץ:**

1. הוספה ספק בקובץ Enums הנמצא בפרויקט Entities, אשר ה.ch.cs של אותו ספק משמש כערך המספרי של שם הספק
2. הוספה קובץ שמרתתו היחידה היא קריית המידע מהקובץ, זאת נעשה בצורה זו משום שבין הספקים אין רק הבדל של סוג קובץ והסדר בתוכו, אלא גם בסוג המידע שהוא מספק. שם הקובץ צריך להיות בניו בצורה הבאה: **SupplierNameFileReaderService**, אשר הגישה אליו נעשית דרך `interface` בלבד!  
לאחר הוספה הקובץ וחיבורו למספק יש לילכת לקובץ `startup.cs` היושב תחת פרויקט `services`, `ManagingDebts`, ולהוסיף תחת `services.AddScope`, את הקבצים החדשניים שנוצרו לאחר מכן יש למלא את הקובץ לפי לוגיקת הקראיה של הקובץ ופיזור הנתונים בטבלה המתאימה (ראה חלק בסיס- מידע לפירוט בנושא)
3. תחת תיקית controllers אשר יושבת בפרויקט `ManagingDebts`, יש לגשת לקובץ `UploadFileController`, בקובץ הנ"ל יש להוסיף את הממשק מסעיף 2 בתחילת הדף ולהוסיף לconstructor של הממשק של controller, לאחר מכן תחת פונקציית `uploadFile`, קיים switch case אשר כל case בוחר מ深切 מיאצג קראיה לממשק של הספק המקשר אליו, יש להוסיף שם case חדש תוך כדי שימוש בenumer מסעיף 1

### וידיוי נתונים:

1. הוספה קובץ יישות תחת Entities המmesh את כל השדות מבוסיס המידע, מומלץ לבצע העתק הדבק מהקובץ של היישות תחת תיקית models הנמצא בפרויקט `dataModel`, יש לוודא שקיים שדה `AmountAfterTax`, שבהתאם לשמו מייצגת את הסכום לאחר חישוב מס.
2. הוספה קובץ שמרתתו היא קבלת נתונים מהבסיס המידע והטיפול בהם, לשם כך יש להוסיף קובץ בעל המבנה השמי `SupplierNameDataConformation`, וממשק התואם לשם הקובץ.
3. בקובץ יש להוסיף את הקראיה מבוסיס הנתונים ואת הפונקציות המכינות את `dictionary` הנשלחה לצד-לקוח לצורך פענוח בעמוד וידי נתונים באפליקציה, `dictionary` בנייתו粲 בדמות הבאה:

מפתח שהינו `string ssoj` של סיכום החובים תחת חוזה מסוים, ניתן לראות דוגמאות בקבי' בזק וחשמל,

וערך שהינו מייצג מראה של הנתונים מהקובץ כמו שהם.

4. בקובץ יש להוסיף פונקציה `match`, שמטרתה לקבלת נתונים קובץ ולעדכן את המצב שלהם, ניתן לראות דוגמאות בקבי' בזק וחשמל
5. לאחר הוספת הקובץ וחיבורו למשك יש לילכת לקובץ `startup.cs`, שמטרתה תחת פרוייקט `services.AddScope`, את הקבצים החדשניים שנוצרו
6. תחת תק"ית controllers הנמצאת בפרויקט `ManagingDebts` יש לגשת לקובץ `ManagingDebtsController` ולהוסיף בחלק העליון של הדף את הקראיה למשק שכרגע יצרנו, לא לשכוח להוסיף קראיה למשק בconstructor.
7. יש למשוך את הפונקציות בswitch הנמצא בפונקציה `GetDataBySummary`, ולהוסיף פונקציה חדשה בשם, **MatchSupplierName**, הקוראת לפונקציית התאמת שיצרנו בקובץ (הכל דרך הממשקים!)

#### דוחות:

1. הוספה case בשני הcases הקיימים בקובץ `dashboardService`
2. הוספה פונקציה נוספת שמטירה היא הבאת כל המידע הקיים במערכת לקובץ אותו לחוזים וליצור יישות המומפה לטבלה `chartEntity`, ניתן לראות דוגמאות איך ניתן לעשות זאת בשורה אחת של קוד על פי פונקציות קודמות

#### יצירת פקודת יומן:

1. יש להוסיף פונקציה נוספת תחת `createPkudatYomanSupplierName`, בשם `SupplierNameService`, הפונקציה אמורה לקבל את המידע מהבסיס מידע על בסיס מספר חשבון, ובכך ליצור סוג של `dictionary` כאשר מספר החשבונית הוא המפתח ולאחר מכן יש לעבור על כל מידע המקיים על ידי חשבונית ולקובץ אותושוב על סマー מספר חוזה לאחר מכן יש לוודא את הקשר הלוגי בין החוזה לסעיף התקציבי וליצור/לסכם את את פרטי החוזה לתוך השורת פקודה של הסעיף התקציבי, הסבר יותר מפורט של הלוגיקה בסוף א'
2. לבסוף יש להמיר הכל למערך ולהעביר חזרה לפקודת היום.  
**חשוב לציין:** לא מומלץ לשנות יותר מדי, מרבית השינוי בין הפונקציות של יצירת תוכן הפקודה הוא השיליפה של המידע מבסיס המידע וצורת הקיבוץ שלו.

#### צד - לקוחות:

##### קריאת קובץ:

1. יש להוסיף לקובץ `billing-summary.service.ts`, פונקציה שמטרתה להחזיר מערך של `TableHeader`, שם הפונקציה צריך להיות `getSupplierNameTableHeaders`, יש למשוך זאת על פי השדות הסיכום הנמצאות בשימוש הרוב ביותר לפי מבנה הקובץ לדוגמא בקובץ בזק יש שימוש בכל העמודות הקיימות בטבלה `GeneralBillingSummary`, אך לא תחת חברת חשמל ולאחר הוספה הפונקציה למשוך קראיה של עלי בסיס ה`switch` הקיים באותו קובץ

##### וידי נתוניים:

1. יש להוסיף לקובץ `bank-matching.service.ts`, פונקציה שמטרתה להחזיר מערך של `TableHeader`, שם הפונקציה צריך להיות `getSupplierNameTableHeaders`, יש למשוך זאת על פי השדות הקיימים בטבלה המייצגת את המידע מהקובץ

2. באופן קובץ יש להוסיף לelow switch תחת פונקציית match ערך או בהתאם לפונקציה בצד-שרת  
**matchSupplierName**

כמובן לא לשוכח ליצור עבור הרשות/רשות הרכונטיות את הספק , פעולה זאת נמצאת בתפריט תחת "מערכת" -> "ספקים"

בosis נתונים:

יצירת טבלה לספק:

הטבלה הנמצאת בbasis המידע אמורה לשמש כמו מראה לשודות הקובץ של הספק, **לא נדרש מידע מעבר לכך** יש להוסיף את השודות הבאים, נא לבדוק בשם **Customer\_id,general\_row\_id,row\_id,is\_matched,journal\_entry\_number**

השילוב של השודות המודגשים צריכים להיות המפתח הראשי בטבלה, כאשר קיים קשר אחד לרבים בין id\_general\_billing\_summary.row\_to general\_row\_id

עדכן צד שרת:

קיים קובץ עם דוגמא לפקודה בתוך הפרויקט dataModel , יש למלא את הנתונים בפקודה בהתאם, ולבצע דרך console Nuget.

לאחר ביצוע הפקודה יוצרו מלא קבצים בפרויקט ManagingDebts, יש למחוק את הקבצים שנוצרו זה זה סתם שכפול

לאחר מכן יש להוסיף את הדברים הבאים:

<http://prntscr.com/re6h64>

<http://prntscr.com/re6hf>

זה קורה כי context מתרען גם הוא בעדכון db, ועל מנת שהפרויקט ישען על הדיסטנס appsettings צריך לבצע את זה

זהו...

מבנה פקודת ימן:

פקודת ימן מייצגת תנועה של כסף בטור הנהלת חשבונות. מכון שלא מדובר בהצהרה של בעל העסק (במקרה זהה עירייה צו או אחרת) עבר תשלום עתידי, אלא הצהרה על תנועה כספית שכבר בוצעה, הפקדוה חייבת להיות מאוזנת בזכות וחובה

אפשר לראות זכות וחובה כמו פועלות פלו"ס ומינוס, כאשר חובה היא פועלות מינוס וזכות היא פועלות פלו"ס.

הסביר יותר נכון הוא:  
אשר בעל עסק מדווח על תנועה כספית, הוא בעצם **מחייב את עצמו**, ומזכה את המוטב בעסקה

דוגמא:

אשר עירייה בני ברק משלם לבזק עבור שירות תשתיות כלו ואחרים, עירייה בני ברק **מחיבת את עצמה ומצחה את בזק**  
וכך בהנהלת החשבונות מתעדכן הסכם שהעירייה שילמה לספק מסוים, ומאפייה היא שילמה אותו

ולכן כאשר אנחנו מעבירים פקודת ימן היא חייבת להיות מאוזנת, מכון שהסכם שゾכה (זאת אומרת שלום)  
לספק חייב להיות קיימם גם בחיוב בעירייה

חלוקת לסעיפים תקציביים:

כל עירייה מתנהלת עם סעיפים תקציביים בצד לדוח ולנהל את ההוצאות שלה,  
למשל לכל עירייה יש סעיף תקציבי שמתעסק בחינוך, או בתשתיות או באבטחה וכו'  
בכל שנה הסעיף התקציבי זהה מתוקצב בכמות מסוימת של כסף, שזו הכמות אותה הוא יכול "לבזבז" בשנות  
הכספיים

ולכן עבור כל ספק קיימם סעיף תקציבי אשר **מחויב על סך העסקאות מולו**.

מהצד של הספק, כל גביה של כסף מתבצעת לרוב אל מול חוזה מסוים, כאשר החוצה יכול לייצג מקום אספקה או  
סוג מסוים של שירות ועוד

ולכן הקשר הלוגי בין חוזים לסעיפים תקציביים הוא כדלקמן:  
"לכל חוזה קיימם לפחות סעיף תקציבי אחד, ותחת סעיף תקציבי אחד יכולים להיות כמה חוזים"  
כלומר, קשר של רבים לרבים.

כאשר יוצרים פקודת ימן אנחנו צריכים להבין מי הסעיף התקציבי **המחויב** עבור חוזה מסוים, ומומלץ לבצע  
 הפרדה של החובות בכל חשבונית וחשבונית  
כלומר, אם סעיף תקציבי מסוים מחויב כמה פעמים בכמה חשבוניות שונות, מומלץ שתהיה שורת חובה של  
הסעיף עבור כל חשבונית

ולבסוף כל שורות החובה האלו צריכים להיות שוות לתנועת **זכות** אחת של הספק  
כלומר, **חייבנו את העירייה וחילקנו את החובה**zzo לשיעיפים תקציביים שונים  
וזיכינו ספק יחיד אל מול הסכם **שחייבנו את העירייה**.

