# שכבת הגישה לבסיס הנתונים – שכבת ה-DAL

השכבה המקשרת בין שכבת הלוגיקה (ה-BL) ובסיס הנתונים עצמו.

השכבה מאפשרת לקוד הפרוייקט לגשת לבסיס הנתונים בצורה שקופה, וללא צורך בהכרת סוג בסיס הנתונים בו נעשה שימוש. השכבה מסתירה את הפרטים הטכניים הקטנים מהמשתמש, ומאפשרת למתכנת לעבוד עם אובייקטים במקום עם טבלאות ורשומות.

השכבה משתמשת ב-Entity Framework לביצוע ההמרה בין האובייקטים הבסיסיים איתם עובדת ADO.NET – DataTables המכילות רשומות, ושאילתות ה-SQL המשמשות לתקשורת עם בסיס הנתונים ושליפת/עדכון אותן טבלאות לאובייקטים אותם היא חושפת למשתמש.

בפרוייקט זה, שכבה זו ממירה את האובייקטים שנוצרים ע"י ה-Entity Framework לאובייקטים המוכרים בשאר הפרוייקט, בספריית Common. אבחנה זו בין Internal Objects – האובייקטים הפנימיים הקיימים ב-EntityFramework ואובייקטים חיצוניים, אליהם פונים שאר רכיבי הפרוייקט, ואותם ניתן לחשוף אף החוצה, לטובת הרחבות עתידיות, מאפשרים להתמודד בקלות עם המקרים בהם האובייקט שנוצר אוטומטית ע"י ה-EntityFramework לא בדיוק מתאים ל-Design שלנו, או מוסיף תכונות ושיטות שאנו לא מעוניינים לחשוף החוצה.

המרה זו מאפשרת להחליף את שכבת ה-DAL בקלות יתרה, כיוון שלא קיים צימוד בין בסיס הנתונים והאובייקטים אותם מייצרת ה-Entity Framework לבין האובייקטים מולם פועל יתר הפרוייקט.

## עקרונות מנחים

השכבה נותנת שירותי שליפה, עדכון והכנסה של נתונים מבסיס הנתונים, באופן שקוף, וללא ביצוע ולידציה, אותנתיקציה או כל שירות אחר הממומש בשכבת הלוגיקה.

שכבה זו, הממומשת כספריה, תימצא על השרת עליו מותקנים רכיבי שכבת הלוגיקה

מאפייני ההתקשרות עם בסיס הנתונים יאוחסנו בקובץ ההגדרות של השכבה, כך שהספרייה לא תכלול את פרטי השרת, בסיס הנתונים וה-credentials המשמשים להזדהות מולו.

כך נוכל לבצע שינויים במיקום בסיס הנתונים (למשל, לעבור לשרת חזק יותר) ללא צורך בהוצאת גרסה חדשה של המערכת

במקרה כישלון כל השיטות בכל המחלקות מחזירות Exception – החריגות יהיו Custom Exceptions כך שגם בהחלפת בסיס הנתונים, מבחינת הקוד בשכבות מעל יוחזרו אותם ה-Exceptions ולא יהיה צורך בשינוי קוד, וההתנהגות תישמר.

## מחלקות כלליות

### Converter

מחלקה המכילה שיטות סטאטיות המשמשות להמרה של אובייקט מ-common לאובייקט של ה-EntityFramework ובחזרה.

השיטות במחלקה כולן מוגדרות Internal ויהיו נגישות רק בתוך ה-DAL

### Exceptions

מחלקה זו מכילה את כל ה-custom exceptions שנועדו לפעפע חריגות בעבודה מול בסיס הנתונים כלפי מעלה. בינהן ניתן למנות את חריגות בשל מידע לא קיים (ניסיון לעדכן רשומה לא קיימת), חריגות בשל כפילויות (ניסיון להכניס רשומה נוספת עם מפתח ראשי זהה לרשומה קיימת) וכן חריגות כלליות שיזרקו בשל אי זמינות של בסיס הנתונים.

כל החריגות הן מחלקות היורשות ממחלקת הבסיס Exception ומממשות את שלושת הקונסטרקטורים הנפוצים:

Exception(), Exception(String message), Exception(String message, Exception inner)

### Exceptions.DALConnectionError

חריגה זו נזרקת כאשר יש תקלה/שגיאה בהתחברות לבסיס הנתונים

Message עשוי לכלול פרטים לגבי השגיאה (שרת לא זמין, משתמש וסיסמא לא נכונים וכולי)

Inner יכלול את החריגה המקורית שנזרקה מבסיס הנתונים

### Exceptions. DALDuplicateKeyError

חריגה זו נזרקת כאשר נעשה ניסיון לבצע insert לרשומה עם מפתח ראשי שכבר בשימוש ע"י רשומה קיימת.

Message עשוי לכלול פרטים לגבי השגיאה (איזה ערכים הועברו, וכד')

Inner יכלול את החריגה המקורית שנזרקה מבסיס הנתונים

### Exceptions. DALZeroRowsAffectedError

חריגה זו נזרקת כאשר עדכון (update) שהורץ מול בסיס הנתונים לא עדכן אף רשומה. מצב זה עשוי לקרות כאשר רשומה שעבדנו על השינוי שלה כבר נמחקה ע"י מישהו אחר – מצב תקין לגמרי. לדוגמא – כאשר נרצה להוסיף תשובה לשאלה ועד שנגמור לערוך את התשובה רכז הקורס כבר מחק את השאלה.

### Exceptions. DALConstraintError

חריגה זו נזרקת כאשר שינוי כלשהו בנתונים (בין אם בהכנסה ובין אם בעדכון) גורם לסטייה מהגבולות המוגדרים בבסיס הנתונים. למשל – אם מנסים להכניס רשומה חדשה בלי לציין ערכים לשדות שלא מאפשרים ערכי null.

## מחלקות לגישה לבסיס הנתונים

### שירותי משתמש

המחלקה מאפשרת פעולות הקשורות לחשבון משתמש, כגון יצירת משתמשים, עדכון פרטי משתמשים, ושליפת פרטי התרעות למשתמשים

#### הוספת משתמש חדש

מקבלת אובייקט משתמש חדש ופרטי הזדהות, ומוסיפה אותו לבסיס הנתונים. לאחר ההוספה מחזירה אובייקט משתמש, עם שדה ID לפי המספור האוטומטי בבסיס הנתונים

public UserInfo CreateNewUser(UserInfo userInfo, UserCredentials userCredentials)

#### כניסה למערכת

מקבלת פרטי הזדהות ומחזירה ערך Boolean

public bool Login(UserCredentials userCredentials)

#### שליפת כל משתמשי המערכת

מחזירה את כל המשתמשים הרשומים בטבלת המשתמשים

public IEnumerable<UserInfo> GetAllUsers()

#### שמירת שינויים בחשבון משתמש

public void SaveUserData(UserInfo userinfo)

#### שמירת שינויים ברשימת חשבונות משתמשים

משתמשת בשמירת שינויים בחשבון משתמש עבור כל החשבונות שהועברו

public void SaveUsersData(IEnumerable<UserInfo> usersInfo)

#### שליפת פרטי משתמש

מחזירה את פרטי המשתמש המבוקש

### public UserInfo GetUserInfo(int userId)

### שירותי POST

מחלקה זו משמשת לפעולות על posts – כלומר שאלות ותשובות במערכת. שליפת שאלות ותשובות, הוספת שאלות ותשובות ועדכון נתונים לגבי שאלות ותשובות.

כל הפעולות שמשנות אובייקט מבצעות את הפעולה או זורקות Exception במקרה שגיאה.

כל הפעולות שמשנות אובייקט שאלה או תשובה מגדילות את ה-version של השאלה ב-1.

#### הוספת שאלה

public Question AddQuestion(Question question)

#### מחיקת שאלה

public void DeleteQuestion(int questionId)

#### הוספת תשובה

public Answer AddAnswer(Answer answer)

#### מחיקת תשובה

public void DeleteAnswer(int answerId)

#### סימון שאלה כמועדפת

public void RecommendQuestion(int questionId)

#### הצבעה חיובית לשאלה

public void VoteUpQuestion(int questionId)

#### הצבעה שלילית לשאלה

public void VoteDownQuestion(int questionId)

#### הצבעה חיובית לתשובה

public void VoteUpAnswer(int answerId)

#### הצבעה שלילית לשאלה

public void VoteDownAnswer(int answerId)

#### העלאת מספר גרסה לשאלה

public void IncrementVersion(int questionId)

#### חיפוש חופשי

מחזירה את כל השאלות שעונות על פרטי החיפוש, אשר מתבצע בבנושא השאלות ובתוכן השאלות.

public IEnumerable<Question> FreeSearch(string searchString)

#### חיפוש לפי תגיות

מחזירה את כל השאלות שאחת מהתגיות שסופקו מקושרת אליה

public IEnumerable<Question> TagsSearch(IEnumerable<string> tags)

#### שליפת כל השאלות של משתמש מסויים

public IEnumerable<Question> GetQuestionsByUser(int userId)

#### שליפת שאלה וכל התשובות שלה

public DiscussionThread GetDiscussionThreadById(int id)

#### שליפת שאלה

מתודה פרטית המשמשת את המתודה הציבורית המאחזרת את כל ה-thread

private Question GetQuestion(int it)

#### שליפת תשובות לשאלה

מתודה פרטית המשמשת את המתודה הציבורית המאחזרת את כל ה-thread

private IEnumerable<Answer> GetAnswers(Question question)

מתודות המשמשות לשמירת היסטוריית השימושים במערכת, כך שמשתמש לא יוכל להעלות דירוג של שאלה יותר מפעם אחת. המתודות נקראות ע"י ה-BL, ולכן ציבוריות.

public void RankQuestion(int questionId, int userId, int rank)

public void RankAnswer(int answerId, int userId, int rank)

מתודות המשמשות לשליפת היסטוריית הדירוג של המשתמש לשאלות / תשובות

public int GetQuestionRankingHistory(int questionId, int userId)

public int GetAnswerRankingHistory(int answerId, int userId)

### שירותי התרעות Notifications

מחלקה זו משמשת לטיפול בנושא ההתרעות. היא מכילה מתודות לשמירה של התרעות לבסיס הנתונים ולשליפת התרעות מבסיס הנתונים.

המחלקה עושה שימוש ב-NotificationTyps enum המכיל את סוגי ההתרעות השונים, אשר מאוחסנים בבסיס הנתונים ע"פ המזהה המספרי שלהם. המתודות הסטאטיות במחלקה Converter משמשות להמרה בין אובייקט ההתרעה החיצוני (מ-common) והפנימי (מה-EntityFramework)

#### שמור התרעה חדשה

מתודה הנקראת ע"י ה-BL עבור כל פעולה שנרצה להתריע בפני המשתמש כי ארעה.

public void SaveNotification(int userId, string message, NotificationType notificationType)

#### מתודה לאחזור התרעות

מתודה לשליפת כל ההתרעות עבור משתמש מאז תאריך/זמן מסויים

public IEnumerable<Notification> GetNotifications(int userId, DateTime since)