מעבדת תרגול – סרט ודירוגים

**סעיף 1 – פתיחת פרויקט חדש**

1. ממשו צד שרת בטכנולוגיית Spring Framework עם הDependencies הבאים:

* Lombok
* Spring Web
* Spring Data / Jpa
* MySQL Driver או H2 או כל מסד נתונים אחר לשיקולכם

1. הגדירו את קובץ application.properties באופן הבא:

על מנת לקצר תהליכים, היעזרו בקונפיגורציה הבאה (מתאימה ל- MySQL): <https://gist.github.com/KobiShashs/c99c7c05c8e1c4fe300786a5d2badb79>

או בקונפיגורציה הבאה (מתאימה ל-H2):  
<https://gist.github.com/KobiShashs/dfefb343b20b629aecc587325a2a1ddc>

1. החליפו את הלוגו של Spring Boot באיורASCII מגניב של ספורט!

**סעיף 2 – הגדרת Beans**

ממשו את המחלקות הבאות (יש להיעזר בספריית Lombok) :

* Rating (Enum)
  + ONE\_STAR("⭐"),
  + TWO\_STARS("⭐⭐"),
  + THREE\_STARS ("⭐⭐⭐"),
  + FOUR\_STARS ("⭐⭐⭐⭐"),
  + FIVE\_STARS("⭐⭐⭐⭐⭐");
* Rate
  + id: int – PK, AI
  + name: string
  + rating: Rating
  + comment: String
* Movie
  + id: int – PK, AI
  + name: String – Not Null, Max Length 40
  + year: int
  + rates: List<Rate> - @OneToMany

**סעיף 3 – הגדרת Repositories**

הגדירו את הממשקים הבאים:

* MovieRepository
* RateRepository

**סעיף 4 – הגדרת נתונים גולמיים**

באמצעות מתודה main או ע"י שימוש ב-CLR הכניסו 3 סרטים בעלי 3 דירוגים לפחות לכל אחד

**סעיף 5 – טיפול ב-Exceptions**

ממשו את המחלקה הבאה:

* CinemaCustomException

**סעיף 6 – הגדרת Service – חלק א**

טיפ של אלופים : הפרידו בין "מה" לעשות ו-"איך" לעשות

הגדירו את הממשק הבא ("מה לעשות"):

* AdminService

הגדירו את המחלקה הבאה ("איך לעשות"):

* AdminServiceImpl

**להלן פונקציונליות השירות:**

1. הוספת סרט חדש
   * הניחו כי סרט חדש מגיע ללא דירוגים.
2. מחיקת סרט לפי שדה id
   * במידה ושדה id לא קיים במסד הנתונים - יש לזרוק חריגה
   * במידה והסרט צבר דירוג ממוצע של 3.5 כוכבים – יש להציג חריגה  
     "סרט איכותי לא מוחקים", אחרת ניתן למחוק
3. קבלת כלל הסרטים
4. קבלת סרט לפי שדה id
   * במידה ושדה id לא קיים במסד הנתונים - יש לזרוק חריגה
5. קבלת הסרט החדש ביותר (במידה ויש כמה החזירו אחד)
   * במידה ואין סרטים – יש לזרוק חריגה
6. קבלת הסרט הותיק ביותר (במידה ויש כמה החזירו אחד)
   * במידה ואין סרטים – יש לזרוק חריגה
7. קבלת כלל הסרטים המתחילות **ברישא** מסוימת
8. קבלת כלל הסרטים הטובים (ממוצע מעל 3.5)
9. קבלת כלל הסרטים הסבירים (ממוצע 2.5 עד 3.5 כולל קצוות)
10. קבלת כלל הסרטים הגרועים (ממוצע מתחת ל2.5)
11. קבלת כלל הסרטים הטובים ממויינים ע"פ סה"כ דירוגים
12. קבלת כלל הסרטים הטובים ממויינים ע"פ מדד הדירוג בסדר עולה
13. קבלת כלל הסרטים הטובים ממויינים ע"פ מדד הדירוג בסדר יורד
14. חישוב סה"כ דירוג **מצטבר** של הסרט המדורג גבוה ביותר

**סעיף 6 – הגדרת Service – חלק ב**

טיפ של אלופים : הפרידו בין "מה" לעשות ו-"איך" לעשות

הגדירו את הממשק הבא ("מה לעשות"):

* CustomerService

הגדירו את המחלקה הבאה ("איך לעשות"):

* CustomrServiceImpl

**הערה חשובה**

**אם תבינו אותה, אז תבינו היטב את ההבדל בין Singleton ל-Prototype**

**את השדה id המזהה את מספר הקבוצה תוסיפו כארגומנט לכל אחת ואחת והפונקציות הבאות.**

**להלן פונקציונליות השירות:**

1. הוספת דירוג לסרט (ע"י מזהה id )
   * במידה ושדה id לא קיים במסד הנתונים - יש לזרוק חריגה
2. עדכון דירוג לסרט (ע"י מזהה id),
   * במידה ושדה id לא קיים במסד הנתונים - יש לזרוק חריגה

**סעיף 7 – כתיבת יחידת בדיקה**

באמצעות מתודה main או ע"י שימוש ב-CLR הוכיחו את הפונקציונליות שלAdmin Service & Customer Service

**סעיף 8 – הגדרת Controller**

ממשו את המחלקה הבאה:

* AdminController
* CustomerController

**סעיף 9 – הגדרת Controller Advice**

ממשו את המחלקה הבאה:

* CinemaControllerAdvice

**סעיף 10 – בדיקות Http Client - Postman**

בדקו את עצמכם יצירת Postman Collection

**סעיף 11 – בדיקות Http Client - Swagger**

בדקו את עצמכם – חשפו את ה-Api ע"י הטעמת SWAGGER

**סעיף 12 – בדיקות Http Client – Rest Template**

בדקו את עצמכם באופן תכנותי ע"י Rest Template, יש לבדוק גם מקרי קצה

**בהצלחה!**