

עבודה עם node.js

כתובת הפונקציה בצורה אי-סינכרונית - פרומיסים (2)

```
connectToDB().then(async()=>{
  console.log('connected!');
  const bookSchema = new mongoose.Schema({
    name:String,
  })
  const book=mongoose.model("book",bookSchema);
  const data = await book.find({name:/ש/}).exec();
  console.log(data)})
```

פונקציות החיפוש:

```
book.find({numPage:$gt:100}).exec()
```

אליציה - בדיקת תקינות נתונים

```
const bookSchema = new mongoose.Schema(
  {
    name:{
      type: String,
      required: true,
    }
  })
```

הודעת שגיאה למשתמש

```
try{
  await shatul.save();
} catch (e) {
  console.error("Error",e_message);
}
```

אפשרות של הקלדה מתוך רשימה - enum

```
typeBook:{
  type: String,
  enum: ['children', 'adult'],
}
```

נדרש רק במקרים מסוימים

```
required:function(){
  return this.pageNum>300;
```

ערכים מסוימים, מינימום ומקסימום

```
pageNum:{
  type: Number,
  min:[100, "its a few pages"],
  max:[500, 'its a lot!!!'],
},
```

תנאי אילו נתונים יוצגו:

```
Where({name:"Israel"})
```

סיונו של מספר מסמכים ראשונים

```
skip(10)
```

מספר המסמכים שיוצגו על המסך

```
limit(2)
```

מיון מהנמוך לקטן או מהקטן לגדול

```
sort({age:1})
```

בחירה אילו שדות יוצגו

```
select({name:true})
```

שימוש בפונקציה עם פרומיסים

```
exec()
```

עבודה

(1) בני סכמה חדשה של מוצרים בחנות שכל המוצרים נמכרים עד 10 ש.

הסכמה תכלול:

- שם המוצר
- מחיר (בין 1-10 ש)
- סוג המוצר (יכול להיות רק כמה אפשריות).
- לאיזה גיל מיועד (שדה נדרש במקרה שמחיר המוצר הוא 10 ש).
- שדה נוסף לפי בחירתך.
- (2) בני מודל לסכמת המוצרים.
- (3) הכניסי לאוסף המוצרים חמישה מוצרים (על פי הסכמה)
- (4) הציגי את כל אוסף המוצרים (על ידי שימוש בפרומיסים). התעלמי מהמוצר הראשון. הציגי רק 2 מוצרים.
- (5) הציגי רק את המוצרים שמחירים שווה ל- 10 ש. השתמשי בפונקציית where
- (6) הציגי את כל המוצרים שמחירים קטן מ- 3 ש. הציגי רק את שם המוצר.
- (7) אם יש שגיאות ולדיציה, הציגי על המסך (השתמשי ב try catch).