

# Roo Code הדרכה על

ברוכים הבאים להדרכה על התוסף Roo Code לעבודה חכמה בעזרת Al ב-Visual Studio

#### מטרות ההדרכה:

- ללמוד את יכולות התוסף ולהבין את הפוטנציאל שלו לשיפור תהליכי עבודה
  - להבין כיצד לשלב את Roo Code בסביבת העבודה היומיומית●
- אריך: 7.7.2025. האריך: NET-ו Angular לראות דוגמאות מעשיות בעבודה עם NET-ו
  - Unit Tests ויצירת Code Review להכיר את היכולות המתקדמות כמו ביצוע

## ?Roo Code מה זה

Roo Code הוא תוסף עוצמתי מבוסס Al ל-VSCode המאפשר להאיץ פיתוח, לבצע סקירות קוד, להריץ טסטים, לאוטומציה של דפדפן ועוד - הכל בשפה טבעית ובאינטגרציה מלאה לסביבת הפיתוח.



### >\_

#### פקודות טרמינל

הרצת פקודות, התקנות חבילות, הפעלת שרתי פיתוח או בדיקות - הכל ישירות מהצ'אט.

## ניהול קבצים וקוד

קריאה וכתיבת קבצים ישירות במרחב העבודה, עריכת קוד, ותיקון שגיאות אוטומטי.

#### תקשורת בשפה טבעית

שאל שאלות, בקש שינויי קוד או ייצור פרויקטים שלמים בשפה טבעית - בדיוק כמו שהיית מסביר למתכנת אחר.



### 7

#### מצבי עבודה מותאמים

התאמת האסיסטנט לתפקידים שונים: מפתח קוד, ארכיטקט מערכת, QA, או מנהל מוצר.

#### אוטומציית דפדפן

פתיחת דפדפן לבדיקת האפליקציה, ביצוע קליקים, מילוי טפסים ולכידת שגיאות.



#### תמיכה במודלים מרובים

עבודה עם OpenAl, Anthropic, Google Gemini ואפילו מודלים מקומיים - הבחירה שלך.

Roo Code Logo

## התקנה וקונפיגורציה



#### שלב 1: התקנת התוסף

- VS Code Marketplace-ב "Roo Code" חפשו
  - לחצו על "Install" להתקנת התוסף
  - VSCodium עבור Open-VSX או התקינו דרך
- לחלופין, הורידו קובץ VSIX והתקינו באופן ידני



#### שלב 4: הגדרת פרופיל ראשוני

שלב 3: פתיחת Boo Code

בסרגל הפעילות Roo Code • לחצו על האייקון של

"Roo Code: Open" ואז "Ctrl+Shift+P • או השתמשו בקיצור

"Roo Code: Open in New Tab" - מומלץ לפתוח בטאב נפרד

- (Code, Architect, Ask) בחרו מצב עבודה
  - הגדירו הרשאות אוטו-אישור לפעולות
- התאימו את הכללים לנוהלי העבודה בארגון
  - שמרו תצורות לפרויקטים שונים



#### שלב 2: הגדרת מודל Al

- Roo Code (Settings) פתחו את הגדרות
- (וכו') OpenAl, Anthropic) למודל הרצוי API Key הוסיפו
  - (וכו') GPT-4, Claude) הגדירו את המודל המועדף
- ניתן להשתמש במודלים מקומיים כמו Ollama/LM Studio



ניתן להגדיר קבצי .clinerules בתיקיית הפרויקט כדי לקבוע הגדרות ספציפיות לפרויקט, כולל הגבלת הרשאות או הנחיות מותאמות אישית למצבי עבודה שונים.

## (Chat Modes) מצבי עבודה

מציע מספר מצבי עבודה ייעודיים המאפשרים לך להתאים את סגנון האינטראקציה בהתאם למשימה Roo Code הנוכחית. לכל מצב יש יכולות והרשאות שונות.



### מצב קוד (Code)

<u>מצב בריר</u>ת המחדל המיועד לכתיבה, עריכה ותיקון קוד. מאפשר הרצת פקודות, יצירת קבצים, ואוטומציה מלאה של משימות פיתוח. אידיאלי לעבודת קידוד יומיומית ופיתוח פרויקטים.

> להריץ פקודות טרמינל 🤡 יכולת לכתוב ולערוך קבצים 🕜



באופן ישיר.

🗷 ללא יכולת לכתוב קוד 🕟 חשיבה ארכיטקטונית

(Architect) מצב ארכיטקט

### מצבים מותאמים אישית (Custom)

יצירת מצבי עבודה ייחודיים המותאמים למשימות ספציפיות. למשל: מהנדס OA לבדיקות, מנהל מוצר, או מעצב UX. ניתן להגדיר את הפרומפטים וההרשאות בכל מצב.

מיועד לחשיבה מעמיקה על עיצוב וארכיטקטורת מערכת. מתמקד בתכנון ברמה

גבוהה, פתרון בעיות מורכבות ותכנון מערכות. אינו כותב קוד או מריץ פקודות

- התאמה אישית מלאה 🛮 🔡 הגדרת הרשאות גמישה 🐼

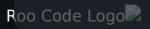




### מצב שאלות (Ask)

מצב המיועד לחקירה ולמידה. מתמקד במתן תשובות מעמיקות, הסברים על קוד קיים, ועזרה בפתרון בעיות. א<sup>י</sup>דיאלי להבנת מושגים או לניתוח בסיס קוד קיים.

🛾 ללא יכולת לכתוב קוד 💮 הסברים והדרכה מעמיקה



## יצירת פרויקט Angular לדוגמה

#### המשתמש 🔼

אור אפליקציית Angular בסיסית עם קומפוננטה bello-world

#### Roo Code 📠

Roo Code

אני אצור עבורך אפליקציית Angular בסיסית עם קומפוננטת hello-world. ראשית אבדוק אם Angular CLI מותקן ואז אתחיל ביצירת הפרויקט.

- הקמת פרויקט Angular CLI והקמת פרויקט

  npm install -g @angular/cli \$

  ng new hello-world-app --routing=false --style=css \$

  cd hello-world-app \$

  ng generate component hello-world \$

  ng serve \$
- hello-world.component.ts

  ;'import { Component } from '@angular/core

  })Component@

  ,'selector: 'app-hello-world

  ,'templateUrl: './hello-world.component.html

  styleUrls: ['./hello-world.component.css']

  {
  } export class HelloWorldComponent

  ;'!Hello World שוב הבאים לקומפוננטת = message: string

  {
  } ()constructor

  } ()changeMessage

  i this.message

  this.message

#### צעדים ליצירת פרויקט:

- בקשה בצ'אט **בקשה בצ'אט** שליחת בקשה טבעית ל-<u>Roo Cod</u>e
- הכנה ואיתחול Angular ווצר פרויקט Roo Code
- יצירת קומפוננטות יצירת hello-world ועדכון
  - **הפעלת האפליקציה** הרצת ng serve ישירות מהצ'אט

#### טיפ 🥊

ניתן לבקש מ-Roo Code להוסיף יכולות נוספות או לשנות את העיצוב בבקשה הבאה



Roo Code Logo

# דוגמה: פרויקט .NET חדש

#### צעדים ליצירת פרויקט .NET:

- בקשה בצ'אט 🚺
- הגדרת הדרישות בשפה טבעית ל-Roo Code
- יצירת פרויקט dotnet new יוצר שלד עם פקודת Roo Code
  - הוספת קונטרולרים 3
- יצירת קונטרולרים והוספת נקודות קצה (Endpoints)
  - הרצת האפליקציה 4 Swagger-ובדיקה dotnet run

טיפ 🍨

ניתן לבקש מ-Roo Code להוסיף שכבות נוספות כמו שירותים, מודלים וחיבור למסד נתונים



#### המשתמש 🔼

צור אפליקציית ASP.NET WebApi פשוטה עם קונטרולר ברירת מחדל

#### Roo Code



אצור עבורך פרויקט ASP.NET WebApi בסיסי עם קונטרולר. נתחיל ביצירת הפרויקט ואז נוסיף את הקונטרולר.

```
ASP.NET WebApi יצירת פרויקט $
               dotnet new webapi -n MyApiProject $
                                 cd MyApiProject $
       dotnet add package Swashbuckle.AspNetCore $
dotnet new controller -n Products -o Controllers $
                                    dotnet build $
                                      dotnet run $
```

#### **ProductsController.cs**

```
;using Microsoft.AspNetCore.Mvc
                                    namespace MyApiProject.Controllers
                                                    [ApiController]
                                        [Route("api/[controller]")]
                  public class ProductsController : ControllerBase
                ()private static readonly List _products = new
,{ Price = 99.99m ,"מוצר ראשון" = Id = 1, Name } new Product
  { Price = 149.99m , "מוצר שני" = Id = 2, Name } new Product
                                                             ;{
                                                      [HttpGet]
                                  ()public ActionResult> GetAll
                                          ;return _products
                                              [HttpGet("{id}")]
                           public ActionResult GetById(int id)
            ;var product = _products.Find(p => p.Id == id)
                   ;()if (product == null) return NotFound
                                           ;return product
                                               public class Product
                                    public int Id { get; set; }
                        ;"" = public string Name { get; set; }
                            public decimal Price { get; set; }
```

## Roo Code עם Code Review ביצוע

מאפשר לבצע סקירות קוד מקצועיות עם הנחיות וניתוחים חכמים לשיפור האיכות, אבטחה ויעילות Roo Code הקוד. הוא יכול להפנות לבעיות, להציע פתרונות, ואף ליישם אותם אוטומטית באישורכם.

```
Roo Code אל ידי אבטחה לסקירת קוד על ידי 'Roo Code '' אים בעיית אבטחה בפונקציית '' Roo Code '' אים בעיית אבטחה בפונקציית '' Roo Code '' אין Injection '' Injection executeQuery(userInput) '' Injection execute(query) '' Injection '' Injection execute(query) '' Injection '' Inject
```

### יתרונות שימוש ב-Roo Code לסקירת קוד:

- איתור בעיות אבטחה, ביצועים ותחזוקה
  - הצעת פתרונות אוטומטיים
- למידה מטעויות וקבלת הסברים מפורטים
  - תמיכה בריבוי שפות ומסגרות פיתוח
  - VSCode אינטגרציה מלאה עם סביבת •

#### ?Code Review איך לבצע

- הוסף קונטקסט 🔟
- file@ כדי להכניס קבצים לסקירה: mentions כדי להכניס קבצים לסקירה: git show HEAD~1@ או src/app.component.ts
- **בקש ביקורת ספציפית** לדוגמה: "בצע ביקורת קוד על הקבצים האלו, התמקד בבעיות אבטחה ויעילות"
  - קבל הערות וסקירה Roo Code ינתח, יספק הערות וימליץ על תיקונים
  - תקן באופן אוטומטי Roo Code אפשר ל-Roo Code לתקן בעיות אוטומטית או בצע אַת התיקונים בעצמך

## יצירת Unit Tests אוטומטית

#### :איך Roo Code מייצר טסטים

- ניתוח הקוד
- זיהוי מבנה הקוד, מחלקות, תלויות, ופונקציות שדורשות בדיקה
  - זיהוי תבניות בדיקה 🌘
- התאמת תבניות בדיקה למבנה הקוד ולטכנולוגיה (NET/xUnit. או Angular/Jest)
  - יצירת מוקים ותמיכה 💅
  - יצירה אוטומטית של מוקים ותמיכות נדרשות לבדיקה מבודדת
    - יצירת מקרי בדיקה
  - יצירת מקרי בדיקה ספציפיים לכל מחלקה ומתודה עם נתוני דוגמה
    - וידוא כיסוי בדיקות
    - מעקב אחר כיסוי הבדיקות והצעת בדיקות נוספות במידת הצורך

#### שילוב מהיר 🖋

Roo Code יכול להריץ את הבדיקות באופן אוטומטי ולתקן שגיאות שמתגלות

#### המשתמש 🔼

תכתוב unit tests עבור ה-Service הזה ב-Angular

#### Roo Code



אשמח לכתוב עבורך unit tests ל-Service. אני אבחן את הקוד ואיצור קובץ טסט מתאים עם Jest.

```
;using Xunit
                       ;using Moq
   ;using System.Threading.Tasks
;using System.Collections.Generic
```

```
public class UserServiceTests
;private readonly Mock _mockRepo
```

UserServiceTests.cs (.NET/xUnit)

```
;private readonly UserService _service
                 ()public UserServiceTests
               ;()mockRepo = new Mock_
e = new UserService(_mockRepo.Object)_
                                    [Fact]
```

```
public async Task GetAllUsers_ReturnsUsers
                           Arrange //
         var expectedUsers = new List
 "ם " = Id = 1, Name } new User
 "2 שם = Id = 2, Name } new User
                                   ;{
```

```
mockRepo.Setup(repo => repo_
              (()GetAllAsync.
;ReturnsAsync(expectedUsers).
```

```
sult = await _service.GetAllUsersAsync
```

```
ual(expectedUsers.Count, result.Count)
expectedUsers[0].Name, result[0].Name)
```

```
user.service.spec.ts (Angular)
```

```
import { TestBed } from '@angular/core/testing
             ,import { HttpClientTestingModule
troller } from '@angular/common/http/testing
 ;'import { UserService } from './user.service
               } <= () ,'describe('UserService</pre>
                    ;let service: UserService
        ;let httpMock: HttpTestingController
                           } <= ())beforeEach</pre>
          })TestBed.configureTestingModule
     ,imports: [HttpClientTestingModule]
                providers: [UserService]
    ;service = TestBed.inject(UserService)
ck = TestBed.inject(HttpTestingController)
             } <= () ,'it('should be created</pre>
             ;()expect(service).toBeTruthy
                                           ;({
} <= () ,'it('should retrieve users from API</pre>
                        ] = const mockUsers
                ,{ 'שם ' :id: 1, name }
                 { 'שם 2' :id: 2, name }
                                          ; [
   } <= service.getUsers().subscribe(users</pre>
       ;expect(users).toEqual(mockUsers)
                                         ;({
onst req = httpMock.expectOne('api/users')
   ;expect(req.request.method).toBe('GET')
                      ;req.flush(mockUsers)
                                          ;({
```

```
$ הרצת הטסטים
```

```
# להרצת טסטים ב-Angular
              ng test $
  # להרצת טסטים ב-.NET
```

dotnet test \$

Made with Genspark 🛨

Roo Code Logo

# יכולות מתקדמות: אוטומציות ודפדפן



#### **Browser Automation**

Roo Code מאפשר לפתוח ולהפעיל דפדפן ישירות מהצ'אט, ובכך להאיץ תהליכי בדיקות וניפוי באגים.

- פתיחת דפדפן לבדיקת אפליקציה מקומית
  - ביצוע פעולות: קליקים, טפסים, ניווט
    - לכידת צילומי מסך ושגיאות קונסול
- אוטומטיות end-to-end הפעלת בדיקות

### ל<u>י> ד</u>וגמת שימוש:

עם מערכות וכלים נוספים.

AWS או Jira חיבור לשירותים חיצוניים כמו•

התאמה לתהליכי עבודה ספציפיים בארגוןהרחבה קלה של יכולות האסיסטנט

• יצירה אוטומטית של כלים חדשים לפי דרישה

#### // פקודה לצ'אט

צור כלי חדש שיאפשר לי לחפש ולקבל נתונים מה-API של Jira ולסנן משימות לפי סטטוס וקדימות.

הוספת כלים מותאמים אישית ל-Roo Code להרחבת היכולות שלו ואינטגרציה

Model Context Protocol (MCP)

#### </> דוגמת שימוש: ⟨/>

#### ע פקודה לצ'אט //

בדוק את האפליקציה בדפדפן, נווט לעמוד הבית ולחץ על כפתור ההרשמה. צלם מסך של התוצאה והדפס שגיאות קונסול אם יש.

### שילוב יכולות מתקדמות

ניתן לשלב את שתי היכולות יחד - למשל, ליצור כלי MCP שמפעיל בדיקות דפדפן אוטומטיות לפי לוח זמנים, או שולף נתונים מ-API חיצוני ומציג אותם בדפדפן.



# (Custom Modes) מותאמים אישית Modes

מאפשרים לך להתאים את Roo Code לתפקידים ומשימות ספציפיות, עם הנחיות, הרשאות וכלים Custom Modes מותאמים לכל מצב עבודה.

#### איך יוצרים מצב מותאם אישית:

#### פתח צ'אט והקלד פקודה 1

'[שם התפקיד] Create a new mode for' הקלד בצ'אט: לדוגמה: 'Create a new mode for QA Engineer'

#### 2 התאם את הפרומפט

עבור לטאב 'Prompts' בתפריט העליון התאם את התיאור, ההוראות והפרסונה של המצב

#### 3 הגדר הרשאות וכלים

בחר אילו כלים המצב יוכל להשתמש בהם הגבל פעולות מסוכנות או הוסף יכולות מיוחדות

#### שמור והתחל להשתמש

בחר את המצב החדש מהתפריט הנפתח בתחתית הצ<sup>י</sup>אט הפעל עם מודל Al מתאים לצרכי המשימה

#### דוגמאות למצבים מותאמים:

### QA Engineer 💃

מתמחה בכתיבת טסטים, חיפוש באגים ואיתור מקרי קצה שלא נלקחו בחשבון

### Code Reviewer **Q**

מתמקד בסקירת קוד מעמיקה, איתור בעיות אבטחה ושיפור יעילות הקוד

### UI/UX Designer

מומחה בעיצוב ממשקים, נגישות ושיפור חוויית משתמש

### **Product Manager ₹Ξ**

עוזר בתכנון מוצרים, כתיבת סיפורי משתמש ותיעדוף יכולות

#### טיפ שימושי 🥝

תוכל לשתף ולייבא מצבים מותאמים אישית מהקהילה ב-Reddit (r/RooCode) או ליצור קבצי .clinerules לשמירת מצבים לשימוש חוזר בפרויקטים שונים.



## טיפים ושיטות עבודה מומלצות





#### בדיקת אוטומציות בקוד חדש

השתמשו בסביבת פיתוח נפרדת לבדיקת הקוד שנוצר ע"י Roo Code. כך תוכלו לוודא את איכות הפתרונות לפני שילובם בענף הראשי של הפרויקט.



בנקודות קריטיות.

#### הגדרות הרשאות מתאימות

Roo Code התאימו את רמת האוטונומיה של לפרויקט. בפרויקטים רגישים הקפידו על אישור ידני של כל פעולה, ובפרויקטים פחות קריטיים אפשר להגדיר auto-approve לפעולות נפוצות.



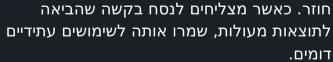
#### שימוש במספר מצבים

נצלו את היכולת לעבור בין מצבי עבודה שונים: השתמשו במצב Architect לתכנון ראשוני, עברו למצב Code ליישום, וסיימו עם מצב CA מותאם אישית לבדיקות.



### Roo Code תיעוד באמצעות

בקשו מ-Roo Code ליצור תיעוד איכותי תוך כדי העבודה. הוא יכול ליצור README, הסברים על הקוד, ואפילו לסייע בכתיבת דוקומנטציה טכנית או מדריכי משתמש.



צרו לעצמכם ספריית פרומפטים יעילים לשימוש

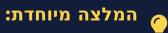
שמירת פרומפטים מוצלחים

סקירה ידנית לפעולות קריטיות

מורכבת או אבטחה. השתמשו ב-Roo Code

כמסייע, אך אל תוותרו על הבקרה האנושית

בצעו בחינה ידנית של קוד שנוגע ללוגיקה עסקית

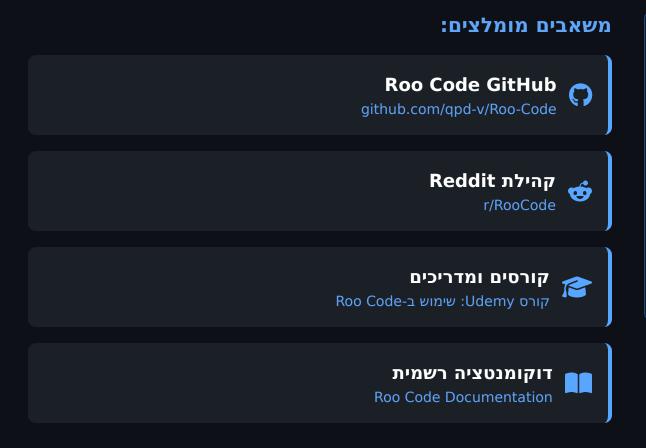


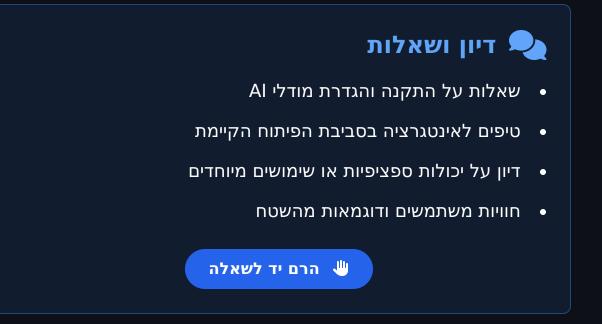
השתמשו בפונקציית @file, @folder ו-@git כדי לתת ל-Roo Code הקשר מלא על הפרויקט ולקבל תוצאות מדויקות יותר.





## שאלות? משאבים נוספים והעמקה







המשיכו לחקור את יכולות Roo Code ולשפר את תהליכי הפיתוח שלכם!