

ASP .Net Web APIS

Identity

TomerBu

ASP.NET Core Web API

המשך הפרוייקט משיעור קודם:

מסמך אפיון של API לכרטיסיות:

[כרטיסיות:](#)

<https://documenter.getpostman.com/view/25008645/2s9YXcd5BE>

[משתמשים](#)

<https://documenter.getpostman.com/view/25008645/2s9YXcd5BL>

קונטרולר לכרטיסיות: הוספת כרטיסיה:

```
[Route("api/[controller]")]
```

```
[ApiController]
```

0 references | [Copy link to this type](#)

```
public class CardsController (IRepository<Card> cardsRepo): ControllerBase  
{
```

```
    [HttpPost]
```

0 references | [Copy link to this method](#)

```
    public async Task<IActionResult> Post([FromBody] CardAddRequest dto)  
{
```

```
        if (!ModelState.IsValid)
```

```
        {
```

```
            return BadRequest("BadBad");
```

```
        }
```

```
        var card = dto.ToCard("TheUserId");
```

```
        var result = await cardsRepo.AddAsync(card);
```

```
        return CreatedAtAction(nameof(Get), new { id = result.Id }, result);
```

```
    }
```

קונטרולר לכרטיסיות: הוספת כרטיסיה:

בדיקת HTTP

Post a new Card:

POST {{ApisModuleLec3_HostAddress}}/api/cards

Content-Type: application/json

```
{
  "title": "a wonderful new card",
  "subtitle": "a test value for this card",
  "description": "a test value for new card\na test value for new card\n",
  "phone": "012-3211234",
  "email": "qwe@gmail.com",
  "web": "www.bing.com",
  "image": {
    "url": "https://img.izismile.com/img/img13/20201030/640/you_have_never_seen_something_like_this_640_36.jpg",
    "alt": "image of something"
  },
  "address": {
    "state": "IL",
    "country": "Israel",
    "city": "Arad",
    "street": "Shoham",
    "houseNumber": 5,
    "zip": 8920435
  }
}
```

קונטרולר לכרטיסיות: הצגת כל הכרטיסיות כולל בדיקת HTTP

```
[HttpGet]  
public async Task<IActionResult> Get()  
{  
    var result = await cardsRepo.GetAllAsync();  
    return Ok(result);  
}
```

```
### Get all the cards:  
GET {{ApisModuleLec3_HostAddress}}/api/cards
```

קונטרולר לכרטיסיות: הצגת כרטיס לפי ID

כולל בדיקת HTTP

```
[HttpGet("{id}")]
public async Task<IActionResult> Get([FromRoute] string id)
{
    var result = await cardsRepo.GetByIdAsync(id);
    if (result == null)
    {
        return NotFound();
    }
    return Ok(result);
}
```

Get a specific card by id:

GET {{ApisModuleLec3_HostAddress}}/api/cards/66a65aff911ef68c782b80eb

מחלקת עזר לתיאור User:

```
public class Name
{
    [Required, MinLength(2), MaxLength(256)]
    0 references | Copy link to this property
    public required string First { get; set; }
    [Required, MinLength(2), MaxLength(256)]
    0 references | Copy link to this property
    public required string Last { get; set; }

    [ValidateNever]
    [MinLength(2), MaxLength(256)]
    0 references | Copy link to this property
    public string? Middle { get; set; } = "";
}
```

עבודה עם MongoDBIdentity

התקנת חבילות NuGet



AspNetCore.Identity.Mongo by Matteo Fabbri
Mongo based identity framework for Asp.Net Core



Microsoft.AspNetCore.Authentication.JwtBearer by Microsoft
ASP.NET Core middleware that enables an application to receive an OpenID Connect bearer token.



MongoDB.Driver by MongoDB Inc.
Official .NET driver for MongoDB.

מחלקה עבור משתמש:

```
public class AppUser() : MongoUser<ObjectId>()
{

    [BsonRequired]
    public required Name Name { get; set; }

    [BsonRequired]
    public required Image Image { get; set; }

    [BsonRequired]
    public required Address Address { get; set; }

    [BsonRequired]
    public required bool IsBusiness { get; set; }

    public bool IsAdmin { get; set; } = false;

    [BsonDateTimeOptions(Kind = DateTimeKind.Utc)]
    public DateTime CreatedAt { get; set; } = DateTime.UtcNow;
}
```

מחלקה עבור Role:

```
public class AppRole() : MongoRole<ObjectId>(){  
  
};
```

DTO להרשמת משתמש חדש

```
public class RegisterDto
{
    [Required]
    public required Name Name { get; set; }

    [Required]
    [StringLength(11, MinimumLength = 9)]
    public required string Phone { get; set; }

    [Required]
    [EmailAddress]
    [StringLength(256)]
    public required string Email { get; set; }

    [Required]
    [StringLength(20, MinimumLength = 7)]
    [RegularExpression(@"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d){6,}$", ErrorMessage = "Password must have at least 6 characters, one uppercase letter, one lowercase letter and one number")]
    public required string Password { get; set; }

    [Required]
    public required Image Image { get; set; }

    [Required]
    public required Address Address { get; set; }

    [Required]
    public bool IsBusiness { get; set; }
}
```

DTO להתחברות

```
public class LoginDto
{
    [Required]
    [EmailAddress]
    public required string Email { get; set; }
    [MinLength(6)]
    [Required]
    [RegularExpression(@"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d){6,}$", ErrorMessage = "Password must have at least 6 characters, one uppercase letter, one lowercase letter and one number")]
    public required string Password { get; set; }
}
```

מיפוי DTO למודל

```
public static class RegisterDtoExtensions
{
    public static AppUser ToAppUser(this RegisterDto dto)
    {
        return new AppUser
        {
            UserName = dto.Email,
            Name = dto.Name,
            PhoneNumber = dto.Phone,
            Email = dto.Email,
            Image = dto.Image,
            Address = dto.Address,
            IsBusiness = dto.IsBusiness
        };
    }
}
```

שם המשתמש במקרה שלנו זה האימייל כי הדרישה היא להתחבר
עם אימייל וסיסמא
במקום שם משתמש וסיסמא.

הספריה מצפינה את הסיסמא ולכן אנחנו לא מכניסים אותה כאן.

הזרקת שירות של הספרייה Identity Program.cs

```
var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");  
var services = builder.Services;
```

```
services.AddIdentityMongoDbProvider<AppUser, AppRole, ObjectId>(identity =>  
{  
    identity.Password.RequiredLength = 6;  
    identity.Password.RequireLowercase = false;  
    identity.Password.RequireUppercase = false;  
    identity.Password.RequireNonAlphanumeric = false;  
    identity.Password.RequireDigit = false;  
    identity.User.RequireUniqueEmail = true;  
    identity.User.AllowedUserNameCharacters =  
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+ ";  
}, mongo =>  
{  
    mongo.ConnectionString = connectionString;  
})  
.AddDefaultTokenProviders();
```

הגדרות לסיסמא
הגדרות למשתמש

הגדרות למונגו: חיבור לדטא-בייס

Auth Controller

```
[ApiController]
[Route("api/[controller]")]
public class AuthController(UserManager<AppUser> userManager, SignInManager<AppUser> signInManager) : ControllerBase
{
    [HttpPost("register")]
    public async Task<IActionResult> Register([FromBody] RegisterDto dto)
    {
        var user = dto.ToAppUser();
        var result = await userManager.CreateAsync(user, dto.Password);

        if (result.Succeeded)
        {
            return Ok(result);
        }

        return BadRequest(result);
    }

    [HttpPost("login")]
    public async Task<IActionResult> Login([FromBody] LoginDto dto)
    {
        var r = await signInManager.PasswordSignInAsync(dto.Email, dto.Password, false, false);

        var user = await userManager.GetUserAsync(User);

        if (r.Succeeded)
        {
            return Ok(user);
        }
        return Unauthorized();
    }
}
```

בדיקות HTTP

Register a new User:

POST {{ApisModuleLec3_HostAddress}}/api/auth/register

Content-Type: application/json

```
{
  "name": {
    "first": "Ell",
    "last": "Vis"
  },
  "phone": "0512345567",
  "email": "ellvis2@email.com",
  "password": "Abc!123Abc",
  "image": {
    "url": "https://fastly.picsum.photos/id/975/200/300.jpg",
    "alt": "The alt"
  },
  "address": {
    "state": "IL",
    "country": "Israel",
    "city": "Arad",
    "street": "Shoham",
    "houseNumber": 5,
    "zip": 8920435
  },
  "isBusiness": true
}
```

login a User:

POST {{ApisModuleLec3_HostAddress}}/api/auth/login

Content-Type: application/json

```
{
  "email": "ellvis2@email.com",
  "password": "Abc!123Abc"
}
```


שיעורי בית:
לחזור על הנושאים הבאים:

Identity Framework:
Login/Registration

מומלץ לממש את הספרייה בפרוייקט חדש כדי לחזור על התהליך בעצמכם ולוודא הבנה טובה.

לתרגול מוזמנים ליצור API שמשתמש ב Repository
ומאפשר פעולות CRUD על שירים, אמנים ואלבומים

כולל הרשמה והתחברות.