



The first Hub for Developers

Authentication/Authorization  
(ASP.NET Core Identity)

Konstantinos Athanasoglou

# Authentication vs Authorization

## Authentication

Process to determine *who* you are

(e.g. Anonymous, Windows Authentication, Active Directory)

## Authorization

*What* are you allowed to do (follows authentication)

(e.g. View Home Page, Create Project, Edit Project, Delete User)

## Why?

Personalization, Security, Auditing/Logging

# ASP.NET Core Identity : Introduction

- Inherit from **IdentityDbContext** (instead of DbContext)
- Default user model: **IdentityUser**
- Default PK type: **string** (Guid's are actually stored in Db)

## Code-First (required to build Db tables)

### Concepts:

- **ModelSnapshot**: Current state of the model
- **Migrations**: Incremental diffs (up/down) compared to ModelSnapshot.

### Basic Commands:

- Add-Migration
- Remove-Migration
- Script-Migration
- Update-Database

# ASP.NET Core Identity: Integration

## Assumptions:

1. DB Model + Model Scaffolding created
2. *(optional)* Controller scaffolding

1. Scaffold Identity
2. Merge two DB Contexts (remember: `base.OnModelCreating()`)
3. *(recommended)* Remove duplicate configuration (*IdentityHostingStartup*)
4. *(optional)* Add *\_LoginPartial.cshtml* in *\_Layout.cshtml*
5. Create Identity DB Schema (Code-First)
6. *(optional)* Customize User + Primary Key (PK) Type  
(use `IdentityDbContext<TUser, TRole, key>`,  
alter registration in `StartUp.cs`, fix *\_LoginPartial.cshtml* + Identity Pages)
7. Delete Migrations + Re-create Identity DB Schema
8. Add new properties in `Register.cshtml.cs`

# Adding Authorization

## **[Authorize]**

Restrict access to Controller/Actions

## **[AllowAnonymous]**

Override previous attribute.

Obtain logged in User's Id:

```
User.FindFirstValue(ClaimTypes.NameIdentifier);
```

Obtain entire User object:

```
Inject: UserManager<T> _user;
```

```
var user = await _user.FindByIdAsync(id);
```

# Thank you!