Quickstart for developing TrackYourFit

Kurs	TINF18B3
Vorlesung	Software Engineering
Team	Tobias Bühler, Toni Einecker, Janis Fix, Johannes Zipfel

Contents

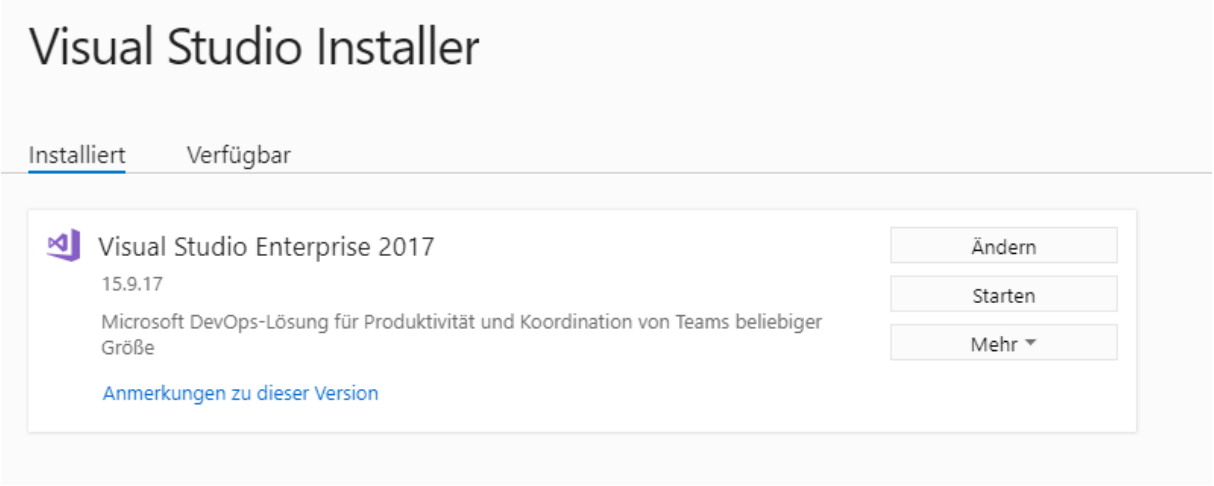
1. Setting up the IDE

First of all you will need to set up your IDE (Visual Studio) with the right packages.

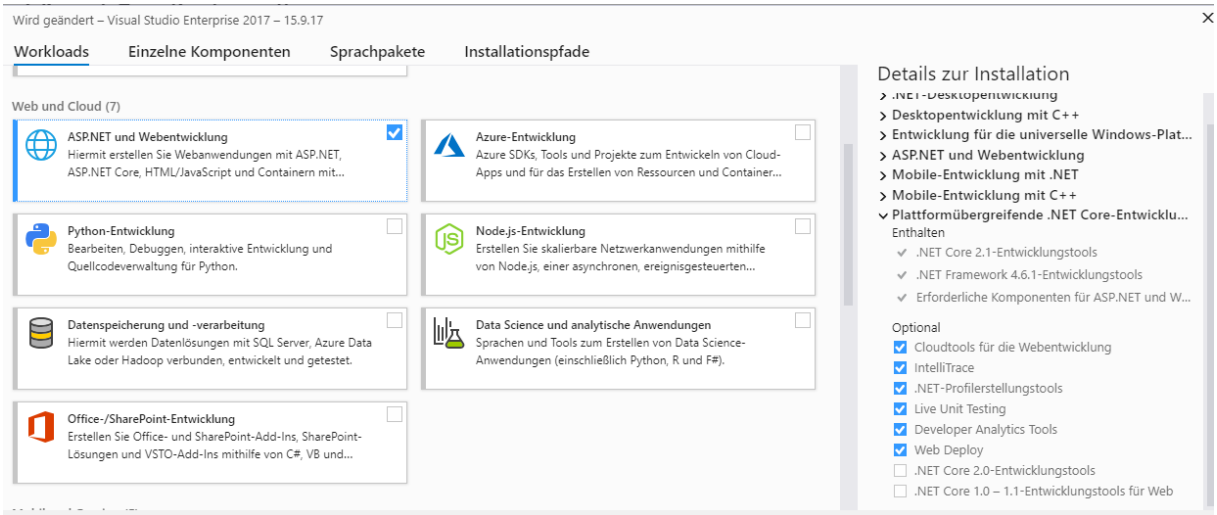
You can find the installer fort he Community Version right here:

https://visualstudio.microsoft.com/de/vs/community/

Being a student of the DHBW, you can also get the enterprise version through the store. If you already have VS you can just click „Ändern“ to get to the packages page.



Now you need to download the required packages, just tick the following package and install it:



Now everything should be set up regarding the IDE.

2. Setting up the database

To begin with, you might need to update the connection string in the file „appsettings.json“. As default the connectionstring is going to be the local DB.

The Categories and Roles provided for the application can be adjusted here too.

When starting the application there will be an admin created on startup. So now you need to specify the E-Mail, Username and Password of the basic-admin. This can be done in the Startup class („Startup.cs“):

```
/// <summary>
/// Creates a basic admin user on startup.
/// </summary>
/// <param name="serviceProvider"></param>
1 reference | Bronzila, 50 days ago | 1 author, 1 change | 0 exceptions
private void CreateBasicAdmin(IServiceProvider serviceProvider)
{
    var roleManager = serviceProvider.GetRequiredService<RoleManager<IdentityRole>>();
    var userManager = serviceProvider.GetRequiredService<UserManager<ApplicationUser>>();
    Task<IdentityResult> roleResult;
    string administratorRoleName = "Administrator";
    string administratorEmail = "";
    string administratorPW = "";

    //Check that there is an Administrator role and create if not
    Task<bool> hasAdminRole = roleManager.RoleExistsAsync(administratorRoleName);
    hasAdminRole.Wait();

    if (!hasAdminRole.Result)
    {
        roleResult = roleManager.CreateAsync(new IdentityRole(administratorRoleName));
        roleResult.Wait();
    }

    //Check if the admin user exists and create it if not
    //Add to the Administrator role
    Task<ApplicationUser> testUser = userManager.FindByEmailAsync(administratorEmail);
    testUser.Wait();

    if (testUser.Result == null)
    {
        ApplicationUser administrator = new ApplicationUser();
        administrator.Email = administratorEmail;
        administrator.UserName = administratorEmail;

        Task<IdentityResult> newUser = userManager.CreateAsync(administrator, administratorPW);
        newUser.Wait();

        if (newUser.Result.Succeeded)
        {
            Task<IdentityResult> newUserRole = userManager.AddToRoleAsync(administrator, administratorRoleName);
            newUserRole.Wait();
        }
    }
}
```

Just update the two strings and you will be set.

Now by debugging the application (with the F5-Key) the database is going tob e created automatically, creating the roles, categories and the basic admin specified by you. You can begin using the application by either registering a new account or just logging in with the admin account.