Dating App

Wednesday, March 3, 2021 3:44 PM

1) VS Code
2) Create new Web API
    a. dotnet new sln [Created new solution in the name of folder]
    b. dotnet new webapi -o API

    ```
    The template "ASP.NET Core Web API" was created successfully.

    Processing post-creation actions...
    Running 'dotnet restore' on API\API.csproj...
      Determining projects to restore...
      Restored D:\To-Git\SelfLearings\UdemyDatingApplication\API\API.csproj (in 10.74 sec).
    Restore succeeded.
    ```

    - Add Project to Solution

      ```
      dotnet sln add API/
      Project `API\API.csproj` added to the solution.
      ```

    - `dotnet dev-certs https --trust`
        - Trusting the HTTPS development certificate was requested. A confirmation prompt will be displayed if the certificate was not previously trusted. Click yes on the prompt to trust the certificate.

C#

Thursday, March 4, 2021 5:52 PM

Ienumerable namespace is System.Collections.Generic
ToList() namespace is system.Linq

  - Search
  - Sort
  - Manipulate

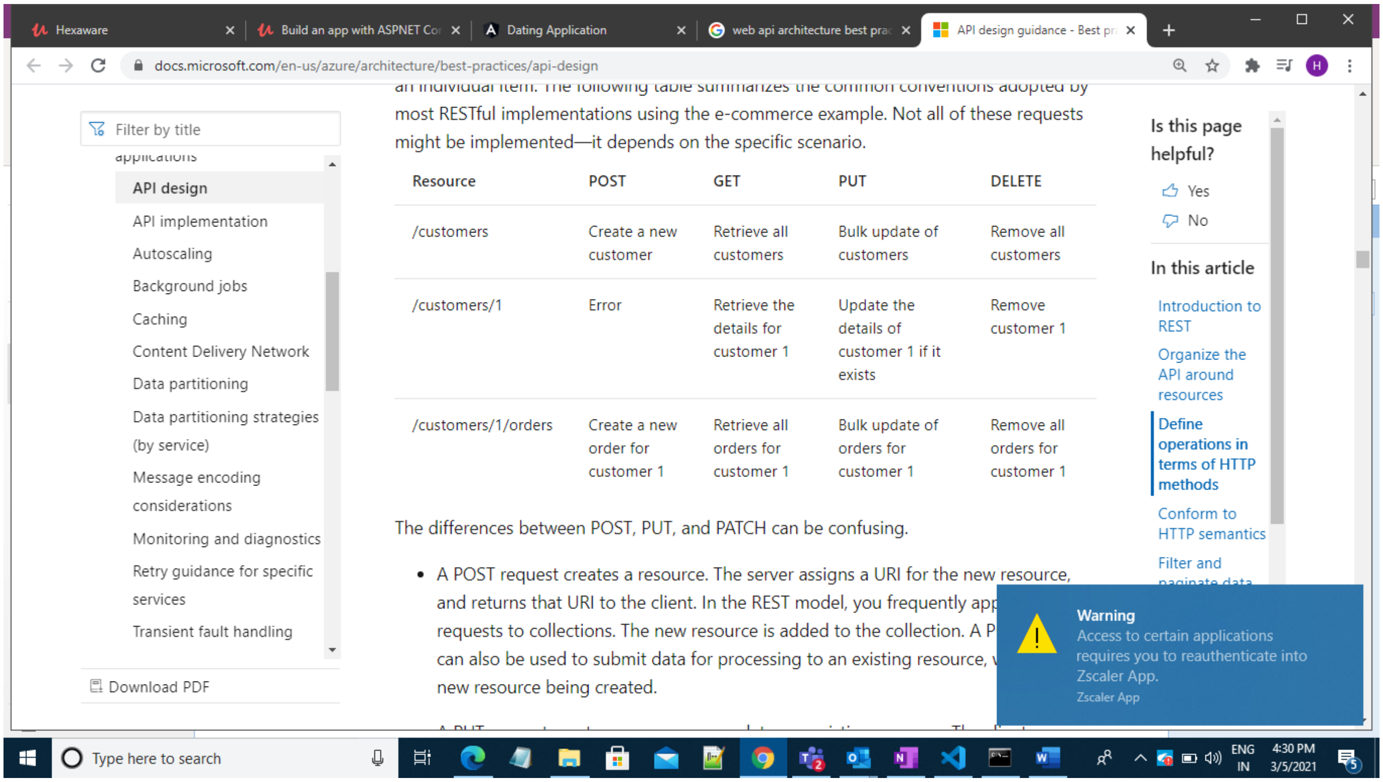
Microsoft.EntityFrameworkCore

  - ToListAsync()

- System.Threading
  - Task<>
  - Async/await

Web API Design Guidelines

Friday, March 5, 2021 4:30 PM


Hexaware
Build an app with ASPNET Co
Dating Application
web api architecture best prac
API design guidance - Best pr
docs.microsoft.com/en-us/azure/architecture/best-practices/api-design
Filter by title
applications
API design
API implementation
Autoscaling
Background jobs
Caching
Content Delivery Network
Data partitioning
Data partitioning strategies (by service)
Message encoding considerations
Monitoring and diagnostics
Retry guidance for specific services
Transient fault handling
Download PDF
Is this page helpful?
Yes
No
In this article
Introduction to REST
Organize the API around resources
Define operations in terms of HTTP methods
Conform to HTTP semantics
Filter and paginate data
Type here to search
ENG
IN
4:30 PM
3/5/2021

an individual item. The following table summarizes the common conventions adopted by most RESTful implementations using the e-commerce example. Not all of these requests might be implemented—it depends on the specific scenario.

Resource	POST	GET	PUT	DELETE
/customers	Create a new customer	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1	Error	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1
/customers/1/orders	Create a new order for customer 1	Retrieve all orders for customer 1	Bulk update of orders for customer 1	Remove all orders for customer 1

The differences between POST, PUT, and PATCH can be confusing.

- A POST request creates a resource. The server assigns a URI for the new resource, and returns that URI to the client. In the REST model, you frequently app requests to collections. The new resource is added to the collection. A P can also be used to submit data for processing to an existing resource, v new resource being created.

Warning
Access to certain applications requires you to reauthenticate into Zscaler App.
Zscaler App

Salted Password and User Check

Wednesday, March 10, 2021 11:10 AM

```
[HttpPost("register")]
    public async Task<ActionResult<AppUser>>Register(RegisterDTO registerDTO )
    {
      if( await UserExists(registerDTO.UserName))  return BadRequest("User Name
not avaliable.");
      //Task<ActionResult<AppUser>>
      HMACSHA512 hmac = new HMACSHA512();

      var user = new AppUser{
        UserName = registerDTO.UserName ,
        PasswordHash =
hmac.ComputeHash(Encoding.UTF8.GetBytes(registerDTO.Password)),
        PasswordSalt = hmac.Key
      };

      datingApplicationContext.ApplicationUsers.Add(user);

      await datingApplicationContext.SaveChangesAsync();

      return user;

    }

    private async Task<bool> UserExists(string userName)
    {
     return await datingApplicationContext.ApplicationUsers.AnyAsync( x =>
x.UserName == userName);
    }
  }
```

Identity - IServiceCollection Extension Class

Thursday, March 11, 2021 10:21 PM

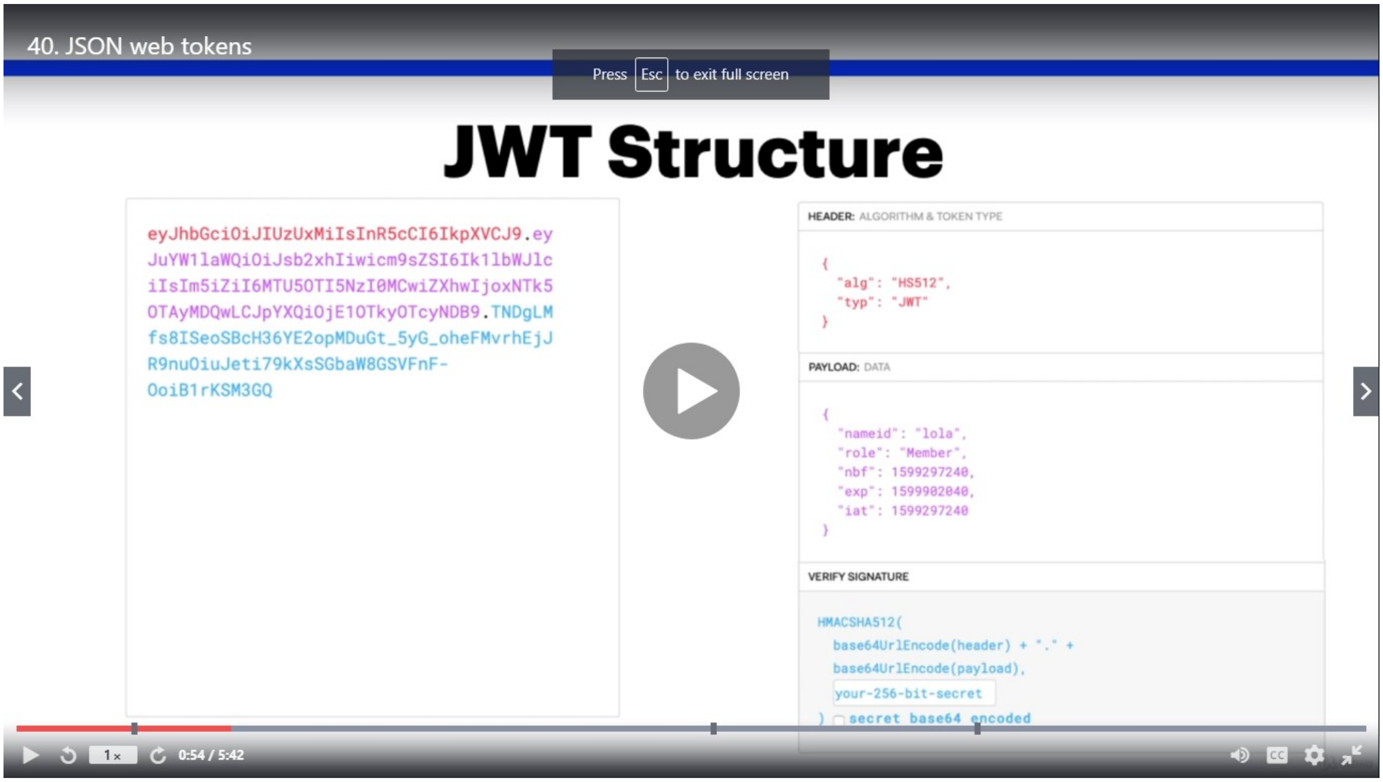
```
namespace API.Extensions
{
   public static class IdentityServiceExtension
   {

      public static IServiceCollection AddIdentityService(this IServiceCollection services,IConfiguration
configuration)
      {

          services.AddAuthentication( option =>
         {
            option.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
            option.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

         }).AddJwtBearer( options =>
             options.TokenValidationParameters = new TokenValidationParameters()
            {
              ValidateAudience = false,
              ValidateIssuer  = false,
             IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(configuration["TokenKey"]))
//Configuration["JwtToken:SecretKey"]
            });

         return services;

      }

   }
}
```

JWT

Wednesday, March 10, 2021 3:13 PM

Credentials
Claims
Other Information

Credentials
Claims
Other Information



JWT Structure

```
eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.ey
JuYW1laWQiOiJsb2xhIiwicm9sZSI6Ik1lbWJlc
iIsIm5iZiI6MTU5OTI5NzI0MCwiZXhwIjoxNTk5
OTAyMDQwLCJpYXQiOjE1OTkyOTcyNDB9.TNDgLM
fs8ISeoSBcH36YE2opMDuGt_5yG_oheFMvrhEjJ
R9nuOiuJeti79kXsSGbaW8GSVFnF-
OoiB1rKSM3GQ
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS512",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "nameid": "lola",
  "role": "Member",
  "nbf": 1599297240,
  "exp": 1599902040,
  "iat": 1599297240
}
```

VERIFY SIGNATURE

```
HMACSHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

Swagger JWT

Thursday, March 11, 2021   6:33 PM

```
 services.AddSwaggerGen(c =>
     {
       c.SwaggerDoc("v1", new OpenApiInfo { Title = "API", Version = "v1" });


        // To Enable authorization using Swagger (JWT)
       c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme()
       {
         Name = "Authorization",
         Type = SecuritySchemeType.ApiKey,
         Scheme = "Bearer",
         BearerFormat = "JWT",
         In = ParameterLocation.Header,
         Description = "JWT Authorization header using the Bearer scheme. \r\n\r\n Enter 'Bearer' [space] and then your token in the text input below.\r\n\r\nExample: \"Bearer 12345abcdef\"",
       });

      c.AddSecurityRequirement(new OpenApiSecurityRequirement
       {
         {
            new OpenApiSecurityScheme
             {
               Reference = new OpenApiReference
               {
                 Type = ReferenceType.SecurityScheme,
                 Id = "Bearer"
               }
             },
             new string[] {}

         }
       });

     });




services.AddAuthentication( option =>
     {
       option.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
       option.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

     }).AddJwtBearer( options =>
        options.TokenValidationParameters = new TokenValidationParameters()
       {
            ValidateAudience = false,
            ValidateIssuer  = false,
           IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration["TokenKey"]))
//Configuration["JwtToken:SecretKey"]
          });
```

Error Handler

Friday, March 12, 2021 1:39 PM

```
app.UseExceptionHandler(
 options => {
   options.Run(
   async context =>
   {
     context.Response.StatusCode = (int)HttpStatusCode.InternalServerError;
     context.Response.ContentType = "text/html";
     var ex = context.Features.Get<IExceptionHandlerFeature>();
     if (ex != null)
     {
       var err = $"<h1>Error: {ex.Error.Message}</h1>{ex.Error.StackTrace }";
       await context.Response.WriteAsync(err).ConfigureAwait(false);
     }
   });
 }
);
```

Action Filter

https://code-maze.com/action-filters-aspnetcore/

Entity Frame Work

Wednesday, March 3, 2021 4:21 PM

Entity[Abstraction of Physical Tables]

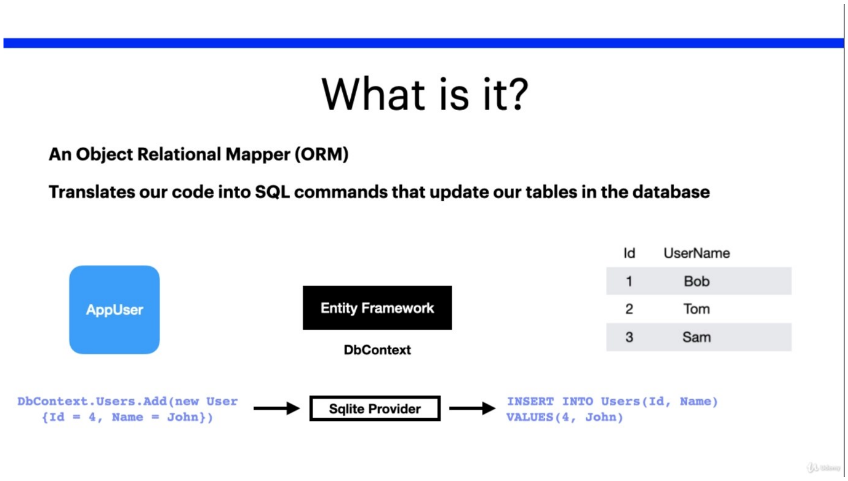
Entity Frame Work ==> Object Relational Mapper

Translate our code to SQL commands

dotnet tool install --global dotnet-ef --version 5.0 [Install the EF manually in .Net Core 3.0]

In Environment variable, Add path "C:\Users\43869\.dotnet\tools" or "
%USERPROFILE%/.dotnet/tools "

Install Package Microsoft.EntityFrameworkCore.Design



What is it?

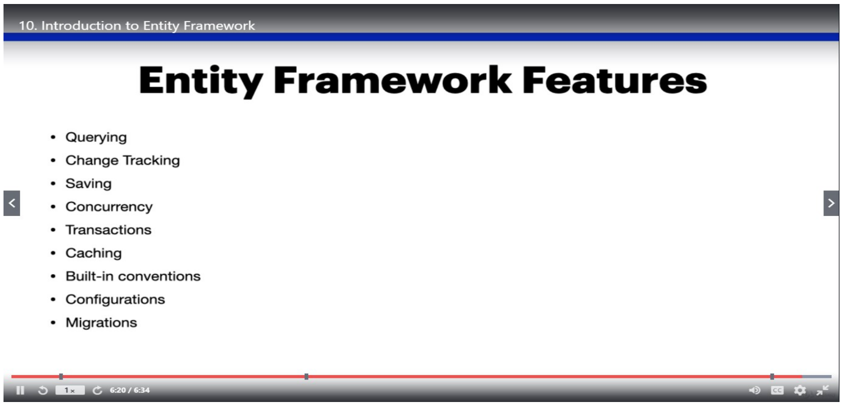
An Object Relational Mapper (ORM)

Translates our code into SQL commands that update our tables in the database

Id	UserName
1	Bob
2	Tom
3	Sam

```
DbContext.Users.Add(new User
  {Id = 4, Name = John})
```

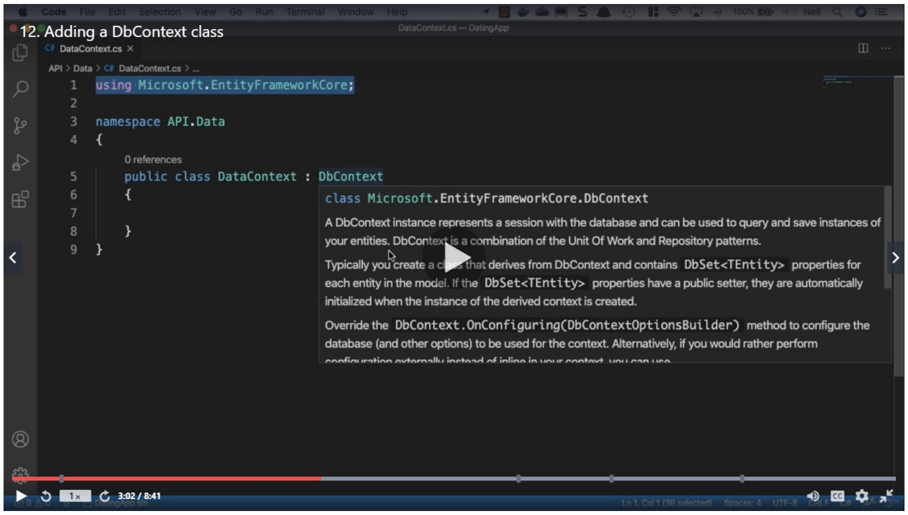
```
INSERT INTO Users(Id, Name)
VALUES(4, John)
```



10. Introduction to Entity Framework

Entity Framework Features

- Querying
- Change Tracking
- Saving
- Concurrency
- Transactions
- Caching
- Built-in conventions
- Configurations
- Migrations



12. Adding a DbContext class

Code File Edit Selection View Go Run Terminal Window Help
DataContext.cs — DatingApp
DataContext.cs
API > Data > DataContext.cs > ...
1×
3:02 / 8:41

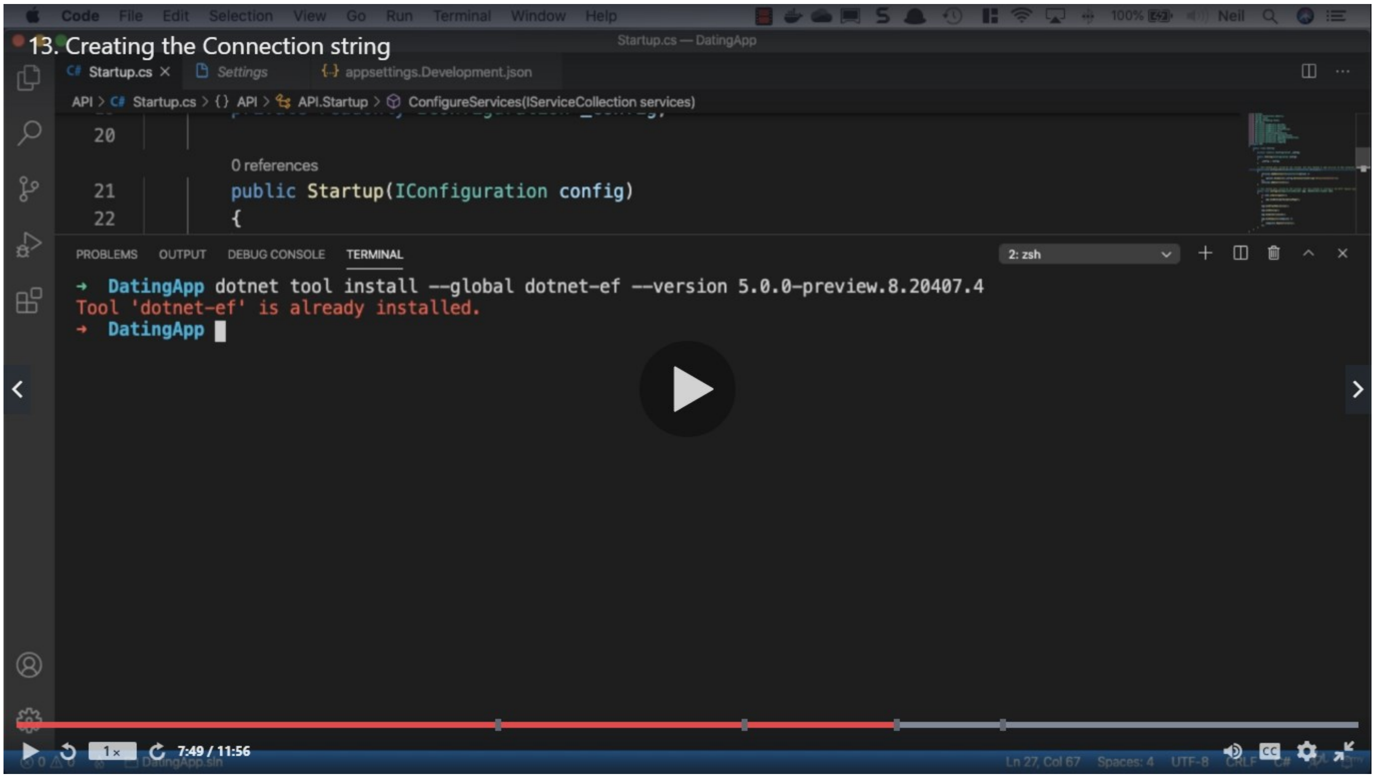
```
using Microsoft.EntityFrameworkCore;

namespace API.Data
{
    0 references
    public class DataContext : DbContext
    {

    }
}
```

`class Microsoft.EntityFrameworkCore.DbContext`

A DbContext instance represents a session with the database and can be used to query and save instances of your entities. DbContext is a combination of the Unit Of Work and Repository patterns.

Typically you create a class that derives from DbContext and contains `DbSet<TEntity>` properties for each entity in the model. If the `DbSet<TEntity>` properties have a public setter, they are automatically initialized when the instance of the derived context is created.

Override the `DbContext.OnConfiguring(DbContextOptionsBuilder)` method to configure the database (and other options) to be used for the context. Alternatively, if you would rather perform configuration externally instead of inline in your context, you can use



```
0 references
public Startup(IConfiguration config)
{
```

```
→ DatingApp dotnet tool install --global dotnet-ef --version 5.0.0-preview.8.20407.4
Tool 'dotnet-ef' is already installed.
→ DatingApp
```

EF Commands

Thursday, March 4, 2021 2:49 PM

Nuget Package

Microsoft.EntityFrameworkCore.Design

DotNet CLI [Migrations can be Add or remove but no update]

```
dotnet tool install --global dotnet-ef --version 5.0
```

```
dotnet ef migrations add InitialCreate -o Data/Migrations
dotnet ef database update
dotnet ef database drop
```

GIT Configuration

Thursday, March 4, 2021 8:16 PM

Add below path in environment variables,
C:\Users\43869\AppData\Local\Programs\Git\bin
C:\Users\43869\AppData\Local\Programs\Git\cmd

Reference: C:\Users\43869\AppData\Roaming\Microsoft\Windows\Start Menu

Add Git Ignore file

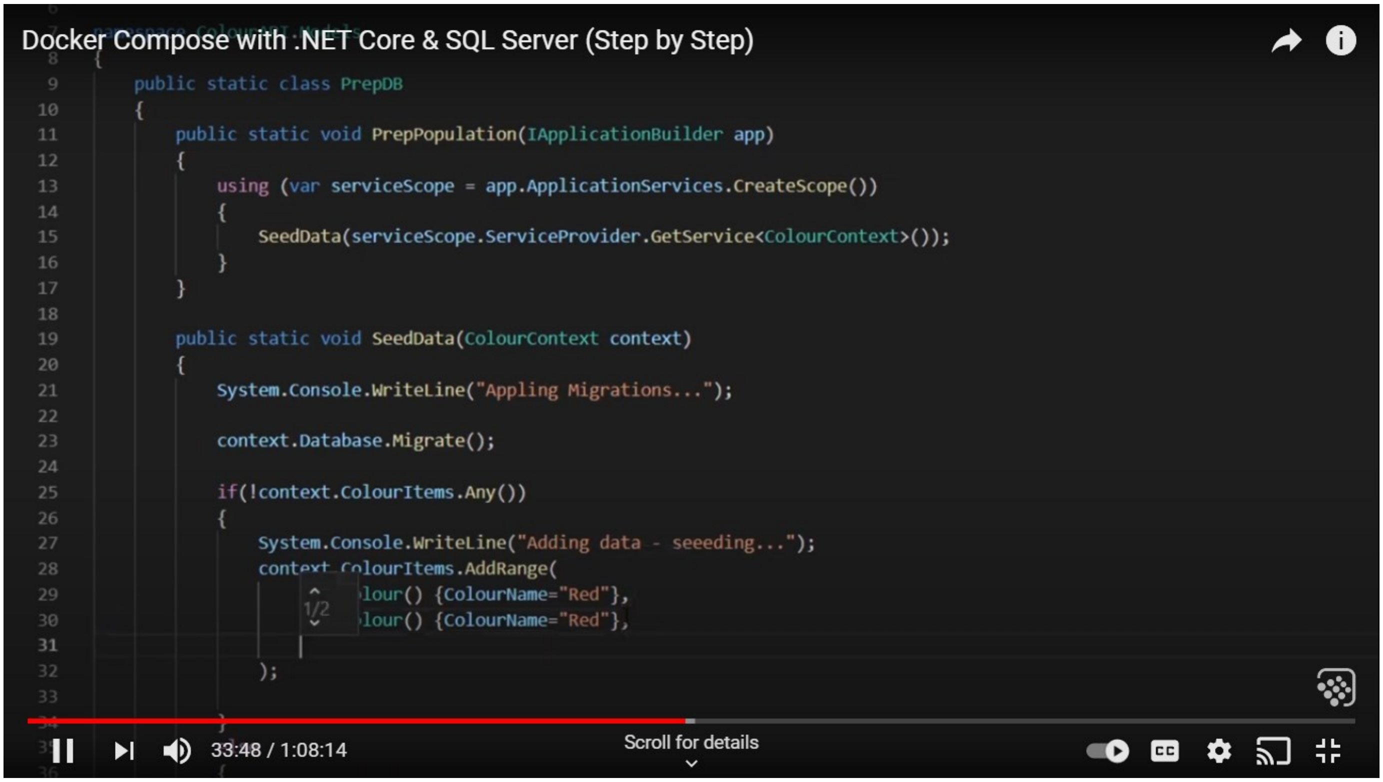
```
dotnet new gitignore
The template "dotnet gitignore file" was created successfully.
```

Create a repository & push the code to code repository.

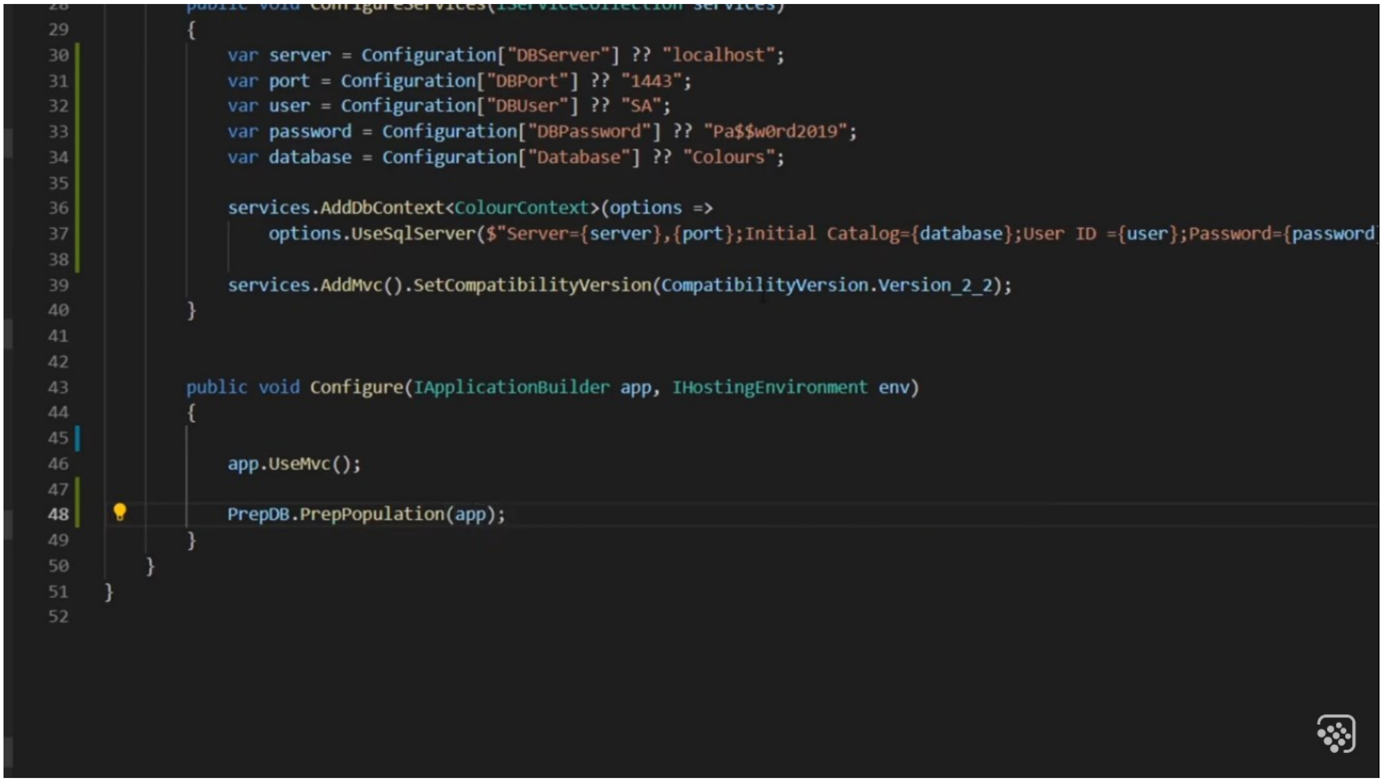
- Initialize GIT Repository
- Stage Code
- Commit Code
- Push the code

Data Seed

Monday, June 7, 2021 3:30 PM



```
namespace ColourAPI.Models
{
    public static class PrepDB
    {
        public static void PrepPopulation(IApplicationBuilder app)
        {
            using (var serviceScope = app.ApplicationServices.CreateScope())
            {
                SeedData(serviceScope.ServiceProvider.GetService<ColourContext>());
            }
        }

        public static void SeedData(ColourContext context)
        {
            System.Console.WriteLine("Appling Migrations...");

            context.Database.Migrate();

            if(!context.ColourItems.Any())
            {
                System.Console.WriteLine("Adding data - seeeding...");
                context.ColourItems.AddRange(
                    lour() {ColourName="Red"},
                    lour() {ColourName="Red"},

                );
```



```
        public void ConfigureServices(IServiceCollection services)
        {
            var server = Configuration["DBServer"] ?? "localhost";
            var port = Configuration["DBPort"] ?? "1443";
            var user = Configuration["DBUser"] ?? "SA";
            var password = Configuration["DBPassword"] ?? "Pa$$w0rd2019";
            var database = Configuration["Database"] ?? "Colours";

            services.AddDbContext<ColourContext>(options =>
                options.UseSqlServer($"Server={server},{port};Initial Catalog={database};User ID ={user};Password={password

            services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);
        }


        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {

            app.UseMvc();

            PrepDB.PrepPopulation(app);
        }
    }
}
```

CORS Configuration

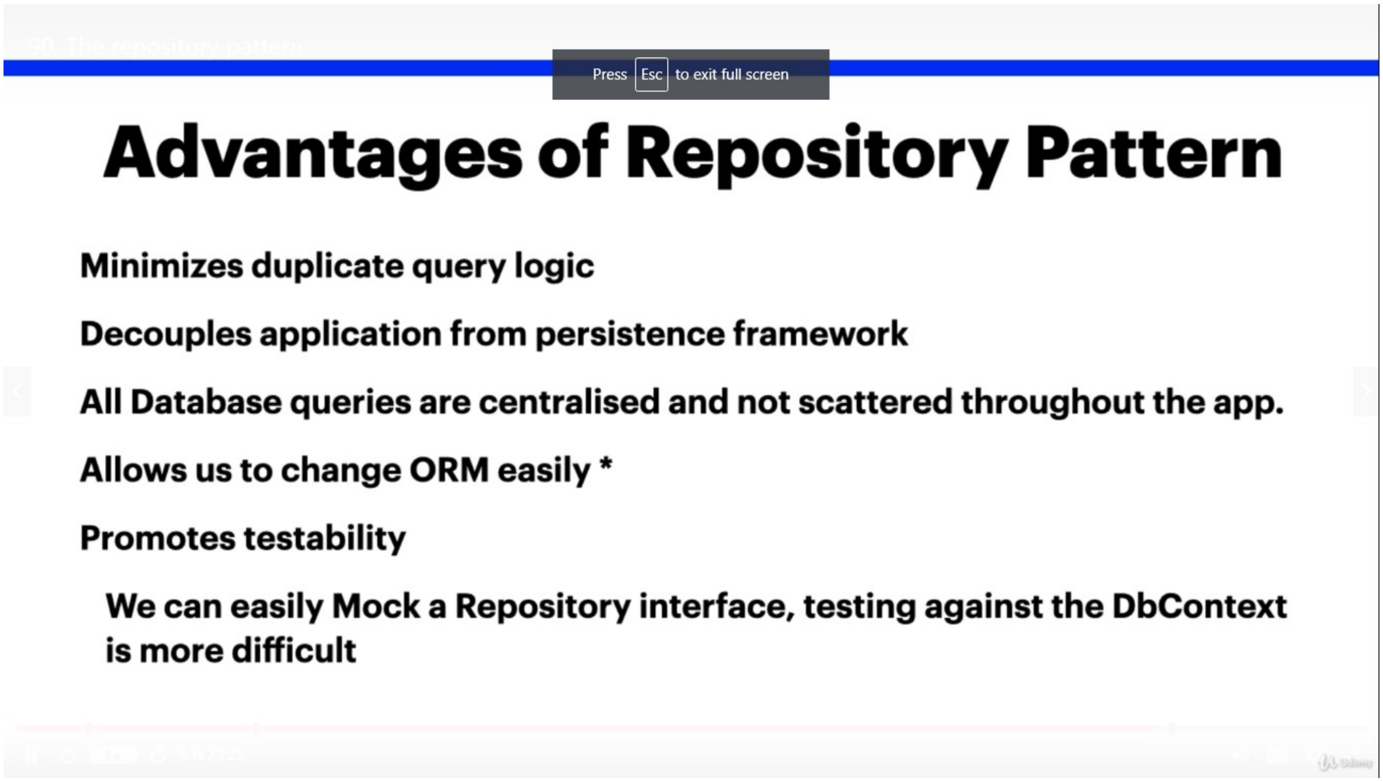
Friday, March 5, 2021 1:39 PM

In **startup.cs**

```
services.AddCors();
app.UseCors( policy => 
policy.AllowAnyHeader().AllowAnyMethod().WithOrigins("http://localhost:4200"));
```

Benefits of Repository Pattern

Monday, March 15, 2021 11:47 AM



Advantages of Repository Pattern

Minimizes duplicate query logic

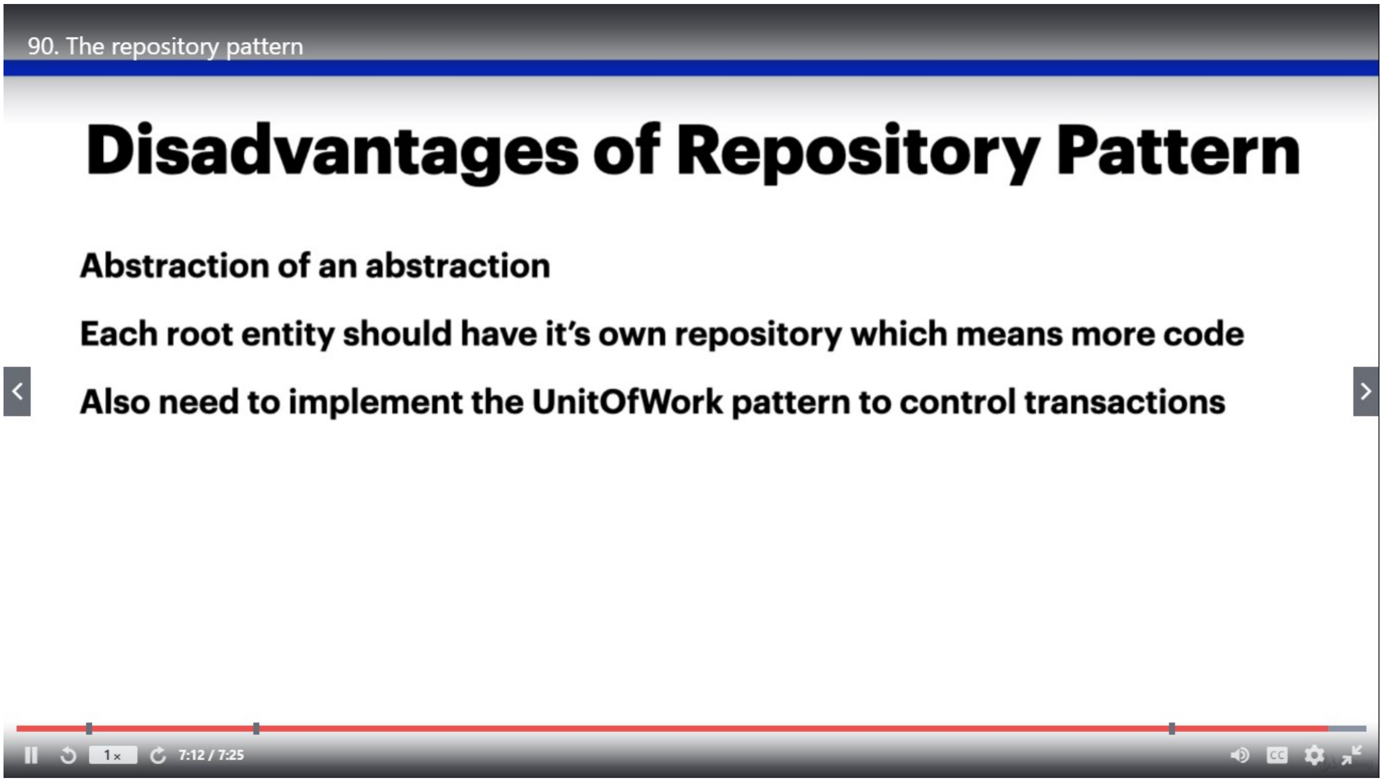
Decouples application from persistence framework

All Database queries are centralised and not scattered throughout the app.

Allows us to change ORM easily *

Promotes testability

We can easily Mock a Repository interface, testing against the DbContext is more difficult



Disadvantages of Repository Pattern

Abstraction of an abstraction

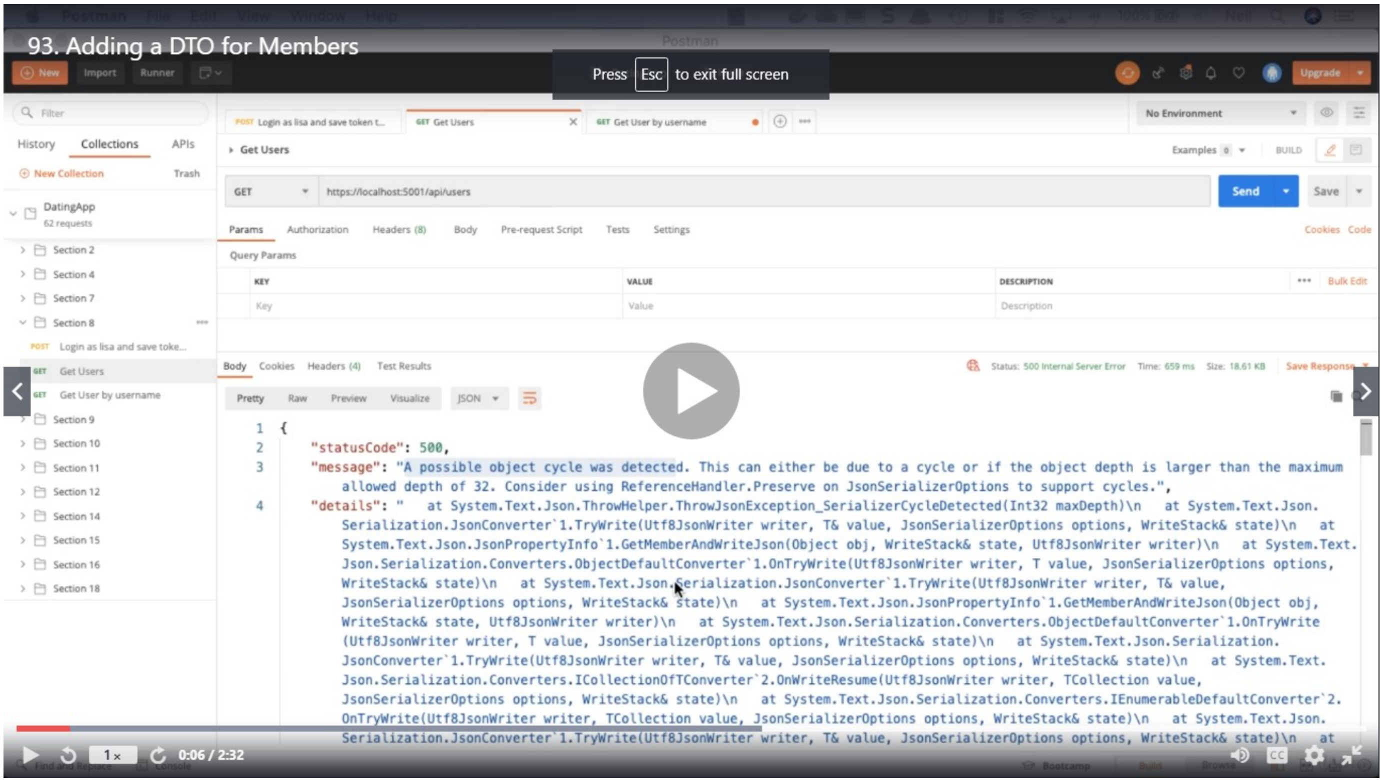
Each root entity should have it's own repository which means more code

Also need to implement the UnitOfWork pattern to control transactions

https://codewithshadman.com/repository-pattern-csharp/#:~:text=Repository%20pattern%20C%23%20is%20a,view%20of%20the%20persistence%20layer.

DTO - To avoid cyclic Issue

Tuesday, March 16, 2021	5:31 PM



Get Users

GET https://localhost:5001/api/users

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 500 Internal Server Error Time: 659 ms Size: 18.61 KB

```
{
    "statusCode": 500,
    "message": "A possible object cycle was detected. This can either be due to a cycle or if the object depth is larger than the maximum
        allowed depth of 32. Consider using ReferenceHandler.Preserve on JsonSerializerOptions to support cycles.",
    "details": "   at System.Text.Json.ThrowHelper.ThrowJsonException_SerializerCycleDetected(Int32 maxDepth)\n   at System.Text.Json.
        Serialization.JsonConverter`1.TryWrite(Utf8JsonWriter writer, T& value, JsonSerializerOptions options, WriteStack& state)\n   at
        System.Text.Json.JsonPropertyInfo`1.GetMemberAndWriteJson(Object obj, WriteStack& state, Utf8JsonWriter writer)\n   at System.Text.
        Json.Serialization.Converters.ObjectDefaultConverter`1.OnTryWrite(Utf8JsonWriter writer, T value, JsonSerializerOptions options,
        WriteStack& state)\n   at System.Text.Json.Serialization.JsonConverter`1.TryWrite(Utf8JsonWriter writer, T& value,
        JsonSerializerOptions options, WriteStack& state)\n   at System.Text.Json.JsonPropertyInfo`1.GetMemberAndWriteJson(Object obj,
        WriteStack& state, Utf8JsonWriter writer)\n   at System.Text.Json.Serialization.Converters.ObjectDefaultConverter`1.OnTryWrite
        (Utf8JsonWriter writer, T value, JsonSerializerOptions options, WriteStack& state)\n   at System.Text.Json.Serialization.
        JsonConverter`1.TryWrite(Utf8JsonWriter writer, T& value, JsonSerializerOptions options, WriteStack& state)\n   at System.Text.
        Json.Serialization.Converters.ICollectionOfTConverter`2.OnWriteResume(Utf8JsonWriter writer, TCollection value,
        JsonSerializerOptions options, WriteStack& state)\n   at System.Text.Json.Serialization.Converters.IEnumerableDefaultConverter`2.
        OnTryWrite(Utf8JsonWriter writer, TCollection value, JsonSerializerOptions options, WriteStack& state)\n   at System.Text.Json.
        Serialization.JsonConverter`1.TryWrite(Utf8JsonWriter writer, T& value, JsonSerializerOptions options, WriteStack& state)\n   at
```

```
var user = await Context.ApplicationUsers.Include( p => p.Photos)
        .Where( x=> x.ID == id)
//Select the property only you want
        .ProjectTo<MembersDTO>(mapper.ConfigurationProvider)
        .FirstOrDefaultAsync();
```

Angular

Friday, March 5, 2021 11:57 AM

```
Node --version
Npm --version
```

Error : Ng' is not recognized as an internal or external command, operable program or batch file.

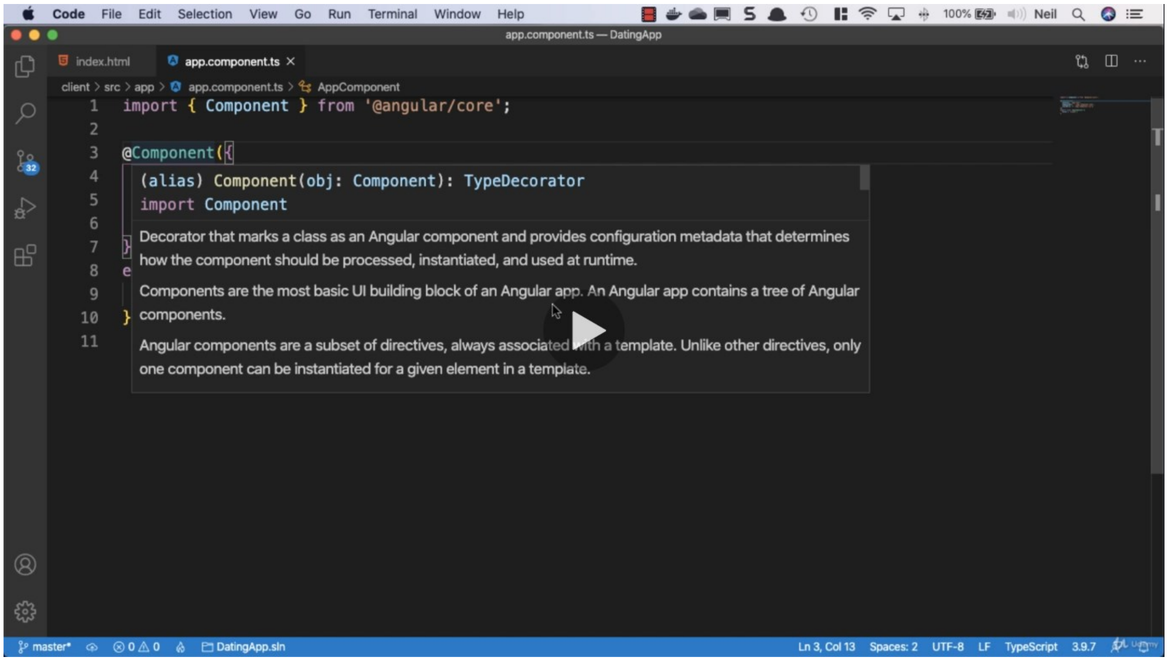
```
npm install -g @angular/cli@latest
```

- Ng new UI
- Ng serve

Angular.json ==> Angular CLI configuration
Package.json ==> Dependency's

Component

Friday, March 5, 2021 12:38 PM


Code File Edit Selection View Go Run Terminal Window Help
100% Neil
app.component.ts — DatingApp
index.html
app.component.ts
client > src > app > app.component.ts > AppComponent
master* ⊗ 0 ⚠ 0 DatingApp.sln
Ln 3, Col 13 Spaces: 2 UTF-8 LF TypeScript 3.9.7

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

@Component({
```

```
(alias) Component(obj: Component): TypeDecorator
import Component
```

Decorator that marks a class as an Angular component and provides configuration metadata that determines how the component should be processed, instantiated, and used at runtime.

Components are the most basic UI building block of an Angular app. An Angular app contains a tree of Angular components.

Angular components are a subset of directives, always associated with a template. Unlike other directives, only one component can be instantiated for a given element in a template.

```
<li *ngFor='let Appusers of user'>{{ Appusers.id}} -- {{Appusers.userName}}</li>
```

Add Bootstrap

Tuesday, March 9, 2021 12:33 PM

```
Npm install
ng add ngx-bootstrap
```

In Angular.json
Look for "Build" section

```
"./node_modules/bootstrap/dist/css/bootstrap.min.css",
            "./node_modules/ngx-bootstrap/datepicker/bs-datepicker.css"
```

```
npm install font-awesome
```

TODO

Thursday, March 4, 2021 7:19 PM

Coding Standards

- Variable Naming Convention
- URL or Routing Convention
-

GIT

Friday, March 19, 2021   12:57 PM

Working Directory --> Stagging Area --> .git Repository --> Remote Repository

Git add . [Move changes to staging Area]
Git commit -m "Initial Message" [Move changes to .Net Repository]
Git Push [Move changes to Remote Repository]
git commit -am "File Rename" [Direct commit from Working Directory to .Net Repository]

Undo Changes

git reset HEAD . [Delete the changes in Stage]
git checkout . [Undo the changes in working Working Directory]
git reset XXXXX

Help Commands
Git log --oneline

Comparing
----
git difftool == working directory Vs Stage
Git diff HEAD or git difftool HEAD== Working Directory vs .Git folder (i.e Last Committed)
Git diff --staged HEAD [Stage Vs .Git Folder]

Compare Commits

Git log --oneline
git diff 2773c97 HEAD

HEAD is last commit

Git Branch

git branch MyFirstBranch [Create "MyFirstBranch" branch]
git branch -a [List all branch]
Git branch -m MyFirstBarnch TestBranch [Rename branch to "TestBranch"]
Git branch -d TestBranch [Delete Test Branch]

git checkout -b MyNewBranch [Create "MyNewBranch" and Checkout "MyNewBranch]

git push https://github.com/dharinath/Pratices.git feature1

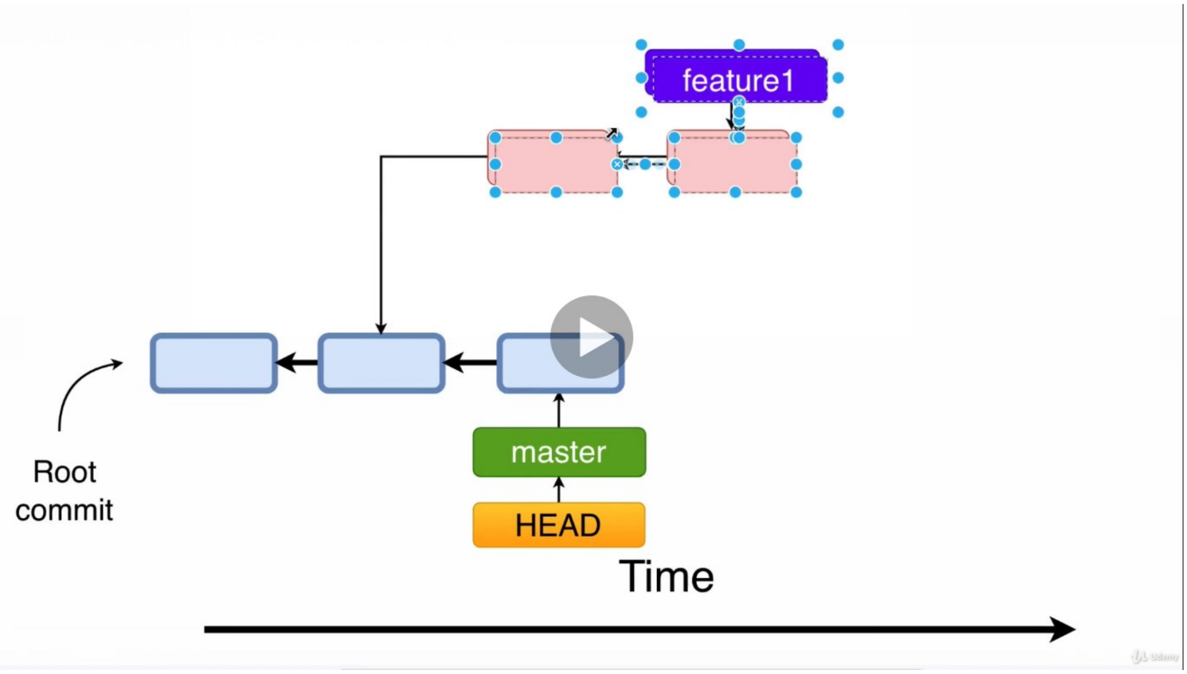
Git Merge

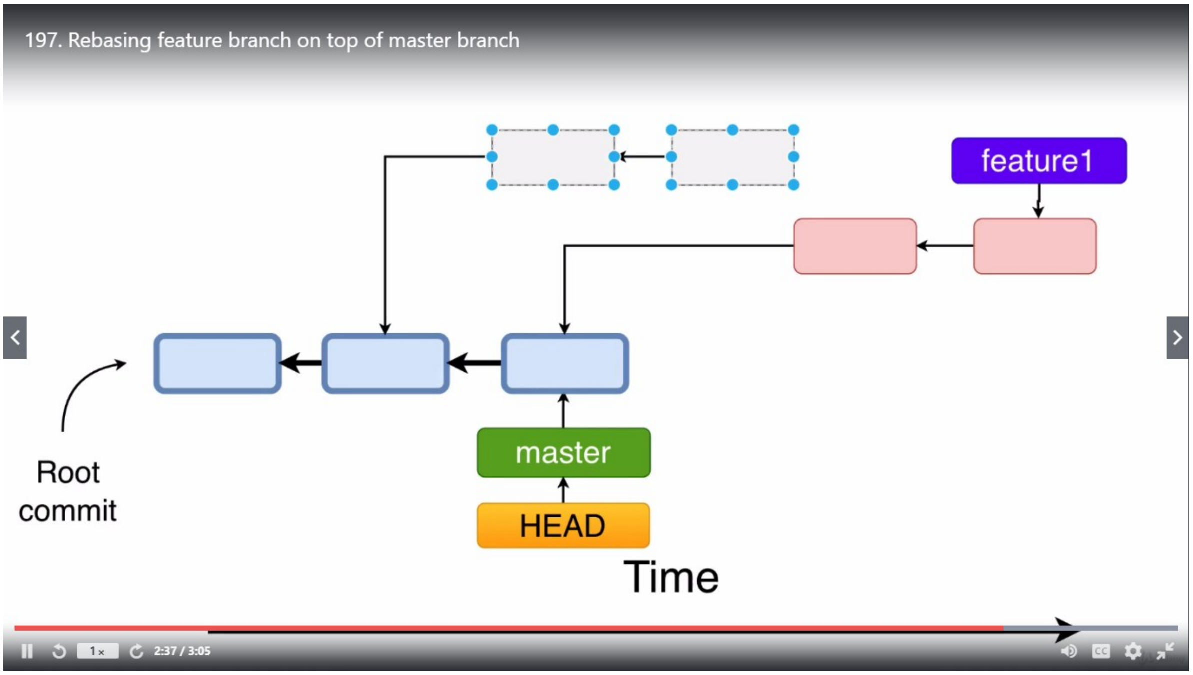
- Git checkout master [Checkout where merge needs to perform]
- Git merge FeatureBranch [Feature branch will be Merge with master]

Git merge FeatureBrach --no-ff [Move the last commit only others will be preserved] git log --oneline --decorate --all --graph [See Preserved Changes]

Git Rebase [Pull changes of Master from Feature Branch]

FeatureBranch >>> Git rebase master [This will merge the master changes to feature branch]




197. Rebasing feature branch on top of master branch

Git Stash

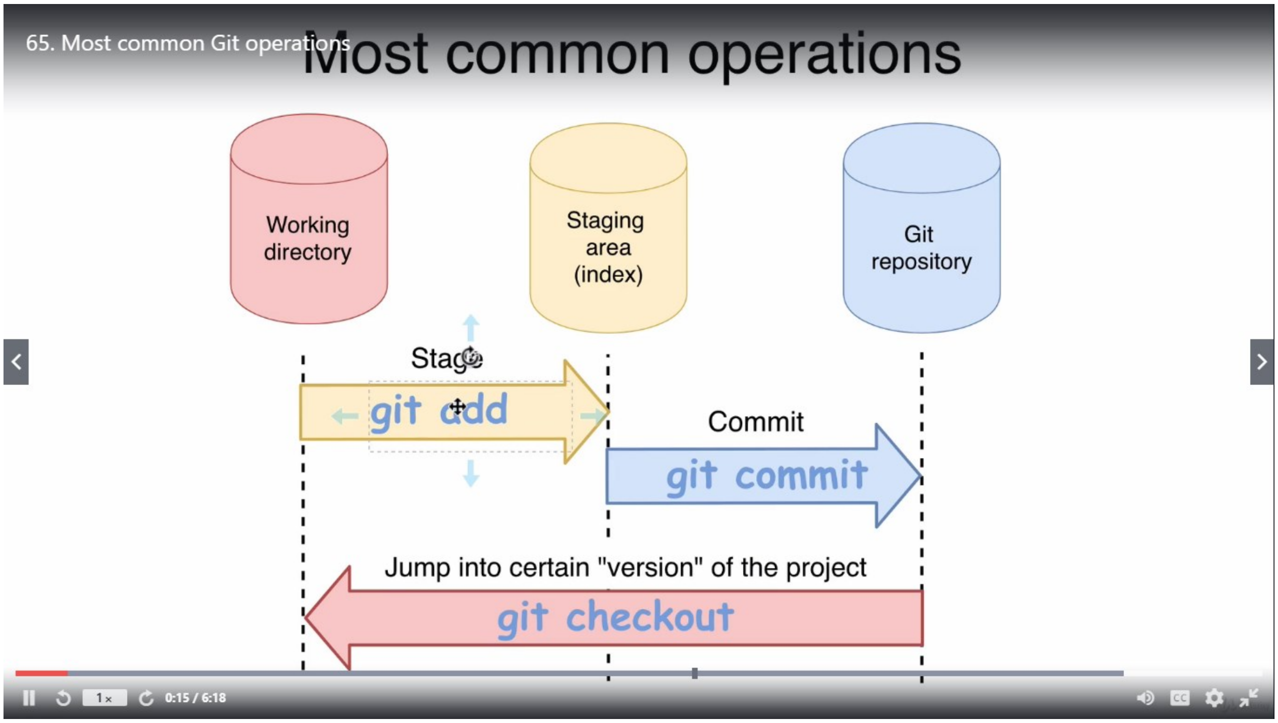
Git stash [Tracked file or modified file only]
Git stash -U [Includes new files or untracked files]
git stash list
Git stash apply
git stash drop

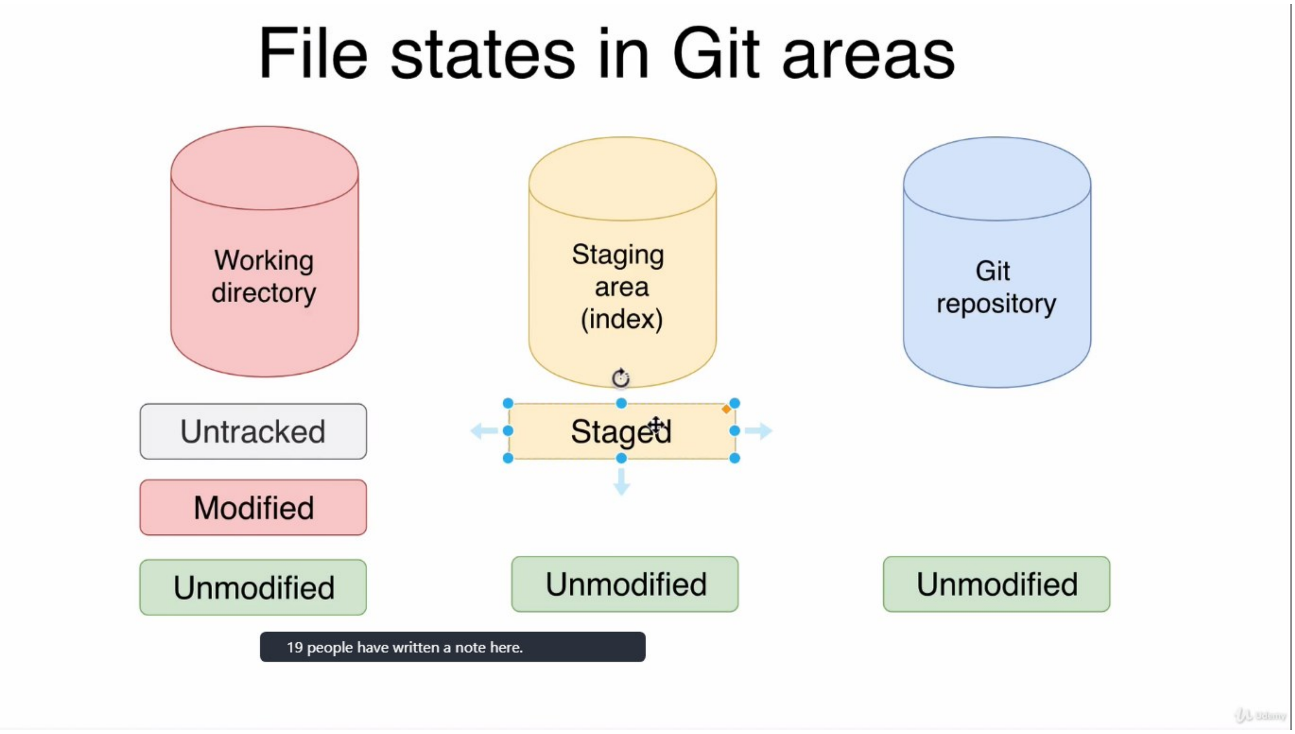
Git stash pop [apply + drop]

Git stash save "Description"
Git stash list
Git stash apply stash{0}

Master Branch >>> Git stash branch existingfeaturebranch [Move stash to "Existing Feature Branch"]

Git pull origin master





File states in Git areas

Working directory

Staging area (index)

Git repository

Untracked

Modified

Unmodified

Staged

Unmodified

Unmodified

19 people have written a note here.

<u>VI Mode Operations</u>

Press I to enter comments
Press "Esc" key and : WQ (save & exit)

<u>Git Staging Commands</u>

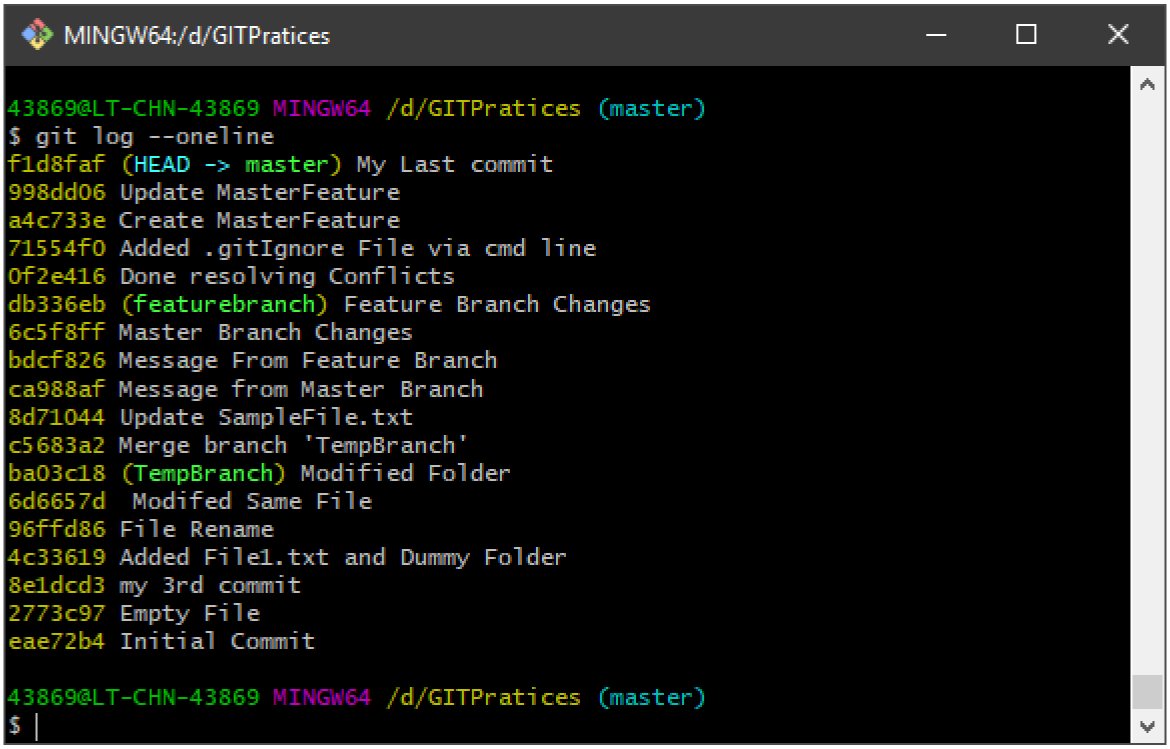
Git ls-files -fs

<u>Git Pull = Git Fetch + Git Merge</u>

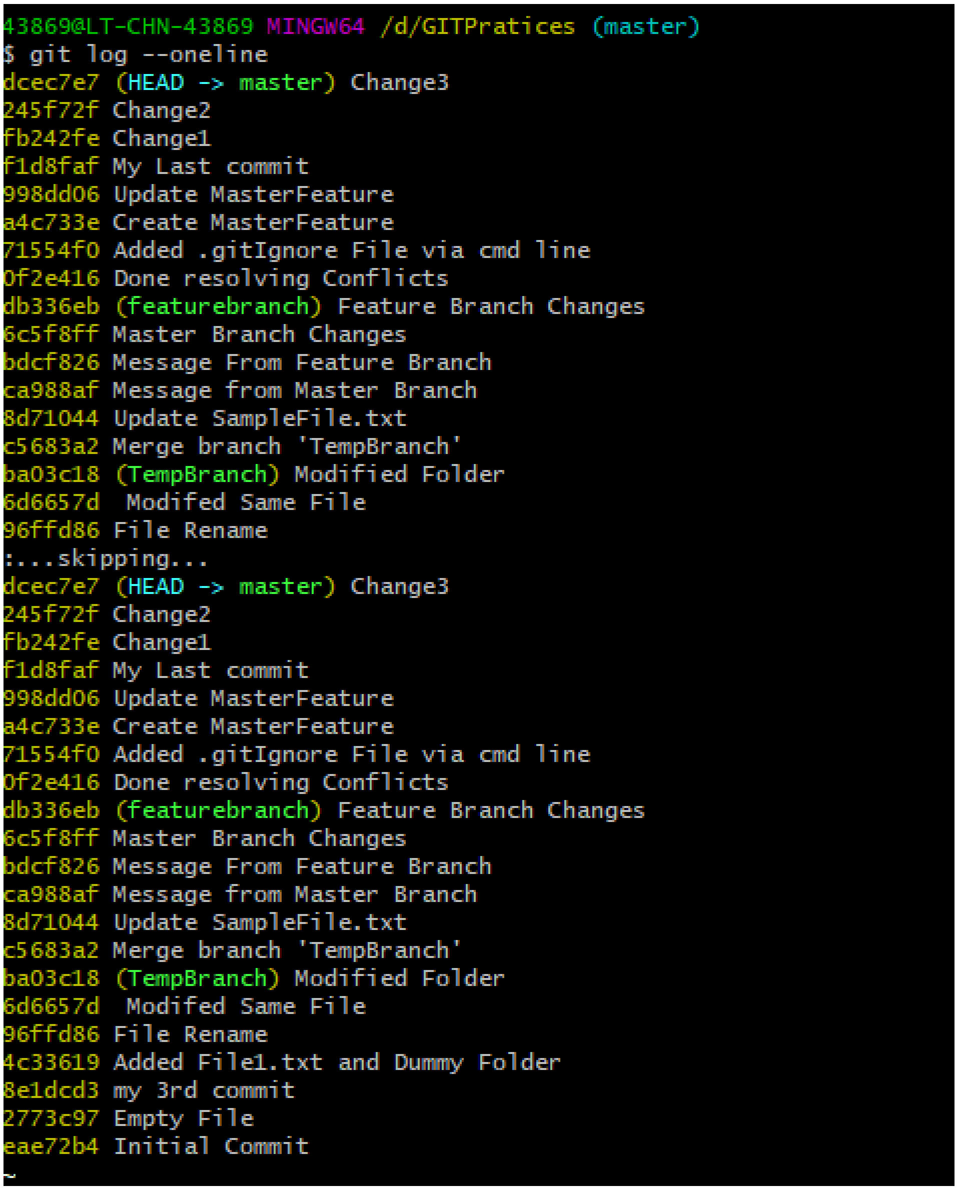
<u>Git reset --hard/soft/mixed(default)</u>

Undoing the Changes

Wednesday, March 24, 2021	1:54 PM



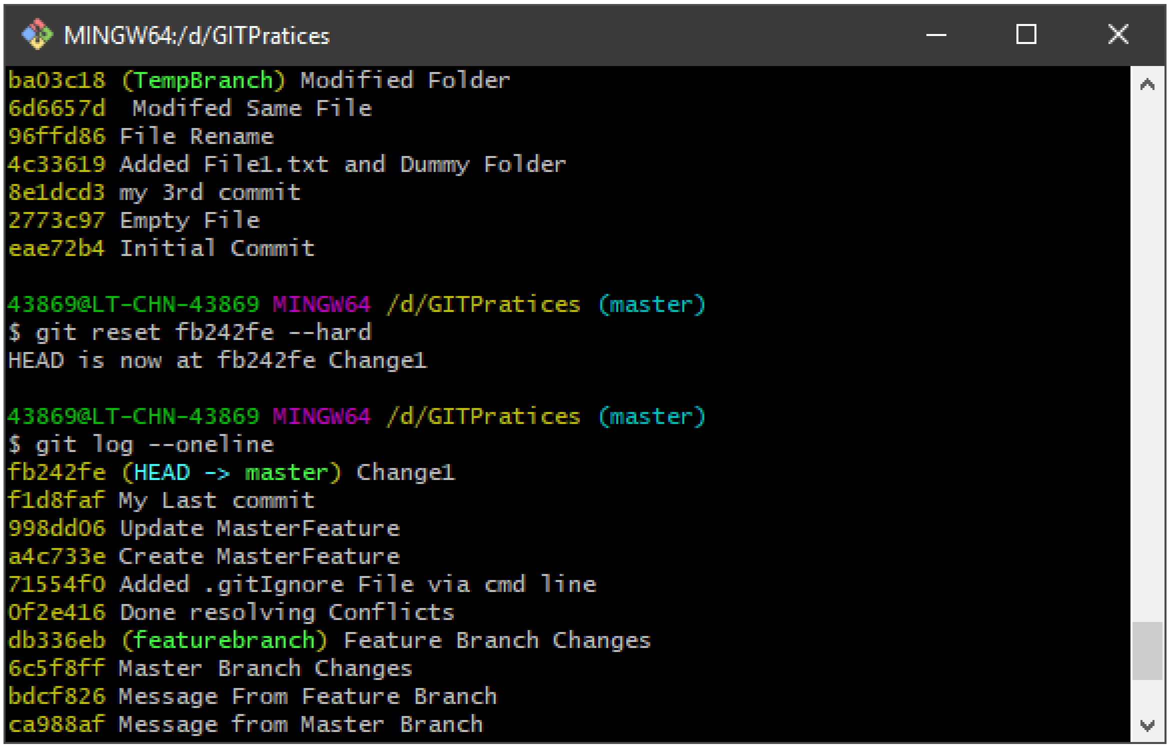
```
43869@LT-CHN-43869 MINGW64 /d/GITPratices (master)
$ git log --oneline
f1d8faf (HEAD -> master) My Last commit
998dd06 Update MasterFeature
a4c733e Create MasterFeature
71554f0 Added .gitIgnore File via cmd line
0f2e416 Done resolving Conflicts
db336eb (featurebranch) Feature Branch Changes
6c5f8ff Master Branch Changes
bdcf826 Message From Feature Branch
ca988af Message from Master Branch
8d71044 Update SampleFile.txt
c5683a2 Merge branch 'TempBranch'
ba03c18 (TempBranch) Modified Folder
6d6657d  Modifed Same File
96ffd86 File Rename
4c33619 Added File1.txt and Dummy Folder
8e1dcd3 my 3rd commit
2773c97 Empty File
eae72b4 Initial Commit

43869@LT-CHN-43869 MINGW64 /d/GITPratices (master)
$ 
```



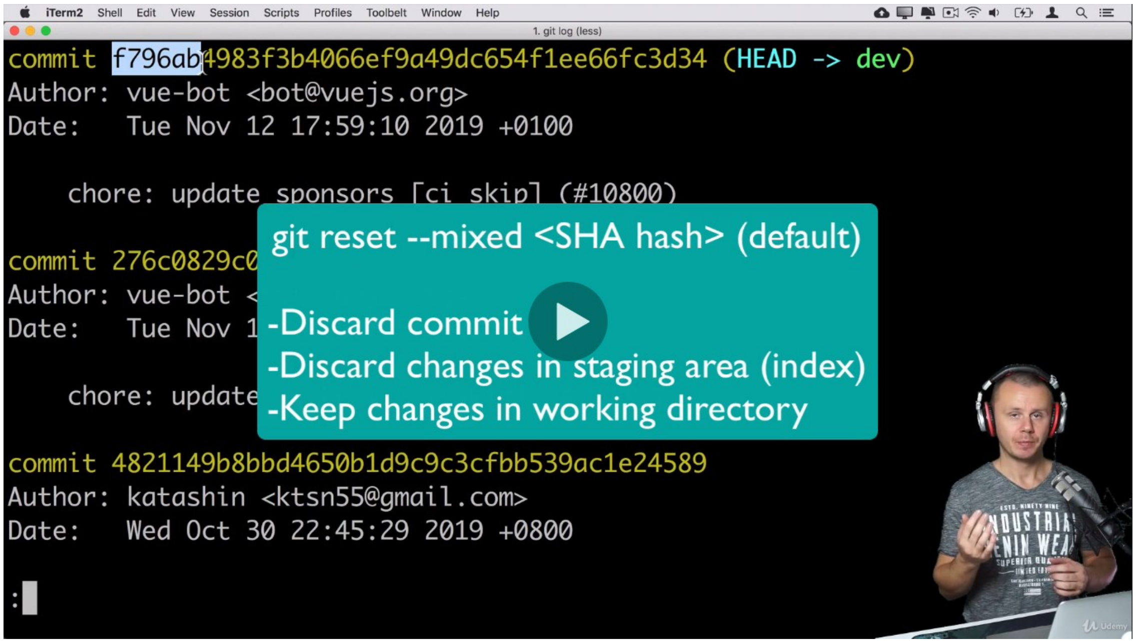
```
43869@LT-CHN-43869 MINGW64 /d/GITPratices (master)
$ git log --oneline
dcec7e7 (HEAD -> master) Change3
245f72f Change2
fb242fe Change1
f1d8faf My Last commit
998dd06 Update MasterFeature
a4c733e Create MasterFeature
71554f0 Added .gitIgnore File via cmd line
0f2e416 Done resolving Conflicts
db336eb (featurebranch) Feature Branch Changes
6c5f8ff Master Branch Changes
bdcf826 Message From Feature Branch
ca988af Message from Master Branch
8d71044 Update SampleFile.txt
c5683a2 Merge branch 'TempBranch'
ba03c18 (TempBranch) Modified Folder
6d6657d  Modifed Same File
96ffd86 File Rename
:...skipping...
dcec7e7 (HEAD -> master) Change3
245f72f Change2
fb242fe Change1
f1d8faf My Last commit
998dd06 Update MasterFeature
a4c733e Create MasterFeature
71554f0 Added .gitIgnore File via cmd line
0f2e416 Done resolving Conflicts
db336eb (featurebranch) Feature Branch Changes
6c5f8ff Master Branch Changes
bdcf826 Message From Feature Branch
ca988af Message from Master Branch
8d71044 Update SampleFile.txt
c5683a2 Merge branch 'TempBranch'
ba03c18 (TempBranch) Modified Folder
6d6657d  Modifed Same File
96ffd86 File Rename
4c33619 Added File1.txt and Dummy Folder
8e1dcd3 my 3rd commit
2773c97 Empty File
eae72b4 Initial Commit
~
```

git rebase --abort

git reset fb242fe --hard [Set the Head to specified commit



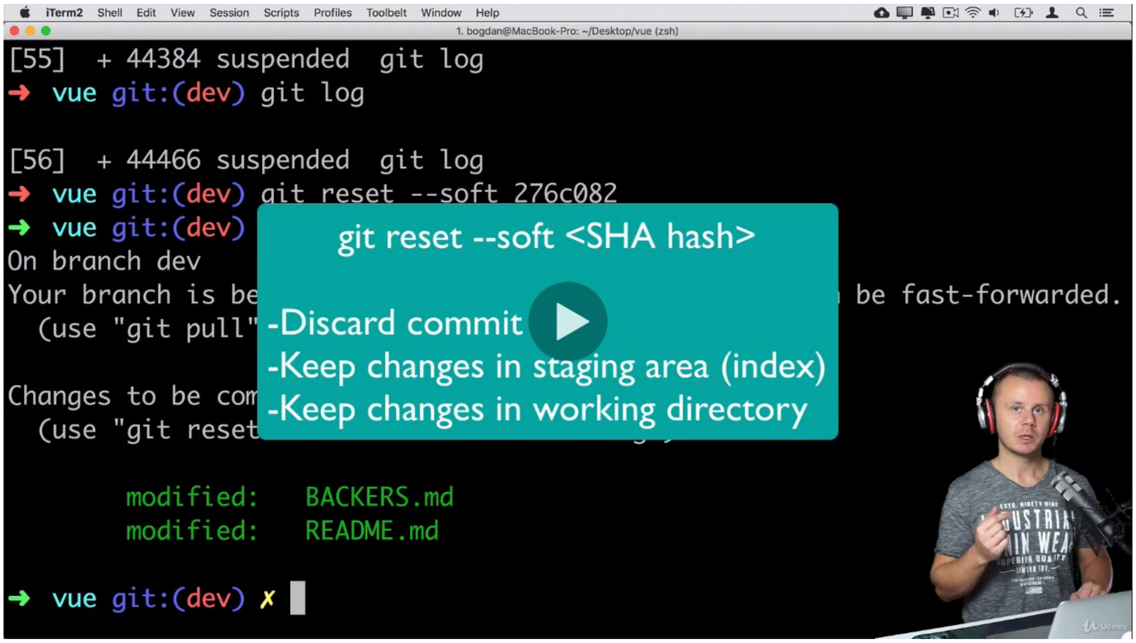
```
ba03c18 (TempBranch) Modified Folder
6d6657d  Modifed Same File
96ffd86 File Rename
4c33619 Added File1.txt and Dummy Folder
8e1dcd3 my 3rd commit
2773c97 Empty File
eae72b4 Initial Commit

43869@LT-CHN-43869 MINGW64 /d/GITPratices (master)
$ git reset fb242fe --hard
HEAD is now at fb242fe Change1

43869@LT-CHN-43869 MINGW64 /d/GITPratices (master)
$ git log --oneline
fb242fe (HEAD -> master) Change1
f1d8faf My Last commit
998dd06 Update MasterFeature
a4c733e Create MasterFeature
71554f0 Added .gitIgnore File via cmd line
0f2e416 Done resolving Conflicts
db336eb (featurebranch) Feature Branch Changes
6c5f8ff Master Branch Changes
bdcf826 Message From Feature Branch
ca988af Message from Master Branch
```


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. git log (less)

```
commit f796ab4983f3b4066ef9a49dc654f1ee66fc3d34 (HEAD -> dev)
Author: vue-bot <bot@vuejs.org>
Date:   Tue Nov 12 17:59:10 2019 +0100

    chore: update sponsors [ci skip] (#10800)

commit 276c0829c0
Author: vue-bot <
Date:   Tue Nov 1

    chore: update

commit 4821149b8bbd4650b1d9c9c3cfbb539ac1e24589
Author: katashin <ktsn55@gmail.com>
Date:   Wed Oct 30 22:45:29 2019 +0800

:
```

git reset --mixed <SHA hash> (default)

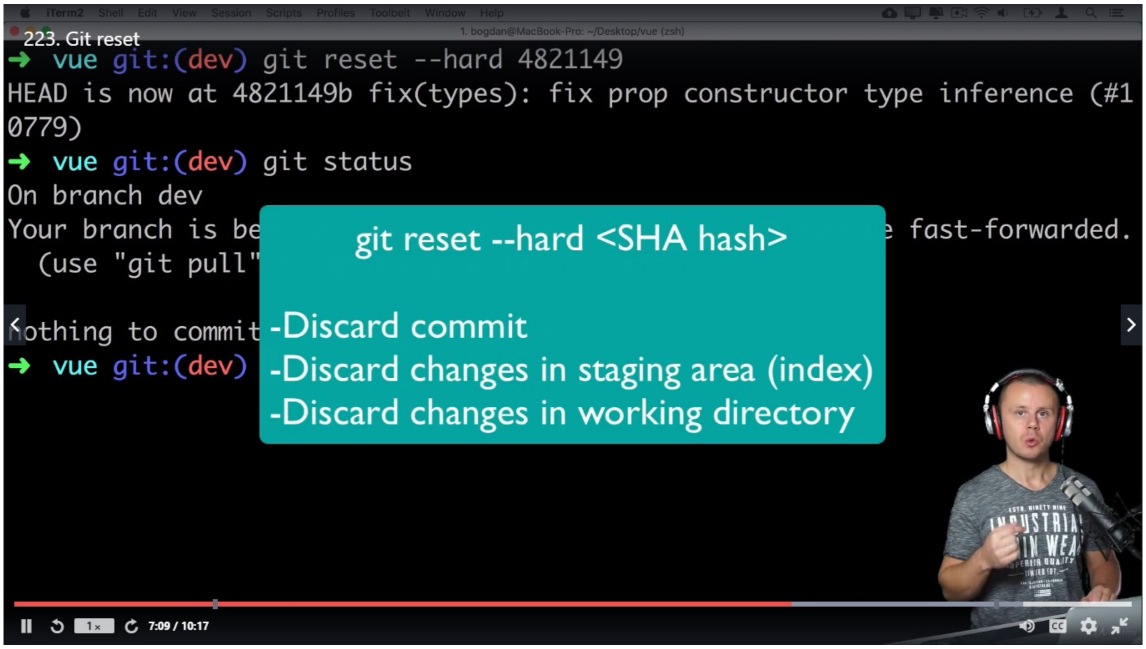
-Discard commit
-Discard changes in staging area (index)
-Keep changes in working directory


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. bogdan@MacBook-Pro: ~/Desktop/vue (zsh)

```
[55]  + 44384 suspended  git log
➜ vue git:(dev) git log

[56]  + 44466 suspended  git log
➜ vue git:(dev) git reset --soft 276c082
➜ vue git:(dev)
On branch dev
Your branch is be                                 be fast-forwarded.
  (use "git pull"

Changes to be com
  (use "git reset

        modified:   BACKERS.md
        modified:   README.md

➜ vue git:(dev) ✗
```

git reset --soft <SHA hash>

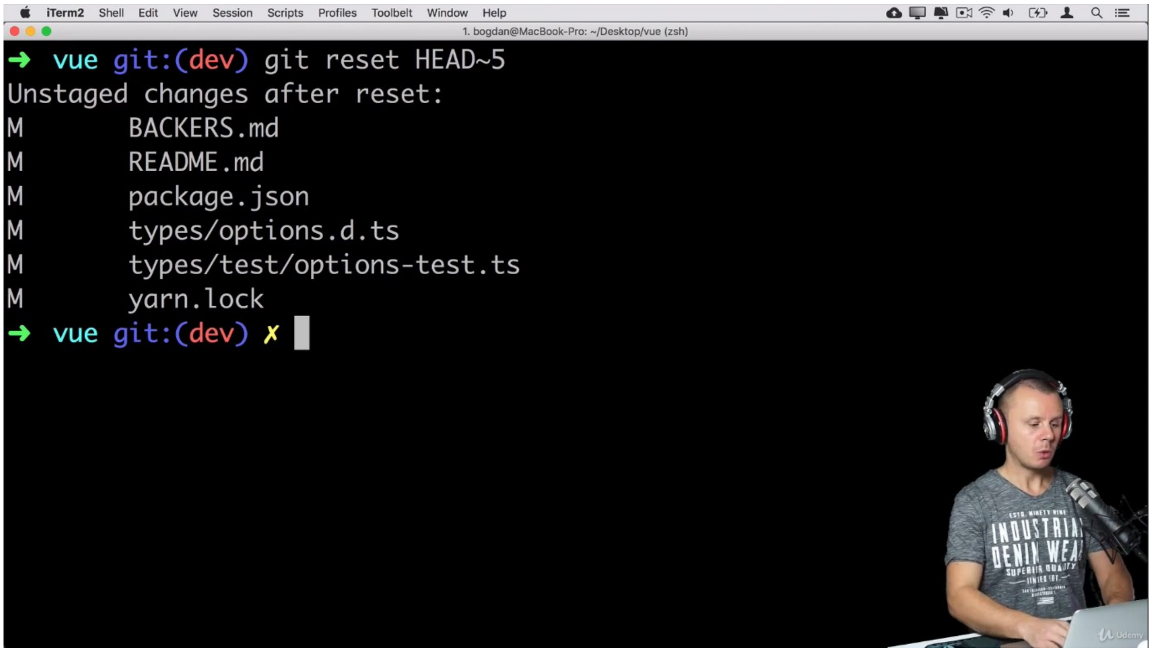
-Discard commit
-Keep changes in staging area (index)
-Keep changes in working directory


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. bogdan@MacBook-Pro: ~/Desktop/vue (zsh)
223. Git reset
1x
7:09 / 10:17

```
➜ vue git:(dev) git reset --hard 4821149
HEAD is now at 4821149b fix(types): fix prop constructor type inference (#1
0779)
➜ vue git:(dev) git status
On branch dev
Your branch is be                                  e fast-forwarded.
  (use "git pull"

nothing to commit
➜ vue git:(dev)
```

git reset --hard <SHA hash>

-Discard commit
-Discard changes in staging area (index)
-Discard changes in working directory



```
➜  vue git:(dev) git reset HEAD~5
Unstaged changes after reset:
M       BACKERS.md
M       README.md
M       package.json
M       types/options.d.ts
M       types/test/options-test.ts
M       yarn.lock
➜  vue git:(dev) ✗ 
```

Swagger Implementation

Thursday, June 10, 2021 2:08 PM

Step 1: Add Package Nswag.AspNetCore
Step 2:

Swagger with .NET Core (1).

Swagger with .NET Core

1) Add the SwashBuckle.AspNetCore package

   It provides 3 packages

   a) Swashbucke.AspNetCore.Swagger
   b) Swashbuckle.AspNetCore.SwaggerGen
   c) Swashbuckle.AspNetCore.SwaggerUI

2) Add the following lines of code in ConfigureServices method

```
services.AddSwaggerGen(options =>
{
    //Customize the Swagger Doc
    options.SwaggerDoc("v1", new Info {
        Title = "SynBlog API",
        Version = "v1",
    });
});
```

3) Add the Middleware for Swagger in Configure method

```
app.UseSwagger();
app.UseSwaggerUI(options =>
{
    options.SwaggerEndpoint("/swagger/v1/swagger.json", "SynBlog API");
    options.RoutePrefix = string.Empty; // to serve swagger UI in root url
});
```

Customize the Swagger Doc

1) Add the following code in Project file (csproj)

```
<PropertyGroup>
  <GenerateDocumentationFile>true</GenerateDocumentationFile>
  <NoWarn>$(NoWarn);1591</NoWarn>
</PropertyGroup>
```

2) Update the program.cs file and add the pragma statements

```
#pragma warning disable CS1591
public class Program
{
    public static void Main(string[] args)
    {
        CreateWebHostBuilder(args).Build().Run();
    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
        WebHost.CreateDefaultBuilder(args)
            .UseStartup<Startup>();
}
#pragma warning restore CS1591
```

3) Add the documentation comment in the API method. Add the <responses > element with status codes and description

```
/// <summary>
/// Adds a new blog to the blogs collection
/// </summary>
/// <remarks>
/// Sample request:
/// POST /api/blogs
/// {
///     "title":"Blog Title",
///     "Content":"Blog content",
///     "AddedBy":"Author email"
///     "AddedDate":"Posted Date"
/// }
/// </remarks>
/// <param name="blog"></param>
/// <returns>A newly created blog item</returns>
/// <response code="201">Returns a newly created blog item</response>
/// <response code="400">If the item is null or invalid</response>
[ProducesResponseType(201)]
[ProducesResponseType(400)]
[HttpPost("",Name ="AddBlog")]
public async Task<ActionResult<Blog>> AddBlog(Blog blog)
{
    if (blog == null)
    {
        return BadRequest();
    }
    var item=await repo.AddAsync(blog);
    return item;
}
```

4) Update the swagger service configuration in ConfigureServices method.

```
services.AddSwaggerGen(options =>
{
    //Customize the Swagger Doc
    options.SwaggerDoc("v1", new Info
    {
        Title = "SynBlog API",
        Version = "v1",
        Description = "Synergetics Blogs API",
        TermsOfService = "None",
        Contact = new Contact()
        {
            Name = "Sonu Sathyadas",
            Email = "sonusathyadas@hotmail.com",
            Url = "https://streamingskills.blog/"

        },
        License = new License
        {
            Name = "MIT License info",
            Url = "https://streamingskills.blog/licence"
        }
    });
    var xmlFile = $"{Assembly.GetExecutingAssembly().GetName().Name}.xml";
    var xmlPath = Path.Combine(AppContext.BaseDirectory, xmlFile);
    options.IncludeXmlComments(xmlPath);
});
```

Customizing the Swagger UI

1) Open the Startup class and add the StaticFiles middleware.

```
app.UseStaticFiles();
```

2) Acquire the contents of the *dist* folder from the Swagger UI GitHub repository. This folder contains the necessary assets for the Swagger UI page. https://github.com/swagger-api/swagger-ui/tree/master/dist
3) Create a ***wwwroot/swagger/ui*** folder, and copy into it the contents of the *dist* folder.
4) Create a ***custom.css*** file, in ***wwwroot/swagger/ui,*** with the following CSS to customize the page header:

```css
.swagger-ui .topbar {
    background-color: #000;
    border-bottom: 3px solid #547f00;
}
```

5) Reference *custom.css* in the *index.html* file, after any other CSS files
6) Browse to the *index.html* page at `http://localhost:<port>/swagger/ui/index.html.` Enter `http://localhost:<port>/swagger/v1/swagger.json` in the header's textbox, and click the **Explore** button.