

Namespace server.Controllers

Classes

[UserController](#)

Контроллер, представляющий методы работы с действиями пользователя (регистрация и авторизация).

Class UserController

Namespace: [server.Controllers](#)

Assembly: server.dll


Контроллер, представляющий методы работы с действиями пользователя (регистрация и авторизация).

```
[Route("api/[controller]")]
[ApiController]
public class UserController : ControllerBase
```

Inheritance

[object](#)  ← [ControllerBase](#)  ← UserController

Inherited Members

[ControllerBase.StatusCode\(int\)](#)  , [ControllerBase.StatusCode\(int, object\)](#)  ,
[ControllerBase.Content\(string\)](#)  , [ControllerBase.Content\(string, string\)](#)  ,
[ControllerBase.Content\(string, string, Encoding\)](#)  ,
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#)  , [ControllerBase.NoContent\(\)](#)  ,
[ControllerBase.Ok\(\)](#)  , [ControllerBase.Ok\(object\)](#)  , [ControllerBase.Redirect\(string\)](#)  ,
[ControllerBase.RedirectPermanent\(string\)](#)  , [ControllerBase.RedirectPreserveMethod\(string\)](#)  ,
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#)  , [ControllerBase.LocalRedirect\(string\)](#)  ,
[ControllerBase.LocalRedirectPermanent\(string\)](#)  , [ControllerBase.LocalRedirectPreserveMethod\(string\)](#)  ,
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#)  , [ControllerBase.RedirectToAction\(\)](#)  ,
[ControllerBase.RedirectToAction\(string\)](#)  , [ControllerBase.RedirectToAction\(string, object\)](#)  ,
[ControllerBase.RedirectToAction\(string, string\)](#)  ,
[ControllerBase.RedirectToAction\(string, string, object\)](#)  ,
[ControllerBase.RedirectToAction\(string, string, string\)](#)  ,
[ControllerBase.RedirectToAction\(string, string, object, string\)](#)  ,
[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string, object\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string, string\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string, string, string\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string, string, object\)](#)  ,
[ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#)  ,
[ControllerBase.RedirectToActionPermanentPreserveMethod\(string, string, object, string\)](#)  ,
[ControllerBase.RedirectToRoute\(string\)](#)  , [ControllerBase.RedirectToRoute\(object\)](#)  ,

[ControllerBase.RedirectToRoute\(string, object\).☐](#) , [ControllerBase.RedirectToRoute\(string, string\).☐](#) ,
[ControllerBase.RedirectToRoute\(string, object, string\).☐](#) ,
[ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\).☐](#) ,
[ControllerBase.RedirectToRoutePermanent\(string\).☐](#) ,
[ControllerBase.RedirectToRoutePermanent\(object\).☐](#) ,
[ControllerBase.RedirectToRoutePermanent\(string, object\).☐](#) ,
[ControllerBase.RedirectToRoutePermanent\(string, string\).☐](#) ,
[ControllerBase.RedirectToRoutePermanent\(string, object, string\).☐](#) ,
[ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\).☐](#) ,
[ControllerBase.RedirectToPage\(string\).☐](#) , [ControllerBase.RedirectToPage\(string, object\).☐](#) ,
[ControllerBase.RedirectToPage\(string, string\).☐](#) , [ControllerBase.RedirectToPage\(string, string, object\).☐](#) ,
[ControllerBase.RedirectToPage\(string, string, string\).☐](#) ,
[ControllerBase.RedirectToPage\(string, string, object, string\).☐](#) ,
[ControllerBase.RedirectToPagePermanent\(string\).☐](#) ,
[ControllerBase.RedirectToPagePermanent\(string, object\).☐](#) ,
[ControllerBase.RedirectToPagePermanent\(string, string\).☐](#) ,
[ControllerBase.RedirectToPagePermanent\(string, string, string\).☐](#) ,
[ControllerBase.RedirectToPagePermanent\(string, string, object, string\).☐](#) ,
[ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\).☐](#) ,
[ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\).☐](#) ,
[ControllerBase.File\(byte\[\], string\).☐](#) , [ControllerBase.File\(byte\[\], string, bool\).☐](#) ,
[ControllerBase.File\(byte\[\], string, string\).☐](#) , [ControllerBase.File\(byte\[\], string, string, bool\).☐](#) ,
[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.File\(Stream, string\).☐](#) , [ControllerBase.File\(Stream, string, bool\).☐](#) ,
[ControllerBase.File\(Stream, string, string\).☐](#) , [ControllerBase.File\(Stream, string, string, bool\).☐](#) ,
[ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.File\(string, string\).☐](#) , [ControllerBase.File\(string, string, bool\).☐](#) ,
[ControllerBase.File\(string, string, string\).☐](#) , [ControllerBase.File\(string, string, string, bool\).☐](#) ,
[ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\).☐](#) ,
[ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\).☐](#) ,
[ControllerBase.PhysicalFile\(string, string\).☐](#) , [ControllerBase.PhysicalFile\(string, string, bool\).☐](#) ,
[ControllerBase.PhysicalFile\(string, string, string\).☐](#) ,

[ControllerBase.PhysicalFile\(string, string, string, bool\)](#),
[ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),
[ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),
[ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),
[ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),
[ControllerBase.Unauthorized\(\)](#), [ControllerBase.Unauthorized\(object\)](#), [ControllerBase.NotFound\(\)](#),
[ControllerBase.NotFound\(object\)](#), [ControllerBase.BadRequest\(\)](#),
[ControllerBase.BadRequest\(object\)](#), [ControllerBase.BadRequest\(ModelStateDictionary\)](#),
[ControllerBase.UnprocessableEntity\(\)](#), [ControllerBase.UnprocessableEntity\(object\)](#),
[ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#), [ControllerBase.Conflict\(\)](#),
[ControllerBase.Conflict\(object\)](#), [ControllerBase.Conflict\(ModelStateDictionary\)](#),
[ControllerBase.Problem\(string, string, int?, string, string\)](#),
[ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#),
[ControllerBase.ValidationProblem\(ModelStateDictionary\)](#), [ControllerBase.ValidationProblem\(\)](#),
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#),
[ControllerBase.Created\(\)](#), [ControllerBase.Created\(string, object\)](#),
[ControllerBase.Created\(Uri, object\)](#), [ControllerBase.CreatedAtAction\(string, object\)](#),
[ControllerBase.CreatedAtAction\(string, object, object\)](#),
[ControllerBase.CreatedAtAction\(string, string, object, object\)](#),
[ControllerBase.CreatedAtRoute\(string, object\)](#), [ControllerBase.CreatedAtRoute\(object, object\)](#),
[ControllerBase.CreatedAtRoute\(string, object, object\)](#), [ControllerBase.Accepted\(\)](#),
[ControllerBase.Accepted\(object\)](#), [ControllerBase.Accepted\(Uri\)](#), [ControllerBase.Accepted\(string\)](#),
[ControllerBase.Accepted\(string, object\)](#), [ControllerBase.Accepted\(Uri, object\)](#),
[ControllerBase.AcceptedAtAction\(string\)](#), [ControllerBase.AcceptedAtAction\(string, string\)](#),
[ControllerBase.AcceptedAtAction\(string, object\)](#),
[ControllerBase.AcceptedAtAction\(string, string, object\)](#),
[ControllerBase.AcceptedAtAction\(string, object, object\)](#),
[ControllerBase.AcceptedAtRoute\(object\)](#), [ControllerBase.AcceptedAtRoute\(string\)](#),
[ControllerBase.AcceptedAtRoute\(string, object\)](#), [ControllerBase.AcceptedAtRoute\(object, object\)](#),
[ControllerBase.AcceptedAtRoute\(string, object, object\)](#), [ControllerBase.Challenge\(\)](#),
[ControllerBase.Challenge\(params string\[\]\)](#), [ControllerBase.Challenge\(AuthenticationProperties\)](#),
[ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#), [ControllerBase.Forbid\(\)](#),
[ControllerBase.Forbid\(params string\[\]\)](#), [ControllerBase.Forbid\(AuthenticationProperties\)](#),
[ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#),
[ControllerBase.SignIn\(ClaimsPrincipal\)](#), [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#),
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#),
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\)](#), [ControllerBase.SignOut\(\)](#),
[ControllerBase.SignOut\(AuthenticationProperties\)](#), [ControllerBase.SignOut\(params string\[\]\)](#),
[ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\)](#),

[ControllerBase.TryUpdateModelAsync<TModel>\(TModel\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\)](#),
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[ControllerBase.TryUpdateModelAsync\(object, Type, string\)](#),
[ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\)](#),
[ControllerBase.TryValidateModel\(object\)](#), [ControllerBase.TryValidateModel\(object, string\)](#),
[ControllerBase.HttpContext](#), [ControllerBase.Request](#), [ControllerBase.Response](#),
[ControllerBase.RouteData](#), [ControllerBase.ModelState](#), [ControllerBase.ControllerContext](#),
[ControllerBase.MetadataProvider](#), [ControllerBase.ModelBinderFactory](#), [ControllerBase.Url](#),
[ControllerBase.ObjectValidator](#), [ControllerBase.ProblemDetailsFactory](#), [ControllerBase.User](#),
[ControllerBase.Empty](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

UserController(IUserService, ApplicationDbContext)

Инициализация нового экземпляра [UserController](#) класса.

```
public UserController(IUserService userService, ApplicationDbContext dbContext)
```

Parameters

userService [IUserService](#)

Сервис пользователя.

dbContext [ApplicationDbContext](#)

Контекст базы данных.

Methods

Authorization(AuthDto)

Авторизация специализированной DTO пользователя.

```
[HttpPost("authorization")]  
public Task<IActionResult> Authorization(AuthDto authDto)
```

Parameters

authDto [AuthDto](#)

DTO авторизации пользователя.

Returns

[Task](#) <[IActionResult](#)>

[IActionResult](#)

Remarks

Позволяет авторизовать пользователя с существующим в системе адресом электронной почты и соответствующим зарегистрированному пользователю паролем.

Register(RegisterDto)

Регистрация специализированной DTO пользователя.

```
[HttpPost("register")]  
public Task<IActionResult> Register(RegisterDto registerDto)
```

Parameters

registerDto [RegisterDto](#)

DTO регистрации пользователя.

Returns

[Task](#) < [IActionResult](#) >

[IActionResult](#)

Remarks

Позволяет зарегистрировать нового пользователя с уникальным никнеймом и адресом электронной почты.

See Also

[ControllerBase](#)

Namespace server.DTOs

Classes

[AuthDto](#)

Aurization Dto class.

[RegisterDto](#)

Register Dto class.

Class AuthDto

Namespace: [server.DTOs](#)

Assembly: server.dll

Authorization Dto class.

```
public class AuthDto
```

Inheritance

[object](#)  ← AuthDto

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Properties

Email

Gets or sets the email.

```
[Required(ErrorMessage = "Email является обязательным для заполнения..")]  
[StringLength(100, ErrorMessage = "Неверный формат почты. Проверьте и попробуйте снова..")]  
[EmailAddress(ErrorMessage = "Неверный формат почты. Проверьте и попробуйте снова..")]  
public string Email { get; set; }
```

Property Value

[string](#) 

The email.

Password

Gets or sets the password.

```
[Required(ErrorMessage = "Пароль является обязательным для заполнения..")]  
[StringLength(100, MinimumLength = 6, ErrorMessage = "Пароль должен содержать минимум  
6 символов..")]  
public string Password { get; set; }
```

Property Value

[string](#) 

The password.

Class RegisterDto


Namespace: [server.DTOs](#)

Assembly: server.dll








Register Dto class.

```
public class RegisterDto
```

Inheritance

[object](#)  ← RegisterDto

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Properties

ConfirmPassword

Gets or sets the confirm password.

```
[Compare("Password", ErrorMessage = "Пароли не совпадают для заполнения..")]  
[Required(ErrorMessage = "Подтверждение пароля является обязательным.")]  
public string ConfirmPassword { get; set; }
```

Property Value

[string](#) 

The confirm password.

Email

Gets or sets the email.

```
[Required(ErrorMessage = "Email является обязательным для заполнения..")]
[StringLength(100, ErrorMessage = "Неверный формат почты. Проверьте и попробуйте снова.")]
[EmailAddress(ErrorMessage = "Неверный формат почты. Проверьте и попробуйте снова.")]
public string Email { get; set; }
```

Property Value

[string](#)

The email.

Nickname

Gets or sets the nickname.

```
[Required(ErrorMessage = "Никнейм является обязательным для заполнения.")]
[MinLength(6, ErrorMessage = "Никнейм должен содержать минимум 6 английских символов.")]
[StringLength(100, ErrorMessage = "Никнейм не может превышать 100 символов.")]
public string Nickname { get; set; }
```

Property Value

[string](#)

The nickname.

Password

Gets or sets the password.

```
[Required(ErrorMessage = "Пароль является обязательным для заполнения..")]
[StringLength(100, MinimumLength = 6, ErrorMessage = "Пароль должен содержать минимум 6 символов.")]
public string Password { get; set; }
```

Property Value

[string](#)

The password.

Namespace server.Interfaces

Interfaces

[IUserService](#)

Interface IUserService

Namespace: [server.Interfaces](#)

Assembly: server.dll

```
public interface IUserService
```

Methods

CreateAsync(User, string)

Creates the asynchronous.

```
Task<bool> CreateAsync(User user, string password)
```

Parameters

user [User](#)

The user.

password [string](#) 

The password.

Returns

[Task](#)  <[bool](#)  >

UserExistInSystemForEmail(string)

Users the exist in system for email.

```
Task<bool> UserExistInSystemForEmail(string email)
```

Parameters

email [string](#)

The email.

Returns

[Task](#) <[bool](#)>

UserExistInSystemForNickname(string)

Users the exist in system for nickname.

```
Task<bool> UserExistInSystemForNickname(string nickname)
```

Parameters

nickname [string](#)

The nickname.

Returns

[Task](#) <[bool](#)>

VerifyAsync(string, string)

Verifies the asynchronous.

```
Task<bool> VerifyAsync(string email, string password)
```

Parameters

email [string](#)

The email.

password [string](#)

The password.

Returns

[Task](#) <[bool](#)>

Namespace server.Models

Classes

[ApplicationDbContext](#)

[History](#)

[Language](#)

[Metric](#)

[MetricsLineOfCodeInterval](#)

[Project](#)

[ProjectMetric](#)

[User](#)

Пользователь.

[UserKey](#)

Class ApplicationDbContext

Namespace: [server.Models](#)









Assembly: server.dll

```
public class ApplicationDbContext : DbContext, IInfrastructure<IServiceProvider>,
    IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,
    IDisposable, IAsyncDisposable
```

Inheritance

[object](#)  ← [DbContext](#)  ← ApplicationDbContext

Implements

[IInfrastructure](#)  <[IServiceProvider](#)  >, [IDbContextDependencies](#) , [IDbSetCache](#) , [IDbContextPoolable](#) ,
[IResettableService](#) , [IDisposable](#) , [IAsyncDisposable](#) 

Inherited Members

[DbContext.Set<TEntity>\(\)](#) , [DbContext.Set<TEntity>\(string\)](#) ,
[DbContext.ConfigureConventions\(ModelConfigurationBuilder\)](#) , [DbContext.SaveChanges\(\)](#) ,
[DbContext.SaveChanges\(bool\)](#) , [DbContext.SaveChangesAsync\(CancellationToken\)](#) ,
[DbContext.SaveChangesAsync\(bool, CancellationToken\)](#) , [DbContext.Dispose\(\)](#) ,
[DbContext.DisposeAsync\(\)](#) , [DbContext.Entry<TEntity>\(TEntity\)](#) , [DbContext.Entry\(object\)](#) ,
[DbContext.Add<TEntity>\(TEntity\)](#) , [DbContext.AddAsync<TEntity>\(TEntity, CancellationToken\)](#) ,
[DbContext.Attach<TEntity>\(TEntity\)](#) , [DbContext.Update<TEntity>\(TEntity\)](#) ,
[DbContext.Remove<TEntity>\(TEntity\)](#) , [DbContext.Add\(object\)](#) ,
[DbContext.AddAsync\(object, CancellationToken\)](#) , [DbContext.Attach\(object\)](#) ,
[DbContext.Update\(object\)](#) , [DbContext.Remove\(object\)](#) , [DbContext.AddRange\(params object\[\]\)](#) ,
[DbContext.AddRangeAsync\(params object\[\]\)](#) , [DbContext.AttachRange\(params object\[\]\)](#) ,
[DbContext.UpdateRange\(params object\[\]\)](#) , [DbContext.RemoveRange\(params object\[\]\)](#) ,
[DbContext.AddRange\(IEnumerable<object>\)](#) ,
[DbContext.AddRangeAsync\(IEnumerable<object>, CancellationToken\)](#) ,
[DbContext.AttachRange\(IEnumerable<object>\)](#) , [DbContext.UpdateRange\(IEnumerable<object>\)](#) ,
[DbContext.RemoveRange\(IEnumerable<object>\)](#) , [DbContext.Find\(Type, params object\[\]\)](#) ,
[DbContext.FindAsync\(Type, params object\[\]\)](#) ,
[DbContext.FindAsync\(Type, object\[\], CancellationToken\)](#) , [DbContext.Find<TEntity>\(params object\[\]\)](#) ,
[DbContext.FindAsync<TEntity>\(params object\[\]\)](#) ,
[DbContext.FindAsync<TEntity>\(object\[\], CancellationToken\)](#) ,
[DbContext.FromExpression<TResult>\(Expression<Func<IQueryable<TResult>>>\)](#) ,
[DbContext.Database](#) , [DbContext.ChangeTracker](#) , [DbContext.Model](#) , [DbContext.ContextId](#) ,

[DbContext.SavingChanges](#) , [DbContext.SavedChanges](#) , [DbContext.SaveChangesFailed](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

ApplicationDbContext()

```
public ApplicationDbContext()
```

ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)

```
public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
```

Parameters

options [DbContextOptions](#) <[ApplicationDbContext](#)>

Properties

Histories

```
public virtual DbSet<History> Histories { get; set; }
```

Property Value

[DbSet](#) <[History](#)>

Languages

```
public virtual DbSet<Language> Languages { get; set; }
```

Property Value

[DbSet](#) <[Language](#)>

Metrics

```
public virtual DbSet<Metric> Metrics { get; set; }
```

Property Value

[DbSet](#) <[Metric](#)>

MetricsLineOfCodeIntervals

```
public virtual DbSet<MetricsLineOfCodeInterval> MetricsLineOfCodeIntervals { get; set; }
```

Property Value

[DbSet](#) <[MetricsLineOfCodeInterval](#)>

ProjectMetrics

```
public virtual DbSet<ProjectMetric> ProjectMetrics { get; set; }
```

Property Value

[DbSet](#) <[ProjectMetric](#)>

Projects

```
public virtual DbSet<Project> Projects { get; set; }
```

Property Value

[DbSet](#) [<Project>](#)

UserKeys

```
public virtual DbSet<UserKey> UserKeys { get; set; }
```

Property Value

[DbSet](#) [<UserKey>](#)

Users

```
public virtual DbSet<User> Users { get; set; }
```

Property Value

[DbSet](#) [<User>](#)

Methods

OnConfiguring(DbContextOptionsBuilder)

Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created. The base implementation does nothing.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

Parameters

optionsBuilder [DbContextOptionsBuilder](#)

A builder used to create or modify options for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure the context.

Remarks

In situations where an instance of [DbContextOptions](#) may or may not have been passed to the constructor, you can use [IsConfigured](#) to determine if the options have already been set, and skip some or all of the logic in [OnConfiguring\(DbContextOptionsBuilder\)](#).

See [DbContext lifetime, configuration, and initialization](#) for more information and examples.

OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in [DbSet<TEntity>](#) properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

modelBuilder [ModelBuilder](#)

The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

Remarks

If a model is explicitly set on the options for this context (via [UseModel\(IModel\)](#)) then this method will not be run. However, it will still run when creating a compiled model.

See [Modeling entity types and relationships](#) for more information and examples.

Class History

Namespace: [server.Models](#)

Assembly: server.dll

```
public class History
```

Inheritance

[object](#) ← History

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Properties

HistoryId

```
public int HistoryId { get; set; }
```

Property Value

[int](#)

Name

```
public string Name { get; set; }
```

Property Value

[string](#)

Project1


```
public virtual Project? Project1 { get; set; }
```

Property Value

[Project](#)

Project1Id

```
public int? Project1Id { get; set; }
```

Property Value

[int](#)?

Project2

```
public virtual Project? Project2 { get; set; }
```

Property Value

[Project](#)

Project2Id

```
public int? Project2Id { get; set; }
```

Property Value

[int](#)?


Class Language

Namespace: [server.Models](#)

Assembly: server.dll

```
public class Language
```

Inheritance

[object](#)  ← Language

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

LanguageId

```
public int LanguageId { get; set; }
```

Property Value

[int](#) 

Name

```
public string Name { get; set; }
```

Property Value

[string](#) 

Projects

```
public virtual ICollection<Project> Projects { get; set; }
```

Property Value

[ICollection](#)  [<Project>](#)

Class Metric

Namespace: [server.Models](#)

Assembly: server.dll

```
public class Metric
```

Inheritance

[object](#) ← Metric

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Properties

IsIgnored

```
public bool IsIgnored { get; set; }
```

Property Value

[bool](#)

MetricTypeId

```
public int? MetricTypeId { get; set; }
```

Property Value

[int](#)?

MetricsId

```
public int MetricsId { get; set; }
```

Property Value

[int](#)

MetricsLineOfCodeIntervals

```
public virtual ICollection<MetricsLineOfCodeInterval> MetricsLineOfCodeIntervals { get; set; }
```

Property Value

[ICollection](#) <[MetricsLineOfCodeInterval](#)>

ProjectMetrics

```
public virtual ICollection<ProjectMetric> ProjectMetrics { get; set; }
```

Property Value

[ICollection](#) <[ProjectMetric](#)>

Projects

```
public virtual ICollection<Project> Projects { get; set; }
```

Property Value

[ICollection](#) <[Project](#)>

Class MetricsLineOfCodeInterval

Namespace: [server.Models](#)

Assembly: server.dll

```
public class MetricsLineOfCodeInterval
```

Inheritance

[object](#) ← MetricsLineOfCodeInterval

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Properties

LineOfCodeIntervalId

```
public int? LineOfCodeIntervalId { get; set; }
```

Property Value

[int](#)?

Metric

```
public virtual Metric? Metric { get; set; }
```

Property Value

[Metric](#)

MetricId

```
public int? MetricId { get; set; }
```

Property Value

[int](#)?

MetricsLineOfCodeIntervalId

```
public int MetricsLineOfCodeIntervalId { get; set; }
```

Property Value

[int](#)

Class Project

Namespace: [server.Models](#)








Assembly: server.dll

```
public class Project
```

Inheritance

[object](#)  ← Project

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

HistoryProject1s

```
public virtual ICollection<History> HistoryProject1s { get; set; }
```

Property Value

[ICollection](#)  <[History](#)>

HistoryProject2s

```
public virtual ICollection<History> HistoryProject2s { get; set; }
```

Property Value

[ICollection](#)  <[History](#)>

Language


```
public virtual Language? Language { get; set; }
```

Property Value

[Language](#)

LanguageId

```
public int? LanguageId { get; set; }
```

Property Value

[int](#)?

Metrics

```
public virtual Metric? Metrics { get; set; }
```

Property Value

[Metric](#)

MetricsId

```
public int? MetricsId { get; set; }
```

Property Value

[int](#)?

Name

```
public string Name { get; set; }
```

Property Value

[string](#)

Patch

```
public string? Patch { get; set; }
```

Property Value

[string](#)

ProjectId

```
public int ProjectId { get; set; }
```

Property Value

[int](#)

ProjectMetrics

```
public virtual ICollection<ProjectMetric> ProjectMetrics { get; set; }
```

Property Value

[ICollection](#) <[ProjectMetric](#)>

Class ProjectMetric

Namespace: [server.Models](#)








Assembly: server.dll

```
public class ProjectMetric
```

Inheritance

[object](#)  ← ProjectMetric

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

Metrics

```
public virtual Metric? Metrics { get; set; }
```

Property Value

[Metric](#)

MetricsId

```
public int? MetricsId { get; set; }
```

Property Value

[int](#)  ?

Project

```
public virtual Project? Project { get; set; }
```

Property Value

[Project](#)

ProjectId

```
public int? ProjectId { get; set; }
```

Property Value

[int](#)?

ProjectMetricsId

```
public int ProjectMetricsId { get; set; }
```

Property Value

[int](#)

Class User

Namespace: [server.Models](#)

Assembly: server.dll

Пользователь.

```
public class User
```

Inheritance

[object](#)  ← User

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

Email

```
public string Email { get; set; }
```

Property Value

[string](#) 

Nickname

```
public string Nickname { get; set; }
```

Property Value

[string](#) 

UserId

```
public int UserId { get; set; }
```

Property Value

[int](#)

UserKeys

```
public virtual ICollection<UserKey> UserKeys { get; set; }
```

Property Value

[ICollection](#) <[UserKey](#)>

Class UserKey

Namespace: [server.Models](#)








Assembly: server.dll

```
public class UserKey
```

Inheritance

[object](#)  ← UserKey

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Properties

Password

```
public byte[] Password { get; set; }
```

Property Value

[byte](#)  []

Salt

```
public byte[] Salt { get; set; }
```

Property Value

[byte](#)  []

User

```
public virtual User? User { get; set; }
```

Property Value

[User](#)

UserId

```
public int? UserId { get; set; }
```

Property Value

[int](#)?

UserKeyId

```
public int UserKeyId { get; set; }
```

Property Value

[int](#)

Namespace server.Services

Classes

[UserService](#)

UserService.

Class UserService

Namespace: [server.Services](#)

Assembly: server.dll

UserService.

```
public class UserService : IUserService
```








Inheritance

[object](#)  ← UserService

Implements

[IUserService](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Constructors

UserService(ApplicationDbContext)

Initializes a new instance of the [UserService](#) class.

```
public UserService(ApplicationDbContext context)
```

Parameters

context [ApplicationDbContext](#)

The context.

Methods

CreateAsync(User, string)

Creates the asynchronous.

```
public Task<bool> CreateAsync(User user, string password)
```

Parameters

user [User](#)

The user.

password [string](#)

The password.

Returns

[Task](#) <[bool](#)>

UserExistInSystemForEmail(string)

Users the exist in system for email.

```
public Task<bool> UserExistInSystemForEmail(string email)
```

Parameters

email [string](#)

The email.

Returns

[Task](#) <[bool](#)>

UserExistInSystemForNickname(string)

Users the exist in system for nickname.

```
public Task<bool> UserExistInSystemForNickname(string nickname)
```

Parameters

nickname [string](#)

The nickname.

Returns

[Task](#) <[bool](#)>

VerifyAsync(string, string)

Verifies the asynchronous.

```
public Task<bool> VerifyAsync(string email, string password)
```

Parameters

email [string](#)

The email.

password [string](#)

The password.

Returns

[Task](#) <[bool](#)>

See Also

[IUserService](#)