Table of Contents

AddressCrud

Namespace SoundParadise.Api.Configuration

Classes

AppConfiguration

ConfigureSwaggerOptions

Class AppConfiguration

Inheritance

object

AppConfiguration

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Configuration

Assembly: SoundParadise.Api.dll

Syntax

```
public class AppConfiguration
```

Properties

AuthenticationApiResources

Declaration

```
public IConfigurationSection AuthenticationApiResources { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

AuthenticationApiScopes

Declaration

```
public IConfigurationSection AuthenticationApiScopes { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

AuthenticationClients

Declaration

```
public IConfigurationSection AuthenticationClients { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

AuthenticationIdentityResources

Declaration

```
public IConfigurationSection AuthenticationIdentityResources { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

BlobOptions

Declaration

```
public IConfigurationSection BlobOptions { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

ConnectionString

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

EncryptionOptions

Declaration

```
public IConfigurationSection EncryptionOptions { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

FondyConfig

Declaration

```
public IConfigurationSection FondyConfig { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

IdentityServerAuthority

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

JwtBearerOptions

Declaration

```
public IConfigurationSection JwtBearerOptions { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

SecretKey

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

SmtpOptions

Declaration

```
public IConfigurationSection SmtpOptions { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

UrlsOptions

Declaration

```
public IConfigurationSection UrlsOptions { get; set; }
```

Property Value

TYPE	DESCRIPTION
IConfigurationSection	

Methods

LoadConfiguration(bool)

Declaration

```
public static AppConfiguration LoadConfiguration(bool isDevelopment)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	isDevelopment	

Returns

TYPE	DESCRIPTION
AppConfiguration	

```
public static AppConfiguration LoadConfiguration(bool isDevelopment)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	isDevelopment	

Class ConfigureSwaggerOptions

Inheritance

object

ConfigureSwaggerOptions

Implements

IConfigureNamedOptions<SwaggerGenOptions>

IConfigureOptions<SwaggerGenOptions>

Inherited Members

[object.Equals(object)](object.Equals)

[object.Equals(object, object)](object.Equals)

[object.GetHashCode()](object.GetHashCode)

[object.GetType()](object.GetType)

[object.MemberwiseClone()](object.MemberwiseClone)

[object.ReferenceEquals(object, object)](object.ReferenceEquals)

[object.ToString()](object.ToString)

Namespace: SoundParadise.Api.Configuration

Assembly: SoundParadise.Api.dll

Syntax

```
public class ConfigureSwaggerOptions : IConfigureNamedOptions<SwaggerGenOptions>,
IConfigureOptions<SwaggerGenOptions>
```

Constructors

ConfigureSwaggerOptions(IApiVersionDescriptionProvider)

Declaration

```
public ConfigureSwaggerOptions(IApiVersionDescriptionProvider provider)
```

Parameters

TYPE	NAME	DESCRIPTION
IApiVersionDescriptionProvider	provider	

Methods

Configure(SwaggerGenOptions)

Configure each API discovered for Swagger Documentation

Declaration

```
public void Configure(SwaggerGenOptions options)
```

Parameters

TYPE	NAME	DESCRIPTION
SwaggerGenOptions	options	

Configure(string?, SwaggerGenOptions)

Configure Swagger Options. Inherited from the Interface

Declaration

```
public void Configure(string? name, SwaggerGenOptions options)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	
SwaggerGenOptions	options	

Implements

IConfigureNamedOptions<TOptions>
IConfigureOptions<TOptions>

```
public void Configure(string? name, SwaggerGenOptions options)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	

Namespace SoundParadise.Api.Constants

Classes

[ApiRoutes](#)

[ApiRoutes.Address](#)

[ApiRoutes.Card](#)

[ApiRoutes.Cart](#)

[ApiRoutes.Category](#)

[ApiRoutes.Order](#)

[ApiRoutes.Product](#)

[ApiRoutes.User](#)

[ApiRoutes.Wishlist](#)

[ApiVersions](#)

[AuthPolicies](#)

Authorization policies.

[NLogVariables](#)

Class ApiRoutes

Inheritance

object

ApiRoutes

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes
```

Fields

Delete

Declaration

```
public const string Delete = "delete"
```

Field Value

TYPE	DESCRIPTION
string	

GetAll

Declaration

```
public const string GetAll = "get-all"
```

Field Value

TYPE	DESCRIPTION
string	

GetById

Declaration

```
public const string GetById = "get-by-id"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Address

Inheritance

[object](#)

ApiRoutes.Address

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: [SoundParadise.Api.Constants](#)

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Address
```

Fields

AddAddress

Declaration

```
public const string AddAddress = "add-address"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

Base

Declaration

```
public const string Base = "addresses"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

DeleteAddress

Declaration

```
public const string DeleteAddress = "delete-address"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

GetAddress

Declaration

```
public const string GetAddress = "get-address"
```

Field Value

TYPE	DESCRIPTION
string	

GetAddressByOption

Declaration

```
public const string GetAddressByOption = "get-address-by-option"
```

Field Value

TYPE	DESCRIPTION
string	

UpdateAddress

Declaration

```
public const string UpdateAddress = "update-address"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Card

Inheritance

[object](#)

ApiRoutes.Card

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: [SoundParadise.Api.Constants](#)

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Card
```

Fields

AddCard

Declaration

```
public const string AddCard = "add-card"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

Base

Declaration

```
public const string Base = "cards"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

DeleteCard

Declaration

```
public const string DeleteCard = "delete-card"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

GetCard

```
public const string GetCard = "get-card"
```

Field Value

TYPE	DESCRIPTION
string	

UpdateCard

Declaration

```
public const string UpdateCard = "update-card"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Cart

Inheritance

[object](#)

ApiRoutes.Cart

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: [SoundParadise.Api.Constants](#)

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Cart
```

Fields

AddToCart

Declaration

```
public const string AddToCart = "add-to-cart"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

AddToCartAuth

Declaration

```
public const string AddToCartAuth = "auth/add-to-cart"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

Base

Declaration

```
public const string Base = "carts"
```

Field Value

TYPE	DESCRIPTION
[string](#)	

GetCart

Declaration

```
public const string GetCart = "get-cart"
```

Field Value

TYPE	DESCRIPTION
string	

GetCartAuth

Declaration

```
public const string GetCartAuth = "auth/get-cart"
```

Field Value

TYPE	DESCRIPTION
string	

RemoveFromCart

Declaration

```
public const string RemoveFromCart = "remove-from-cart"
```

Field Value

TYPE	DESCRIPTION
string	

RemoveFromCartAuth

Declaration

```
public const string RemoveFromCartAuth = "auth/remove-from-cart"
```

Field Value

TYPE	DESCRIPTION
string	

UpdateCartItemQuantity

Declaration

```
public const string UpdateCartItemQuantity = "update-cart-item-quantity"
```

Field Value

TYPE	DESCRIPTION
string	

UpdateCartItemQuantityAuth

Declaration

```
public const string UpdateCartItemQuantityAuth = "auth/update-cart-item-quantity"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Category

Inheritance

object

ApiRoutes.Category

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Category
```

Fields

Base

Declaration

```
public const string Base = "categories"
```

Field Value

TYPE	DESCRIPTION
string	

GetCategoryName

Declaration

```
public const string GetCategoryName = "get-category-name"
```

Field Value

TYPE	DESCRIPTION
string	

GetSubcategoriesOfCategory

Declaration

```
public const string GetSubcategoriesOfCategory = "get-subcategories-of-category"
```

Field Value

TYPE	DESCRIPTION
string	

GetSubcategoryName

Declaration

```
public const string GetSubcategoryName = "get-subcategory-name"
```

Field Value

TYPE	DESCRIPTION
string	

Declaration

```
public const string GetSubcategoryName = "get-subcategory-name"
```

Field Value

TYPE	DESCRIPTION

Class ApiRoutes.Order

Inheritance

object

ApiRoutes.Order

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Order
```

Fields

Base

Declaration

```
public const string Base = "orders"
```

Field Value

TYPE	DESCRIPTION
string	

ConfirmOrder

Declaration

```
public const string ConfirmOrder = "confirm-order"
```

Field Value

TYPE	DESCRIPTION
string	

GetOrder

Declaration

```
public const string GetOrder = "get-order"
```

Field Value

TYPE	DESCRIPTION
string	

PostOrder

Declaration

```
public const string PostOrder = "post-order"
```

Field Value

TYPE	DESCRIPTION
string	

PostOrderAuth

Declaration

```
public const string PostOrderAuth = "auth/post-order"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Product

Inheritance

object

ApiRoutes.Product

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Product
```

Fields

Base

Declaration

```
public const string Base = "products"
```

Field Value

TYPE	DESCRIPTION
string	

DeleteImage

Declaration

```
public const string DeleteImage = "delete-image"
```

Field Value

TYPE	DESCRIPTION
string	

GetByCategoryId

Declaration

```
public const string GetByCategoryId = "get-by-category-id"
```

Field Value

TYPE	DESCRIPTION
string	

GetByCategoryName

Declaration

```
public const string GetByCategoryName = "get-by-category-name"
```

Field Value

TYPE	DESCRIPTION
string	

GetBySlug

Declaration

```
public const string GetBySlug = "get-by-slug"
```

Field Value

TYPE	DESCRIPTION
string	

GetBySubcategoryId

Declaration

```
public const string GetBySubcategoryId = "get-by-subcategory-id"
```

Field Value

TYPE	DESCRIPTION
string	

GetBySubcategoryName

Declaration

```
public const string GetBySubcategoryName = "get-by-subcategory-name"
```

Field Value

TYPE	DESCRIPTION
string	

GetPopular

Declaration

```
public const string GetPopular = "get-popular"
```

Field Value

TYPE	DESCRIPTION
string	

GetRecentBuys

Declaration

```
public const string GetRecentBuys = "get-recent-buys"
```

Field Value

TYPE	DESCRIPTION
string	

GetSpecialOffers

Declaration

```
public const string GetSpecialOffers = "get-special-offers"
```

Field Value

TYPE	DESCRIPTION
string	

SearchProducts

Declaration

```
public const string SearchProducts = "search-products"
```

Field Value

TYPE	DESCRIPTION
string	

SearchSuggestions

Declaration

```
public const string SearchSuggestions = "search-suggestions"
```

Field Value

TYPE	DESCRIPTION
string	

UploadImage

Declaration

```
public const string UploadImage = "upload-image"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.User

Inheritance

object

ApiRoutes.User

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.User
```

Fields

Base

Declaration

```
public const string Base = "users"
```

Field Value

TYPE	DESCRIPTION
string	

ConfirmUser

Declaration

```
public const string ConfirmUser = "confirm-user"
```

Field Value

TYPE	DESCRIPTION
string	

DeleteUser

Declaration

```
public const string DeleteUser = "delete-user"
```

Field Value

TYPE	DESCRIPTION
string	

GetCurrentUser

Declaration

```
public const string GetCurrentUser = "get-current-user"
```

Field Value

TYPE	DESCRIPTION
string	

LoginUser

Declaration

```
public const string LoginUser = "login-user"
```

Field Value

TYPE	DESCRIPTION
string	

LogoutUser

Declaration

```
public const string LogoutUser = "logout-user"
```

Field Value

TYPE	DESCRIPTION
string	

RegisterUser

Declaration

```
public const string RegisterUser = "register-user"
```

Field Value

TYPE	DESCRIPTION
string	

SecureEndpoint

Declaration

```
public const string SecureEndpoint = "secure-endpoint"
```

Field Value

TYPE	DESCRIPTION
string	

UpdateUser

Declaration

```
public const string UpdateUser = "update-user"
```

Field Value

TYPE	DESCRIPTION
string	

UploadAvatar

Declaration

```
public const string UploadAvatar = "upload-avatar"
```

Field Value

TYPE	DESCRIPTION
string	

Class ApiRoutes.Wishlist

Inheritance

object

ApiRoutes.Wishlist

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiRoutes.Wishlist
```

Fields

AddToWishlist

Declaration

```
public const string AddToWishlist = "add-product-to-wishlist"
```

Field Value

TYPE	DESCRIPTION
string	

Base

Declaration

```
public const string Base = "wishlists"
```

Field Value

TYPE	DESCRIPTION
string	

GetWishlist

Declaration

```
public const string GetWishlist = "get-wishlist"
```

Field Value

TYPE	DESCRIPTION
string	

RemoveFromWishlist

Declaration

```
public const string RemoveFromWishlist = "remove-product-from-wishlist"
```

Field Value

TYPE	DESCRIPTION
string	

Declaration

```
public const string RemoveFromWishlist = "remove-product-from-wishlist"
```

Field Value

TYPE	DESCRIPTION

Class ApiVersions

Inheritance

object

ApiVersions

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class ApiVersions
```

Fields

ApiHeader

Header name for API versioning.

Declaration

```
public const string ApiHeader = "x-api-version"
```

Field Value

TYPE	DESCRIPTION
string	

V1

Current API version.

Declaration

```
public const string V1 = "1.0"
```

Field Value

TYPE	DESCRIPTION
string	

V2

Current API version.

Declaration

```
public const string V2 = "2.0"
```

Field Value

TYPE	DESCRIPTION
string	

Class AuthPolicies

Authorization policies.

Inheritance

object

AuthPolicies

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class AuthPolicies
```

Fields

AdminOrCustomerPolicy

Declaration

```
public const string AdminOrCustomerPolicy = "AdminOrCustomerPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

AdminOrSellerPolicy

Declaration

```
public const string AdminOrSellerPolicy = "AdminOrSellerPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

AdminPolicy

Admin policy. This is the most privileged policy.

Declaration

```
public const string AdminPolicy = "AdminPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

CustomerPolicy

Customer policy. This is the least privileged policy.

Declaration

```
public const string CustomerPolicy = "CustomerPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

SellerOrCustomerPolicy

Declaration

```
public const string SellerOrCustomerPolicy = "SellerOrCustomerPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

SellerPolicy

Seller policy. This is the second most privileged policy.

Declaration

```
public const string SellerPolicy = "SellerPolicy"
```

Field Value

TYPE	DESCRIPTION
string	

Class NLogVariables

Inheritance

object

NLogVariables

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Constants

Assembly: SoundParadise.Api.dll

Syntax

```
public static class NLogVariables
```

Fields

DatabaseConnectionString

Declaration

```
public const string DatabaseConnectionString = "databaseConnectionString"
```

Field Value

TYPE	DESCRIPTION
string	

DatabaseName

Declaration

```
public const string DatabaseName = "databaseName"
```

Field Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Controllers.Api

Classes

BaseApiController

Base API controller

BaseRouteAttribute

Base Route Attribute

RequestResult

Request result

Class BaseApiController

Base API controller

Inheritance

object

ControllerBase

BaseApiController

AddressController

CardController

CartController

CategoryController

OrderController

ProductController

UserController

WishlistController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiController]
public abstract class BaseApiController : ControllerBase
```

Class BaseRouteAttribute

Base Route Attribute

Inheritance

object

Attribute

RouteAttribute

BaseRouteAttribute

Implements

IRouteTemplateProvider

Inherited Members

RouteAttribute.Template

RouteAttribute.Order

RouteAttribute.Name

Attribute.Equals(object)

Attribute.GetCustomAttribute(Assembly, Type)

Attribute.GetCustomAttribute(Assembly, Type, bool)

Attribute.GetCustomAttribute(MemberInfo, Type)

Attribute.GetCustomAttribute(MemberInfo, Type, bool)

Attribute.GetCustomAttribute(Module, Type)

Attribute.GetCustomAttribute(Module, Type, bool)

Attribute.GetCustomAttribute(ParameterInfo, Type)

Attribute.GetCustomAttribute(ParameterInfo, Type, bool)

Attribute.GetCustomAttributes(Assembly)

Attribute.GetCustomAttributes(Assembly, bool)

Attribute.GetCustomAttributes(Assembly, Type)

Attribute.GetCustomAttributes(Assembly, Type, bool)

Attribute.GetCustomAttributes(MemberInfo)

Attribute.GetCustomAttributes(MemberInfo, bool)

Attribute.GetCustomAttributes(MemberInfo, Type)

Attribute.GetCustomAttributes(MemberInfo, Type, bool)

Attribute.GetCustomAttributes(Module)

Attribute.GetCustomAttributes(Module, bool)

Attribute.GetCustomAttributes(Module, Type)

Attribute.GetCustomAttributes(Module, Type, bool)

Attribute.GetCustomAttributes(ParameterInfo)

Attribute.GetCustomAttributes(ParameterInfo, bool)

Attribute.GetCustomAttributes(ParameterInfo, Type)

Attribute.GetCustomAttributes(ParameterInfo, Type, bool)

Attribute.GetHashCode()

Attribute.IsDefaultAttribute()

Attribute.IsDefined(Assembly, Type)

Attribute.IsDefined(Assembly, Type, bool)

Attribute.IsDefined(MemberInfo, Type)

Attribute.IsDefined(MemberInfo, Type, bool)

Attribute.IsDefined(Module, Type)

Attribute.IsDefined(Module, Type, bool)

Attribute.IsDefined(ParameterInfo, Type)

Attribute.IsDefined(ParameterInfo, Type, bool)

Attribute.Match(object)

Attribute.TypeId

object.Equals(object, object)

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Syntax

```
[AttributeUsage(AttributeTargets.Class)]
public class BaseRouteAttribute : RouteAttribute, IRouteTemplateProvider
```

Constructors

BaseRouteAttribute(string)

Base Route Attribute constructor

Declaration

```
public BaseRouteAttribute(string template)
```

Parameters

TYPE	NAME	DESCRIPTION
string	template	

Implements

IRouteTemplateProvider

Class RequestResult

Request result

Inheritance

object

RequestResult

[CreateOrderResult](#)

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Api.Controllers.Api

Assembly: SoundParadise.Api.dll

Syntax

```
public class RequestResult
```

Constructors

RequestResult(bool, string, HttpStatusCode)

Declaration

```
protected RequestResult(bool isSuccess, string message, HttpStatusCode httpStatus)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	isSuccess	
string	message	
HttpStatusCode	httpStatus	

Properties

HttpStatus

Declaration

```
public HttpStatusCode HttpStatus { get; set; }
```

Property Value

TYPE	DESCRIPTION
HttpStatusCode	

IsSuccess

Declaration

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

Property Value

TYPE	DESCRIPTION
bool	

Message

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Methods

Error(string, HttpStatusCode)

Declaration

```
public static RequestResult Error(string message = "", HttpStatusCode httpStatus = HttpStatusCode.BadRequest)
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	
HttpStatusCode	httpStatus	

Returns

TYPE	DESCRIPTION
RequestResult	

Success(string, HttpStatusCode)

Declaration

```
public static RequestResult Success(string message = "", HttpStatusCode httpStatus = HttpStatusCode.OK)
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	
HttpStatusCode	httpStatus	

Returns

TYPE	DESCRIPTION
RequestResult	

Namespace SoundParadise.Api.Controllers.Api.v1

Classes

AddressController

Address API v1 controller

CardController

Card API v1 controller

CartController

Cart API v1 controller

CategoryController

Category API v1 controller

OrderController

Order API v1 controller

ProductController

Product API v1 controller

UserController

User API v1 controller

WishlistController

Wishlist API v1 controller

Class AddressController

Address API v1 controller

Inheritance

object

ControllerBase

BaseApiController

AddressController

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.AddressToRoute(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(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.AcceptedAtAction(string, 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()

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("addresses")]
public class AddressController : BaseApiController
```

Constructors

AddressController(IAddressCrud)

Constructor for AddressController v1.

Declaration

```
public AddressController(IAddressCrud addressCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
IAddressCrud	addressCrud	

Methods

AddAddress(AddressDto)

This method is used to create an order for an unauthenticated user.

Declaration

```
[Authorize]
[HttpPost("add-address")]
[SwaggerResponse(200, "Returns the message that the address was successfully added", typeof(string))]
public IActionResult AddAddress(AddressDto address)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressDto	address	

Returns

TYPE	DESCRIPTION
IActionResult	Successfully added address

DeleteAddress(Guid)

Declaration

```
[Authorize]
[HttpDelete("delete-address")]
[SwaggerResponse(200, "Returns the message that the address was successfully deleted", typeof(string))]
public IActionResult DeleteAddress(Guid address)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	address	

Returns

TYPE	DESCRIPTION
IActionResult	

GetAddressByOption(DeliveryOptionEnum)

This method is used to get addresses of the authenticated user by delivery option.

Declaration

```
[Authorize]
[HttpGet("get-address-by-option")]
[SwaggerResponse(200, "Returns the list of addresses of the authenticated user by delivery option",
typeof(List<AddressDto>))]
public IActionResult GetAddressByOption(DeliveryOptionEnum option)
```

Parameters

TYPE	NAME	DESCRIPTION
DeliveryOptionEnum	option	

Returns

TYPE	DESCRIPTION
IActionResult	List of AddressDtos

GetAddresses()

This method is used to get addresses of the authenticated user.

Declaration

```
[Authorize]
[HttpGet("get-all")]
[SwaggerResponse(200, "Returns the list of addresses of the authenticated user", typeof(List<AddressDto>))]
public IActionResult GetAddresses()
```

Returns

TYPE	DESCRIPTION
IActionResult	List of all Addresses of User

Class CardController

Card API v1 controller

Inheritance

object

ControllerBase

BaseApiController

CardController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("cards")]
public class CardController : BaseApiController
```

Constructors

CardController(ICardCrud, CardDtoValidator)

Constructor for CardController v1.

Declaration

```
public CardController(ICardCrud cardCrud, CardDtoValidator cardValidator)
```

Parameters

TYPE	NAME	DESCRIPTION
ICardCrud	cardCrud	
CardDtoValidator	cardValidator	

Methods

AddCard(CardDto)

Adds a new card to the user's account.

Declaration

```
[Authorize]
[HttpPost("add-card")]
[SwaggerResponse(200, "The card is successfully added.", typeof(string))]
[SwaggerResponse(400, "Invalid card data. Returns an error message.", typeof(string))]
public IActionResult AddCard(CardDto cardDto)
```

Parameters

TYPE	NAME	DESCRIPTION
CardDto	cardDto	

Returns

TYPE	DESCRIPTION
IActionResult	Invalid card data. Returns an error message.

DeleteCard(Guid)

Deletes the card from the user's account.

Declaration

```
[Authorize]
[HttpDelete("delete-card")]
[SwaggerResponse(200, "The card is successfully deleted.", typeof(string))]
[SwaggerResponse(400, "Invalid card data. Returns an error message.", typeof(string))]
public IActionResult DeleteCard(Guid cardId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	

Returns

TYPE	DESCRIPTION
IActionResult	Card deleted successfully.

GetCard(Guid)

Returns the card information in the form of a CardDto object.

Declaration

```
[Authorize]
[HttpGet("get-card")]
[SwaggerResponse(200, "Returns the card information in the form of a CardDto object.", typeof(CardDto))]
public IActionResult GetCard(Guid cardId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	

Returns

TYPE	DESCRIPTION
IActionResult	Returns CardDto.

GetCards()

Returns all cards for related user in the form of a CardDto object.

Declaration

```
[Authorize]
[HttpGet("get-all")]
[SwaggerResponse(200, "Returns all cards for related user in the form of a CardDto object.", typeof(CardDto))]
public IActionResult GetCards()
```

Returns

TYPE	DESCRIPTION
IActionResult	Returns list of dtos.

UpdateCard(CardDto)

Updates the card information in the user's account.

Declaration

```
[Authorize]
[HttpPut("update-card")]
[SwaggerResponse(200, "The card is successfully updated.", typeof(string))]
[SwaggerResponse(400, "Invalid card data. Returns an error message.", typeof(string))]
public IActionResult UpdateCard(CardDto cardDto)
```

Parameters

TYPE	NAME	DESCRIPTION
CardDto	cardDto	

Returns

TYPE	DESCRIPTION
IActionResult	Card updated.

Class CartController

Cart API v1 controller

Inheritance

object

ControllerBase

BaseApiController

CartController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("carts")]
public class CartController : BaseApiController
```

Constructors

CartController(ICartCrud, IProductCrud, CartItemDtoValidator, IUserCrud)

Constructor for CartController v1.

Declaration

```
public CartController(ICartCrud cartCrud, IProductCrud productCrud, CartItemDtoValidator cartItemDtoValidator,
IUserCrud userCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
ICartCrud	cartCrud	
IProductCrud	productCrud	
CartItemDtoValidator	cartItemDtoValidator	
IUserCrud	userCrud	

Methods

AddToCart(Guid)

Adds the product to the cart.

Declaration

```
[AllowAnonymous]
[HttpPost("add-to-cart")]
[SwaggerResponse(201, "The product is successfully added to cart.", typeof(CartItemDto))]
public IActionResult AddToCart(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	Product added to cart successfully.

AddToCartAuth(Guid)

Adds a product to the cart by authenticated user.

Declaration

```
[Authorize]
[HttpPost("auth/add-to-cart")]
[SwaggerResponse(201, "The product is successfully added to cart of authenticated user.",
typeof(CartItemDto))]
public IActionResult AddToCartAuth(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
IActionResult	Product added successfully.

GetAllCarts()

Gets all carts.

Declaration

```
[Authorize(Policy = "AdminPolicy")]
[HttpGet("get-all")]
[SwaggerResponse(200, "The carts are successfully retrieved.", typeof(CartModel))]
public IActionResult GetAllCarts()
```

Returns

TYPE	DESCRIPTION
IActionResult	Returns all carts.

GetCart()

Gets the cart by id.

Declaration

```
[HttpGet("get-cart")]
[SwaggerResponse(200, "The cart is successfully retrieved for user.", typeof(CartItemDtoProduct))]
public IActionResult GetCart()
```

Returns

TYPE	DESCRIPTION
IActionResult	Return CartItemDtoProduct.

GetCartAuth()

Gets the cart by user id.

Declaration

```
[Authorize]
[HttpGet("auth/get-cart")]
[SwaggerResponse(200, "The cart is successfully retrieved for authenticated user.", typeof(CartModel))]
public IActionResult GetCartAuth()
```

Returns

TYPE	DESCRIPTION
IActionResult	Returns CartModel.

RemoveFromCart(Guid)

Removes a cart item from the cart.

Declaration

```
[HttpDelete("remove-from-cart")]
[SwaggerResponse(200, "Item removed from cart successfully by user.", null)]
public IActionResult RemoveFromCart(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	Item removed from cart successfully.

RemoveFromCartAuth(Guid)

Removes a cart item from the cart.

Declaration

```
[Authorize]
[HttpDelete("auth/remove-from-cart")]
[SwaggerResponse(200, "Item removed from cart successfully by authenticated user.", null)]
public IActionResult RemoveFromCartAuth(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	Item removed from cart successfully.

UpdateCartItemQuantity(Guid, short)

Updates the quantity of a cart item.

Declaration

```
[HttpPut("update-cart-item-quantity")]
[SwaggerResponse(200, "Item quantity updated successfully.", null)]
public IActionResult UpdateCartItemQuantity(Guid productId, short quantity = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

TYPE	NAME	DESCRIPTION
short	quantity	

Returns

TYPE	DESCRIPTION
IActionResult	Cart item quantity updated successfully.

UpdateCartItemQuantityAuth(Guid, short)

Updates the quantity of a cart item by authenticated user.

Declaration

```
[Authorize]
[HttpPut("auth/update-cart-item-quantity")]
[SwaggerResponse(200, "Item quantity updated successfully by authenticated user.", null)]
public IActionResult UpdateCartItemQuantityAuth(Guid cartItemId, short quantity = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartItemId	
short	quantity	

Returns

TYPE	DESCRIPTION
IActionResult	Cart item quantity updated successfully.

Class CategoryController

Category API v1 controller

Inheritance

object

ControllerBase

BaseApiController

CategoryController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("categories")]
public class CategoryController : BaseApiController
```

Constructors

CategoryController(ICategoryCrud)

Constructor for CategoryController v1.

Declaration

```
public CategoryController(ICategoryCrud categoryCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
ICategoryCrud	categoryCrud	

Methods

GetAllCategories()

Gets the list of all categories.

Declaration

```
[HttpGet("get-all")]
[SwaggerResponse(200, "Returns the list of all categories.", typeof(List<CategoryDto>))]
public IActionResult GetAllCategories()
```

Returns

TYPE	DESCRIPTION
IActionResult	List of CategoryModel.

GetCategoryName(Guid)

Return name of the category by ID

Declaration

```
[HttpGet("get-category-name")]
[SwaggerResponse(200, "Returns the name of the category", typeof(string))]
public IActionResult GetCategoryName(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
IActionResult	Name of the Ccategory

GetSubcategoriesOfCategory(Guid)

Declaration

```
[HttpGet("get-subcategories-of-category")]
[SwaggerResponse(200, "Returns the list of all subcategories of the category.", typeof(List<SubcategoryDto>))]
public IActionResult GetSubcategoriesOfCategory(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
IActionResult	

GetSubcategoryName(Guid)

Returns name of the subcategory by ID

Declaration

```
[HttpGet("get-subcategory-name")]
[SwaggerResponse(200, "Returns the name of the subcategory", typeof(string))]
public IActionResult GetSubcategoryName(Guid subcategoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	subcategoryId	

Returns

TYPE	DESCRIPTION
IActionResult	The name of the Subcategory

Class OrderController

Order API v1 controller

Inheritance

object

ControllerBase

BaseApiController

OrderController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("orders")]
public class OrderController : BaseApiController
```

Constructors

OrderController(IOrderCrud)

Constructor for OrderController v1.

Declaration

```
public OrderController(IOrderCrud orderCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
IOrderCrud	orderCrud	

Methods

ConfirmOrder(dynamic)

Confirming order

Declaration

```
[AllowAnonymous]
[HttpPost("confirm-order")]
[SwaggerResponse(200, "Callback endpoint for confirming order", typeof(string))]
public IActionResult ConfirmOrder(dynamic data)
```

Parameters

TYPE	NAME	DESCRIPTION
dynamic	data	

Returns

TYPE	DESCRIPTION
IActionResult	Order was confirmed

GetOrder(Guid)

Gets the order by ID

Declaration

```
[Authorize]
[HttpPost("get-order")]
[SwaggerResponse(200, "Get order by ID", typeof(OrderModel))]
public IActionResult GetOrder(Guid orderId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	

Returns

TYPE	DESCRIPTION
IActionResult	Returns the OrderModel

GetOrders()

This method is used to get orders of the authenticated user.

Declaration

```
[Authorize]
[HttpGet("get-all")]
[SwaggerResponse(200, "Returns the list of orders of the authenticated user", typeof(List<OrderModel>))]
public IActionResult GetOrders()
```

Returns

TYPE	DESCRIPTION
IActionResult	List of all orders of the authenticated user.

PostOrder(OrderModel)

This method is used to create an order for an unauthenticated user.

Declaration

```
[AllowAnonymous]
[HttpPost("post-order")]
[SwaggerResponse(200, "Order wad created for the user.", typeof(string))]
public IActionResult PostOrder(OrderModel order)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderModel	order	

Returns

TYPE	DESCRIPTION
IActionResult	The success message response.

PostOrderAuth(OrderDto)

This method is used to create an order for an authenticated user.

Declaration

```
[Authorize]
[HttpPost("auth/post-order")]
[SwaggerResponse(200, "Order was created for the authenticated user.", typeof(string))]
public IActionResult PostOrderAuth(OrderDto order)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderDto	order	

Returns

TYPE	DESCRIPTION
IActionResult	The success message response.

Class ProductController

Product API v1 controller

Inheritance

object

ControllerBase

BaseApiController

ProductController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("products")]
public class ProductController : BaseApiController
```

Constructors

ProductController(IProductCrud, IUserCrud, ImageService)

Constructor for ProductController v1.

Declaration

```
public ProductController(IProductCrud productCrud, IUserCrud userCrud, ImageService imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
IProductCrud	productCrud	
IUserCrud	userCrud	
ImageService	imageService	

Methods

DeleteImageFromProduct(Guid, Guid)

Deletes an image from a specific product.

Declaration

```
[Authorize]
[HttpDelete("delete-image")]
[SwaggerResponse(200, "The image is successfully deleted.", typeof(string))]
[SwaggerResponse(401, "Unauthorized. The user is not the owner of the product.", typeof(string))]
[SwaggerResponse(404, "Not found. The user or product is not found.", typeof(string))]
public IActionResult DeleteImageFromProduct(Guid productId, Guid imageId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	The ID of the product to delete the image from.
Guid	imageId	The ID of the image to delete.

Returns

TYPE	DESCRIPTION
IActionResult	A response indicating the status of the operation.

GetPopular(int, int)

Declaration

```
[AllowAnonymous]
[HttpGet("get-popular")]
[SwaggerResponse(200, "Returns the list of products with the specified subcategory ID.",
typeof(List<ProductDto>))]
public IActionResult GetPopular(int page = 1, int limit = 3)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	

GetProductById(Guid)

Gets the product by ID.

Declaration

```
[AllowAnonymous]
[HttpGet("get-by-id")]
[SwaggerResponse(200, "Returns the product with the specified ID.", typeof(ProductModel))]
public IActionResult GetProductById(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	The ProductModel.

GetProductBySlug(string)

Gets the product by slug.

Declaration

```
[HttpGet("get-by-slug")]
[SwaggerResponse(200, "Returns the product with the specified slug.", typeof(ProductModel))]
public IActionResult GetProductBySlug(string slug)
```

Parameters

TYPE	NAME	DESCRIPTION
string	slug	

Returns

TYPE	DESCRIPTION
IActionResult	ProductModel

GetProductsByCategoryId(SortOptionsEnum, Guid, int, int)

Gets the list of products by category ID.

Declaration

```
[AllowAnonymous]
[HttpGet("get-by-category-id")]
[SwaggerResponse(200, "Returns the list of products with the specified category ID.",
typeof(List<ProductModel>))]
public IActionResult GetProductsByCategoryId(SortOptionsEnum sort, Guid categoryId, int page = 1, int limit =
20)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
Guid	categoryId	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	The list of products.

GetProductsByCategoryName(SortOptionsEnum, string, int, int)

Gets the list of products by category name.

Declaration

```
[AllowAnonymous]
[HttpGet("get-by-category-name")]
[SwaggerResponse(200, "Returns the list of products with the specified category name.",
typeof(List<ProductModel>))]
public IActionResult GetProductsByCategoryName(SortOptionsEnum sort, string categoryName = "category-name",
int page = 1, int limit = 20)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	categoryName	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	The list of products.

GetProductsBySubcategoryId(SortOptionsEnum, Guid, int, int)

Gets the list of products by subcategory ID.

Declaration

```
[HttpGet("get-by-subcategory-id")]
[SwaggerResponse(200, "Returns the list of products with the specified subcategory ID.",
typeof(List<ProductModel>))]
public IActionResult GetProductsBySubcategoryId(SortOptionsEnum sort, Guid subcategoryId, int page = 1, int
limit = 20)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
Guid	subcategoryId	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	The list of products.

GetProductsBySubcategoryName(SortOptionsEnum, string, int, int)

Gets the list of products by subcategory name.

Declaration

```
[HttpGet("get-by-subcategory-name")]
[SwaggerResponse(200, "Returns the list of products with the specified subcategory name.",
typeof(List<ProductModel>))]
public IActionResult GetProductsBySubcategoryName(SortOptionsEnum sort, string subcategoryName = "subcategory-
name", int page = 1, int limit = 20)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	subcategoryName	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	The list of products.

GetRecentBuys(int, int)

Gets the list of products by subcategory ID.

Declaration

```
[AllowAnonymous]
[HttpGet("get-recent-buys")]
[SwaggerResponse(200, "Returns the list of products with the specified subcategory ID.",
typeof(List<ProductDto>))]
public IActionResult GetRecentBuys(int page = 1, int limit = 3)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	List of ProductDtos

GetSpecialOffers(SortOptionsEnum, int, int)

Gets the list of products by subcategory ID.

Declaration

```
[AllowAnonymous]
[HttpGet("get-special-offers")]
[SwaggerResponse(200, "Returns the list of products with the specified subcategory ID.",
typeof(List<ProductDto>))]
public IActionResult GetSpecialOffers(SortOptionsEnum sort, int page = 1, int limit = 3)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	List of ProductDtos

SearchProducts(SortOptionsEnum, string, int, int)

Searches for products by name.

Declaration

```
[AllowAnonymous]
[HttpGet("search-products")]
[SwaggerResponse(200, "Returns the list of products matching the search query.", typeof(List<ProductDto>))]
public IActionResult SearchProducts(SortOptionsEnum sort, string query = "query-example", int page = 1, int limit = 20)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	query	
int	page	
int	limit	

Returns

TYPE	DESCRIPTION
IActionResult	List of ProductDtos

SearchSuggestions(string, int)

Searches for products by name.

Declaration

```
[HttpGet("search-suggestions")]
[SwaggerResponse(200, "Returns the list of products matching the search query.", typeof(List<SuggestionProductDto>))]
public IActionResult SearchSuggestions(string query = "query-example", int limit = 4)
```

Parameters

TYPE	NAME	DESCRIPTION
string	query	The search query.
int	limit	The maximum number of results to retrieve (default is 4).

Returns

TYPE	DESCRIPTION
IActionResult	The list of products matching the search query.

UploadImage(Guid, IFormFile)

Uploads an image for a specific product.

Declaration

```
[Authorize]
[HttpPost("upload-image")]
[SwaggerResponse(200, "Returns the URL of the uploaded image.", typeof(string))]
[SwaggerResponse(400, "Invalid file or an error occurred while uploading.", typeof(string))]
public Task<IActionResult> UploadImage(Guid productId, IFormFile file)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	The ID of the product to upload the image for.
IFormFile	file	The image file to upload.

Returns

TYPE	DESCRIPTION
Task<IActionResult>	The URL of the uploaded image.

Class UserController

User API v1 controller

Inheritance

object

ControllerBase

BaseApiController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("users")]
public class UserController : BaseApiController
```

Constructors

UserController(IUserCrud, UserAuthenticationDtoValidator, UserCreateDtoValidator, IOptions<UrlsOptions>, ImageService)

Constructor for UserController v1.

Declaration

```
public UserController(IUserCrud userCrud, UserAuthenticationDtoValidator userAuthenticationDtoValidator,
UserCreateDtoValidator userCreateDtoValidator, IOptions<UrlsOptions> options, ImageService imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
IUserCrud	userCrud	The UserCrud instance.
UserAuthenticationDtoValidator	userAuthenticationDtoValidator	The UserAuthenticationDtoValidator instance.
UserCreateDtoValidator	userCreateDtoValidator	The UserCreateDtoValidator instance.
IOptions<UrlsOptions>	options	
ImageService	imageService	

Methods

ConfirmUser(Guid, string)

Confirms a user's registration using the provided user ID and token.

Declaration

```
[HttpGet("confirm-user")]
[SwaggerResponse(200, "The user confirmation is successful. Returns an HTML page with a redirect to the login page.", typeof(string))]
[SwaggerResponse(400, "Invalid user ID or token. Returns a bad request with an error message.", typeof(string))]
public IActionResult ConfirmUser(Guid userId, string token)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	The ID of the user to confirm.
string	token	The confirmation token.

Returns

TYPE	DESCRIPTION
IActionResult	The user confirmation is successful. Returns an HTML page with a redirect to the login page.

DeleteUser(Guid)

Deletes a user with the specified user ID.

Declaration

```
[HttpDelete("delete")]
[SwaggerResponse(200, "The user is successfully deleted.", typeof(string))]
[SwaggerResponse(400, "Invalid user ID. Returns an error message.", typeof(string))]
public IActionResult DeleteUser(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	The ID of the user to delete.

Returns

TYPE	DESCRIPTION
IActionResult	The user is successfully deleted.

GetCurrentUser()

Gets the currently authenticated user.

Declaration

```
[Authorize]
[HttpPost("get-current-user")]
[SwaggerResponse(200, "Getting the current user", typeof(UserDto))]
public IActionResult GetCurrentUser()
```

Returns

TYPE	DESCRIPTION
IActionResult	UserDto

Login(UserAuthenticationDto)

Authenticates a user by validating their credentials.

Declaration

```
[HttpPost("login-user")]
[SwaggerResponse(200, "The user is successfully authenticated. Returns an authentication token.",
typeof(string))]
[SwaggerResponse(400, "Invalid user credentials. Returns an error message.", typeof(string))]
public IActionResult Login(UserAuthenticationDto authenticationDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserAuthenticationDto	authenticationDto	The user authentication data.

Returns

TYPE	DESCRIPTION
IActionResult	The user is successfully authenticated. Returns an authentication token.

Logout()

Logs out the currently authenticated user.

Declaration

```
[Authorize]
[HttpPost("logout-user")]
[SwaggerResponse(200, "The user is successfully logged out.", typeof(string))]
public IActionResult Logout()
```

Returns

TYPE	DESCRIPTION
IActionResult	The user is successfully logged out.

RegisterUser(UserCreateDto)

Registers a new user with the provided details.

Declaration

```
[HttpPost("register-user")]
[SwaggerResponse(200, "The user is successfully registered. Returns the registered user object.",
typeof(UserModel))]
[SwaggerResponse(400, "Invalid user registration data. Returns an error message with validation errors.",
typeof(string))]
public IActionResult RegisterUser(UserCreateDto userCreateDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userCreateDto	The user registration data.

Returns

TYPE	DESCRIPTION
IActionResult	The user is successfully registered. Returns the registered user object.

SecureEndpoint()

Retrieves the details of the authenticated user.

Declaration

```
[Authorize]
[HttpGet("secure-endpoint")]
[Authorize(Policy = "SellerPolicy")]
[SwaggerResponse(200, "Returns the user information in the form of a UserDto object.", typeof(UserDto))]
public IActionResult SecureEndpoint()
```

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
IActionResult	Details of the user.

UploadAvatar(IFormFile)

Uploads a user's avatar.

Declaration

```
[Authorize]
[HttpPost("upload-avatar")]
[SwaggerResponse(200, "Uploading user's avatar", typeof(string))]
public IActionResult UploadAvatar(IFormFile file)
```

Parameters

TYPE	NAME	DESCRIPTION
IFormFile	file	

Returns

TYPE	DESCRIPTION
IActionResult	Image url

Class WishlistController

Wishlist API v1 controller

Inheritance

object

ControllerBase

BaseApiController

WishlistController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v1

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("1.0")]
[BaseRoute("wishlists")]
public class WishlistController : BaseApiController
```

Constructors

WishlistController(IWishlistCrud, IProductCrud)

Constructor for WishlistController v1

Declaration

```
public WishlistController(IWishlistCrud wishlistCrud, IProductCrud productCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
IWishlistCrud	wishlistCrud	
IProductCrud	productCrud	

Methods

AddToWishlist(Guid)

Adds the product to the wishlist.

Declaration

```
[Authorize]
[HttpPut("add-product-to-wishlist")]
[SwaggerResponse(200, "The product is successfully added to the wishlist.", typeof(string))]
[SwaggerResponse(400, "The product is already in the wishlist.", typeof(string))]
public IActionResult AddToWishlist(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	Product added to the wishlist.

GetWishlist()

Gets the wishlist of the authenticated user.

Declaration

```
[Authorize]
[HttpGet("get-wishlist")]
[SwaggerResponse(200, "The wishlist is successfully retrieved.", typeof(WishlistModel))]
public IActionResult GetWishlist()
```

Returns

TYPE	DESCRIPTION
IActionResult	Returns list of ProductDtos.

RemoveFromWishlist(Guid)

Removes the product from the wishlist.

Declaration

```
[Authorize]
[HttpPut("remove-product-from-wishlist")]
[SwaggerResponse(200, "The product is successfully removed from the wishlist.", typeof(string))]
[SwaggerResponse(400, "The product is not in the wishlist.", typeof(string))]
public IActionResult RemoveFromWishlist(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
IActionResult	Product removed from wishlist.

Namespace SoundParadise.Api.Controllers.Api.v2

Classes

UserController

User API controller

Class UserController

User API controller

Inheritance

object

ControllerBase

BaseApiController

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(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.AcceptedAtAction(string, 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()

Namespace: SoundParadise.Api.Controllers.Api.v2

Assembly: SoundParadise.Api.dll

Syntax

```
[ApiVersion("2.0")]
[BaseRoute("users")]
public class UserController : BaseApiController
```

Constructors

UserController(UserCrud, UserAuthenticationDtoValidator, UserCreateDtoValidator)

Constructor for UserController

Declaration

```
public UserController(UserCrud userCrud, UserAuthenticationDtoValidator userAuthenticationDtoValidator,
UserCreateDtoValidator userCreateDtoValidator)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCrud	userCrud	
UserAuthenticationDtoValidator	userAuthenticationDtoValidator	
UserCreateDtoValidator	userCreateDtoValidator	

Methods

RegisterUser(UserCreateDto)

Just a test method for checking if the API Versioning is working

Declaration

```
[HttpPost("register-user")]
public IActionResult RegisterUser(UserCreateDto userCreateDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userCreateDto	

Returns

TYPE	DESCRIPTION
IActionResult	Returns a string

Namespace SoundParadise.Api.Data

Classes

SoundParadiseDbContext

Data base context.

Class SoundParadiseDbContext

Data base context.

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.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Data

Assembly: SoundParadise.Api.dll

Syntax

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

Constructors

SoundParadiseDbContext(DbContextOptions<SoundParadiseDbContext>, IHostEnvironment, HashingService)

Constructor for SoundParadiseDbContext.

Declaration

```
public SoundParadiseDbContext(DbContextOptions<SoundParadiseDbContext> options, IHostEnvironment environment,
HashingService hashingService)
```

Parameters

TYPE	NAME	DESCRIPTION
DbContextOptions<SoundParadiseDbContext>	options	Option.
IHostEnvironment	environment	Environment.
HashingService	hashingService	Hashing service.

Properties

AllSalesAccumulators

Dbset all sales accumulator.

Declaration

```
public DbSet<AllSalesAccumulator> AllSalesAccumulators { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<AllSalesAccumulator>	

Cards

Dbset card model.

Declaration

```
public DbSet<CardModel> Cards { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<CardModel>	

CartItems

Dbset CartItems.

Declaration

```
public DbSet<CartItemModel> CartItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<CartItemModel>	

Carts

Dbset Carts.

Declaration

```
public DbSet<CartModel> Carts { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<CartModel>	

Categories

Dbset categories.

Declaration

```
public DbSet<CategoryModel> Categories { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<CategoryModel>	

Comments

Dbset Comments.

Declaration

```
public DbSet<CommentModel> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<CommentModel>	

DeliveryAddresses

Dbset delivery address.

Declaration

```
public DbSet<AddressModel> DeliveryAddresses { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<AddressModel>	

Images

Dbset Images.

Declaration

```
public DbSet<ImageModel> Images { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<ImageModel>	

Logs

Dbset logs.

Declaration

```
public DbSet<LogModel> Logs { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<LogModel>	

Orders

Dbset orders.

Declaration

```
public DbSet<OrderModel> Orders { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<OrderModel>	

Parameters

Dbset parameters.

Declaration

```
public DbSet<ProductParameterModel> Parameters { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<ProductParameterModel>	

Payments

Dbset payments.

Declaration

```
public DbSet<PaymentModel> Payments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<PaymentModel>	

ProductSalesAccumulators

Dbset product sales accumulator.

Declaration

```
public DbSet<ProductSalesAccumulator> ProductSalesAccumulators { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<ProductSalesAccumulator>	

Products

Dbset products..

Declaration

```
public DbSet<ProductModel> Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<ProductModel>	

Subcategories

Dbset subcategory.

Declaration

```
public DbSet<SubcategoryModel> Subcategories { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<SubcategoryModel>	

TokenJournals

Dbset token journals.

Declaration

```
public DbSet<TokenJournalModel> TokenJournals { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TokenJournalModel>	

Tokens

Dbset tokens users.

Declaration

```
public DbSet<TokenModel> Tokens { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TokenModel>	

UserAccumulators

Dbset user accumulator.

Declaration

```
public DbSet<UserAccumulator> UserAccumulators { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<UserAccumulator>	

Users

Dbset users.

Declaration

```
public DbSet<UserModel> Users { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<UserModel>	

WishlistProducts

Dbset whislistproducts.

Declaration

```
public DbSet<WishlistProductsModel> WishlistProducts { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<WishlistProductsModel>	

Wishlists

Dbset whishlist.

Declaration

```
public DbSet<WishlistModel> Wishlists { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<WishlistModel>	

Methods

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.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
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.

Overrides

DbContext.OnModelCreating(ModelBuilder)

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.

SaveChangesAsync(CancellationToken)

Saves all changes made in this context to the database.

Declaration

```
public override Task<int> SaveChangesAsync(CancellationToken cancellationToken = default)
```

Parameters

TYPE	NAME	DESCRIPTION
CancellationToken	cancellationToken	A CancellationToken to observe while waiting for the task to complete.

Returns

TYPE	DESCRIPTION
Task<int>	A task that represents the asynchronous save operation. The task result contains the number of state entries written to the database.

Overrides

DbContext.SaveChangesAsync(CancellationToken)

Remarks

This method will automatically call DetectChanges() to discover any changes to entity instances before saving to the underlying database. This can be disabled via AutoDetectChangesEnabled.

Entity Framework Core does not support multiple parallel operations being run on the same DbContext instance. This includes both parallel execution of async queries and any explicit concurrent use from multiple threads. Therefore, always await async calls immediately, or use separate DbContext instances for operations that execute in parallel. See Avoiding DbContext threading issues for more information and examples.

See Saving data in EF Core for more information and examples.

Exceptions

TYPE	CONDITION
DbUpdateException	An error is encountered while saving to the database.
DbUpdateConcurrencyException	A concurrency violation is encountered while saving to the database. A concurrency violation occurs when an unexpected number of rows are affected during save. This is usually because the data in the database has been modified since it was loaded into memory.
OperationCanceledException	If the CancellationToken is canceled.

Implements

IInfrastructure<T>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Namespace SoundParadise.Api.Dto.CartItem

Classes

CartItemDto

CartItem dto.

CartItemDtoProduct

CartItem dto product.

OrderItemDtoProduct

OrderItem dto product.

Class CartItemDto

CartItem dto.

Syntax

```
public class CartItemDto
```

Properties

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
[Guid](#)	

ProductId

Product Id.

Declaration

```
[JsonIgnore]
public Guid ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
[Guid](#)	

Quantity

Quantity.

Declaration

```
public short Quantity { get; set; }
```

Property Value

TYPE	DESCRIPTION
short	

Declaration

```
public short Quantity { get; set; }
```

Property Value

TYPE	DESCRIPTION

Class CartItemDtoProduct

CartItem dto product.

object

CartItemDto

CartItemDtoProduct

Inherited Members

CartItemDto.Id

CartItemDto.ProductId

CartItemDto.Quantity

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.CartItem

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartItemDtoProduct : CartItemDto
```

Properties

Product

Product.

Declaration

```
public ProductCartDto Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductCartDto	

Class OrderItemDtoProduct

OrderItem dto product.

Inheritance

object

CartItemDto

OrderItemDtoProduct

Inherited Members

CartItemDto.Id

CartItemDto.ProductId

CartItemDto.Quantity

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.CartItem

Assembly: SoundParadise.Api.dll

Syntax

```
public class OrderItemDtoProduct : CartItemDto
```

Properties

Product

Product.

Declaration

```
public ProductDto Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductDto	

SellerName

Seller name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Dto.Category

Classes

[CategoryDto](#)

Category dto.

Class CategoryDto

Category dto.

Inheritance

object

CategoryDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Category

Assembly: SoundParadise.Api.dll

Syntax

```
public class CategoryDto
```

Properties

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Image

Image.

Declaration

```
public ImageDto Image { get; set; }
```

Property Value

TYPE	DESCRIPTION
ImageDto	

Name

Name.

Declaration

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

TYPE	DESCRIPTION
string	

Subcategories

List of subcategorydto objects.

Declaration

```
public List<SubcategoryDto> Subcategories { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<SubcategoryDto>	

Subcategories

List of subcategorydto objects.

Namespace SoundParadise.Api.Dto.Comment

Classes

CommentDto

Class CommentDto

object

CommentDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Comment

Assembly: SoundParadise.Api.dll

Syntax

```
public class CommentDto
```

Namespace SoundParadise.Api.Dto.Image

Classes

[ImageDto](#)

Image dto.

Class ImageDto

Image dto.

Inheritance

object

ImageDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Image

Assembly: SoundParadise.Api.dll

Syntax

```
public class ImageDto
```

Properties

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Path

Path.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Dto.Order

Classes

OrderDto

Order dto.

OrderHistoryDto

Order history dto.

Class OrderDto

Order dto.

Inheritance

object

OrderDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Order

Assembly: SoundParadise.Api.dll

Syntax

```
public class OrderDto
```

Properties

Address

Address navigation property.

Declaration

```
public AddressDto? Address { get; set; }
```

Property Value

TYPE	DESCRIPTION
AddressDto	

AddressId

Addres Id.

Declaration

```
public Guid? AddressId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

CheckoutUrl

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

City

City.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Comment

Comment

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

CustomerName

Customer name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

CustomerSurname

Customer surname.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

DeliveryOption

Delivery option enum.

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
public DeliveryOptionEnum DeliveryOption { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryOptionEnum	

DeliveryType

Delivery type enum.

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
public DeliveryTypeEnum DeliveryType { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryTypeEnum	

OrderDate

Declaration

```
public DateTime? OrderDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

OrderItems

Declaration

```
public List<OrderItemDtoProduct>? OrderItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<OrderItemDtoProduct>	

OrderStatus

Declaration

```
public OrderStatusEnum? OrderStatus { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderStatusEnum?	

PaymentType

Payment type enum.

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
public PaymentTypeEnum PaymentType { get; set; }
```

Property Value

TYPE	DESCRIPTION
PaymentTypeEnum	

PhoneNumber

Phone number.

Declaration

```
[RegularExpression("^[0-9]*$", ErrorMessage = "Phone number must contain only numeric characters.")]
public string PhoneNumber { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

PostOfficeAddress

Post office address.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

TotalPrice

Declaration

```
public decimal? TotalPrice { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal?	

UserId

User Id.

Declaration

```
[JsonIgnore]
public Guid? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

UserId

User Id.

Declaration

```
[JsonIgnore]
public Guid? UserId { get; set; }
```

Class OrderHistoryDto

Order history dto.

Inheritance

object

OrderHistoryDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Order

Assembly: SoundParadise.Api.dll

Syntax

```
public class OrderHistoryDto
```

Properties

CheckoutUrl

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

OrderDate

Order date.

Declaration

```
public DateTime OrderDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

OrderItems

List of order items.

Declaration

```
public List<OrderItemDtoProduct>? OrderItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<OrderItemDtoProduct>	

OrderStatus

Order status enum.

Declaration

```
public OrderStatusEnum OrderStatus { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderStatusEnum	

TotalPrice

Total price.

Declaration

```
public decimal TotalPrice { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Namespace SoundParadise.Api.Dto.Payment

Classes

PaymentDto

Class PaymentDto

Inheritance

object

PaymentDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Payment

Assembly: SoundParadise.Api.dll

Syntax

```
public class PaymentDto
```

Namespace SoundParadise.Api.Dto.Product

Classes

[ProductCartDto](#)

ProductCrat dto.

[ProductDto](#)

[SuggestionProductDto](#)

Suggestion product dto.

Class ProductCartDto

ProductCrat dto.

Inheritance

object

ProductCartDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Product

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductCartDto
```

Properties

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

ImagePath

Image path.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Price

Nice.

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Class ProductDto

Inheritance

object

ProductDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Product

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductDto
```

Properties

CommentsCount

Comments count.

Declaration

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

Property Value

TYPE	DESCRIPTION
int?	

Description

Description

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

ImagePath

Image path.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

InCart

In cart status.

Declaration

```
public bool? InCart { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool?	

InStock

In stock status.

Declaration

```
public bool? InStock { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool?	

InWishlist

In wishlist status.

Declaration

```
public bool? InWishlist { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool?	

IsNew

Is new status.

Declaration

```
public bool? IsNew { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool?	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Price

Price.

Declaration

```
public decimal? Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal?	

Quantity

Quantity.

Declaration

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

Property Value

TYPE	DESCRIPTION
int?	

Rating

Rating.

Declaration

```
public decimal? Rating { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal?	

Slug

Slug.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Class SuggestionProductDto

Suggestion product dto.

Inheritance

object

SuggestionProductDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Product

Assembly: SoundParadise.Api.dll

Syntax

```
public class SuggestionProductDto
```

Properties

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Slug

Slug.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Dto.Requests

Classes

CreateOrderResult

Creatte order results.

SearchProductDtoResult

Search product dto result.

SearchProductResult

Search product result.

Class CreateOrderResult

Creatte order results.

Inherited Members

RequestResult.IsSuccess

RequestResult.Message

RequestResult.HttpStatus

RequestResult.Success(string, HttpStatusCode)

RequestResult.Error(string, HttpStatusCode)

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Requests

Assembly: SoundParadise.Api.dll

Syntax

```
public class CreateOrderResult : RequestResult
```

Properties

CheckoutUrl

Checkout url.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Id

Create order result Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Methods

Error(string, HttpStatusCode, Guid, string)

Error.

Declaration

```
public static CreateOrderResult Error(string message, HttpStatusCode httpStatus = HttpStatusCode.BadRequest,
Guid id = default, string checkoutUrl = "")
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	Message.
HttpStatusCode	httpStatus	HttpStatusCode object.
Guid	id	Id.
string	checkoutUrl	Checkout url.

Returns

TYPE	DESCRIPTION
CreateOrderResult	

Success(string, Guid, HttpStatusCode, string)

Success.

Declaration

```
public static CreateOrderResult Success(string message = "", Guid id = default, HttpStatusCode httpStatus =
HttpStatusCode.OK, string checkoutUrl = "")
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	String message.
Guid	id	Id.
HttpStatusCode	httpStatus	HttpStatusCode object.
string	checkoutUrl	CheckOut url.

Returns

TYPE	DESCRIPTION
CreateOrderResult	CreateOrderResult object

Returns

TYPE	DESCRIPTION
CreateOrderResult	CreateOrderResult object

Class SearchProductDtoResult

Search product dto result.

Inheritance

object

SearchProductDtoResult

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Requests

Assembly: SoundParadise.Api.dll

Syntax

```
public class SearchProductDtoResult
```

Properties

Products

List of products.

Declaration

```
public List<ProductDto>? Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductDto>	

TotalCount

Total count.

Declaration

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

Property Value

TYPE	DESCRIPTION
int?	

Class SearchProductResult

Search product result.

Syntax

```
public class SearchProductResult
```

Properties

Products

List of products.

Declaration

```
public List<ProductModel>? Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductModel>	

TotalCount

Total count.

Declaration

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

Property Value

TYPE	DESCRIPTION
int?	

Namespace SoundParadise.Api.Dto.Subcategory

Classes

SubcategoryDto

Subcategory dto.

Class SubcategoryDto

Subcategory dto.

Inheritance

object

SubcategoryDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.Subcategory

Assembly: SoundParadise.Api.dll

Syntax

```
public class SubcategoryDto
```

Properties

Id

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Path

Path.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Dto.User

Classes

UserAuthenticationDto

User Authentication dto.

UserCreateDto

User create dto.

UserDto

User dto.

Class UserAuthenticationDto

User Authentication dto.

Inheritance

object

UserAuthenticationDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Dto.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserAuthenticationDto
```

Properties

Email

Email.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Password

Password.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Class UserCreateDto

User create dto.

Inheritance

object
UserDto
UserCreateDto

Inherited Members

UserDto.Id
UserDto.Name
UserDto.Surname
UserDto.Username
UserDto.Email
UserDto.ImageUrl
UserDto.PhoneNumber
UserDto.Role
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: SoundParadise.Api.Dto.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserCreateDto : UserDto
```

Properties

Password

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Class UserDto

User dto.

Syntax

```
public class UserDto
```

Properties

Email

Email.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public Guid? Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

ImageUrl

Image Url.

Declaration

```
public string ImageUrl { get; init; }
```

Property Value

TYPE	DESCRIPTION
string	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

PhoneNumber

Phone Number.

Declaration

```
public string PhoneNumber { get; init; }
```

Property Value

TYPE	DESCRIPTION
string	

Role

User role enum.

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
public UserRoleEnum Role { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserRoleEnum	

Surname

Surname.

Declaration

```
public string Surname { get; init; }
```

Property Value

TYPE	DESCRIPTION
string	

Username

Username.

Declaration

```
public string Username { get; init; }
```

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Extensions

Classes

SessionExtensions

Session extensions.

Class SessionExtensions

Session extensions.

Inheritance

object

SessionExtensions

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Extensions

Assembly: SoundParadise.Api.dll

Syntax

```
public static class SessionExtensions
```

Methods

Get<T>(ISession, string)

Get session data.

Declaration

```
public static T Get<T>(this ISession session, string key)
```

Parameters

TYPE	NAME	DESCRIPTION
ISession	session	Session.
string	key	Session key.

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

Set<T>(ISession, string, T)

Set session data.

Declaration

```
public static void Set<T>(this ISession session, string key, T value)
```

Parameters

TYPE	NAME	DESCRIPTION
ISession	session	Session.
string	key	Session key.
T	value	Session value.

Type Parameters

NAME	DESCRIPTION
T	

Declaration

```
public static void Set<T>(this ISession session, string key, T value)
```

Parameters

TYPE	NAME	DESCRIPTION
ISession	session	Session.
string	key	Session key.
T	value	Session value.

Namespace SoundParadise.Api.Helpers

Classes

[CartItemValidationHelper](#)

CartItem validator service helper.

[SlugHelper](#)

Slug helper.

[UserValidationHelper](#)

User validator helper.

Class CartItemValidationHelper

CartItem validator service helper.

Inheritance

object

CartItemValidationHelper

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Helpers

Assembly: SoundParadise.Api.dll

Syntax

```
public static class CartItemValidationHelper
```

Methods

IsCanBeAddedToCart(Guid, short, SoundParadiseDbContext)

Is can be added to cart.

Declaration

```
public static bool IsCanBeAddedToCart(Guid productId, short quantity, SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.
short	quantity	Quantity.
SoundParadiseDbContext	context	

Returns

TYPE	DESCRIPTION
bool	True if can be added, false if not.

IsProductExists(Guid, SoundParadiseDbContext)

Is Product exists.

Declaration

```
public static bool IsProductExists(Guid productId, SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.
SoundParadiseDbContext	context	Db context.

Returns

TYPE	DESCRIPTION
bool	True if exists, false if not.

Class SlugHelper

Slug helper.

Inheritance

object

SlugHelper

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Helpers

Assembly: SoundParadise.Api.dll

Syntax

```
public static class SlugHelper
```

Methods

GenerateSlug(string, SoundParadiseDbContext)

Generate Slug.

Declaration

```
public static string GenerateSlug(string name, SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	
SoundParadiseDbContext	context	Db context.

Returns

TYPE	DESCRIPTION
string	

Class UserValidationHelper

User validator helper.

Inheritance

object

UserValidationHelper

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Helpers

Assembly: SoundParadise.Api.dll

Syntax

```
public static class UserValidationHelper
```

Methods

EmailIsNotTaken(string, SoundParadiseDbContext)

Email is not taken.

Declaration

```
public static bool EmailIsNotTaken(string email, SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
string	email	Email.
SoundParadiseDbContext	context	Db context.

Returns

TYPE	DESCRIPTION
bool	True if ok, false if not.

PhoneNumberIsNotTaken(string, SoundParadiseDbContext)

Phone number is not taken.

Declaration

```
public static bool PhoneNumberIsNotTaken(string phoneNumber, SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
string	phoneNumber	Phone number.
SoundParadiseDbContext	context	Db context.

Returns

TYPE	DESCRIPTION
bool	True if ok, false if not.

ValidateEmail(string)

Validate email.

Declaration

```
public static bool ValidateEmail(string email)
```

Parameters

TYPE	NAME	DESCRIPTION
string	email	String email.

Returns

TYPE	DESCRIPTION
bool	

ValidatePassword(string)

Validate password.

Declaration

```
public static bool ValidatePassword(string password)
```

Parameters

TYPE	NAME	DESCRIPTION
string	password	Password.

Returns

TYPE	DESCRIPTION
bool	True if validate, false if not.

ValidatePhoneNumber(string)

Validate phone number.

Declaration

```
public static bool ValidatePhoneNumber(string phoneNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
string	phoneNumber	

Returns

TYPE	DESCRIPTION
bool	True if ok, false if not.

ValidatePhoneNumber(string)

Validate phone number.

Namespace SoundParadise.Api.Interfaces

Interfaces

IAddressCrud

AddressCrud implements that interface.

ICardCrud

CardCrud implements that interface.

ICartCrud

CartCrud implements that interface.

ICartItemCrud

CartItemCrud implements that interface.

ICategoryCrud

CategoryCrud implements that interface.

ICommentCrud

CommentCrud implements that interface.

IEncryptionService

Encryption Service implements that interface.

ILoggingService<T>

Logging service implements that interface.

IOrderCrud

OrderCrud implements that interface.

IPaymentService

Payment Service implements that interface.

IProductCrud

ProductCrud implements that interface.

IProductParameterCrud

ProductParameterCrud implements that interface.

ISubcategoryCrud

SubcategoryCrud implements that interface.

IUserCrud

UserCrud implements that interface.

IWishlistCrud

WishlistCrud implements that interface.

Interface IAddressCrud

AddressCrud implements that interface.

Syntax

```
public interface IAddressCrud
```

Methods

AddAddress(AddressDto, Guid)

Declaration

```
RequestResult AddAddress(AddressDto addressDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressDto	addressDto	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

CreateAddress(AddressModel)

Declaration

```
RequestResult CreateAddress(AddressModel address)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressModel	address	

Returns

TYPE	DESCRIPTION
RequestResult	

DeleteAddress(Guid, Guid)

Declaration

```
RequestResult DeleteAddress(Guid addressId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	addressId	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

GetAddressById(Guid?, Guid?)

Declaration

```
AddressModel GetAddressById(Guid? addressId, Guid? userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	addressId	
Guid?	userId	

Returns

TYPE	DESCRIPTION
AddressModel	

GetAddressDtoById(Guid?, Guid?)

Declaration

```
AddressDto GetAddressDtoById(Guid? addressId, Guid? userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	addressId	
Guid?	userId	

Returns

TYPE	DESCRIPTION
AddressDto	

GetAddressesByDeliveryOption(DeliveryOptionEnum, Guid)

Declaration

```
List<AddressDto> GetAddressesByDeliveryOption(DeliveryOptionEnum deliveryOption, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
DeliveryOptionEnum	deliveryOption	
Guid	userId	

Returns

TYPE	DESCRIPTION
List<AddressDto>	

GetAddressesByUserId(Guid)

Declaration

```
List<AddressDto> GetAddressesByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<AddressDto>	

Interface ICardCrud

CardCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ICardCrud
```

Methods

AddCard(CardDto, Guid)

Declaration

```
RequestResult AddCard(CardDto cardDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
CardDto	cardDto	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

CardExists(IEnumerable, Guid)

Declaration

```
RequestResult CardExists(IEnumerable cardNumber, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable	cardNumber	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

CardExists(Guid, Guid)

Declaration

```
RequestResult CardExists(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

DeleteCard(Guid, Guid)

Declaration

```
RequestResult DeleteCard(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

GetCardById(Guid, Guid)

Declaration

```
CardDto GetCardById(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	
Guid	userId	

Returns

TYPE	DESCRIPTION
CardDto	

GetCardsByUserId(Guid)

Declaration

```
List<CardDto> GetCardsByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<CardDto>	

UpdateCard(CardDto, Guid)

Declaration

```
RequestResult UpdateCard(CardDto cardDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
CardDto	cardDto	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

Interface ICartCrud

CartCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ICartCrud
```

Methods

AddCartItem(Guid, Guid)

Declaration

```
RequestResult AddCartItem(Guid cartId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

CartItemExists(Guid, out Guid)

Declaration

```
bool CartItemExists(Guid cartItemId, out Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartItemId	
Guid	cartId	

Returns

TYPE	DESCRIPTION
bool	

CreateCart(CartModel)

Declaration

```
bool CreateCart(CartModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartModel	cart	

Returns

TYPE	DESCRIPTION
bool	

DeleteCart(Guid)

Declaration

```
bool DeleteCart(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	

Returns

TYPE	DESCRIPTION
bool	

GetAllCarts()

Declaration

```
List<CartModel> GetAllCarts()
```

Returns

TYPE	DESCRIPTION
List<CartModel>	

GetCartById(Guid)

Declaration

```
CartModel GetCartById(Guid Id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	Id	

Returns

TYPE	DESCRIPTION
CartModel	

GetCartByUserId(Guid)

Declaration

```
List<CartItemDtoProduct> GetCartByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<CartItemDtoProduct>	

GetCartIdByUserId(Guid)

Declaration

```
Guid GetCartIdByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
Guid	

GetCartItems(Guid)

Declaration

```
List<CartItemModel> GetCartItems(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	

Returns

TYPE	DESCRIPTION
List<CartItemModel>	

RemoveCartItem(Guid, Guid)

Declaration

```
RequestResult RemoveCartItem(Guid cartId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

UpdateCart(CartModel)

Declaration

```
bool UpdateCart(CartModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartModel	cart	

Returns

TYPE	DESCRIPTION
bool	

UpdateCartItemQuantity(Guid, Guid, short)

Declaration

```
RequestResult UpdateCartItemQuantity(Guid cartId, Guid cartItemId, short quantity)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	
Guid	cartItemId	
short	quantity	

Returns

TYPE	DESCRIPTION
RequestResult	

Interface ICartItemCrud

CartItemCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ICartItemCrud
```

Methods

CreateCartItem(CartItemModel)

Declaration

```
bool CreateCartItem(CartItemModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartItemModel	cart	

Returns

TYPE	DESCRIPTION
bool	

DeleteCartItem(Guid)

Declaration

```
bool DeleteCartItem(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	

Returns

TYPE	DESCRIPTION
bool	

GetAllCartItems()

Declaration

```
List<CartItemModel> GetAllCartItems()
```

Returns

TYPE	DESCRIPTION
List<CartItemModel>	

GetCartItemByCartId(Guid)

Declaration

```
CartItemModel GetCartItemByCartId(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	

Returns

TYPE	DESCRIPTION
CartItemModel	

GetCartItemById(Guid)

Declaration

```
CartItemModel? GetCartItemById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
CartItemModel	

GetCartItemsByCartId(Guid?)

Declaration

```
List<CartItemModel> GetCartItemsByCartId(Guid? cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	cartId	

Returns

TYPE	DESCRIPTION
List<CartItemModel>	

GetTotalPriceOfOrder(IEnumerable<CartItemDto>)

Declaration

```
decimal GetTotalPriceOfOrder(IEnumerable<CartItemDto> cartItems)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable<CartItemDto>	cartItems	

Returns

TYPE	DESCRIPTION
decimal	

GetTotalPriceOfOrderAuth(IEnumerable<CartItemModel>)

Declaration

```
decimal GetTotalPriceOfOrderAuth(IEnumerable<CartItemModel> cartItems)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable<CartItemModel>	cartItems	

Returns

TYPE	DESCRIPTION
decimal	

UpdateCartItem(CartItemModel)

Declaration

```
bool UpdateCartItem(CartItemModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartItemModel	cart	

Returns

TYPE	DESCRIPTION
bool	

Interface ICategoryCrud

CategoryCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ICategoryCrud
```

Methods

CreateCategory(CategoryModel)

Declaration

```
bool CreateCategory(CategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
CategoryModel	category	

Returns

TYPE	DESCRIPTION
bool	

DeleteCategory(Guid)

Declaration

```
bool DeleteCategory(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
bool	

GetAllCategories()

Declaration

```
List<CategoryDto> GetAllCategories()
```

Returns

TYPE	DESCRIPTION
List<CategoryDto>	

GetCategoryById(Guid)

Declaration

```
CategoryModel? GetCategoryById(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
CategoryModel	

GetCategoryByName(string)

Declaration

```
CategoryModel? GetCategoryByName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	

Returns

TYPE	DESCRIPTION
CategoryModel	

GetCategoryNameById(Guid)

Declaration

```
string GetCategoryNameById(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
string	

GetSubcategoriesOfCategory(Guid)

Declaration

```
List<SubcategoryDto> GetSubcategoriesOfCategory(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	

Returns

TYPE	DESCRIPTION
List<SubcategoryDto>	

GetSubcategoryNameById(Guid)

Declaration

```
string GetSubcategoryNameById(Guid subcategoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	subcategoryId	

Returns

TYPE	DESCRIPTION
string	

Interface ICommentCrud

CommentCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ICommentCrud
```

Methods

CreateComment(CommentModel)

Declaration

```
bool CreateComment(CommentModel comment)
```

Parameters

TYPE	NAME	DESCRIPTION
CommentModel	comment	

Returns

TYPE	DESCRIPTION
bool	

DeleteComment(Guid)

Declaration

```
bool DeleteComment(Guid commentId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	commentId	

Returns

TYPE	DESCRIPTION
bool	

GetAllComments()

Declaration

```
List<CommentModel> GetAllComments()
```

Returns

TYPE	DESCRIPTION
List<CommentModel>	

GetCommentByProductId(Guid)

Declaration

```
CommentModel GetCommentByProductId(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
CommentModel	

GetCommentByUserId(Guid)

Declaration

```
CommentModel GetCommentByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
CommentModel	

GetCommentsCount(Guid)

Declaration

```
int GetCommentsCount(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
int	

Interface IEncryptionService

Encryption Service implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface IEncryptionService
```

Methods

DecryptData(byte[], ICryptoTransform)

Declaration

```
string DecryptData(byte[] encryptedData, ICryptoTransform decryptor)
```

Parameters

TYPE	NAME	DESCRIPTION
byte[]	encryptedData	
ICryptoTransform	decryptor	

Returns

TYPE	DESCRIPTION
string	

EncryptData(string, ICryptoTransform)

Declaration

```
byte[] EncryptData(string data, ICryptoTransform encryptor)
```

Parameters

TYPE	NAME	DESCRIPTION
string	data	
ICryptoTransform	encryptor	

Returns

TYPE	DESCRIPTION
byte[]	

Interface ILoggingService<T>

Logging service implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ILoggingService<T>
```

Type Parameters

NAME	DESCRIPTION
T	

Methods

Log(LogLevel, string)

Declaration

```
void Log(LogLevel logLevel, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
LogLevel	logLevel	
string	message	

LogError(string)

Declaration

```
void LogError(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	

LogException(Exception, string)

Declaration

```
void LogException(Exception exception, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
Exception	exception	
string	message	

Interface IOrderCrud

OrderCrud implements that interface.

Syntax

```
public interface IOrderCrud
```

Methods

ConfirmOrder(dynamic)

Declaration

```
RequestResult ConfirmOrder(dynamic response)
```

Parameters

TYPE	NAME	DESCRIPTION
dynamic	response	

Returns

TYPE	DESCRIPTION
RequestResult	

CreateOrder(OrderModel)

Declaration

```
RequestResult CreateOrder(OrderModel order)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderModel	order	

Returns

TYPE	DESCRIPTION
RequestResult	

CreateOrderAuth(OrderDto)

Declaration

```
CreateOrderResult CreateOrderAuth(OrderDto order)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderDto	order	

Returns

TYPE	DESCRIPTION
CreateOrderResult	

DeleteComment(Guid)

Declaration

```
bool DeleteComment(Guid orderId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	

Returns

TYPE	DESCRIPTION
bool	

GetAllOrders()

Declaration

```
List<OrderModel> GetAllOrders()
```

Returns

TYPE	DESCRIPTION
List<OrderModel>	

GetOrderById(Guid, Guid)

Declaration

```
OrderDto GetOrderById(Guid orderId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	
Guid	userId	

Returns

TYPE	DESCRIPTION
OrderDto	

GetOrdersByUserId(Guid)

Declaration

```
List<OrderHistoryDto> GetOrdersByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<OrderHistoryDto>	

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Interface IPaymentService

Payment Service implements that interface.

Syntax

```
public interface IPaymentService
```

Methods

Checkout(Guid, string, decimal, PaymentProvider)

Declaration

```
RequestResult Checkout(Guid orderId, string description, decimal amount, PaymentProvider paymentProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	
string	description	
decimal	amount	
PaymentProvider	paymentProvider	

Returns

TYPE	DESCRIPTION
RequestResult	

Interface IProductCrud

ProductCrud implements that interface.

Syntax

```
public interface IProductCrud
```

Methods

CreateComment(CommentModel)

Declaration

```
CommentModel CreateComment(CommentModel comment)
```

Parameters

TYPE	NAME	DESCRIPTION
CommentModel	comment	

Returns

TYPE	DESCRIPTION
CommentModel	

CreateProduct(ProductModel)

Declaration

```
ProductModel CreateProduct(ProductModel product)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductModel	product	

Returns

TYPE	DESCRIPTION
ProductModel	

DeleteImage(Guid, Guid)

Declaration

```
RequestResult DeleteImage(Guid imageId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	imageId	

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

DeleteProduct(Guid)

Declaration

```
RequestResult DeleteProduct(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

GetAllProducts(SortOptionsEnum, int, int, Guid)

Declaration

```
SearchProductResult GetAllProducts(SortOptionsEnum sort, int page = 1, int pageSize = 3, Guid userId =
default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
int	page	
int	pageSize	
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductResult	

GetPopularProducts(int, int, Guid)

Declaration

```
List<ProductDto> GetPopularProducts(int page = 1, int limit = 3, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	
int	limit	
Guid	userId	

Returns

TYPE	DESCRIPTION
List<ProductDto>	

GetProductById(Guid, Guid)

Declaration

```
ProductModel GetProductById(Guid productId, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	
Guid	userId	

Returns

TYPE	DESCRIPTION
ProductModel	

GetProductBySlug(string, Guid)

Declaration

```
ProductModel GetProductBySlug(string slug, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
string	slug	
Guid	userId	

Returns

TYPE	DESCRIPTION
ProductModel	

GetProductCartDtoById(Guid)

Declaration

```
ProductCartDto GetProductCartDtoById(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
ProductCartDto	

GetProductsByCategoryId(SortOptionsEnum, Guid, int, int, Guid)

Declaration

```
SearchProductDtoResult GetProductsByCategoryId(SortOptionsEnum sort, Guid categoryId, int page = 1, int
pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
Guid	categoryId	
int	page	
int	pageSize	
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	

GetProductsByCategoryName(SortOptionsEnum, string, int, int, Guid)

Declaration

```
SearchProductDtoResult GetProductsByCategoryName(SortOptionsEnum sort, string categoryName, int page = 1, int
pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	categoryName	
int	page	
```

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| int | pageSize | |
| Guid | userId | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SearchProductDtoResult | |

## GetProductsBySubcategoryId(SortOptionsEnum, Guid, int, int, Guid)

Declaration

```
SearchProductDtoResult GetProductsBySubcategoryId(SortOptionsEnum sort, Guid subcategoryId, int page = 1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
Guid	subcategoryId	
int	page	
int	pageSize	
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	

GetProductsBySubcategoryName(SortOptionsEnum, string, int, int, Guid)

Declaration

```
SearchProductDtoResult GetProductsBySubcategoryName(SortOptionsEnum sort, string subcategoryName, int page = 1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	subcategoryName	
int	page	
int	pageSize	

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	

GetProductsByUserId(Guid)

Declaration

```
SearchProductResult GetProductsByUserId(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
SearchProductResult	

GetRecentBuys(int, int, Guid)

Declaration

```
List<ProductDto> GetRecentBuys(int page = 1, int limit = 3, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	
int	limit	
Guid	userId	

Returns

TYPE	DESCRIPTION
List<ProductDto>	

GetSpecial0ffers(SortOptionsEnum, int, int, Guid)

Declaration

```
List<ProductDto> GetSpecial0ffers(SortOptionsEnum sort, int page = 1, int limit = 3, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
int	page	
int	limit	
Guid	userId	

Returns

TYPE	DESCRIPTION
List<ProductDto>	

IsProductInCart(Guid, Guid)

Declaration

```
bool IsProductInCart(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	
Guid	productId	

Returns

TYPE	DESCRIPTION
bool	

IsProductInWishlist(Guid, Guid)

Declaration

```
bool IsProductInWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	
Guid	productId	

Returns

TYPE	DESCRIPTION
bool	

IsQuantityValid(Guid, short)

Declaration

```
bool IsQuantityValid(Guid productId, short quantity)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	
short	quantity	

Returns

TYPE	DESCRIPTION
bool	

ProductExists(Guid)

Declaration

```
bool ProductExists(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
bool	

SearchProducts(SortOptionsEnum, string, int, int, Guid)

Declaration

```
SearchProductDtoResult SearchProducts(SortOptionsEnum sort, string query, int page = 1, int pageSize = 20,
Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	
string	query	
int	page	
int	pageSize	
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	

SearchSuggestions(string, int)

Declaration

```
List<SuggestionProductDto> SearchSuggestions(string query, int limit)
```

Parameters

TYPE	NAME	DESCRIPTION
string	query	
int	limit	

Returns

TYPE	DESCRIPTION
List<SuggestionProductDto>	

UpdateProduct(ProductModel)

Declaration

```
bool UpdateProduct(ProductModel productChange)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductModel	productChange	

Returns

TYPE	DESCRIPTION
bool	

UploadImage(Guid, string)

Declaration

```
RequestResult UploadImage(Guid productId, string image)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	
string	image	

Returns

TYPE	DESCRIPTION
RequestResult	

Interface IProductParameterCrud

ProductParameterCrud implements that interface.

Syntax

```
public interface IProductParameterCrud
```

Methods

CreateParameter(ProductParameterModel)

Declaration

```
bool CreateParameter(ProductParameterModel parameter)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductParameterModel	parameter	

Returns

TYPE	DESCRIPTION
bool	

DeleteParameter(Guid)

Declaration

```
bool DeleteParameter(Guid parameterId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	parameterId	

Returns

TYPE	DESCRIPTION
bool	

GetAllParameters()

Declaration

```
List<ProductParameterModel> GetAllParameters()
```

Returns

TYPE	DESCRIPTION
List<ProductParameterModel>	

GetParameterByProductId(Guid)

Declaration

```
List<ProductParameterModel> GetParameterByProductId(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
List<ProductParameterModel>	

GetParameterByProductId(Guid)

Declaration

```
List<ProductParameterModel> GetParameterByProductId(Guid productId)
```

Interface ISubcategoryCrud

SubcategoryCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface ISubcategoryCrud
```

Methods

CreateSubcategory(SubcategoryModel)

Declaration

```
bool CreateSubcategory(SubcategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
SubcategoryModel	category	

Returns

TYPE	DESCRIPTION
bool	

DeleteSubcategory(Guid)

Declaration

```
RequestResult DeleteSubcategory(Guid subcategoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	subcategoryId	

Returns

TYPE	DESCRIPTION
RequestResult	

GetAllSubcategories()

Declaration

```
List<SubcategoryModel> GetAllSubcategories()
```

Returns

TYPE	DESCRIPTION
List<SubcategoryModel>	

GetSubcategoryByCategory(CategoryModel)

Declaration

```
List<SubcategoryModel> GetSubcategoryByCategory(CategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
CategoryModel	category	

Returns

TYPE	DESCRIPTION
List<SubcategoryModel>	

GetSubcategoryById(Guid)

Declaration

```
SubcategoryModel GetSubcategoryById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
SubcategoryModel	

GetSubcategoryByName(string)

Declaration

```
SubcategoryModel GetSubcategoryByName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	

Returns

TYPE	DESCRIPTION
SubcategoryModel	

Interface IUserCrud

UserCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface IUserCrud
```

Methods

AuthenticateUser(UserAuthenticationDto)

Declaration

```
RequestResult AuthenticateUser(UserAuthenticationDto authenticationDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserAuthenticationDto	authenticationDto	

Returns

TYPE	DESCRIPTION
RequestResult	

ConfirmUser(Guid, string)

Declaration

```
RequestResult ConfirmUser(Guid userId, string token)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	
string	token	

Returns

TYPE	DESCRIPTION
RequestResult	

CreateUser(UserCreateDto)

Declaration

```
bool CreateUser(UserCreateDto userDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userDto	

Returns

TYPE	DESCRIPTION
bool	

DeleteUser(Guid)

Declaration

```
RequestResult DeleteUser(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
RequestResult	

GenerateConfirmationLink(UserModel)

Declaration

```
string GenerateConfirmationLink(UserModel user)
```

Parameters

TYPE	NAME	DESCRIPTION
UserModel	user	

Returns

TYPE	DESCRIPTION
string	

GenerateUniqueToken()

Declaration

```
string GenerateUniqueToken()
```

Returns

TYPE	DESCRIPTION
string	

GetAllUsers()

Declaration

```
List<UserModel> GetAllUsers()
```

Returns

TYPE	DESCRIPTION
List<UserModel>	

GetUserByCredentials(UserAuthenticationDto)

Declaration

```
UserModel? GetUserByCredentials(UserAuthenticationDto authenticationDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserAuthenticationDto	authenticationDto	

Returns

TYPE	DESCRIPTION
UserModel	

GetUserById(Guid)

Declaration

```
UserModel GetUserById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
UserModel	

GetUserByToken(string)

Declaration

```
UserModel GetUserByToken(string token)
```

Parameters

TYPE	NAME	DESCRIPTION
string	token	

Returns

TYPE	DESCRIPTION
UserModel	

GetUserDtoById(Guid)

Declaration

```
UserDto GetUserDtoById(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
UserDto	

Logout(string)

Declaration

```
RequestResult Logout(string token)
```

Parameters

TYPE	NAME	DESCRIPTION
string	token	

Returns

TYPE	DESCRIPTION
RequestResult	

RegisterUser(UserCreateDto, List<CartItemDto>)

Declaration

```
UserModel RegisterUser(UserCreateDto userDto, List<CartItemDto> cartItemDtos)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userDto	
List<CartItemDto>	cartItemDtos	

Returns

TYPE	DESCRIPTION
UserModel	

UpdateUser(UserDto)

Declaration

```
RequestResult UpdateUser(UserDto userDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserDto	userDto	

Returns

TYPE	DESCRIPTION
RequestResult	

UploadAvatar(IFormFile, UserModel)

Declaration

```
RequestResult UploadAvatar(IFormFile file, UserModel userModel)
```

Parameters

TYPE	NAME	DESCRIPTION
IFormFile	file	
UserModel	userModel	

Returns

TYPE	DESCRIPTION
RequestResult	

Interface IWishlistCrud

WishlistCrud implements that interface.

Namespace: SoundParadise.Api.Interfaces

Assembly: SoundParadise.Api.dll

Syntax

```
public interface IWishlistCrud
```

Methods

AddToWishlist(Guid, Guid)

Declaration

```
RequestResult AddToWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

CreateWishlist(WishlistModel)

Declaration

```
RequestResult CreateWishlist(WishlistModel wishlist)
```

Parameters

TYPE	NAME	DESCRIPTION
WishlistModel	wishlist	

Returns

TYPE	DESCRIPTION
RequestResult	

GetWishlistById(Guid)

Declaration

```
WishlistModel GetWishlistById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
WishlistModel	

GetWishlistByUserId(Guid)

Declaration

```
List<ProductDto> GetWishlistByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<ProductDto>	

RemoveFromWishlist(Guid, Guid)

Declaration

```
RequestResult RemoveFromWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	

Namespace SoundParadise.Api.Middlewares

Classes

[TokenExpirationMiddleware](#)

Token expiration middleware.

Class TokenExpirationMiddleware

Token expiration middleware.

Inheritance

object

TokenExpirationMiddleware

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Middlewares

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenExpirationMiddleware
```

Constructors

TokenExpirationMiddleware(RequestDelegate)

Constructor for token expiration middleware

Declaration

```
public TokenExpirationMiddleware(RequestDelegate next)
```

Parameters

TYPE	NAME	DESCRIPTION
RequestDelegate	next	Request Delegate.

Methods

InvokeAsync(HttpContext, TokenService, IServiceProvider)

Invoke.

Declaration

```
public Task InvokeAsync(HttpContext context, TokenService tokenService, IServiceProvider serviceProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
HttpContext	context	Http context.
TokenService	tokenService	Token service.

TYPE	NAME	DESCRIPTION
IServiceProvider	serviceProvider	Service provider.

Returns

TYPE	DESCRIPTION
Task	Task object.

Namespace SoundParadise.Api.Migrations

Classes

Initial

A base class inherited by each EF Core migration.

Class Initial

A base class inherited by each EF Core migration.

Inheritance

object
Migration
Initial

Inherited Members

Migration.InitialDatabase
Migration.TargetModel
Migration.UpOperations
Migration.DownOperations
Migration.ActiveProvider
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: SoundParadise.Api.Migrations

Assembly: SoundParadise.Api.dll

Syntax

```
[DbContext(typeof(SoundParadiseDbContext))]
[Migration("20230712235142_Initial")]
public class Initial : Migration
```

Remarks

See Database migrations for more information and examples.

Methods

BuildTargetModel(ModelBuilder)

Implemented to build the TargetModel.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
ModelBuilder	modelBuilder	The ModelBuilder to use to build the model.

Overrides

Migration.BuildTargetModel(ModelBuilder)

Remarks

See Database migrations for more information and examples.

Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

Declaration

```
protected override void Down(MigrationBuilder migrationBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
MigrationBuilder	migrationBuilder	The MigrationBuilder that will build the operations.

Overrides

Migration.Down(MigrationBuilder)

Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from Migration if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See Database migrations for more information and examples.

Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

Declaration

```
protected override void Up(MigrationBuilder migrationBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
MigrationBuilder	migrationBuilder	The MigrationBuilder that will build the operations.

Overrides

Migration.Up(MigrationBuilder)

Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from Migration.

See Database migrations for more information and examples.

Namespace SoundParadise.Api.Models.Accumulators.AllSalesAccumulator

Classes

AllSalesAccumulator

AllSalesAccumulator data model.

Class AllSalesAccumulator

AllSalesAccumulator data model.

Inheritance

object

AllSalesAccumulator

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Accumulators.AllSalesAccumulator

Assembly: SoundParadise.Api.dll

Syntax

```
public class AllSalesAccumulator
```

Properties

AllCount

Sales ount.

Declaration

```
[Column("count")]
public int AllCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Date

Date.

Declaration

```
[Key]
[Column("date")]
public DateTime Date { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Namespace SoundParadise.Api.Models.Accumulators.ProductSalesAccumulator

Classes

ProductSalesAccumulator

ProductSales data model.

Class ProductSalesAccumulator

ProductSales data model.

Inheritance

object

ProductSalesAccumulator

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Accumulators.ProductSalesAccumulator

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductSalesAccumulator
```

Properties

AllCount

Sales count.

Declaration

```
[Column("count")]
public int AllCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Date

Date.

Declaration

```
[Key]
[Column("date")]
public DateTime Date { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

ProductName

Product Name.

Declaration

```
[Column("product_name")]
public string ProductName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Declaration

```
[Column("product_name")]
public string ProductName { get; set; }
```

Property Value

TYPE	DESCRIPTION

Namespace SoundParadise.Api.Models.Accumulators.UserAccumulator

Classes

UserAccumulator

UserAccumulator data model.

Class UserAccumulator

UserAccumulator data model.

Inheritance

object

UserAccumulator

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Accumulators.UserAccumulator

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserAccumulator
```

Properties

Authorized

Authorized users.

Declaration

```
[Column("authorized")]
public int Authorized { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Date

Date.

Declaration

```
[Key]
[Column("date")]
public DateTime Date { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Registered

Registered users.

Declaration

```
[Column("registered")]
public int Registered { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Declaration

```
[Column("registered")]
public int Registered { get; set; }
```

Property Value

TYPE	DESCRIPTION

Namespace SoundParadise.Api.Models.Address

Classes

AddressModel

Address data model.

Class AddressModel

Address data model.

object
AddressModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Address

Assembly: SoundParadise.Api.dll

Syntax

```
public class AddressModel
```

Properties

City

City.

Declaration

```
[Column("city")]
public string? City { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DeliveryOption

DeliveryOption enum.

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
[Column("delivery_option")]
public DeliveryOptionEnum DeliveryOption { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryOptionEnum	

Id

Address Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("address_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Orders

Order navigation property.

Declaration

```
[JsonIgnore]
public List<OrderModel>? Orders { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<OrderModel>	

PostOfficeAddress

PostOfficeAddress.

Declaration

```
[Column("post_office_address")]
public string? PostOfficeAddress { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

User

User navigation property.

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id.

Declaration

```
[Column("user_id")]
public Guid? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Declaration

```
[Column("user_id")]
public Guid? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Namespace SoundParadise.Api.Models.Card

Classes

CardCrud

CardModel CRUD.

CardModel

Card data model.

Class CardCrud

CardModel CRUD.

Inheritance

[object](#)
CardCrud

Implements

[ICardCrud](#)

Inherited Members

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

Namespace: [SoundParadise.Api.Models.Card](#)

Assembly: SoundParadise.Api.dll

Syntax

```
public class CardCrud : ICardCrud
```

Constructors

CardCrud(SoundParadiseDbContext, ILoggingService<CardCrud>, IUserCrud, CardEncryptionService)

CardCrud constructor.

Declaration

```
public CardCrud(SoundParadiseDbContext context, ILoggingService<CardCrud> loggingService, IUserCrud userCrud,
CardEncryptionService cardEncryptionService)
```

Parameters

TYPE	NAME	DESCRIPTION
[SoundParadiseDbContext](#)	context	Db context
[ILoggingService](#)<[CardCrud](#)>	loggingService	Service for logging errors
[IUserCrud](#)	userCrud	UserCrud
[CardEncryptionService](#)	cardEncryptionService	Service for encryption card

Methods

AddCard(CardDto, Guid)

Add card in data base.

Declaration

```
public RequestResult AddCard(CardDto cardDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
CardDto	cardDto	CardDto object
Guid	userId	User Id

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

CardExists(IEnumerable, Guid)

Check if card exists.

Declaration

```
public RequestResult CardExists(IEnumerable cardNumber, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable	cardNumber	Card number.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object.

CardExists(Guid, Guid)

Check if card exists.

Declaration

```
public RequestResult CardExists(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	Card Id.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object.

DeleteCard(Guid, Guid)

Delete card from data base.

Declaration

```
public RequestResult DeleteCard(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	Card Id
Guid	userId	User Id

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

GetCardById(Guid, Guid)

Get Cards by Id.

Declaration

```
public CardDto GetCardById(Guid cardId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cardId	Card Id.

TYPE	NAME	DESCRIPTION
[Guid](#)	userId	User Id.

Returns

TYPE	DESCRIPTION
[CardDto](#)	CardDto object

GetCardsByUserId(Guid)

Get Cards by User Id.

Declaration

```
public List<CardDto> GetCardsByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
[Guid](#)	userId	User Id.

Returns

TYPE	DESCRIPTION
[List](#)<[CardDto](#)>	List of CardDto object

UpdateCard(CardDto, Guid)

Update card in data base.

Declaration

```
public RequestResult UpdateCard(CardDto cardDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
[CardDto](#)	cardDto	CardDto object
[Guid](#)	userId	User Id

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
RequestResult	RequestResult object

Implements

ICardCrud

Class CardModel

Card data model.

Syntax

```
public class CardModel
```

Properties

EncryptedCVV

Encrypted cvv.

Declaration

```
[Column("encrypted_cvv")]
public byte[] EncryptedCVV { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

EncryptedCardNumber

Encrypted card number.

Declaration

```
[Column("encrypted_card_number")]
public byte[] EncryptedCardNumber { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

EncryptedExpiryDate

Encrypted expiry date.

Declaration

```
[Column("encrypted_expiry_date")]
public byte[] EncryptedExpiryDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

Id

Card Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("card_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

User

User navigation property.

Declaration

```
[JsonIgnore]
public UserModel User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id.

Declaration

```
[Column("user_id")]
public Guid UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	
```

Namespace SoundParadise.Api.Models.Cart

Classes

[CartCrud](#)

[CartModel](#)

Cart data model.

Class CartCrud

Inheritance

object

CartCrud

Implements

ICartCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Cart

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartCrud : ICartCrud
```

Constructors

CartCrud(SoundParadiseDbContext, ILoggingService<CartCrud>, IProductCrud, ImageService)

CartCrud constructor.

Declaration

```
public CartCrud(SoundParadiseDbContext context, ILoggingService<CartCrud> loggingService, IProductCrud
productCrud, ImageService imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<CartCrud>	loggingService	Service for logging errors.
IProductCrud	productCrud	Product Crud.
ImageService	imageService	Service for image.

Methods

AddCartItem(Guid, Guid)

Get cart Id by user Id.

Declaration

```
public RequestResult AddCartItem(Guid cartId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.
Guid	productId	

Returns

TYPE	DESCRIPTION
RequestResult	List of CartItemModel objects

CartItemExists(Guid, out Guid)

Check CartItem exists.

Declaration

```
public bool CartItemExists(Guid cartItemId, out Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartItemId	CartItem Id.
Guid	cartId	

Returns

TYPE	DESCRIPTION
bool	True if exsists, false if not

CreateCart(CartModel)

Add cart to data base.

Declaration

```
public bool CreateCart(CartModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartModel	cart	CartModel object.

Returns

TYPE	DESCRIPTION
bool	True if add, false if not

DeleteCart(Guid)

Delete Cart from data base.

Declaration

```
public bool DeleteCart(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.

Returns

TYPE	DESCRIPTION
bool	True if delete, false if error.

GetAllCarts()

Get all carts.

Declaration

```
public List<CartModel> GetAllCarts()
```

Returns

TYPE	DESCRIPTION
List<CartModel>	List of CartModel objects

GetCartById(Guid)

Get cart by Id.

Declaration

```
public CartModel GetCartById(Guid Id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	Id	Cart Id.

Returns

TYPE	DESCRIPTION
CartModel	List of CartModel objects

GetCartByUserId(Guid)

Get cart by user Id.

Declaration

```
public List<CartItemDtoProduct> GetCartByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
List<CartItemDtoProduct>	List of CartItemDtoProduct objects

GetCartIdByUserId(Guid)

Get cart Id by user Id.

Declaration

```
public Guid GetCartIdByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
Guid	Cart Id

GetCartItems(Guid)

Get CartItems by cart Id.

Declaration

```
public List<CartItemModel> GetCartItems(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.

Returns

TYPE	DESCRIPTION
List<CartItemModel>	List of CartItemModel objects

RemoveCartItem(Guid, Guid)

Delete Cart from data base.

Declaration

```
public RequestResult RemoveCartItem(Guid cartId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object.

UpdateCart(CartModel)

Update Cart in data base.

Declaration

```
public bool UpdateCart(CartModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartModel	cart	

Returns

TYPE	DESCRIPTION
bool	List of CartItemModel objects

UpdateCartItemQuantity(Guid, Guid, short)

Update CartItem quantity in data base.

Declaration

```
public RequestResult UpdateCartItemQuantity(Guid cartId, Guid cartItemId, short quantity)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.
Guid	cartItemId	CartItem Id.
short	quantity	Quantity

Returns

TYPE	DESCRIPTION
RequestResult	List of CartItemModel objects

Implements

ICartCrud

Class CartModel

Cart data model.

Syntax

```
public class CartModel
```

Properties

CartItems

CartItems navigation property.

Declaration

```
public List<CartItemModel>? CartItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CartItemModel>	

Id

Cart Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("cart_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Order

Order navigation property.

```
[JsonIgnore]
public OrderModel? Order { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderModel	

OrderId

Order Id.

Declaration

```
[Column("order_id")]
public Guid? OrderId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

User

User navigation property.

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id.

Declaration

```
[Column("user_id")]
public Guid UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Api.Models.CartItem

Classes

CartItemCrud

CartItemModel CRUD.

CartItemModel

CartItem data model.

Class CartItemCrud

CartItemModel CRUD.

Inheritance

object

CartItemCrud

Implements

ICartItemCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.CartItem

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartItemCrud : ICartItemCrud
```

Constructors

CartItemCrud(SoundParadiseDbContext, ILoggingService<CartItemCrud>, IProductCrud)

CartItemCrud constructor.

Declaration

```
public CartItemCrud(SoundParadiseDbContext context, ILoggingService<CartItemCrud> loggingService, IProductCrud
productCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<CartItemCrud>	loggingService	Service for logging errors.
IProductCrud	productCrud	

Methods

CreateCartItem(CartItemModel)

Add new CartItem to data base.

Declaration

```
public bool CreateCartItem(CartItemModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartItemModel	cart	Db context.

Returns

TYPE	DESCRIPTION
bool	True if add, false if error

DeleteCartItem(Guid)

Delete cartitem from data base.

Declaration

```
public bool DeleteCartItem(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	CartItem Id.

Returns

TYPE	DESCRIPTION
bool	True if delete, false if not

GetAllCartItems()

Get all cartitems.

Declaration

```
public List<CartItemModel> GetAllCartItems()
```

Returns

TYPE	DESCRIPTION
List<CartItemModel>	List of CartItem models.

GetCartItemByCartId(Guid)

Get CartItem by Cart Id.

Declaration

```
public CartItemModel GetCartItemByCartId(Guid cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	cartId	Cart Id.

Returns

TYPE	DESCRIPTION
CartItemModel	CartItem model.

GetCartItemById(Guid)

Get CartItem by Id.

Declaration

```
public CartItemModel? GetCartItemById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	CartItem Id.

Returns

TYPE	DESCRIPTION
CartItemModel	CartItem model.

GetCartItemsByCartId(Guid?)

Get CartItems by Cart Id.

Declaration

```
public List<CartItemModel> GetCartItemsByCartId(Guid? cartId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	cartId	Cart Id.

Returns

TYPE	DESCRIPTION
List<CartItemModel>	List of CartItem model.

GetTotalPriceOfOrder(IEnumerable<CartItemDto>)

Get total price of order.

Declaration

```
public decimal GetTotalPriceOfOrder(IEnumerable<CartItemDto> cartItems)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable<CartItemDto>	cartItems	CartItemDto object.

Returns

TYPE	DESCRIPTION
decimal	Total price

GetTotalPriceOfOrderAuth(IEnumerable<CartItemModel>)

Get total price of order auth.

Declaration

```
public decimal GetTotalPriceOfOrderAuth(IEnumerable<CartItemModel> cartItems)
```

Parameters

TYPE	NAME	DESCRIPTION
IEnumerable<CartItemModel>	cartItems	CartItemDto object.

Returns

TYPE	DESCRIPTION
decimal	Total price

UpdateCartItem(CartItemModel)

Update cartitem in data base.

Declaration

```
public bool UpdateCartItem(CartItemModel cart)
```

Parameters

TYPE	NAME	DESCRIPTION
CartItemModel	cart	CartModel object.

Returns

TYPE	DESCRIPTION
bool	True if update, false if not

Implements

ICartItemCrud

Class CartItemModel

CartItem data model.

Inheritance

object

CartItemModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.CartItem

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartItemModel
```

Properties

Cart

Cart navigation property.

Declaration

```
[JsonIgnore]
public CartModel? Cart { get; set; }
```

Property Value

TYPE	DESCRIPTION
CartModel	

CartId

Cart Id.

Declaration

```
[Column("cart_id")]
public Guid? CartId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Id

CartItem Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("cart_item_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Order

Order navigation property.

Declaration

```
[JsonIgnore]
public OrderModel? Order { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderModel	

OrderId

Order Id.

Declaration

```
[Column("order_id")]
public Guid? OrderId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Product

Product navigation property.

Declaration

```
public ProductModel? Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductModel	

ProductId

Product Id.

Declaration

```
[Column("product_id")]
public Guid ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Quantity

Quantity.

Declaration

```
[Column("quantity")]
public short Quantity { get; set; }
```

Property Value

TYPE	DESCRIPTION
short	

Namespace SoundParadise.Api.Models.Category

Classes

CategoryCrud

CategoryModel CRUD.

CategoryModel

Category data model.

Class CategoryCrud

CategoryModel CRUD.

Inheritance

object

CategoryCrud

Implements

ICategoryCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Category

Assembly: SoundParadise.Api.dll

Syntax

```
public class CategoryCrud : ICategoryCrud
```

Constructors

CategoryCrud(SoundParadiseDbContext, ILoggingService<CategoryCrud>, ImageService)

CategoryCrud constructor.

Declaration

```
public CategoryCrud(SoundParadiseDbContext context, ILoggingService<CategoryCrud> loggingService, ImageService
imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<CategoryCrud>	loggingService	Service for logging errors.
ImageService	imageService	Service for images.

Methods

CreateCategory(CategoryModel)

Add category in data base.

Declaration

```
public bool CreateCategory(CategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
CategoryModel	category	Category model.

Returns

TYPE	DESCRIPTION
bool	True if create, false if not

DeleteCategory(Guid)

Delete category from data base.

Declaration

```
public bool DeleteCategory(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	Category Id.

Returns

TYPE	DESCRIPTION
bool	True if delete, false if not

GetAllCategories()

Get all categories from data base.

Declaration

```
public List<CategoryDto> GetAllCategories()
```

Returns

TYPE	DESCRIPTION
List<CategoryDto>	List CategoryDto

GetCategoryById(Guid)

Get category by Id.

Declaration

```
public CategoryModel? GetCategoryById(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	Category Id

Returns

TYPE	DESCRIPTION
CategoryModel	Category model

GetCategoryByName(string)

Get category by name.

Declaration

```
public CategoryModel? GetCategoryByName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	Category name.

Returns

TYPE	DESCRIPTION
CategoryModel	Category model

GetCategoryNameById(Guid)

Get category name by Id.

Declaration

```
public string GetCategoryNameById(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	Category Id.

Returns

TYPE	DESCRIPTION
string	Name string

GetSubcategoriesOfCategory(Guid)

Get subcategory of category.

Declaration

```
public List<SubcategoryDto> GetSubcategoriesOfCategory(Guid categoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	categoryId	CategoryId

Returns

TYPE	DESCRIPTION
List<SubcategoryDto>	List SubcategoryDto

GetSubcategoryNameById(Guid)

Get subcategory name by Id.

Declaration

```
public string GetSubcategoryNameById(Guid subcategoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	subcategoryId	Subcategory Id.

Returns

TYPE	DESCRIPTION
string	name string

Implements

ICategoryCrud

Class CategoryModel

Category data model.

[object](#)

CategoryModel

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: [SoundParadise.Api.Models.Category](#)

Assembly: SoundParadise.Api.dll

Syntax

```
public class CategoryModel
```

Properties

Id

Category Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("category_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
[Guid](#)	

Image

Image navigation property.

Declaration

```
[JsonIgnore]
public ImageModel? Image { get; set; }
```

Property Value

TYPE	DESCRIPTION
[ImageModel](#)	

ImageId

Image Id.

```
[Column("image_id")]
public Guid? ImageId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Name

Category name.

Declaration

```
[Column("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Subcategories

Subcategories navigation property.

Declaration

```
[JsonIgnore]
public List<SubcategoryModel>? Subcategories { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<SubcategoryModel>	

Namespace SoundParadise.Api.Models.Comment

Classes

CommentCrud

CommentModel CRUD.

CommentModel

Comment data model

Class CommentCrud

CommentModel CRUD.

Inheritance

object

CommentCrud

Implements

ICommentCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Comment

Assembly: SoundParadise.Api.dll

Syntax

```
public class CommentCrud : ICommentCrud
```

Constructors

CommentCrud(SoundParadiseDbContext, ILoggingService<CommentCrud>)

CommentCrud constructor.

Declaration

```
public CommentCrud(SoundParadiseDbContext context, ILoggingService<CommentCrud> loggingService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<CommentCrud>	loggingService	Service for logging errors.

Methods

CreateComment(CommentModel)

Create comment in data base.

Declaration

```
public bool CreateComment(CommentModel comment)
```

Parameters

TYPE	NAME	DESCRIPTION
CommentModel	comment	Comment bodel.

Returns

TYPE	DESCRIPTION
bool	True if create, false if not.

DeleteComment(Guid)

Delete comment in data base.

Declaration

```
public bool DeleteComment(Guid commentId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	commentId	Comment Id.

Returns

TYPE	DESCRIPTION
bool	True if create, false if not.

GetAllComments()

Get all comments.

Declaration

```
public List<CommentModel> GetAllComments()
```

Returns

TYPE	DESCRIPTION
List<CommentModel>	List of comment model

GetCommentByProductId(Guid)

Get comment by product Id.

Declaration

```
public CommentModel GetCommentByProductId(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id

Returns

TYPE	DESCRIPTION
CommentModel	Comment model

GetCommentByUserId(Guid)

Get comment by user Id.

Declaration

```
public CommentModel GetCommentByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
CommentModel	Comment model

GetCommentsCount(Guid)

Get comment count.

Declaration

```
public int GetCommentsCount(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id

Returns

TYPE	DESCRIPTION
int	Comment model

Implements

ICommentCrud

Class CommentModel

Comment data model

Inheritance

object

CommentModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Comment

Assembly: SoundParadise.Api.dll

Syntax

```
public class CommentModel
```

Properties

Content

Content

Declaration

```
[Column("content")]
public string Content { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CreatedAt

Create date

Declaration

```
[Column("created_at")]
public DateTime CreatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

DeletedAt

Delete date

Declaration

```
[Column("deleted_at")]
public DateTime? DeletedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

Id

Comment Id

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("comment_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

IsApproved

Is approved status

Declaration

```
[Column("is_approved")]
public bool IsApproved { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsBlocked

Is blocked status

Declaration

```
[Column("is_blocked")]
public bool IsBlocked { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsDeleted

Is delete status

Declaration
```

```
[Column("is_deleted")]
public bool IsDeleted { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsEdited

Is edited status

Declaration

```
[Column("is_edited")]
public bool IsEdited { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Product

Product navigation property

Declaration

```
[JsonIgnore]
public ProductModel? Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductModel	

ProductId

Product Id

Declaration

```
[Column("product_id")]
public Guid ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

UpdatedAt

Update date

Declaration

```
[Column("updated_at")]
public DateTime? UpdatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

User

User navigation property

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id

Declaration

```
[Column("user_id")]
public Guid UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Api.Models.Enums

Enums

DeliveryOptionEnum

Delivery Option enum

DeliveryTypeEnum

Delivery type enum

OrderStatusEnum

Order status enum

PaymentProvider

Payment provider enum

PaymentTypeEnum

Payment type enum

SortOptionsEnum

Sort option enum

UserRoleEnum

User role enum

Enum DeliveryOptionEnum

Delivery Option enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum DeliveryOptionEnum
```

Fields

NAME	DESCRIPTION
NovaPoshta	
UkrPoshta	

Enum DeliveryTypeEnum

Delivery type enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum DeliveryTypeEnum
```

Fields

NAME	DESCRIPTION
Courier	
Department	
ParcelLocker	

Enum OrderStatusEnum

Order status enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum OrderStatusEnum
```

Fields

NAME	DESCRIPTION
Canceled	
Delivered	
Pending	
Processing	
Shipped	

Enum PaymentProvider

Payment provider enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum PaymentProvider
```

Fields

NAME	DESCRIPTION
Fondy	
LiqPay	

Enum PaymentTypeEnum

Payment type enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum PaymentTypeEnum
```

Fields

NAME	DESCRIPTION
CardPayment	
CashOnDelivery	

Enum SortOptionsEnum

Sort option enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum SortOptionsEnum
```

Fields

NAME	DESCRIPTION
NameAsc	
NameDesc	
PriceAsc	
PriceDesc	

Enum UserRoleEnum

User role enum

Namespace: SoundParadise.Api.Models.Enums

Assembly: SoundParadise.Api.dll

Syntax

```
[JsonConverter(typeof(StringEnumConverter))]
public enum UserRoleEnum
```

Fields

NAME	DESCRIPTION
Admin	
Customer	
Seller	

Namespace SoundParadise.Api.Models.Image

Classes

ImageModel

Image data model.

Class ImageModel

Image data model.

Namespace: SoundParadise.Api.Models.Image

Assembly: SoundParadise.Api.dll

Syntax

```
public class ImageModel
```

Properties

Category

Category navigation property.

Declaration

```
[JsonIgnore]
public CategoryModel? Category { get; set; }
```

Property Value

TYPE	DESCRIPTION
CategoryModel	

CategoryId

Category Id.

Declaration

```
[Column("category_id")]
public Guid? CategoryId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Id

Image Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("image_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Path

Path.

Declaration

```
[Column("image_path")]
public string Path { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Product

Product navigation property.

Declaration

```
[JsonIgnore]
public ProductModel? Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductModel	

ProductId

Product Id.

Declaration

```
[Column("product_id")]
public Guid? ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Subcategory

Subcategory navigation property.

Declaration

```
[JsonIgnore]
public SubcategoryModel? Subcategory { get; set; }
```

Property Value

TYPE	DESCRIPTION
SubcategoryModel	

SubcategoryId

Subcategory Id.

Declaration

```
[Column("subcategory_id")]
public Guid? SubcategoryId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

User

User navigation property.

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

Declaration

```
[Column("user_id")]
public Guid? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	
```

Namespace SoundParadise.Api.Models.Log

Classes

LogCrud

LogModel CRUD.

LogModel

Log data model.

Class LogCrud

LogModel CRUD.

Inheritance

object

LogCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Log

Assembly: SoundParadise.Api.dll

Syntax

```
public class LogCrud
```

Constructors

LogCrud(SoundParadiseDbContext)

Declaration

```
public LogCrud(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	

Class LogModel

Log data model.

Inheritance

object

LogModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Log

Assembly: SoundParadise.Api.dll

Syntax

```
public class LogModel
```

Properties

Application

Log Application.

Declaration

```
[Column("application")]
public string Application { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Exception

Exception.

Declaration

```
[Column("exception")]
public string Exception { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Log Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("log_id")]
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Level

Error level.

Declaration

```
[Column("level")]
public string Level { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Logged

Logged string.

Declaration

```
[Column("logged")]
public string Logged { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Message

Message string.

Declaration

```
[Column("message")]
public string Message { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Models.Order

Classes

OrderCrud

OrderModel CRUD.

OrderModel

Order data model.

Class OrderCrud

OrderModel CRUD.

Inheritance

object

OrderCrud

Implements

IOrderCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Order

Assembly: SoundParadise.Api.dll

Syntax

```
public class OrderCrud : IOrderCrud
```

Constructors

OrderCrud(SoundParadiseDbContext, ILoggingService<OrderCrud>, ICartItemCrud, IPaymentService, ICommentCrud, ImageService, IAddressCrud)

OrderCrud Constructor.

Declaration

```
public OrderCrud(SoundParadiseDbContext context, ILoggingService<OrderCrud> loggingService, ICartItemCrud
cartItemCrud, IPaymentService paymentService, ICommentCrud commentCrud, ImageService imageService,
IAddressCrud addressCrud)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context
ILoggingService<OrderCrud>	loggingService	Service for logging errors
ICartItemCrud	cartItemCrud	CartItemCrud
IPaymentService	paymentService	Payment service
ICommentCrud	commentCrud	Comment crud

TYPE	NAME	DESCRIPTION
[ImageService](#)	imageService	Image service
[IAddressCrud](#)	addressCrud	AddressCrud

Methods

ConfirmOrder(dynamic)

Confirm order.

Declaration

```
public RequestResult ConfirmOrder(dynamic response)
```

Parameters

TYPE	NAME	DESCRIPTION
dynamic	response	Response

Returns

TYPE	DESCRIPTION
[RequestResult](#)	RequestResult object

CreateOrder(OrderModel)

Create order.

Declaration

```
public RequestResult CreateOrder(OrderModel order)
```

Parameters

TYPE	NAME	DESCRIPTION
[OrderModel](#)	order	Order Model

Returns

TYPE	DESCRIPTION
[RequestResult](#)	RequestResult object

CreateOrderAuth(OrderDto)

Create order auth.

Declaration

```
public CreateOrderResult CreateOrderAuth(OrderDto order)
```

Parameters

TYPE	NAME	DESCRIPTION
OrderDto	order	OrderDto

Returns

TYPE	DESCRIPTION
CreateOrderResult	CreateOrderResult object

DeleteComment(Guid)

OrderCrud Constructor.

Declaration

```
public bool DeleteComment(Guid orderId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	Order Id

Returns

TYPE	DESCRIPTION
bool	True if succes delete, false if not

GetAllOrders()

Get all orders.

Declaration

```
public List<OrderModel> GetAllOrders()
```

Returns

TYPE	DESCRIPTION
List<OrderModel>	List OrderModel object

GetOrderById(Guid, Guid)

Get orders by Id.

Declaration

```
public OrderDto GetOrderById(Guid orderId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	orderId	Order Id
Guid	userId	User Id

Returns

TYPE	DESCRIPTION
OrderDto	OrderModel object

GetOrdersByUserId(Guid)

Get orders by user id.

Declaration

```
public List<OrderHistoryDto> GetOrdersByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id

Returns

TYPE	DESCRIPTION
List<OrderHistoryDto>	List of OrderHistoryDto object

Implements

IOrderCrud

Class OrderModel

Order data model.

object
OrderModel

Inherited Members

object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: SoundParadise.Api.Models.Order

Assembly: SoundParadise.Api.dll

Syntax

```
public class OrderModel
```

Properties

Address

Delivery Address navigation property.

Declaration

```
[JsonIgnore]
public AddressModel Address { get; set; }
```

Property Value

TYPE	DESCRIPTION
AddressModel	

CartItems

Navigation property for List CartItemModel.

Declaration

```
[JsonIgnore]
public List<CartItemModel>? CartItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CartItemModel>	

CheckoutUrl

Declaration

```
[Column("checkout_url")]
public string? CheckoutUrl { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Comment

Comment.

Declaration

```
[Column("comment")]
public string? Comment { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CustomerName

Customer name.

Declaration

```
[Column("customer_name")]
public string CustomerName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CustomerSurname

Customer Surname.

Declaration

```
[Column("customer_surname")]
public string CustomerSurname { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DeliveryAddressId

Delivery Address Id.

Declaration

```
[Column("delivery_address_id")]
public Guid DeliveryAddressId { get; set; }
```

TYPE	DESCRIPTION
Guid	

DeliveryOption

Delivery Enum option.

Declaration

```
[Column("delivery_option")]
public DeliveryOptionEnum DeliveryOption { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryOptionEnum	

DeliveryType

Delivery Enum type.

Declaration

```
[Column("delivery_type")]
public DeliveryTypeEnum DeliveryType { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryTypeEnum	

Id

Order Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("order_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

IsPaid

Is paid status.

Declaration

```
[Column("is_paid")]
public bool IsPaid { get; set; }
```

TYPE	DESCRIPTION
bool	

OrderDate

Order date.

Declaration

```
[Column("order_date")]
public DateTime OrderDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

OrderStatus

Order status enum.

Declaration

```
[Column("order_status")]
public OrderStatusEnum OrderStatus { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderStatusEnum	

Payment

Payment navigation property.

Declaration

```
[JsonIgnore]
public PaymentModel? Payment { get; set; }
```

Property Value

TYPE	DESCRIPTION
PaymentModel	

PaymentId

Payment Id.

Declaration

```
[Column("payment_id")]
public Guid? PaymentId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

PaymentType

Payment Enum type.

Declaration

```
[Column("payment_type")]
public PaymentTypeEnum PaymentType { get; set; }
```

Property Value

TYPE	DESCRIPTION
PaymentTypeEnum	

PhoneNumber

Customer phone number.

Declaration

```
[Column("customer_phone_number")]
[RegularExpression("^[0-9]*$", ErrorMessage = "Phone number must contain only numeric characters.")]
public string PhoneNumber { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TotalPrice

Total price.

Declaration

```
[Column("total_price")]
public decimal TotalPrice { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

User

User navigation property.

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id.

Declaration

```
[Column("user_id")]
public Guid? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

Namespace SoundParadise.Api.Models.PaymentDetails

Classes

PaymentModel

Payment data model.

Class PaymentModel

Payment data model.

Syntax

```
public class PaymentModel
```

Properties

Amount

Amount.

Declaration

```
[Column("amount")]
public decimal Amount { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

CardType

Card Type.

Declaration

```
[Column("card_type")]
public string CardType { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Currency

Currency.

Declaration

```
[Column("currency")]
public string Currency { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Payment Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("payment_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

MaskedCard

Masked card string.

Declaration

```
[Column("masked_card_number")]
public string MaskedCard { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Order

Order navigation property.

Declaration

```
[JsonIgnore]
public OrderModel Order { get; set; }
```

Property Value

TYPE	DESCRIPTION
OrderModel	

OrderId

Order Id.

Declaration
```

```
[Column("order_id")]
public Guid OrderId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

TransactionId

Transaction Id.

Declaration

```
[Column("transaction_id")]
public int TransactionId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Namespace SoundParadise.Api.Models.Product

Classes

ProductCrud

ProductModel CRUD.

ProductModel

Product data model.

Class ProductCrud

ProductModel CRUD.

Inheritance

object
ProductCrud

Implements

IProductCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Product

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductCrud : IProductCrud
```

Constructors

ProductCrud(SoundParadiseDbContext, ILoggingService<ProductCrud>, ImageService)

ProductCrud constructor.

Declaration

```
public ProductCrud(SoundParadiseDbContext context, ILoggingService<ProductCrud> loggingService, ImageService
imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context
ILoggingService<ProductCrud>	loggingService	Servuce for logging errors
ImageService	imageService	Image service

Methods

CreateComment(CommentModel)

Add comment to product.

Declaration

```
public CommentModel CreateComment(CommentModel comment)
```

Parameters

TYPE	NAME	DESCRIPTION
CommentModel	comment	Comment model.

Returns

TYPE	DESCRIPTION
CommentModel	New comment

CreateProduct(ProductModel)

Add new product in data base.

Declaration

```
public ProductModel CreateProduct(ProductModel product)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductModel	product	Product model.

Returns

TYPE	DESCRIPTION
ProductModel	New product

DeleteImage(Guid, Guid)

Delete image by imageId and productId.

Declaration

```
public RequestResult DeleteImage(Guid imageId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	imageId	Image Id.
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequstResult object

DeleteProduct(Guid)

Delete product by Id.

Declaration

```
public RequestResult DeleteProduct(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequstResult object

GetAllProducts(SortOptionsEnum, int, int, Guid)

Get all products.

Declaration

```
public SearchProductResult GetAllProducts(SortOptionsEnum sort, int page, int pageSize, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
int	page	Page count.
int	pageSize	Page size
Guid	userId	User Id.Default value

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
SearchProductResult	SearchProductResult object

GetPopularProducts(int, int, Guid)

Get popular products.

Declaration

```
public List<ProductDto> GetPopularProducts(int page = 1, int limit = 3, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	Page count.Default 1.
int	limit	Product limit.Default 3.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
List<ProductDto>	List of ProductDto

GetProductById(Guid, Guid)

Get product by user Id.

Declaration

```
public ProductModel GetProductById(Guid productId, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
ProductModel	ProductModel object

GetProductBySlug(string, Guid)

Get product by slug.

Declaration

```
public ProductModel GetProductBySlug(string slug, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
string	slug	Slug string.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
ProductModel	ProductModel object

GetProductCartDtoById(Guid)

Get product cart by Id.

Declaration

```
public ProductCartDto GetProductCartDtoById(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
ProductCartDto	ProductCartDto object

GetProductsByCategoryId(SortOptionsEnum, Guid, int, int, Guid)

Get product by category Id.

Declaration

```
public SearchProductDtoResult GetProductsByCategoryId(SortOptionsEnum sort, Guid categoryId, int page = 1, int
pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
Guid	categoryId	Category Id.
int	page	Page count.Default 1
int	pageSize	Page count.Default 20
Guid	userId	User Id.Default value

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	SearchProductDtoResult object

GetProductsByCategoryName(SortOptionsEnum, string, int, int, Guid)

Get product by category name.

Declaration

```
public SearchProductDtoResult GetProductsByCategoryName(SortOptionsEnum sort, string categoryName, int page =
1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
string	categoryName	Category name.
int	page	Page count.Default 1
int	pageSize	Page count.Default 20

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.Default value

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	SearchProductDtoResult object

GetProductsBySubcategoryId(SortOptionsEnum, Guid, int, int, Guid)

Get product by subcategory Id.

Declaration

```
public SearchProductDtoResult GetProductsBySubcategoryId(SortOptionsEnum sort, Guid subcategoryId, int page =
1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
Guid	subcategoryId	Subcategory Id.
int	page	Page count.Default 1
int	pageSize	Page size.Default 20
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	SearchProductDtoResult object

GetProductsBySubcategoryName(SortOptionsEnum, string, int, int, Guid)

Get product by subcategory name.

Declaration

```
public SearchProductDtoResult GetProductsBySubcategoryName(SortOptionsEnum sort, string subcategoryName, int
page = 1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
string	subcategoryName	Subcategory name.
int	page	Page count.Default 1
int	pageSize	Page count.Default 20
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	SearchProductDtoResult object

GetProductsByUserId(Guid)

Get products by User Id.

Declaration

```
public SearchProductResult GetProductsByUserId(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	User Id.

Returns

TYPE	DESCRIPTION
SearchProductResult	SearchProductResult object

GetRecentBuys(int, int, Guid)

Get recent products.

Declaration

```
public List<ProductDto> GetRecentBuys(int page = 1, int limit = 3, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
int	page	Page count.Default 1.
int	limit	Product limit.Default 3.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
List<ProductDto>	List of ProductDto

GetSpecial0ffers(SortOptionsEnum, int, int, Guid)

Get special offers products.

Declaration

```
public List<ProductDto> GetSpecial0ffers(SortOptionsEnum sort, int page = 1, int limit = 3, Guid userId =
default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
int	page	Page count.Default 1.
int	limit	Product limit.Default 3.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
List<ProductDto>	List of ProductDto

IsProductInCart(Guid, Guid)

Check is product in cart.

Declaration

```
public bool IsProductInCart(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
bool	True if in cart. False if not in cart

IsProductInWishlist(Guid, Guid)

Check is product in wishlist.

Declaration

```
public bool IsProductInWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
bool	True if in wishlist. False if not in wishlist

IsQuantityValid(Guid, short)

Check quantity valid.

Declaration

```
public bool IsQuantityValid(Guid productId, short quantity)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.
short	quantity	Quantity.

Returns

TYPE	DESCRIPTION
bool	True if valid. False if not valid

ProductExists(Guid)

Check is product exists.

Declaration

```
public bool ProductExists(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product Id.

Returns

TYPE	DESCRIPTION
bool	True if product exists. False if not exists

SearchProducts(SortOptionsEnum, string, int, int, Guid)

Search ProductDtoResult.

Declaration

```
public SearchProductDtoResult SearchProducts(SortOptionsEnum sort, string query, int page = 1, int pageSize = 20, Guid userId = default)
```

Parameters

TYPE	NAME	DESCRIPTION
SortOptionsEnum	sort	Enum of sort option.
string	query	String query.

TYPE	NAME	DESCRIPTION
int	page	Result for page.Default 1 page.
int	pageSize	
Guid	userId	

Returns

TYPE	DESCRIPTION
SearchProductDtoResult	List SuggestionProductDto

SearchSuggestions(string, int)

Search suggestion.

Declaration

```
public List<SuggestionProductDto> SearchSuggestions(string query, int limit)
```

Parameters

TYPE	NAME	DESCRIPTION
string	query	String query.
int	limit	Count results.

Returns

TYPE	DESCRIPTION
List<SuggestionProductDto>	List SuggestionProductDto

UpdateProduct(ProductModel)

Update product in data base.

Declaration

```
public bool UpdateProduct(ProductModel productchange)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductModel	productchange	Product model object

Returns

TYPE	DESCRIPTION
bool	True if Ok, False if error

UploadImage(Guid, string)

Add image to product.

Declaration

```
public RequestResult UploadImage(Guid productId, string image)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	Product model Id.
string	image	Path to image.

Returns

TYPE	DESCRIPTION
RequestResult	RequstResult object

Implements

IProductCrud

Class ProductModel

Product data model.

Syntax

```
public class ProductModel
```

Fields

CommentsCount

Comments count.

Declaration

```
[NotMapped]
public int CommentsCount
```

Field Value

TYPE	DESCRIPTION
int	

Properties

CartItems

List of cart items, where is the product.

Declaration

```
[JsonIgnore]
public List<CartItemModel>? CartItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CartItemModel>	

Code

Product code.

Declaration

```
[Column("code")]
public int Code { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Color

Product color.

Declaration

```
[Column("color")]
public string Color { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Comments

List of comments about product.

Declaration

```
[JsonIgnore]
public List<CommentModel>? Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CommentModel>	

Country

Product country.

Declaration

```
[Column("country")]
public string Country { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CreatedAt

Date of product create.

Declaration

```
[Column("created_at")]
public DateTime CreatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

DeletedAt

Date of product delete.

Declaration

```
[Column("deleted_at")]
public DateTime? DeletedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

Description

Product description.

Declaration

```
[Column("description")]
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Discount

Discount sum.

Declaration

```
[Column("discount")]
public decimal Discount { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Height

Product height.

Declaration

```
[Column("height")]
public decimal Height { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Id

Product Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("product_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Images

List of product images.

Declaration

```
public List<ImageModel>? Images { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ImageModel>	

InCart

In cart status.

Declaration

```
[NotMapped]
public bool InCart { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

InStock

In stok status.

Declaration

```
[Column("in_stock")]
public bool InStock { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

InWishlist

In wishlist status.

Declaration

```
[NotMapped]
public bool InWishlist { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsDiscount

Discount status.

Declaration

```
[Column("is_discount")]
public bool IsDiscount { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsNew

Is new status.

Declaration

```
[Column("is_new")]
public bool IsNew { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Length

Product lenght.

Declaration

```
[Column("length")]
public decimal Length { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Manufacturer

Product manufacturer.

Declaration

```
[Column("manufacturer")]
public string Manufacturer { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Name

Product's name.

Declaration

```
[Column("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Parameters

List of parameters.

Declaration

```
[JsonIgnore]
public List<ProductParameterModel>? Parameters { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductParameterModel>	

Price

Product price.

Declaration

```
[Column("price")]
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Quantity

Product quantity.

Declaration

```
[Column("quantity")]
public short Quantity { get; set; }
```

Property Value

TYPE	DESCRIPTION
short	

Rating

Rating of product.

Declaration

```
[Column("rating")]
public decimal Rating { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

RatingCount

Rating count of product.

Declaration

```
[Column("rating_count")]
public int? RatingCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

Seller

Seller navigation property.

Declaration

```
[JsonIgnore]
public UserModel? Seller { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

SellerId

Seller Id.

Declaration

```
[Column("seller_id")]
public Guid SellerId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Slug

Slug's Id.

Declaration

```
[Column("slug_id")]
public string Slug { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Subcategory

Subcategory navigation property.

Declaration

```
[JsonIgnore]
public SubcategoryModel? Subcategory { get; set; }
```

Property Value

TYPE	DESCRIPTION
SubcategoryModel	

SubcategoryId

Subcategory Id.

Declaration

```
[Column("subcategory_id")]
public Guid? SubcategoryId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

UpdatedAt

Date of product update.

Declaration

```
[Column("updated_at")]
public DateTime? UpdatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

Weight

Product weight.

Declaration

```
[Column("weight")]
public decimal Weight { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Width

Product widtht.

Declaration

```
[Column("width")]
public decimal Width { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

WishlistProducts

List of WishListProduct objects.

Declaration

```
[JsonIgnore]
public List<WishlistProductsModel>? WishlistProducts { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<WishlistProductsModel>	

Namespace SoundParadise.Api.Models.ProductParameter

Classes

ProductParameterCrud

ProductParameter CRUD.

ProductParameterModel

ProductParameter data model.

Class ProductParameterCrud

ProductParameter CRUD.

Inheritance

object

ProductParameterCrud

Implements

IProductParameterCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.ProductParameter

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductParameterCrud : IProductParameterCrud
```

Constructors

ProductParameterCrud(SoundParadiseDbContext, ILoggingService<ProductParameterCrud>)

ProductParameter constructor.

Declaration

```
public ProductParameterCrud(SoundParadiseDbContext context, ILoggingService<ProductParameterCrud>
loggingService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<ProductParameterCrud>	loggingService	Service for logging errors.

Methods

CreateParameter(ProductParameterModel)

Add ProductParameter to data base.

Declaration

```
public bool CreateParameter(ProductParameterModel parameter)
```

Parameters

TYPE	NAME	DESCRIPTION
ProductParameterModel	parameter	ProductParameter model.

Returns

TYPE	DESCRIPTION
bool	True if Ok. False if error.

DeleteParameter(Guid)

Delete ProductParameter model from data base.

Declaration

```
public bool DeleteParameter(Guid parameterId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	parameterId	ProductParameter Id.

Returns

TYPE	DESCRIPTION
bool	True if Ok. False if error.

GetAllParameters()

Get all ProductParameter models from data base.

Declaration

```
public List<ProductParameterModel> GetAllParameters()
```

Returns

TYPE	DESCRIPTION
List<ProductParameterModel>	List of ProductParameter model.

GetParameterByProductId(Guid)

Get ProductParameter by Id.

Declaration

```
public List<ProductParameterModel> GetParameterByProductId(Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	productId	

Returns

TYPE	DESCRIPTION
List<ProductParameterModel>	List of ProductParameter model.

Implements

IProductParameterCrud

Class ProductParameterModel

ProductParameter data model.

Inheritance

[object](#)

ProductParameterModel

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Api.Models.ProductParameter

Assembly: SoundParadise.Api.dll

Syntax

```
public class ProductParameterModel
```

Properties

Id

ProductParameter Id.

Declaration

```
[Column("parameter_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
[Guid](#)	

ParameterName

Parameter's name.

Declaration

```
[Column("parameter_name")]
public string ParameterName { get; set; }
```

Property Value

TYPE	DESCRIPTION
[string](#)	

ParameterValue

Parameter's value.

Declaration

```
[Column("parameter_value")]
public string ParameterValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Product

Product's navigation property.

Declaration

```
[JsonIgnore]
public ProductModel? Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductModel	

ProductId

Product's Id.

Declaration

```
[Column("product_id")]
public Guid ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Api.Models.Subcategory

Classes

SubcategoryCrud

Subcategory CRUD.

SubcategoryModel

Subcategoty data model.

Class SubcategoryCrud

Subcategory CRUD.

Inheritance

object

SubcategoryCrud

Implements

ISubcategoryCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Subcategory

Assembly: SoundParadise.Api.dll

Syntax

```
public class SubcategoryCrud : ISubcategoryCrud
```

Constructors

SubcategoryCrud(SoundParadiseDbContext, ILoggingService<SubcategoryCrud>)

Subcategory constructor.

Declaration

```
public SubcategoryCrud(SoundParadiseDbContext context, ILoggingService<SubcategoryCrud> loggingService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.
ILoggingService<SubcategoryCrud>	loggingService	Service for logging errors.

Methods

CreateSubcategory(SubcategoryModel)

Declaration

```
public bool CreateSubcategory(SubcategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
SubcategoryModel	category	

Returns

TYPE	DESCRIPTION
bool	

DeleteSubcategory(Guid)

Delete Subcategory.

Declaration

```
public RequestResult DeleteSubcategory(Guid subcategoryId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	subcategoryId	Subcategory's Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

GetAllSubcategories()

Get all subcategory from data base.

Declaration

```
public List<SubcategoryModel> GetAllSubcategories()
```

Returns

TYPE	DESCRIPTION
List<SubcategoryModel>	List of SubcategoryModel object.

GetSubcategoryByCategory(CategoryModel)

Get Subcategory by category.

Declaration

```
public List<SubcategoryModel> GetSubcategoryByCategory(CategoryModel category)
```

Parameters

TYPE	NAME	DESCRIPTION
CategoryModel	category	Category model.

Returns

TYPE	DESCRIPTION
List<SubcategoryModel>	SubcategoryModel object.

GetSubcategoryById(Guid)

Get Subcategory by Id.

Declaration

```
public SubcategoryModel GetSubcategoryById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	Subcategory Id.

Returns

TYPE	DESCRIPTION
SubcategoryModel	SubcategoryModel object.

GetSubcategoryByName(string)

Get Subcategory by name.

Declaration

```
public SubcategoryModel GetSubcategoryByName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
string	name	Subcategory name.

Returns

TYPE	DESCRIPTION
SubcategoryModel	SubcategoryModel object.

Implements

ISubcategoryCrud

Class SubcategoryModel

Subcategoty data model.

Inheritance

object

SubcategoryModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Subcategory

Assembly: SoundParadise.Api.dll

Syntax

```
public class SubcategoryModel
```

Properties

Category

Category's navigation property.

Declaration

```
[JsonIgnore]
public CategoryModel? Category { get; set; }
```

Property Value

TYPE	DESCRIPTION
CategoryModel	

CategoryId

Category's Id.

Declaration

```
[Column("category_id")]
public Guid CategoryId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Id

Subcategoty's Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("subcategory_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Image

Image navigation property.

Declaration

```
public ImageModel? Image { get; set; }
```

Property Value

TYPE	DESCRIPTION
ImageModel	

ImageId

Image Id.

Declaration

```
[Column("image_id")]
public Guid ImageId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Name

Name of subcategory.

Declaration

```
[Column("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Products

Products navigation property.

Declaration

```
[JsonIgnore]
public List<ProductModel>? Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductModel>	

```
[JsonIgnore]
public List<ProductModel>? Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductModel>	

Namespace SoundParadise.Api.Models.Token

Classes

TokenCrud

Token CRUD.

TokenModel

Token data model.

Class TokenCrud

Token CRUD.

Inheritance

object

TokenCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Token

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenCrud
```

Constructors

TokenCrud(SoundParadiseDbContext)

Declaration

```
public TokenCrud(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	

Class TokenModel

Token data model.

Inheritance

object

TokenModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Token

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenModel
```

Properties

ExpirationDate

Expiration date.

Declaration

```
[Column("expiration_date")]
public DateTime ExpirationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Token Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("token_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Token

Token string.

Declaration

```
[Column("token")]
public string Token { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TokenJournals

TokenJournal navigation property.

Declaration

```
[JsonIgnore]
public List<TokenJournalModel>? TokenJournals { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<TokenJournalModel>	

Namespace SoundParadise.Api.Models.TokenJournal

Classes

TokenJournalCrud

TokenJournal CRUD.

TokenJournalModel

TokenJournal data model.

Class TokenJournalCrud

TokenJournal CRUD.

Inheritance

object

TokenJournalCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.TokenJournal

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenJournalCrud
```

Constructors

TokenJournalCrud(SoundParadiseDbContext)

Declaration

```
public TokenJournalCrud(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	

Class TokenJournalModel

TokenJournal data model.

Inheritance

object

TokenJournalModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.TokenJournal

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenJournalModel
```

Properties

ActivatedAt

Activated date.

Declaration

```
[Column("activated_at")]
public DateTime ActivatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

DeactivatedAt

Deactivated date.

Declaration

```
[Column("deactivated_at")]
public DateTime DeactivatedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

TokenJournal's Id.

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("token_journal_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

IsActive

Active status.

Declaration

```
[Column("is_active")]
public bool IsActive { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Token

Token's navigation property.

Declaration

```
[JsonIgnore]
public TokenModel? Token { get; set; }
```

Property Value

TYPE	DESCRIPTION
TokenModel	

TokenId

Token's Id.

Declaration

```
[Column("token_id")]
public Guid TokenId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

User

User's navigation property.

Declaration
```

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User's Id.

Declaration

```
[Column("user_id")]
public Guid UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Api.Models.User

Classes

UserCrud

User CRUD

UserModel

User data model

Class UserCrud

User CRUD

Inheritance

object

UserCrud

Implements

IUserCrud

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserCrud : IUserCrud
```

Constructors

UserCrud(SoundParadiseDbContext, HashingService, TokenService, SmtpService, ILoggingService<UserCrud>, IUrlHelperFactory, IActionContextAccessor, ImageService)

User CRUD constructor

Declaration

```
public UserCrud(SoundParadiseDbContext context, HashingService hashingService, TokenService tokenService,
SmtpService smtpService, ILoggingService<UserCrud> loggingService, IUrlHelperFactory urlHelperFactory,
IActionContextAccessor actionContextAccessor, ImageService imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	
HashingService	hashingService	
TokenService	tokenService	
SmtpService	smtpService	
ILoggingService<UserCrud>	loggingService	
IUrlHelperFactory	urlHelperFactory	
IActionContextAccessor	actionContextAccessor	
ImageService	imageService	

Methods

AuthenticateUser(UserAuthenticationDto)

User authorizationм

Declaration

```
public RequestResult AuthenticateUser(UserAuthenticationDto authenticationDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserAuthenticationDto	authenticationDto	User'datas

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

ConfirmUser(Guid, string)

Confirm user

Declaration

```
public RequestResult ConfirmUser(Guid userId, string token)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User's Id
string	token	User's token

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

CreateUser(UserCreateDto)

Add new user to data base

Declaration

```
public bool CreateUser(UserCreateDto userDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userDto	Contains user data

Returns

TYPE	DESCRIPTION
bool	True if Ok, false if error

DeleteUser(Guid)

Declaration

```
public RequestResult DeleteUser(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	

Returns

TYPE	DESCRIPTION
RequestResult	

GenerateConfirmationLink(UserModel)

Generate link to confirm user

Declaration

```
public string GenerateConfirmationLink(UserModel user)
```

Parameters

TYPE	NAME	DESCRIPTION
UserModel	user	UserModel object

Returns

TYPE	DESCRIPTION
string	String object

GenerateUniqueToken()

Generate unique token

Declaration

```
public string GenerateUniqueToken()
```

Returns

TYPE	DESCRIPTION
string	String object

GetAllUsers()

Get list of all users

Declaration

```
public List<UserModel> GetAllUsers()
```

Returns

TYPE	DESCRIPTION
List<UserModel>	List of users

GetUserByCredentials(UserAuthenticationDto)

Get user by credentials

Declaration

```
public UserModel? GetUserByCredentials(UserAuthenticationDto authenticationDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserAuthenticationDto	authenticationDto	User credentials

Returns

TYPE	DESCRIPTION
UserModel	UserModel object

GetUserById(Guid)

Get user by Id

Declaration

```
public UserModel GetUserById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	User Id

Returns

TYPE	DESCRIPTION
UserModel	UserModel object

GetUserByToken(string)

Get user by token

Declaration

```
public UserModel GetUserByToken(string token)
```

Parameters

TYPE	NAME	DESCRIPTION
string	token	User token

Returns

TYPE	DESCRIPTION
UserModel	UserModel object

GetUserDtoById(Guid)

Get UserDto by Id

Declaration

```
public UserDto GetUserDtoById(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
UserDto	

Logout(string)

Logout user

Declaration

```
public RequestResult Logout(string token)
```

Parameters

TYPE	NAME	DESCRIPTION
string	token	Token

Returns

TYPE	DESCRIPTION
RequestResult	String object

RegisterUser(UserCreateDto, List<CartItemDto>)

Registers a new user

Declaration

```
public UserModel RegisterUser(UserCreateDto userDto, List<CartItemDto> cartItemDtos)
```

Parameters

TYPE	NAME	DESCRIPTION
UserCreateDto	userDto	User data
List<CartItemDto>	cartItemDtos	User's products in cart

Returns

TYPE	DESCRIPTION
UserModel	UserModel object

UpdateUser(UserDto)

Update user in data base

Declaration

```
public RequestResult UpdateUser(UserDto userDto)
```

Parameters

TYPE	NAME	DESCRIPTION
UserDto	userDto	Contains user data

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
RequestResult	True if Ok, false if error

UploadAvatar(IFormFile, UserModel)

Declaration

```
public RequestResult UploadAvatar(IFormFile file, UserModel user)
```

Parameters

TYPE	NAME	DESCRIPTION
IFormFile	file	
UserModel	user	

Returns

TYPE	DESCRIPTION
RequestResult	

Implements

IUserCrud

Class UserModel

User data model

Inheritance

object

UserModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserModel
```

Properties

Cards

�ards for online payment

Declaration

```
[JsonIgnore]
public List<CardModel>? Cards { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CardModel>	

Cart

User's cart navigation property

Declaration

```
[JsonIgnore]
public CartModel? Cart { get; set; }
```

Property Value

TYPE	DESCRIPTION
CartModel	

CartId

User's cart id

Declaration

```
[Column("cart_id")]
public Guid CartId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Comments

Navigation property for user's comments

Declaration

```
[JsonIgnore]
public List<CommentModel>? Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CommentModel>	

ConfirmationToken

Token that is issued after confirmation of mail

Declaration

```
[Column("confirmation_token")]
public string? ConfirmationToken { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DeliveryAddresses

Address where user orders are delivered

Declaration

```
[JsonIgnore]
public List<AddressModel>? DeliveryAddresses { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<AddressModel>	

Email

User email, need to confirm account

Declaration

```
[Column("email")]
public string Email { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

User ID

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("user_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Image

Declaration

```
[JsonIgnore]
public ImageModel? Image { get; set; }
```

Property Value

TYPE	DESCRIPTION
ImageModel	

ImageId

Declaration

```
public Guid? ImageId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid?	

IsConfirmed

�onfirmation status

Declaration

```
[Column("is_confirmed")]
public bool IsConfirmed { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Name

Real name of user

Declaration

```
[Column("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Orders

Orders that belong to the user

Declaration

```
[JsonIgnore]
public List<OrderModel>? Orders { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<OrderModel>	

PasswordHash

Password after hashing

Declaration

```
[Column("password_hash")]
public byte[] PasswordHash { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

PasswordSalt

Salt for password hashing

Declaration

```
[Column("password_salt")]
public byte[] PasswordSalt { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

PhoneNumber

Phone number, need to contact with user

Declaration

```
[Column("phone_number")]
[RegularExpression("^[0-9]*$", ErrorMessage = "Phone number must contain only numeric characters.")]
public string PhoneNumber { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Products

Navigation property for products the user owns

Declaration

```
[JsonIgnore]
public List<ProductModel>? Products { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ProductModel>	

Role

User role, can be admin, seller, default user

Declaration

```
[JsonConverter(typeof(StringEnumConverter))]
[Column("role")]
public UserRoleEnum Role { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserRoleEnum	

Surname

Real surname of user

Declaration

```
[Column("surname")]
public string Surname { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TokenJournals

Tokens that were issued and are issued to the user

Declaration

```
[JsonIgnore]
public List<TokenJournalModel>? TokenJournals { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<TokenJournalModel>	

Username

User name

Declaration

```
[Column("username")]
public string Username { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Wishlist

User's wishlist navigation property

Declaration

```
[JsonIgnore]
public WishlistModel? Wishlist { get; set; }
```

Property Value

TYPE	DESCRIPTION
WishlistModel	

WishlistId

User's wishlist id

Declaration

```
[Column("wishlist_id")]
public Guid WishlistId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Methods

Confirm()

Declaration

```
public void Confirm()
```

Namespace SoundParadise.Api.Models.Wishlist

Classes

WishlistCrud

Whislist CRUD.

WishlistModel

Whislist data model

WishlistProductsModel

WhislistProuct data model

Class WishlistCrud

Whislist CRUD.

Inheritance

[object](#)

WishlistCrud

Implements

[IWishlistCrud](#)

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Api.Models.Wishlist

Assembly: SoundParadise.Api.dll

Syntax

```
public class WishlistCrud : IWishlistCrud
```

Constructors

WishlistCrud(SoundParadiseDbContext, ILoggingService<WishlistCrud>, IProductCrud, ICommentCrud, ImageService)

Constructor for WishlistCrus.

Declaration

```
public WishlistCrud(SoundParadiseDbContext context, ILoggingService<WishlistCrud> loggingService, IProductCrud
productCrud, ICommentCrud commentCrud, ImageService imageService)
```

Parameters

TYPE	NAME	DESCRIPTION
[SoundParadiseDbContext](#)	context	Db context.
[ILoggingService](#)<[WishlistCrud](#)>	loggingService	Service for logging errors.
[IProductCrud](#)	productCrud	Product crud.
[ICommentCrud](#)	commentCrud	Comment crud.
[ImageService](#)	imageService	Image service.

Methods

AddToWishlist(Guid, Guid)

Add product to whishlist

Declaration

```
public RequestResult AddToWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id
Guid	productId	Product Id

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

CreateWishlist(WishlistModel)

Add new Whishlist to data base

Declaration

```
public RequestResult CreateWishlist(WishlistModel wishlist)
```

Parameters

TYPE	NAME	DESCRIPTION
WishlistModel	wishlist	Whishlist data model

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

GetWishlistById(Guid)

Get Whishlist from data base by Id

Declaration

```
public WishlistModel GetWishlistById(Guid id)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	id	Whishkist Id

Returns

TYPE	DESCRIPTION
WishlistModel	WhishlistModel object

GetWishlistByUserId(Guid)

Get Whishlist from data base by user Id

Declaration

```
public List<ProductDto> GetWishlistByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User's Id

Returns

TYPE	DESCRIPTION
List<ProductDto>	List of ProductDto

RemoveFromWishlist(Guid, Guid)

Remove product from whishlist

Declaration

```
public RequestResult RemoveFromWishlist(Guid userId, Guid productId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	User Id
Guid	productId	Product Id

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
RequestResult	RequestResult object

Implements

IWishlistCrud

Class WishlistModel

Whislist data model

Inheritance

object

WishlistModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Wishlist

Assembly: SoundParadise.Api.dll

Syntax

```
public class WishlistModel
```

Properties

Id

Whislist's Id

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("wishlist_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

User

User navigation property

Declaration

```
[JsonIgnore]
public UserModel? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserModel	

UserId

User Id

Declaration

```
[Column("user_id")]
public Guid UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

WishlistProducts

Navigation property for whislist products

Declaration

```
public List<WishlistProductsModel>? WishlistProducts { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<WishlistProductsModel>	

Class WishlistProductsModel

WhislistProuct data model

Inheritance

object

WishlistProductsModel

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Models.Wishlist

Assembly: SoundParadise.Api.dll

Syntax

```
public class WishlistProductsModel
```

Properties

Id

WishlistProducts Id

Declaration

```
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[Column("wishlist_products_id")]
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Product

Product's navigation property

Declaration

```
public ProductModel? Product { get; set; }
```

Property Value

TYPE	DESCRIPTION
ProductModel	

ProductId

Product's Id

Declaration

```
[Column("product_id")]
public Guid ProductId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Wishlist

Wishlist's navigation property

Declaration

```
[JsonIgnore]
public WishlistModel? Wishlist { get; set; }
```

Property Value

TYPE	DESCRIPTION
WishlistModel	

WishlistId

Wishlist's Id

Declaration

```
[Column("wishlists_id")]
public Guid WishlistId { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Api.Options

Classes

BlobOptions

EncryptionOptions

Encryption options.

SmtpOptions

TokenOptions

Token options.

UrlsOptions

Urls options.

Class BlobOptions

Inheritance

[object](#)

BlobOptions

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Api.Options

Assembly: SoundParadise.Api.dll

Syntax

```
public class BlobOptions
```

Properties

BaseUrl

Base url.

Declaration

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

Property Value

TYPE	DESCRIPTION
[string](#)	

ConnectionString

Connection string to Blob.

Declaration

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

Property Value

TYPE	DESCRIPTION
[string](#)	

ContainerName

Container name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

TokenSas

Token Sas.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Methods

Validate()

Validate.

Declaration

```
public void Validate()
```

Exceptions

TYPE	CONDITION
ArgumentNullException	

Class EncryptionOptions

Encryption options.

object

EncryptionOptions

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Syntax

```
public class EncryptionOptions
```

Properties

IV

Encryption vector.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Key

Key.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Methods

Validate()

Validate.

Declaration

```
public void Validate()
```

Exceptions

TYPE	CONDITION
ArgumentNullException	

```
public void Validate()
```

Exceptions

TYPE	CONDITION
ArgumentNullException	

Class SmtpOptions

Inheritance

object

SmtpOptions

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Options

Assembly: SoundParadise.Api.dll

Syntax

```
public class SmtpOptions
```

Properties

Email

Email.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Host

Host.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Name

Name.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Password

Password.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Port

Port.

Declaration

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

Property Value

TYPE	DESCRIPTION
int	

Class TokenOptions

Token options.

Inheritance

object

TokenOptions

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Options

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenOptions
```

Properties

SecretKey

Secret key.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Methods

Validate()

Validate.

Declaration

```
public void Validate()
```

Exceptions

TYPE	CONDITION
ArgumentNullException	

Class UrlsOptions

Urls options.

Inheritance

object

UrlsOptions

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Options

Assembly: SoundParadise.Api.dll

Syntax

```
public class UrlsOptions
```

Properties

Client

Client.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Server

Server.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Api.Services

Classes

[CardEncryptionService](#)

Card encryption service.

[EncryptionService](#)

Encription service.

[HashingService](#)

Hashing service.

[ImageService](#)

Image service.

[LoggingService<T>](#)

Logging service.

[PaymentService](#)

Payment service.

[SmtpService](#)

Smtp service.

[TokenService](#)

Token service.

Class CardEncryptionService

Card encryption service.

Inheritance

object

EncryptionService

CardEncryptionService

Implements

IEncryptionService

Inherited Members

EncryptionService.EncryptData(string, ICryptoTransform)

EncryptionService.DecryptData(byte[], ICryptoTransform)

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class CardEncryptionService : EncryptionService, IEncryptionService
```

Constructors

CardEncryptionService(IOptions<EncryptionOptions>)

Constructor for card encrypt service.

Declaration

```
public CardEncryptionService(IOptions<EncryptionOptions> options)
```

Parameters

TYPE	NAME	DESCRIPTION
IOptions<EncryptionOptions>	options	Encrypt option.

Methods

DecryptCardData(CardModel)

Decrypt card data.

Declaration

```
public CardDto DecryptCardData(CardModel card)
```

Parameters

TYPE	NAME	DESCRIPTION
[CardModel](#)	card	CardModel object.

Returns

TYPE	DESCRIPTION
[CardDto](#)	CartDto object.

EncryptCardData(CardDto)

Encrypt card data.

Declaration

```
public CardModel EncryptCardData(CardDto cardDto)
```

Parameters

TYPE	NAME	DESCRIPTION
[CardDto](#)	cardDto	CardDto object.

Returns

TYPE	DESCRIPTION
[CardModel](#)	CardModel object.

Implements

[IEncryptionService](#)

Class EncryptionService

Encription service.

Inheritance

object

EncryptionService

CardEncryptionService

Implements

IEncryptionService

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class EncryptionService : IEncryptionService
```

Methods

DecryptData(byte[], ICryptoTransform)

Decrypt data.

Declaration

```
public string DecryptData(byte[] encryptedData, ICryptoTransform decryptor)
```

Parameters

TYPE	NAME	DESCRIPTION
byte[]	encryptedData	Encrypted data.
ICryptoTransform	decryptor	Decryptor data.

Returns

TYPE	DESCRIPTION
string	Decrypt data in string

EncryptData(string, ICryptoTransform)

Encrypt data.

Declaration

```
public byte[] EncryptData(string data, ICryptoTransform encryptor)
```

Parameters

TYPE	NAME	DESCRIPTION
string	data	Data to encrypt.
ICryptoTransform	encryptor	Encryptor.

Returns

TYPE	DESCRIPTION
byte[]	Encrypt data in byte array.

Implements

IEncryptionService

Class HashingService

Hashing service.

Inheritance

[object](#)

HashingService

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class HashingService
```

Methods

CreatePasswordHash(string, out byte[], out byte[])

Create password hash.

Declaration

```
public void CreatePasswordHash(string password, out byte[] passwordSalt, out byte[] passwordHash)
```

Parameters

TYPE	NAME	DESCRIPTION
[string](#)	password	Password.
[byte](#)[]	passwordSalt	Password salt.
[byte](#)[]	passwordHash	Password hash.

VerifyPasswordHash(string, byte[], byte[])

Verify password hesh.

Declaration

```
public bool VerifyPasswordHash(string password, byte[] passwordHash, byte[] passwordSalt)
```

Parameters

TYPE	NAME	DESCRIPTION
string	password	Password.
byte[]	passwordHash	Password hash.
byte[]	passwordSalt	PAssword salt.

Returns

TYPE	DESCRIPTION
bool	True if ok, false if not.

Class ImageService

Image service.

Inheritance

object

ImageService

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class ImageService
```

Constructors

ImageService(IOptions<BlobOptions>)

Constructor for image service.

Declaration

```
public ImageService(IOptions<BlobOptions> options)
```

Parameters

TYPE	NAME	DESCRIPTION
IOptions<BlobOptions>	options	BlopOption option.

Methods

GetImageUrl(string)

Get image url.

Declaration

```
public string GetImageUrl(string fileName)
```

Parameters

TYPE	NAME	DESCRIPTION
string	fileName	Image file name.

Returns

TYPE	DESCRIPTION
string	Url string.

UploadImage(IFormFile, Guid)

Uploads an image file to the blob storage container.

Declaration

```
public RequestResult UploadImage(IFormFile file, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
IFormFile	file	The image file to upload.
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	The uploaded image URL or an error message.

UploadImage(Stream, string)

Upload image.

Declaration

```
public Task<string> UploadImage(Stream imageStream, string fileName)
```

Parameters

TYPE	NAME	DESCRIPTION
Stream	imageStream	Image stream.
string	fileName	Image file name.

Returns

TYPE	DESCRIPTION
Task<string>	Task object.

Class LoggingService\<T>

Logging service.

Inheritance

[object](#)

LoggingService\<T>

Implements

[ILoggingService](#)\<T>

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Syntax

```
public class LoggingService<T> : ILoggingService<T>
```

Type Parameters

NAME	DESCRIPTION
T	

Constructors

LoggingService()

Constructor for LoggingService

Declaration

```
public LoggingService()
```

Methods

Log(LogLevel, string)

Log error.

Declaration

```
public void Log(LogLevel logLevel, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
LogLevel	logLevel	Error level.

TYPE	NAME	DESCRIPTION
string	message	Message about error.

LogError(string)

Log error.

Declaration

```
public void LogError(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
string	message	Message error.

LogException(Exception, string)

Log exception

Declaration

```
public void LogException(Exception exception, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
Exception	exception	Exception.
string	message	Message error.

Implements

ILoggingService<T>

Class PaymentService

Payment service.

Inheritance

object

PaymentService

Implements

IPaymentService

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class PaymentService : IPaymentService
```

Constructors

PaymentService(ILoggingService<PaymentService>)

Constructor for ppayment service.

Declaration

```
public PaymentService(ILoggingService<PaymentService> loggingService)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggingService<PaymentService>	loggingService	Service for logging errors.

Methods

Checkout(Guid, string, decimal, PaymentProvider)

Checkout.

Declaration

```
public RequestResult Checkout(Guid orderId, string description, decimal amount, PaymentProvider paymentProvider)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Guid	orderId	Order Id.
string	description	Description.
decimal	amount	Amount.
PaymentProvider	paymentProvider	Payment provider.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object.

Exceptions

TYPE	CONDITION
NotImplementedException	
ArgumentOutOfRangeException	

Implements

IPaymentService

Class SmtpService

Smtp service.

Inheritance

object

SmtpService

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class SmtpService
```

Constructors

SmtpService(IOptions<SmtpOptions>)

Constructor for smtp service.

Declaration

```
public SmtpService(IOptions<SmtpOptions> options)
```

Parameters

TYPE	NAME	DESCRIPTION
IOptions<SmtpOptions>	options	Smtp option.

Methods

SendConfirmationEmail(string, string)

Confirmation email.

Declaration

```
public Task SendConfirmationEmail(string recipientEmail, string confirmationLink)
```

Parameters

TYPE	NAME	DESCRIPTION
string	recipientEmail	Email.
string	confirmationLink	Confirmation link.

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
Task	Task object.

SendOrderConfirmationEmail(string, string)

Send order confirmation url.

Declaration

```
public Task SendOrderConfirmationEmail(string recipientEmail, string orderNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
string	recipientEmail	Email.
string	orderNumber	Order nubmer.

Returns

TYPE	DESCRIPTION
Task	Task object.

SendPasswordChangeEmail(string, string)

Send password change email.

Declaration

```
public Task SendPasswordChangeEmail(string recipientEmail, string confirmationLink)
```

Parameters

TYPE	NAME	DESCRIPTION
string	recipientEmail	Email.
string	confirmationLink	Confirmation link.

Returns

TYPE	DESCRIPTION
Task	Task object.

TYPE	DESCRIPTION
Task	Task object.

Class TokenService

Token service.

Inheritance

object

TokenService

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Services

Assembly: SoundParadise.Api.dll

Syntax

```
public class TokenService
```

Constructors

TokenService(IOptions<TokenOptions>)

TokenService constructor.

Declaration

```
public TokenService(IOptions<TokenOptions> options)
```

Parameters

TYPE	NAME	DESCRIPTION
IOptions<TokenOptions>	options	Token option.

Methods

GenerateToken(UserModel, out TokenJournalModel)

Generate new token.

Declaration

```
public TokenModel GenerateToken(UserModel user, out TokenJournalModel tokenJournal)
```

Parameters

TYPE	NAME	DESCRIPTION
UserModel	user	UserModel object.
TokenJournalModel	tokenJournal	TokenJournalModel object

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
TokenModel	TokenModel object.

ValidateToken(string)

Token validator.

Declaration

```
public ClaimsPrincipal? ValidateToken(string token)
```

Parameters

TYPE	NAME	DESCRIPTION
string	token	Token.

Returns

TYPE	DESCRIPTION
ClaimsPrincipal	TokenModel object.

Namespace SoundParadise.Api.Validators.Card

Classes

[CardDtoValidator](#)

CartDto validator.

Class CardDtoValidator

CartDto validator.

Inheritance

object

AbstractValidator<CardDto>

CardDtoValidator

Implements

IValidator<CardDto>

IValidator

IEnumerable<IValidationRule>

IEnumerable

Inherited Members

AbstractValidator<CardDto>.Validate(CardDto)

AbstractValidator<CardDto>.ValidateAsync(CardDto, CancellationToken)

AbstractValidator<CardDto>.Validate(ValidationContext<CardDto>)

AbstractValidator<CardDto>.ValidateAsync(ValidationContext<CardDto>, CancellationToken)

AbstractValidator<CardDto>.CreateDescriptor()

AbstractValidator<CardDto>.RuleFor<TProperty>(Expression<Func<CardDto, TProperty>>)

AbstractValidator<CardDto>.Transform<TProperty, TTransformed>(Expression<Func<CardDto, TProperty>>, Func<TProperty, TTransformed>)

AbstractValidator<CardDto>.Transform<TProperty, TTransformed>(Expression<Func<CardDto, TProperty>>, Func<CardDto, TProperty, TTransformed>)

AbstractValidator<CardDto>.RuleForEach<TElement>(Expression<Func<CardDto, IEnumerable<TElement>>>)

AbstractValidator<CardDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CardDto, IEnumerable<TElement>>>, Func<TElement, TTransformed>)

AbstractValidator<CardDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CardDto, IEnumerable<TElement>>>, Func<CardDto, TElement, TTransformed>)

AbstractValidator<CardDto>.RuleSet(string, Action)

AbstractValidator<CardDto>.When(Func<CardDto, bool>, Action)

AbstractValidator<CardDto>.When(Func<CardDto, ValidationContext<CardDto>, bool>, Action)

AbstractValidator<CardDto>.Unless(Func<CardDto, bool>, Action)

AbstractValidator<CardDto>.Unless(Func<CardDto, ValidationContext<CardDto>, bool>, Action)

AbstractValidator<CardDto>.WhenAsync(Func<CardDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<CardDto>.WhenAsync(Func<CardDto, ValidationContext<CardDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<CardDto>.UnlessAsync(Func<CardDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<CardDto>.UnlessAsync(Func<CardDto, ValidationContext<CardDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<CardDto>.Include(IValidator<CardDto>)

AbstractValidator<CardDto>.Include<TValidator>(Func<CardDto, TValidator>)

AbstractValidator<CardDto>.GetEnumerator()

AbstractValidator<CardDto>.EnsureInstanceNotNull(object)

AbstractValidator<CardDto>.PreValidate(ValidationContext<CardDto>, ValidationResult)

AbstractValidator<CardDto>.RaiseValidationException(ValidationContext<CardDto>, ValidationResult)

AbstractValidator<CardDto>.CascadeMode

AbstractValidator<CardDto>.ClassLevelCascadeMode

AbstractValidator<CardDto>.RuleLevelCascadeMode

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Validators.Card

Assembly: SoundParadise.Api.dll

Syntax

```
public class CardDtoValidator : AbstractValidator<CardDto>, IValidator<CardDto>, IValidator,
IEnumerable<IValidationRule>, IEnumerable
```

Constructors

CardDtoValidator()

Validate.

Declaration

```
public CardDtoValidator()
```

Implements

FluentValidation.IValidator<T>

FluentValidation.IValidator

IEnumerable<T>

IEnumerable

Namespace SoundParadise.Api.Validators.CartItem

Classes

[CartItemDtoValidator](#)

CartItemDto validator.

Class CartItemDtoValidator

CartItemDto validator.

Inheritance

object

AbstractValidator<CartItemDto>

CartItemDtoValidator

Implements

IValidator<CartItemDto>

IValidator

IEnumerable<IValidationRule>

IEnumerable

Inherited Members

AbstractValidator<CartItemDto>.Validate(CartItemDto)

AbstractValidator<CartItemDto>.ValidateAsync(CartItemDto, CancellationToken)

AbstractValidator<CartItemDto>.Validate(ValidationContext<CartItemDto>)

AbstractValidator<CartItemDto>.ValidateAsync(ValidationContext<CartItemDto>, CancellationToken)

AbstractValidator<CartItemDto>.CreateDescriptor()

AbstractValidator<CartItemDto>.RuleFor<TProperty>(Expression<Func<CartItemDto, TProperty>>)

AbstractValidator<CartItemDto>.Transform<TProperty, TTransformed>(Expression<Func<CartItemDto, TProperty>>, Func<TProperty, TTransformed>)

AbstractValidator<CartItemDto>.Transform<TProperty, TTransformed>(Expression<Func<CartItemDto, TProperty>>, Func<CartItemDto, TProperty, TTransformed>)

AbstractValidator<CartItemDto>.RuleForEach<TElement>(Expression<Func<CartItemDto, IEnumerable<TElement>>>)

AbstractValidator<CartItemDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CartItemDto, IEnumerable<TElement>>>, Func<TElement, TTransformed>)

AbstractValidator<CartItemDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CartItemDto, IEnumerable<TElement>>>, Func<CartItemDto, TElement, TTransformed>)

AbstractValidator<CartItemDto>.RuleSet(string, Action)

AbstractValidator<CartItemDto>.When(Func<CartItemDto, bool>, Action)

AbstractValidator<CartItemDto>.When(Func<CartItemDto, ValidationContext<CartItemDto>, bool>, Action)

AbstractValidator<CartItemDto>.Unless(Func<CartItemDto, bool>, Action)

AbstractValidator<CartItemDto>.Unless(Func<CartItemDto, ValidationContext<CartItemDto>, bool>, Action)

AbstractValidator<CartItemDto>.WhenAsync(Func<CartItemDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<CartItemDto>.WhenAsync(Func<CartItemDto, ValidationContext<CartItemDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<CartItemDto>.UnlessAsync(Func<CartItemDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<CartItemDto>.UnlessAsync(Func<CartItemDto, ValidationContext<CartItemDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<CartItemDto>.Include(IValidator<CartItemDto>)

AbstractValidator<CartItemDto>.Include<TValidator>(Func<CartItemDto, TValidator>)

AbstractValidator<CartItemDto>.GetEnumerator()

AbstractValidator<CartItemDto>.EnsureInstanceNotNull(object)

AbstractValidator<CartItemDto>.PreValidate(ValidationContext<CartItemDto>, ValidationResult)

AbstractValidator<CartItemDto>.RaiseValidationException(ValidationContext<CartItemDto>, ValidationResult)

AbstractValidator<CartItemDto>.CascadeMode

AbstractValidator<CartItemDto>.ClassLevelCascadeMode

AbstractValidator<CartItemDto>.RuleLevelCascadeMode

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Validators.CartItem

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartItemDtoValidator : AbstractValidator<CartItemDto>, IValidator<CartItemDto>, IValidator,
IEnumerable<IValidationRule>, IEnumerable
```

Constructors

CartItemDtoValidator(SoundParadiseDbContext)

Validator.

Declaration

```
public CartItemDtoValidator(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	

Implements

FluentValidation.IValidator<T>

FluentValidation.IValidator

IEnumerable<T>

IEnumerable

Namespace SoundParadise.Api.Validators.User

Classes

UserAuthenticationDtoValidator

UserAuthenticationDto validator.

UserCreateDtoValidator

UserDtoValidator<TUserDto>

UserDto validator.

Class UserAuthenticationDtoValidator

UserAuthenticationDto validator.

Inheritance

object

AbstractValidator<UserAuthenticationDto>

UserAuthenticationDtoValidator

Implements

IValidator<UserAuthenticationDto>

IValidator

IEnumerable<IValidationRule>

IEnumerable

Inherited Members

AbstractValidator<UserAuthenticationDto>.Validate(UserAuthenticationDto)

AbstractValidator<UserAuthenticationDto>.ValidateAsync(UserAuthenticationDto, CancellationToken)

AbstractValidator<UserAuthenticationDto>.Validate(ValidationContext<UserAuthenticationDto>)

AbstractValidator<UserAuthenticationDto>.ValidateAsync(ValidationContext<UserAuthenticationDto>, CancellationToken)

AbstractValidator<UserAuthenticationDto>.CreateDescriptor()

AbstractValidator<UserAuthenticationDto>.RuleFor<TProperty>(Expression<Func<UserAuthenticationDto, TProperty>>)

AbstractValidator<UserAuthenticationDto>.Transform<TProperty, TTransformed>(Expression<Func<UserAuthenticationDto, TProperty>>, Func<TProperty, TTransformed>)

AbstractValidator<UserAuthenticationDto>.Transform<TProperty, TTransformed>(Expression<Func<UserAuthenticationDto, TProperty>>, Func<UserAuthenticationDto, TProperty, TTransformed>)

AbstractValidator<UserAuthenticationDto>.RuleForEach<TElement>(Expression<Func<UserAuthenticationDto, IEnumerable<TElement>>>)

AbstractValidator<UserAuthenticationDto>.TransformForEach<TElement, TTransformed>(Expression<Func<UserAuthenticationDto, IEnumerable<TElement>>>, Func<TElement, TTransformed>)

AbstractValidator<UserAuthenticationDto>.TransformForEach<TElement, TTransformed>(Expression<Func<UserAuthenticationDto, IEnumerable<TElement>>>, Func<UserAuthenticationDto, TElement, TTransformed>)

AbstractValidator<UserAuthenticationDto>.RuleSet(string, Action)

AbstractValidator<UserAuthenticationDto>.When(Func<UserAuthenticationDto, bool>, Action)

AbstractValidator<UserAuthenticationDto>.When(Func<UserAuthenticationDto, ValidationContext<UserAuthenticationDto>, bool>, Action)

AbstractValidator<UserAuthenticationDto>.Unless(Func<UserAuthenticationDto, bool>, Action)

AbstractValidator<UserAuthenticationDto>.Unless(Func<UserAuthenticationDto, ValidationContext<UserAuthenticationDto>, bool>, Action)

AbstractValidator<UserAuthenticationDto>.WhenAsync(Func<UserAuthenticationDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<UserAuthenticationDto>.WhenAsync(Func<UserAuthenticationDto, ValidationContext<UserAuthenticationDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<UserAuthenticationDto>.UnlessAsync(Func<UserAuthenticationDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<UserAuthenticationDto>.UnlessAsync(Func<UserAuthenticationDto, ValidationContext<UserAuthenticationDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<UserAuthenticationDto>.Include(IValidator<UserAuthenticationDto>)

AbstractValidator<UserAuthenticationDto>.Include<TValidator>(Func<UserAuthenticationDto, TValidator>)

AbstractValidator<UserAuthenticationDto>.GetEnumerator()

AbstractValidator<UserAuthenticationDto>.EnsureInstanceNotNull(object)

AbstractValidator<UserAuthenticationDto>.PreValidate(ValidationContext<UserAuthenticationDto>, ValidationResult)

AbstractValidator<UserAuthenticationDto>.RaiseValidationException(ValidationContext<UserAuthenticationDto>, ValidationResult)

AbstractValidator<UserAuthenticationDto>.CascadeMode

AbstractValidator<UserAuthenticationDto>.ClassLevelCascadeMode

AbstractValidator<UserAuthenticationDto>.RuleLevelCascadeMode

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Validators.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserAuthenticationDtoValidator : AbstractValidator<UserAuthenticationDto>,
IValidator<UserAuthenticationDto>, IValidator, IEnumerable<IValidationRule>, IEnumerable
```

Constructors

UserAuthenticationDtoValidator(SoundParadiseDbContext)

Validator.

Declaration

```
public UserAuthenticationDtoValidator(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context

Implements

FluentValidation.IValidator<T>

FluentValidation.IValidator

IEnumerable<T>

IEnumerable

Class UserCreateDtoValidator

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Validators.User

Assembly: SoundParadise.Api.dll

Syntax

```
public class UserCreateDtoValidator : UserDtoValidator<UserCreateDto>, IValidator<UserCreateDto>, IValidator,
IEnumerable<IValidationRule>, IEnumerable
```

Constructors

UserCreateDtoValidator(SoundParadiseDbContext)

Constructor for the UserCreateDtoValidator

Declaration

```
public UserCreateDtoValidator(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.

Implements

FluentValidation.IValidator<T>

FluentValidation.IValidator

IEnumerable<T>

IEnumerable

Class UserDtoValidator<TUserDto>

UserDto validator.

Inheritance

object

AbstractValidator<TUserDto>

UserDtoValidator<TUserDto>

UserCreateDtoValidator

Implements

IValidator<TUserDto>

IValidator

IEnumerable<IValidationRule>

IEnumerable

Inherited Members

AbstractValidator<TUserDto>.Validate(TUserDto)

AbstractValidator<TUserDto>.ValidateAsync(TUserDto, CancellationToken)

AbstractValidator<TUserDto>.Validate(ValidationContext<TUserDto>)

AbstractValidator<TUserDto>.ValidateAsync(ValidationContext<TUserDto>, CancellationToken)

AbstractValidator<TUserDto>.CreateDescriptor()

AbstractValidator<TUserDto>.RuleFor<TProperty>(Expression<Func<TUserDto, TProperty>>)

AbstractValidator<TUserDto>.Transform<TProperty, TTransformed>(Expression<Func<TUserDto, TProperty>>, Func<TProperty, TTransformed>)

AbstractValidator<TUserDto>.Transform<TProperty, TTransformed>(Expression<Func<TUserDto, TProperty>>, Func<TUserDto, TProperty, TTransformed>)

AbstractValidator<TUserDto>.RuleForEach<TElement>(Expression<Func<TUserDto, IEnumerable<TElement>>>)

AbstractValidator<TUserDto>.TransformForEach<TElement, TTransformed>(Expression<Func<TUserDto, IEnumerable<TElement>>>, Func<TElement, TTransformed>)

AbstractValidator<TUserDto>.TransformForEach<TElement, TTransformed>(Expression<Func<TUserDto, IEnumerable<TElement>>>, Func<TUserDto, TElement, TTransformed>)

AbstractValidator<TUserDto>.RuleSet(string, Action)

AbstractValidator<TUserDto>.When(Func<TUserDto, bool>, Action)

AbstractValidator<TUserDto>.When(Func<TUserDto, ValidationContext<TUserDto>, bool>, Action)

AbstractValidator<TUserDto>.Unless(Func<TUserDto, bool>, Action)

AbstractValidator<TUserDto>.Unless(Func<TUserDto, ValidationContext<TUserDto>, bool>, Action)

AbstractValidator<TUserDto>.WhenAsync(Func<TUserDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<TUserDto>.WhenAsync(Func<TUserDto, ValidationContext<TUserDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<TUserDto>.UnlessAsync(Func<TUserDto, CancellationToken, Task<bool>>, Action)

AbstractValidator<TUserDto>.UnlessAsync(Func<TUserDto, ValidationContext<TUserDto>, CancellationToken, Task<bool>>, Action)

AbstractValidator<TUserDto>.Include(IValidator<TUserDto>)

AbstractValidator<TUserDto>.Include<TValidator>(Func<TUserDto, TValidator>)

AbstractValidator<TUserDto>.GetEnumerator()

AbstractValidator<TUserDto>.EnsureInstanceNotNull(object)

AbstractValidator<TUserDto>.PreValidate(ValidationContext<TUserDto>, ValidationResult)

AbstractValidator<TUserDto>.RaiseValidationException(ValidationContext<TUserDto>, ValidationResult)

AbstractValidator<TUserDto>.CascadeMode

AbstractValidator<TUserDto>.ClassLevelCascadeMode

AbstractValidator<TUserDto>.RuleLevelCascadeMode

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Api.Validators.User

Assembly: SoundParadise.Api.dll

Syntax

```
public abstract class UserDtoValidator<TUserDto> : AbstractValidator<TUserDto>, IValidator<TUserDto>,
IValidator, IEnumerable<IValidationRule>, IEnumerable where TUserDto : UserDto
```

Type Parameters

NAME	DESCRIPTION
TUserDto	Parent class parameter.

Constructors

UserDtoValidator(SoundParadiseDbContext)

Constructor for UserDtoValidator.

Declaration

```
protected UserDtoValidator(SoundParadiseDbContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
SoundParadiseDbContext	context	Db context.

Implements

FluentValidation.IValidator<T>

FluentValidation.IValidator

IEnumerable<T>

IEnumerable

Namespace SoundParadise.Web.Dto.Address

Classes

[AddressDto](#)

Address Dto.

Class AddressDto

Address Dto.

Inheritance

object

AddressDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Web.Dto.Address

Assembly: SoundParadise.Api.dll

Syntax

```
public class AddressDto
```

Properties

City

City.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

DeliveryOption

Delivery option enum.

Declaration

```
public DeliveryOptionEnum DeliveryOption { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryOptionEnum	

DeliveryType

Delivery type enum.

Declaration

```
public DeliveryTypeEnum DeliveryType { get; set; }
```

Property Value

TYPE	DESCRIPTION
DeliveryTypeEnum	

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

PostOfficeAddress

Post office address.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Namespace SoundParadise.Web.Dto.Card

Classes

CardDto

Card Dto.

Class CardDto

Card Dto.

Inheritance

object

CardDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Web.Dto.Card

Assembly: SoundParadise.Api.dll

Syntax

```
public class CardDto
```

Properties

CVV

CVV code.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

CardNumber

Card number.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

ExpirationDate

Expiration date.

Declaration

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

Property Value

TYPE	DESCRIPTION
string	

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Web.Dto.Cart

Classes

[CartDto](#)

Cart dto.

Class CartDto

Cart dto.

Inheritance

object

CartDto

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: SoundParadise.Web.Dto.Cart

Assembly: SoundParadise.Api.dll

Syntax

```
public class CartDto
```

Properties

CartItems

List of cartitemobjects.

Declaration

```
public List<CartItemDto> CartItems { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CartItemDto>	

Id

Id.

Declaration

```
public Guid Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
Guid	

Namespace SoundParadise.Web.Models.Address

Classes

AddressCrud

AddressModel CRUD .

Class AddressCrud

AddressModel CRUD .

Inheritance

[object](#)

AddressCrud

Implements

[IAddressCrud](#)

Inherited Members

[object.Equals(object)](#)

[object.Equals(object, object)](#)

[object.GetHashCode()](#)

[object.GetType()](#)

[object.MemberwiseClone()](#)

[object.ReferenceEquals(object, object)](#)

[object.ToString()](#)

Namespace: SoundParadise.Web.Models.Address

Assembly: SoundParadise.Api.dll

Syntax

```
public class AddressCrud : IAddressCrud
```

Constructors

AddressCrud(SoundParadiseDbContext, ILoggingService<AddressCrud>)

AddressCrud.

Declaration

```
public AddressCrud(SoundParadiseDbContext context, ILoggingService<AddressCrud> loggingService)
```

Parameters

TYPE	NAME	DESCRIPTION
[SoundParadiseDbContext](#)	context	Db context
[ILoggingService](#)<[AddressCrud](#)>	loggingService	Service for logging errors.

Methods

AddAddress(AddressDto, Guid)

Add new address to data base.

Declaration

```
public RequestResult AddAddress(AddressDto addressDto, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressDto	addressDto	AddressDto object.
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

CreateAddress(AddressModel)

Create new address to data base.

Declaration

```
public RequestResult CreateAddress(AddressModel address)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressModel	address	AddressModel object.

Returns

TYPE	DESCRIPTION
RequestResult	RequestResult object

DeleteAddress(Guid, Guid)

Declaration

```
public RequestResult DeleteAddress(Guid addressId, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	addressId	
Guid	userId	

Returns

TYPE	DESCRIPTION
RequestResult	

GetAddressById(Guid?, Guid?)

Get address from data base.

Declaration

```
public AddressModel GetAddressById(Guid? addressId, Guid? userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	addressId	Address Id.
Guid?	userId	User Id.

Returns

TYPE	DESCRIPTION
AddressModel	AddressModel object

GetAddressDtoById(Guid?, Guid?)

Declaration

```
public AddressDto GetAddressDtoById(Guid? addressId, Guid? userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid?	addressId	
Guid?	userId	

Returns

TYPE	DESCRIPTION
AddressDto	

GetAddressesByDeliveryOption(DeliveryOptionEnum, Guid)

Create new address to data base.

Declaration

```
public List<AddressDto> GetAddressesByDeliveryOption(DeliveryOptionEnum deliveryOption, Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
DeliveryOptionEnum	deliveryOption	Delivery option enum.

TYPE	NAME	DESCRIPTION
Guid	userId	User Id.

Returns

TYPE	DESCRIPTION
List<AddressDto>	List of AddressDto object

GetAddressesByUserId(Guid)

Declaration

```
public List<AddressDto> GetAddressesByUserId(Guid userId)
```

Parameters

TYPE	NAME	DESCRIPTION
Guid	userId	

Returns

TYPE	DESCRIPTION
List<AddressDto>	

UpdateAddress(AddressModel)

Declaration

```
public RequestResult UpdateAddress(AddressModel address)
```

Parameters

TYPE	NAME	DESCRIPTION
AddressModel	address	

Returns

TYPE	DESCRIPTION
RequestResult	

Implements

IAddressCrud