

# Namespace Agora.API.Controllers

## Classes

### [PostCategoriesController](#)

Manages operations for post categories, including listing and retrieving post category data.

### [PostsController](#)

Handles creation, updating, deletion, and retrieval of posts offered or requested by users.

### [TransactionStatusController](#)

Handles operations related to transaction status management, such as updating status definitions or retrieving status information.

### [TransactionsController](#)

Manages transactions between users, including creation, updates, and status changes.

### [UsersController](#)

Handles operations related to user management such as registration, login, logout, and retrieval of user information.

# Class PostCategoriesController

Namespace: [Agora.API.Controllers](#)

Assembly: Agora.API.dll

Manages operations for post categories, including listing and retrieving post category data.

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

## Inheritance

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

## 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.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#) ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) ,  [ControllerBase.ValidationProblem\(\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#) ,  
 [ControllerBase.Created\(\)](#) ,  [ControllerBase.Created\(string, object\)](#) ,  
 [ControllerBase.Created\(Uri, object\)](#) ,  [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#) ,  [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) ,  [ControllerBase.Accepted\(\)](#) ,  
 [ControllerBase.Accepted\(object\)](#) ,  [ControllerBase.Accepted\(Uri\)](#) ,  [ControllerBase.Accepted\(string\)](#) ,  
 [ControllerBase.Accepted\(string, object\)](#) ,  [ControllerBase.Accepted\(Uri, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string\)](#) ,  [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.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\(\)](#)

## Extension Methods

[ControllerBaseExtensions.MapErrorResult\(ControllerBase, Result\)](#)

# Constructors

## PostCategoriesController(IPostCategoryService, IMapper)

Manages operations for post categories, including listing and retrieving post category data.

```
public PostCategoriesController(IPostCategoryService postCategoryService, IMapper mapper)
```

## Parameters

postCategoryService [IPostCategoryService](#)

## Methods

### CreatePostCategory(CreatePostCategoryDto)

Creates a new post category.

```
[Authorize(Roles = "Admin")]
[HttpPost]
public Task<ActionResult<PostCategoryDetailsDto>> CreatePostCategory(CreatePostCategoryDto
dto)
```

#### Parameters

dto [CreatePostCategoryDto](#)

The post category data transfer object containing creation details.

#### Returns

[Task](#) <[ActionResult](#) <[PostCategoryDetailsDto](#)>>

Returns **201 Created** with the newly created post category details. Returns **400 Bad Request** if input or business rules validation fails. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user is not allowed to manage the post category (not admin).

### DeletePostCategory(long)

Deletes a post category.

```
[Authorize(Roles = "Admin")]
[HttpDelete("{id:long}")]
public Task<ActionResult> DeletePostCategory(long id)
```

#### Parameters

id [long](#)

The identifier of the post category to delete.

Returns

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

Returns **204 No Content** on successful deletion. Returns **400 Bad Request** if input validation fails.

Returns **404 Not Found** if the post category or a related object does not exist.

## GetAllPostCategories(PostCategoryQueryParameters)

Retrieves all post categories, optionally filtered and sorted by query parameters.

```
[HttpGet]  
public Task<ActionResult<IReadOnlyList<PostCategorySummaryDto>>>  
GetAllPostCategories(PostCategoryQueryParameters queryParameters)
```

Parameters

**queryParameters** [PostCategoryQueryParameters](#)

Optional filters to apply to the post categories list.

Returns

[Task](#) <[ActionResult](#)> <[IReadOnlyList](#)> <[PostCategorySummaryDto](#)>>>

Returns **200 OK** with a list of post categories.

## GetPostCategory(long)

Retrieves detailed information of a specific post category by its identifier, like all related posts.

```
[Authorize(Roles = "Admin")]  
[HttpGet("{id:long}")]  
public Task<ActionResult<PostCategoryDetailsDto>> GetPostCategory(long id)
```

Parameters

[id](#) [long](#)

The identifier of the post category to retrieve.

Returns

[Task](#) <[ActionResult](#)> <[PostCategoryDetailsDto](#)>>

Returns [200 OK](#) with the post category details if found and authorized. Returns [401 Unauthorized](#) if the user is not authenticated. Returns [403 Forbidden](#) if the user is not allowed to view the post category (not admin). Returns [404 Not Found](#) if the post category does not exist.

## UpdatePostCategoryName(long, UpdatePostCategoryDto)

Updates name of an existing post category.

```
[Authorize(Roles = "Admin")]
[HttpPatch("{id:long}")]
public Task<ActionResult> UpdatePostCategoryName(long id, UpdatePostCategoryDto dto)
```

Parameters

[id](#) [long](#)

The identifier of the post category to update.

[dto](#) [UpdatePostCategoryDto](#)

Dto containing the new name to apply to the post category.

Returns

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

Returns [204 No Content](#) on successful update. Returns [400 Bad Request](#) if input validation fails. Returns [404 Not Found](#) if the post category or a related object does not exist.

# Class PostsController

Namespace: [Agora.API.Controllers](#)

Assembly: Agora.API.dll

Handles creation, updating, deletion, and retrieval of posts offered or requested by users.

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

## Inheritance

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

## 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.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#) ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) ,  [ControllerBase.ValidationProblem\(\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#) ,  
 [ControllerBase.Created\(\)](#) ,  [ControllerBase.Created\(string, object\)](#) ,  
 [ControllerBase.Created\(Uri, object\)](#) ,  [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#) ,  [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) ,  [ControllerBase.Accepted\(\)](#) ,  
 [ControllerBase.Accepted\(object\)](#) ,  [ControllerBase.Accepted\(Uri\)](#) ,  [ControllerBase.Accepted\(string\)](#) ,  
 [ControllerBase.Accepted\(string, object\)](#) ,  [ControllerBase.Accepted\(Uri, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string\)](#) ,  [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.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\(\)](#)

## Extension Methods

[ControllerExtensions.MapErrorResult\(ControllerBase, Result\)](#)

## Constructors

**PostsController(IPostService, IMapper, IInputValidator, IUserContextService)**

Handles creation, updating, deletion, and retrieval of posts offered or requested by users.

```
public PostsController(IPostService postService, IMapper mapper, IInputValidator  
inputValidator, IUserContextService userContextService)
```

## Parameters

`postService` [IPostService](#)

`mapper` IMapper

`inputValidator` [IInputValidator](#)

`userContextService` [IUserContextService](#)

## Methods

### ActivatePost(long)

Make a post public by setting its status to `Active`.

```
[Authorize]
[HttpPut("{id:long}/activate")]
public Task<IActionResult> ActivatePost(long id)
```

#### Parameters

`id` [long](#)

The unique identifier of the post to make public.

#### Returns

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

Returns `204 No Content` if the operation succeeds. Returns `401 Unauthorized` if the user is not authenticated. Returns `403 Forbidden` if the user does not have permission to activate this post. Returns `404 Not Found` if the post does not exist or is not visible to the user. Returns `400 Bad Request` if the action violates business rules.

### CreatePost(CreatePostDto)

Creates a new post.

```
[Authorize]
[HttpPost]
```

```
public Task<ActionResult<PostDetailsDto>> CreatePost(CreatePostDto dto)
```

## Parameters

### dto [CreatePostDto](#)

The post data transfer object containing creation details.

## Returns

### [Task](#) <[ActionResult](#) <[PostDetailsDto](#)>>

Returns **201 Created** with the newly created post details. Returns **400 Bad Request** if input or business rules validation fails. Returns **401 Unauthorized** if the user is not authenticated.

## DeactivatePost(long)

Hide a post from the public by setting its status to **Inactive**.

```
[Authorize]  
[HttpPut("{id:long}/deactivate")]  
public Task<IActionResult> DeactivatePost(long id)
```

## Parameters

### id [long](#)

The unique identifier of the post to hide.

## Returns

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

Returns **204 No Content** if the operation succeeds. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user does not have permission to deactivate this post. Returns **404 Not Found** if the post does not exist or is not visible to the user. Returns **400 Bad Request** if the action violates business rules.

## DeletePost(long)

Deletes a post.

```
[Authorize]
[HttpDelete("{id:long}")]
public Task<ActionResult> DeletePost(long id)
```

Parameters

`id long`

The identifier of the post to delete.

Returns

`Task<ActionResult>`

Returns `204 No Content` on successful deletion. Returns `400 Bad Request` if input validation fails.  
Returns `404 Not Found` if the post or a related object does not exist.

## GetAllPosts(PostQueryParameters)

Retrieves all posts from all users, regardless of their status. This action is restricted to administrators only.

```
[Authorize(Roles = "Admin")]
[HttpGet("all")]
public Task<ActionResult<IReadOnlyList<PostSummaryDto>>> GetAllPosts(PostQueryParameters
queryParameters)
```

Parameters

`queryParameters PostQueryParameters`

Optional filtering parameters such as title, price range, category, user, etc.

Returns

`Task<ActionResult<IReadOnlyList<PostSummaryDto>>>`

Returns `200 OK` with a list of summarized post information across the platform. Returns `401 Unauthorized` if the user is not authenticated. Returns `403 Forbidden` if the user does not have admin privileges.

## GetCurrentUserPosts(PostQueryParameters)

Retrieves all posts created by the currently authenticated user. Includes posts with status "Active" or "Inactive", but not "Deleted".

```
[Authorize]  
[HttpGet("me")]  
public Task<ActionResult<IReadOnlyList<PostSummaryDto>>>  
GetCurrentUserPosts(PostQueryParameters queryParameters)
```

### Parameters

**queryParameters** [PostQueryParameters](#)

Optional filtering parameters such as title, price range, category, etc.

### Returns

[Task](#) <ActionResult <IReadOnlyList <PostSummaryDto>>>

Returns **200 OK** with a list of summarized post information owned by the current user. Returns **401 Unauthorized** if the user is not authenticated.

## GetPost(long)

Retrieves detailed information of a specific post by its identifier.

```
[HttpGet("{id:long}")]
public Task<ActionResult<PostDetailsDto>> GetPost(long id)
```

### Parameters

**id** [long](#)

The identifier of the post to retrieve.

### Returns

[Task](#) <ActionResult <PostDetailsDto>>

Returns **200 OK** with the post details if found and authorized. Returns **401 Unauthorized** if the user is not allowed to view the post. Returns **404 Not Found** if the post does not exist.

## GetPostsCatalogue(PostQueryParameters)

Retrieves all public posts visible in the catalog (i.e., with status "Active"), optionally filtered and sorted using query parameters.

```
[HttpGet("catalog")]
public Task<ActionResult< IReadOnlyList<PostSummaryDto>>>
GetPostsCatalogue(PostQueryParameters queryParameters)
```

### Parameters

**queryParameters** [PostQueryParameters](#)

Optional filtering parameters such as title, price range, category, etc.

### Returns

[Task](#) <[ActionResult](#) <[IReadOnlyList](#) <[PostSummaryDto](#)>>>

Returns **200 OK** with a list of summarized post information available to all users.

## UpdatePostDetails(long, UpdatePostDetailsDto)

Updates details of an existing post partially. Only allowed if the current user has modification rights.

```
[Authorize]
[HttpPatch("{id:long}")]
public Task<ActionResult> UpdatePostDetails(long id, UpdatePostDetailsDto dto)
```

### Parameters

**id** [long](#)

The identifier of the post to update.

**dto** [UpdatePostDetailsDto](#)

Partial post data to update.

## Returns

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

Returns **204 No Content** on successful update. Returns **400 Bad Request** if input validation fails. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user has not the modification rights for this post. Returns **404 Not Found** if the post or a related object does not exist.

# Class TransactionStatusController

Namespace: [Agora.API.Controllers](#)

Assembly: Agora.API.dll

Handles operations related to transaction status management, such as updating status definitions or retrieving status information.

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

## Inheritance

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

## 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\(string, string\)](#)  ,  [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.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#) ,  
[ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
[ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) , [ControllerBase.ValidationProblem\(\)](#) ,  
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#) ,  
[ControllerBase.Created\(\)](#) , [ControllerBase.Created\(string, object\)](#) ,  
[ControllerBase.Created\(Uri, object\)](#) , [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
[ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
[ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
[ControllerBase.CreatedAtRoute\(string, object\)](#) , [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
[ControllerBase.CreatedAtRoute\(string, object, object\)](#) , [ControllerBase.Accepted\(\)](#) ,  
[ControllerBase.Accepted\(object\)](#) , [ControllerBase.Accepted\(Uri\)](#) , [ControllerBase.Accepted\(string\)](#) ,  
[ControllerBase.Accepted\(string, object\)](#) , [ControllerBase.Accepted\(Uri, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string\)](#) , [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
[ControllerBase.AcceptedAtRoute\(object\)](#) , [ControllerBase.AcceptedAtRoute\(string\)](#) ,  
[ControllerBase.AcceptedAtRoute\(string, object\)](#) , [ControllerBase.AcceptedAtRoute\(object, object\)](#) ,  
[ControllerBase.AcceptedAtRoute\(string, object, object\)](#) , [ControllerBase.Challenge\(\)](#) ,  
[ControllerBase.Challenge\(params string\[\]\)](#) , [ControllerBase.Challenge\(AuthenticationProperties\)](#) ,  
[ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#) , [ControllerBase.Forbid\(\)](#) ,  
[ControllerBase.Forbid\(params string\[\]\)](#) , [ControllerBase.Forbid\(AuthenticationProperties\)](#) ,  
[ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal\)](#) , [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#) ,

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

## Extension Methods

[ControllerBaseExtensions.MapErrorResult\(ControllerBase, Result\)](#)

# Constructors

## TransactionStatusController(IMapper, ITransactionStatusService)

Handles operations related to transaction status management, such as updating status definitions or retrieving status information.

```
public TransactionStatusController(IMapper mapper, ITransactionStatusService  
transactionStatusService)
```

## Parameters

`mapper IMapper`

`transactionStatusService ITransactionStatusService`

## Methods

### GetAllTransactionStatus(TransactionStatusQueryParameters)

Retrieves all transaction status, optionally filtered and sorted by query parameters.

```
[HttpGet]  
public Task<ActionResult< IReadOnlyList< TransactionStatusSummaryDto >>>  
GetAllTransactionStatus(TransactionStatusQueryParameters queryParameters)
```

#### Parameters

`queryParameters TransactionStatusQueryParameters`

Optional filters to apply to the transaction status list.

#### Returns

`Task<ActionResult< IReadOnlyList< TransactionStatusSummaryDto >>>`

Returns `200 OK` with a list of transaction status.

### GetTransactionStatus(long)

Retrieves detailed information of a specific transaction status by its identifier, like all related transactions.

```
[Authorize(Roles = "Admin")]  
[HttpGet("{id:long}")]  
public Task<ActionResult< TransactionStatusDetailsDto >> GetTransactionStatus(long id)
```

#### Parameters

`id long`

The identifier of the transaction status to retrieve.

## Returns

[Task](#) <[ActionResult](#)> <[TransactionStatusDetailsDto](#)>>

Returns **200 OK** with the transaction status details if found and authorized. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user is not allowed to view the transaction status (not admin). Returns **404 Not Found** if the transaction status does not exist.

## UpdateTransactionStatus(long, UpdateTransactionStatusDetailsDto)

Updates details of an existing transaction status partially (name and/or description). Only allowed if the current user is an admin.

```
[Authorize(Roles = "Admin")]
[HttpPut("{id:long}")]
public Task<ActionResult> UpdateTransactionStatus(long id, UpdateTransactionStatusDetailsDto
dto)
```

## Parameters

**id** [long](#)

The identifier of the transaction status to update.

**dto** [UpdateTransactionStatusDetailsDto](#)

Partial transaction status data to update.

## Returns

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

Returns **204 No Content** on successful update. Returns **400 Bad Request** if input validation fails. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user has not the modification rights. Returns **404 Not Found** if the transaction status or a related object does not exist.

# Class TransactionsController

Namespace: [Agora.API.Controllers](#)

Assembly: Agora.API.dll

Manages transactions between users, including creation, updates, and status changes.

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

## Inheritance

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

## 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.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#) ,  
 [ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
 [ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) ,  [ControllerBase.ValidationProblem\(\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
 [ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#) ,  
 [ControllerBase.Created\(\)](#) ,  [ControllerBase.Created\(string, object\)](#) ,  
 [ControllerBase.Created\(Uri, object\)](#) ,  [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object\)](#) ,  [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
 [ControllerBase.CreatedAtRoute\(string, object, object\)](#) ,  [ControllerBase.Accepted\(\)](#) ,  
 [ControllerBase.Accepted\(object\)](#) ,  [ControllerBase.Accepted\(Uri\)](#) ,  [ControllerBase.Accepted\(string\)](#) ,  
 [ControllerBase.Accepted\(string, object\)](#) ,  [ControllerBase.Accepted\(Uri, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string\)](#) ,  [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
 [ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
 [ControllerBase.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\(\)](#)

## Extension Methods

[ControllerBaseExtensions.MapErrorResult\(ControllerBase, Result\)](#)

# Constructors

**TransactionsController(IMapper, IInputValidator, ITransactionService, IUserContextService)**

Manages transactions between users, including creation, updates, and status changes.

```
public TransactionsController(IMapper mapper, IInputValidator inputValidator,
ITransactionService transactionService, IUserContextService userContextService)
```

## Parameters

`mapper IMapper`

`inputValidator IInputValidator`

`transactionService ITransactionService`

`userContextService IUserContextService`

## Methods

### ChangeTransactionStatus(long, ChangeTransactionStatusDto)

Changes the status of an existing transaction (e.g., from Pending to Accepted). This operation is allowed only for the buyer, seller, or administrators.

```
[Authorize]
[HttpPut("{id:long}/status")]
public Task<IActionResult> ChangeTransactionStatus(long id, ChangeTransactionStatusDto dto)
```

#### Parameters

`id long`

The identifier of the transaction whose status is to be changed.

`dto ChangeTransactionStatusDto`

The data transfer object containing the new status.

#### Returns

`Task<IActionResult>`

Returns **204 No Content** on successful status change. Returns **400 Bad Request** if input validation fails or status change is invalid. Returns **401 Unauthorized** if the user is not allowed to change the status. Returns **404 Not Found** if the transaction or a related object does not exist.

### CreateTransaction(CreateTransactionDto)

Creates a new transaction between two users.

```
[Authorize]
[HttpPost]
public Task<ActionResult<TransactionDetailsDto>> CreateTransaction(CreateTransactionDto dto)
```

## Parameters

dto [CreateTransactionDto](#)

The transaction data transfer object containing creation details.

## Returns

[Task](#) <[ActionResult](#) <[TransactionDetailsDto](#)>>

Returns **201 Created** with the newly created transaction details. Returns **400 Bad Request** if input or business rules validation fails. Returns **401 Unauthorized** if the user is not authenticated.

## GetAllTransactions(TransactionQueryParameters)

Retrieves all transactions visible to the current user, optionally filtered by query parameters.

```
[Authorize]
[HttpGet]
public Task<ActionResult<IReadOnlyList<TransactionSummaryDto>>>
GetAllTransactions(TransactionQueryParameters queryParameters)
```

## Parameters

queryParameters [TransactionQueryParameters](#)

Optional filters to apply to the transaction list.

## Returns

[Task](#) <[ActionResult](#) <[IReadOnlyList](#) <[TransactionSummaryDto](#)>>>

Returns **200 OK** with a list of transaction summaries visible to the user.

## GetTransaction(long)

Retrieves detailed information of a specific transaction by its identifier, if the current user is authorized to view it.

```
[Authorize]  
[HttpGet("{id:long}")]  
public Task<ActionResult<TransactionDetailsDto>> GetTransaction(long id)
```

## Parameters

[id \[long\]\(#\)](#)

The identifier of the transaction to retrieve.

## Returns

[Task](#) <[ActionResult](#) <[TransactionDetailsDto](#)>>

Returns [200 OK](#) with the transaction details if found and authorized. Returns [401 Unauthorized](#) if the user is not allowed to view the transaction. Returns [404 Not Found](#) if the transaction does not exist.

## UpdateTransactionDetails([long](#), [UpdateTransactionDetailsDto](#))

Updates details of an existing transaction partially. Only allowed if the current user has modification rights.

```
[Authorize]  
[HttpPatch("{id:long}")]  
public Task<ActionResult> UpdateTransactionDetails(long id, UpdateTransactionDetailsDto dto)
```

## Parameters

[id \[long\]\(#\)](#)

The identifier of the transaction to update.

[dto \[UpdateTransactionDetailsDto\]\(#\)](#)

Partial transaction data to update.

## Returns

## Task <ActionResult>

Returns **204 No Content** on successful update. Returns **400 Bad Request** if input validation fails. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user has not the modification rights for this transaction. Returns **404 Not Found** if the transaction or a related object does not exist.

# Class UsersController

Namespace: [Agora.API.Controllers](#)

Assembly: Agora.API.dll

Handles operations related to user management such as registration, login, logout, and retrieval of user information.

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

## Inheritance

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

## 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\(string, string\)](#)  ,  [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.Problem\(string, string, int?, string, string, IDictionary<string, object>\)](#) ,  
[ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#) ,  
[ControllerBase.ValidationProblem\(ModelStateDictionary\)](#) , [ControllerBase.ValidationProblem\(\)](#) ,  
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#) ,  
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary, IDictionary<string, object>\)](#) ,  
[ControllerBase.Created\(\)](#) , [ControllerBase.Created\(string, object\)](#) ,  
[ControllerBase.Created\(Uri, object\)](#) , [ControllerBase.CreatedAtAction\(string, object\)](#) ,  
[ControllerBase.CreatedAtAction\(string, object, object\)](#) ,  
[ControllerBase.CreatedAtAction\(string, string, object, object\)](#) ,  
[ControllerBase.CreatedAtRoute\(string, object\)](#) , [ControllerBase.CreatedAtRoute\(object, object\)](#) ,  
[ControllerBase.CreatedAtRoute\(string, object, object\)](#) , [ControllerBase.Accepted\(\)](#) ,  
[ControllerBase.Accepted\(object\)](#) , [ControllerBase.Accepted\(Uri\)](#) , [ControllerBase.Accepted\(string\)](#) ,  
[ControllerBase.Accepted\(string, object\)](#) , [ControllerBase.Accepted\(Uri, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string\)](#) , [ControllerBase.AcceptedAtAction\(string, string\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, string, object\)](#) ,  
[ControllerBase.AcceptedAtAction\(string, object, object\)](#) ,  
[ControllerBase.AcceptedAtRoute\(object\)](#) , [ControllerBase.AcceptedAtRoute\(string\)](#) ,  
[ControllerBase.AcceptedAtRoute\(string, object\)](#) , [ControllerBase.AcceptedAtRoute\(object, object\)](#) ,  
[ControllerBase.AcceptedAtRoute\(string, object, object\)](#) , [ControllerBase.Challenge\(\)](#) ,  
[ControllerBase.Challenge\(params string\[\]\)](#) , [ControllerBase.Challenge\(AuthenticationProperties\)](#) ,  
[ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\)](#) , [ControllerBase.Forbid\(\)](#) ,  
[ControllerBase.Forbid\(params string\[\]\)](#) , [ControllerBase.Forbid\(AuthenticationProperties\)](#) ,  
[ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\)](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal\)](#) , [ControllerBase.SignIn\(ClaimsPrincipal, string\)](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\)](#) ,

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

## Extension Methods

[ControllerBaseExtensions.MapErrorResult\(ControllerBase, Result\)](#)

## Constructors

UsersController(IAuthService, IMapper, IUserService,  
IInputValidator, IUserContextService)

Handles operations related to user management such as registration, login, logout, and retrieval of user information.

```
public UsersController(IAuthService authService, IMapper mapper, IUserService userService,  
IInputValidator inputValidator, IUserContextService userContextService)
```

## Parameters

authService [IAuthService](#)

mapper IMapper

userService [IUserService](#)

inputValidator [IInputValidator](#)

userContextService [IUserContextService](#)

## Methods

### GetAllUsers(IUserFilter)

Retrieves a list of all users in the system, optionally filtered and sorted using query parameters.

```
[Authorize(Roles = "Admin")]
[HttpGet]
public Task<ActionResult< IReadOnlyList<UserSummaryDto>>> GetAllUsers(IUserFilter
queryParameters)
```

## Parameters

queryParameters [IUserFilter](#)

Optional filtering and sorting parameters.

## Returns

[Task](#)<[ActionResult](#)<[IReadOnlyList](#)<[UserSummaryDto](#)>>>

Returns [200 OK](#) with a list of summarized user information. Returns [403 Forbidden](#) if the user does not have admin privileges.

### GetCurrentUserAsync()

Retrieves the currently authenticated user's details.

```
[Authorize]
[HttpGet("me")]
public Task<ActionResult<UserDetailsDto>> GetCurrentUserAsync()
```

Returns

[Task](#) <[ActionResult](#) <[UserDetailsDto](#)>>

Returns **200 OK** with the authenticated user's details. Returns **401 Unauthorized** if the user is not authenticated. Returns **404 Not Found** if the user cannot be retrieved.

## GetUser(string)

Retrieves detailed information about a specific user by their ID.

```
[Authorize(Roles = "Admin")]
[HttpGet("{id}")]
public Task<ActionResult<UserDetailsDto>> GetUser(string id)
```

Parameters

[id](#) [string](#)

The unique identifier (GUID) of the user.

Returns

[Task](#) <[ActionResult](#) <[UserDetailsDto](#)>>

Returns **200 OK** with the user's details. Returns **400 BadRequest** if the ID format is invalid. Returns **401 Unauthorized** if the user is not authenticated. Returns **403 Forbidden** if the user does not have admin privileges. Returns **404 Not Found** if the user cannot be retrieved.

## Login(SignInDto)

Authenticates a user and starts a session.

```
[HttpPost("login")]
```

```
public Task<ActionResult> Login(SignInDto dto)
```

## Parameters

dto [SignInDto](#)

The login credentials: email and password.

## Returns

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

Returns **204 No Content** on successful login. Returns **401 Unauthorized** if the credentials are invalid.

## Logout()

Logs out the currently authenticated user.

```
[Authorize]  
[HttpPost("logout")]  
public Task<ActionResult> Logout()
```

## Returns

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

Returns **204 No Content** on successful logout. Returns **401 Unauthorized** if the user was not authenticated.

## Register(RegisterDto)

Registers a new **User** in the system.

```
[HttpPost("register")]  
public Task<ActionResult<UserDetailsDto>> Register(RegisterDto dto)
```

## Parameters

dto [RegisterDto](#)

The registration data including username, email, and password.

## Returns

[Task](#) <[ActionResult](#)> <[UserDetailsDto](#)>>

Returns **200 OK** with the new user's details, if the user was successfully registered. Returns **400 BadRequest** if input validation fails or the user already exists. Returns **500 Internal Server Error** if the user was saved but could not be retrieved afterwards.

# Namespace Agora.API.DTOs.Post

## Classes

[CreatePostDto](#)

[PostDetailsDto](#)

[PostSummaryDto](#)

[UpdatePostDetailsDto](#)

# Class CreatePostDto

Namespace: [Agora.API.DTOs.Post](#)

Assembly: Agora.API.dll

```
public class CreatePostDto
```

## Inheritance

[object](#) ← CreatePostDto

## Inherited Members

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

# Properties

## Description

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(5, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(2000, ErrorMessage = "{0} must be less than {1} characters.")]  
public string Description { get; set; }
```

## Property Value

[string](#)

## PostCategoryId

```
[Required]  
public long PostCategoryId { get; set; }
```

## Property Value

[long](#)

## Price

```
[Required]  
[Range(0, 100000, ErrorMessage = "{0} must be between {1} and {2}.")]  
public int Price { get; set; }
```

### Property Value

[int ↗](#)

## Title

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(100, ErrorMessage = "{0} must be less than {1} characters.")]  
public string Title { get; set; }
```

### Property Value

[string ↗](#)

## Type

```
[Required]  
public string Type { get; set; }
```

### Property Value

[string ↗](#)

# Class PostDetailsDto

Namespace: [Agora.API.DTOs.Post](#)

Assembly: Agora.API.dll

```
public class PostDetailsDto
```

## Inheritance

[object](#) ← PostDetailsDto

## Inherited Members

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

# Properties

## CreatedAt

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

### Property Value

[DateTime](#)

## Description

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

### Property Value

[string](#)

## Id

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

Property Value

[long](#)

PostCategoryName

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

Property Value

[string](#)

Price

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

Property Value

[int](#)

StatusName

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

Property Value

[string](#)

Title

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

Property Value

[string](#)

## Transactions

```
public List<TransactionSummaryDto> Transactions { get; set; }
```

Property Value

[List](#)<[TransactionSummaryDto](#)>

## TypeName

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

Property Value

[string](#)

## UpdatedAt

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

Property Value

[DateTime](#)?

## UserName

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

Property Value

string ↗

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string ↗

A string that represents the current object.

# Class PostSummaryDto

Namespace: [Agora.API.DTOs.Post](#)

Assembly: Agora.API.dll

```
public class PostSummaryDto
```

## Inheritance

[object](#) ← PostSummaryDto

## Inherited Members

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

# Properties

## Id

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

## Property Value

[long](#)

## PostCategoryName

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

## Property Value

[string](#)

## Price

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

Property Value

[int ↗](#)

StatusName

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

Property Value

[string ↗](#)

Title

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

Property Value

[string ↗](#)

TypeName

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

Property Value

[string ↗](#)

UserName

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

Property Value

[string](#) ↗

# Class UpdatePostDetailsDto

Namespace: [Agora.API.DTOs.Post](#)

Assembly: Agora.API.dll

```
public class UpdatePostDetailsDto
```

## Inheritance

[object](#) ← UpdatePostDetailsDto

## Inherited Members

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

# Properties

## Description

```
[MinLength(5, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(2000, ErrorMessage = "{0} must be less than {1} characters.")]  
public string? Description { get; set; }
```

## Property Value

[string](#)

## PostCategoryId

```
public long? PostCategoryId { get; set; }
```

## Property Value

[long](#)?

## Price

```
[Range(0, 100000, ErrorMessage = "{0} must be between {1} and {2}.")]
public int? Price { get; set; }
```

## Property Value

[int](#)?

## Title

```
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]
[MaxLength(100, ErrorMessage = "{0} must be less than {1} characters.")]
public string? Title { get; set; }
```

## Property Value

[string](#)?

## Type

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

## Property Value

[string](#)?

# Namespace Agora.API.DTOs.PostCategory

## Classes

[CreatePostCategoryDto](#)

[PostCategoryDetailsDto](#)

[PostCategorySummaryDto](#)

[UpdatePostCategoryDto](#)

# Class CreatePostCategoryDto

Namespace: [Agora.API.DTOs.PostCategory](#)

Assembly: Agora.API.dll

```
public class CreatePostCategoryDto
```

## Inheritance

[object](#) ← CreatePostCategoryDto

## Inherited Members

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

# Properties

## Name

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(100, ErrorMessage = "{0} must be less than {1} characters.")]  
public string Name { get; set; }
```

## Property Value

[string](#)

# Class PostCategoryDetailsDto

Namespace: [Agora.API.DTOs.PostCategory](#)

Assembly: Agora.API.dll

```
public class PostCategoryDetailsDto
```

## Inheritance

[object](#) ← PostCategoryDetailsDto

## Inherited Members

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

# Properties

## Id

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

### Property Value

[long](#)

## Name

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

### Property Value

[string](#)

## Posts

```
public List<PostSummaryDto> Posts { get; set; }
```

## Property Value

[List](#) <[PostSummaryDto](#)>

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class PostCategorySummaryDto

Namespace: [Agora.API.DTOs.PostCategory](#)

Assembly: Agora.API.dll

```
public class PostCategorySummaryDto
```

## Inheritance

[object](#) ← PostCategorySummaryDto

## Inherited Members

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

# Properties

## Id

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

### Property Value

[long](#)

## Name

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

### Property Value

[string](#)

# Class UpdatePostCategoryDto

Namespace: [Agora.API.DTOs.PostCategory](#)

Assembly: Agora.API.dll

```
public class UpdatePostCategoryDto
```

## Inheritance

[object](#) ← UpdatePostCategoryDto

## Inherited Members

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

# Properties

## NewName

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(100, ErrorMessage = "{0} must be less than {1} characters.")]  
public string NewName { get; set; }
```

## Property Value

[string](#)

# Namespace Agora.API.DTOs.Transaction

## Classes

[ChangeTransactionStatusDto](#)

[CreateTransactionDto](#)

[TransactionDetailsDto](#)

[TransactionSummaryDto](#)

[UpdateTransactionDetailsDto](#)

# Class ChangeTransactionStatusDto

Namespace: [Agora.API.DTOs.Transaction](#)

Assembly: Agora.API.dll

```
public class ChangeTransactionStatusDto
```

## Inheritance

[object](#) ← ChangeTransactionStatusDto

## Inherited Members

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

## Fields

### NewStatus

[Required]

```
public TransactionStatusEnum NewStatus
```

## Field Value

[TransactionStatusEnum](#)

# Class CreateTransactionDto

Namespace: [Agora.API.DTOs.Transaction](#)

Assembly: Agora.API.dll

```
public class CreateTransactionDto
```

## Inheritance

[object](#) ← CreateTransactionDto

## Inherited Members

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

# Properties

## BuyerId

[Required]  
public string BuyerId { get; set; }

### Property Value

[string](#)

## PostId

[Required]  
public long? PostId { get; set; }

### Property Value

[long](#)?

## Price

```
[Required]  
[Range(0, 100000, ErrorMessage = "{0} must be between {1} and {2}.")]  
public int Price { get; set; }
```

## Property Value

[int](#)

## SellerId

```
[Required]  
public string SellerId { get; set; }
```

## Property Value

[string](#)

## Title

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(10, ErrorMessage = "{0} must be at least {1} characters." )]  
[MaxLength(200, ErrorMessage = "{0} must be less than {1} characters." )]  
public string Title { get; set; }
```

## Property Value

[string](#)

## TransactionDate

```
[DataType(DataType.Date)]  
public DateTime? TransactionDate { get; set; }
```

Property Value

[DateTime](#)?

# Class TransactionDetailsDto

Namespace: [Agora.API.DTOs.Transaction](#)

Assembly: Agora.API.dll

```
public class TransactionDetailsDto
```

## Inheritance

[object](#) ← TransactionDetailsDto

## Inherited Members

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

# Properties

## BuyerConfirmed

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

### Property Value

[bool](#)

## BuyerUsername

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

### Property Value

[string](#)

## CompletedAt

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

Property Value

[DateTime](#)?

CreatedAt

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

Property Value

[DateTime](#)

Id

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

Property Value

[long](#)

Post

```
public PostSummaryDto Post { get; set; }
```

Property Value

[PostSummaryDto](#)

Price

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

Property Value

[int ↗](#)

## SellerConfirmed

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

Property Value

[bool ↗](#)

## SellerUsername

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

Property Value

[string ↗](#)

## Title

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

Property Value

[string ↗](#)

## TransactionDate

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

Property Value

[DateTime](#)?

## TransactionStatusName

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

Property Value

[string](#)

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class TransactionSummaryDto

Namespace: [Agora.API.DTOs.Transaction](#)

Assembly: Agora.API.dll

```
public class TransactionSummaryDto
```

## Inheritance

[object](#) ← TransactionSummaryDto

## Inherited Members

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

# Properties

## BuyerUsername

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

### Property Value

[string](#)

## Id

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

### Property Value

[long](#)

## PostTitle

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

Property Value

[string](#)

Price

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

Property Value

[int](#)

SellerUsername

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

Property Value

[string](#)

Title

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

Property Value

[string](#)

TransactionStatusName

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

Property Value

[string](#) ↗

# Class UpdateTransactionDetailsDto

Namespace: [Agora.API.DTOs.Transaction](#)

Assembly: Agora.API.dll

```
public class UpdateTransactionDetailsDto
```

## Inheritance

[object](#) ← UpdateTransactionDetailsDto

## Inherited Members

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

# Properties

## PostId

```
public long? PostId { get; set; }
```

### Property Value

[long](#)?

## Price

```
[Range(0, 100000, ErrorMessage = "{0} must be between {1} and {2}.")]
public int? Price { get; set; }
```

### Property Value

[int](#)?

## Title

```
[MinLength(10, ErrorMessage = "{0} must be at least {1} characters." )]  
[MaxLength(200, ErrorMessage = "{0} must be less than {1} characters." )]  
public string? Title { get; set; }
```

Property Value

[string](#) ↗

## TransactionDate

```
[DataType(DataType.Date)]  
public DateTime? TransactionDate { get; set; }
```

Property Value

[DateTime](#) ↗?

# Namespace Agora.API.DTOs.TransactionStatus

## Classes

[TransactionStatusDetailsDto](#)

[TransactionStatusSummaryDto](#)

[UpdateTransactionStatusDetailsDto](#)

# Class TransactionStatusDetailsDto

Namespace: [Agora.API.DTOs.TransactionStatus](#)

Assembly: Agora.API.dll

```
public class TransactionStatusDetailsDto
```

## Inheritance

[object](#) ← TransactionStatusDetailsDto

## Inherited Members

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

# Properties

## Description

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

### Property Value

[string](#)

## Id

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

### Property Value

[long](#)

## IsFinal

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

Property Value

[bool](#)

## IsSuccess

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

Property Value

[bool](#)

## Name

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

Property Value

[string](#)

## Transactions

```
public ICollection<TransactionSummaryDto> Transactions { get; set; }
```

Property Value

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

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string ↗

A string that represents the current object.

# Class TransactionStatusSummaryDto

Namespace: [Agora.API.DTOs.TransactionStatus](#)

Assembly: Agora.API.dll

```
public class TransactionStatusSummaryDto
```

## Inheritance

[object](#) ← TransactionStatusSummaryDto

## Inherited Members

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

## Properties

### Description

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

#### Property Value

[string](#)

### Id

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

#### Property Value

[long](#)

### IsFinal

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

Property Value

[bool](#)

IsSuccess

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

Property Value

[bool](#)

Name

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

Property Value

[string](#)

# Class UpdateTransactionStatusDetailsDto

Namespace: [Agora.API.DTOs.TransactionStatus](#)

Assembly: Agora.API.dll

```
public class UpdateTransactionStatusDetailsDto
```

## Inheritance

[object](#) ← UpdateTransactionStatusDetailsDto

## Inherited Members

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

# Properties

## Description

```
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(500, ErrorMessage = "{0} must be less than {1} characters.")]  
public string? Description { get; set; }
```

## Property Value

[string](#)

## Name

```
[MinLength(3, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(50, ErrorMessage = "{0} must be less than {1} characters.")]  
public string? Name { get; set; }
```

## Property Value

[string](#)

# Namespace Agora.API.DTOs.User

## Classes

[RegisterDto](#)

[SignInDto](#)

[UserDetailsDto](#)

[UserSummaryDto](#)

# Class RegisterDto

Namespace: [Agora.API.DTOs.User](#)

Assembly: Agora.API.dll

```
public class RegisterDto
```

## Inheritance

[object](#) ← RegisterDto

## Inherited Members

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

# Properties

## Email

```
[Required]  
[NotEmptyOrWhitespace]  
[EmailAddress]  
[MinLength(5, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(255, ErrorMessage = "{0} must be less than {1} characters.")]  
public string Email { get; set; }
```

## Property Value

[string](#)

## Password

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(8, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(255, ErrorMessage = "{0} must be less than {1} characters.")]  
public string Password { get; set; }
```

Property Value

[string](#) ↗

## UserName

```
[Required]  
[NotEmptyOrWhitespace]  
[MinLength(5, ErrorMessage = "{0} must be at least {1} characters.")]  
[MaxLength(50, ErrorMessage = "{0} must be less than {1} characters.")]  
public string UserName { get; set; }
```

Property Value

[string](#) ↗

# Class SignInDto

Namespace: [Agora.API.DTOs.User](#)

Assembly: Agora.API.dll

```
public class SignInDto
```

## Inheritance

[object](#) ← SignInDto

## Inherited Members

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

# Properties

## Email

```
[Required]  
[EmailAddress]  
public string Email { get; set; }
```

## Property Value

[string](#)

## Password

```
[Required]  
public string Password { get; set; }
```

## Property Value

[string](#)

# Class UserDetailsDto

Namespace: [Agora.API.DTOs.User](#)

Assembly: Agora.API.dll

```
public class UserDetailsDto
```

## Inheritance

[object](#) ← UserDetailsDto

## Inherited Members

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

# Properties

## CreatedAt

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

### Property Value

[DateTime](#)

## Credit

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

### Property Value

[int](#)

## Email

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

Property Value

[string](#)

Id

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

Property Value

[string](#)

LastLoginAt

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

Property Value

[DateTime](#)

Posts

```
public List<PostSummaryDto> Posts { get; set; }
```

Property Value

[List](#)<[PostSummaryDto](#)>

TransactionsAsBuyer

```
public List<TransactionSummaryDto> TransactionsAsBuyer { get; set; }
```

Property Value

[List](#) <[TransactionSummaryDto](#)>

## TransactionsAsSeller

```
public List<TransactionSummaryDto> TransactionsAsSeller { get; set; }
```

Property Value

[List](#) <[TransactionSummaryDto](#)>

## UserName

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

Property Value

[string](#)

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class UserSummaryDto

Namespace: [Agora.API.DTOs.User](#)

Assembly: Agora.API.dll

```
public class UserSummaryDto
```

## Inheritance

[object](#) ← UserSummaryDto

## Inherited Members

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

# Properties

## Credit

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

### Property Value

[int](#)

## Email

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

### Property Value

[string](#)

## Id

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

Property Value

[string](#) ↗

## UserName

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

Property Value

[string](#) ↗

# Namespace Agora.API.Extensions

## Classes

[ControllerExtensions](#)

# Class ControllerExtensions

Namespace: [Agora.API.Extensions](#)

Assembly: Agora.API.dll

```
public static class ControllerExtensions
```

## Inheritance

[object](#) ← ControllerExtensions

## Inherited Members

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

## Methods

### MapErrorResult(ControllerBase, Result)

```
public static ActionResult MapErrorResult(this ControllerBase controller, Result result)
```

#### Parameters

controller [ControllerBase](#)

result [Result](#)

#### Returns

[ActionResult](#)

# Namespace Agora.API.Filters

## Classes

[LogActionFilter](#)

# Class LogActionFilter

Namespace: [Agora.API.Filters](#)

Assembly: Agora.API.dll

```
public class LogActionFilter : IActionFilter, IFilterMetadata
```

## Inheritance

[object](#) ← LogActionFilter

## Implements

[IActionFilter](#), [IFilterMetadata](#)

## Inherited Members

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

## Constructors

### LogActionFilter(ILogger<LogActionFilter>)

```
public LogActionFilter(ILogger<LogActionFilter> logger)
```

## Parameters

logger [ILogger](#)<[LogActionFilter](#)>

## Methods

### OnActionExecuted(ActionExecutedContext)

Called after the action executes, before the action result.

```
public void OnActionExecuted(ActionExecutedContext context)
```

## Parameters

`context` [ActionExecutedContext](#)

The [ActionExecutedContext](#).

## OnActionExecuting(ActionExecutingContext)

Called before the action executes, after model binding is complete.

```
public void OnActionExecuting(ActionExecutingContext context)
```

### Parameters

`context` [ActionExecutingContext](#)

The [ActionExecutingContext](#).

# Namespace Agora.API.InputValidation

## Classes

[InputValidationResult](#)

[InputValidator](#)

# Class InputValidationResult

Namespace: [Agora.API.InputValidation](#)

Assembly: Agora.API.dll

```
public class InputValidationResult
```

## Inheritance

[object](#) ← InputValidationResult

## Inherited Members

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

# Properties

## Errors

```
public List<string> Errors { get; }
```

## Property Value

[List](#)<[string](#)>

## IsValid

```
public bool IsValid { get; }
```

## Property Value

[bool](#)

# Class InputValidator

Namespace: [Agora.API.InputValidation](#)

Assembly: Agora.API.dll

```
public class InputValidator : IInputValidator
```

## Inheritance

[object](#) ← InputValidator

## Implements

[IInputValidator](#)

## Inherited Members

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

## Constructors

InputValidator(IUserRepository, IPostCategoryRepository,  
IPostRepository, ITransactionStatusRepository)

```
public InputValidator(IUserRepository userRepo, IPostCategoryRepository postCategoryRepo,  
IPostRepository postRepo, ITransactionStatusRepository transactionStatusRepo)
```

## Parameters

userRepo [IUserRepository](#)

postCategoryRepo [IPostCategoryRepository](#)

postRepo [IPostRepository](#)

transactionStatusRepo [ITransactionStatusRepository](#)

## Methods

## ValidateChangeTransactionStatusDto(ChangeTransactionStatusDto)

Validates the input for a transaction status change request. Ensures that the provided transaction status exists within the TransactionStatusEnum.

```
public InputValidationResult ValidateChangeTransactionStatusDto(ChangeTransactionStatusDto  
dto)
```

Parameters

dto [ChangeTransactionStatusDto](#)

Returns

[InputValidationResult](#)

## ValidateCreatePostDtoAsync(CreatePostDto)

Validates the input data for creating a new [Post](#). Ensures the post type is valid and the specified post category exists.

```
public Task<InputValidationResult> ValidateCreatePostDtoAsync(CreatePostDto dto)
```

Parameters

dto [CreatePostDto](#)

The DTO containing post creation data.

Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateCreateTransactionDtoAsync(CreateTransactionDto)

Validates the input for a new transaction request. Ensures that the involved users and the optional linked post exist in database

```
public Task<InputValidationResult> ValidateCreateTransactionDtoAsync(CreateTransactionDto  
dto)
```

Parameters

dto [CreateTransactionDto](#)

Returns

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

## ValidateRegisterDtoAsync(RegisterDto)

Validates the input data for a new [User](#) registration. Checks for username and email uniqueness.

```
public Task<InputValidationResult> ValidateRegisterDtoAsync(RegisterDto dto)
```

Parameters

dto [RegisterDto](#)

The DTO containing registration data.

Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUpdatePostDtoAsync(UpdatePostDetailsDto)

Validates the input data for updating an existing [Post](#). Checks for valid post type and verifies the existence of the specified post category if provided.

```
public Task<InputValidationResult> ValidateUpdatePostDtoAsync(UpdatePostDetailsDto dto)
```

## Parameters

**dto** [UpdatePostDetailsDto](#)

The DTO containing updated post details.

## Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUpdateTransactionDetailsDtoAsync(UpdateTransactionDetailsDto)

Validates the input data for updating an existing [Transaction](#). Ensures the referenced post (if any) exists.

```
public Task<InputValidationResult>
ValidateUpdateTransactionDetailsDtoAsync(UpdateTransactionDetailsDto dto)
```

## Parameters

**dto** [UpdateTransactionDetailsDto](#)

The DTO containing updated transaction data.

## Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUserId(string)

Validates that the user id respects the required format (GUID).

```
public InputValidationResult ValidateUserId(string id)
```

## Parameters

**id** [string](#)

The user id to check

Returns

[InputValidationResult](#)

An [InputValidationResult](#) containing any validation errors.

# Namespace Agora.API.InputValidation.Interfaces

## Interfaces

[IInputValidator](#)

# Interface IInputValidator

Namespace: [Agora.API.InputValidation.Interfaces](#)

Assembly: Agora.API.dll

```
public interface IInputValidator
```

## Methods

### ValidateChangeTransactionStatusDto(ChangeTransactionStatusDto)

Validates the input data for changing the status of a **Transaction**. Ensures that the provided new status exists in **TransactionStatusEnum**.

```
InputValidationResult ValidateChangeTransactionStatusDto(ChangeTransactionStatusDto dto)
```

#### Parameters

**dto** [ChangeTransactionStatusDto](#)

The DTO containing the requested status change.

#### Returns

[InputValidationResult](#)

An [InputValidationResult](#) containing any validation errors.

### ValidateCreatePostDtoAsync(CreatePostDto)

Validates the input data for creating a new **Post**. Ensures the post type is valid and the specified post category exists.

```
Task<InputValidationResult> ValidateCreatePostDtoAsync(CreatePostDto dto)
```

## Parameters

**dto** [CreatePostDto](#)

The DTO containing post creation data.

## Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateCreateTransactionDtoAsync(CreateTransactionDto)

Validates the input data for creating a new [Transaction](#). Ensures that the referenced post (if any) exists, and that both buyer and seller users are valid.

`Task<InputValidationResult> ValidateCreateTransactionDtoAsync(CreateTransactionDto dto)`

## Parameters

**dto** [CreateTransactionDto](#)

The DTO containing transaction creation data.

## Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateRegisterDtoAsync(RegisterDto)

Validates the input data for a new [User](#) registration. Checks for username and email uniqueness.

`Task<InputValidationResult> ValidateRegisterDtoAsync(RegisterDto dto)`

## Parameters

## **dto** [RegisterDto](#)

The DTO containing registration data.

Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUpdatePostDtoAsync(UpdatePostDetailsDto)

Validates the input data for updating an existing [Post](#). Checks for valid post type and verifies the existence of the specified post category if provided.

```
Task<InputValidationResult> ValidateUpdatePostDtoAsync(UpdatePostDetailsDto dto)
```

Parameters

### **dto** [UpdatePostDetailsDto](#)

The DTO containing updated post details.

Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUpdateTransactionDetailsDtoAsync(UpdateTransactionDetailsDto)

Validates the input data for updating an existing [Transaction](#). Ensures the referenced post (if any) exists.

```
Task<InputValidationResult>
```

```
ValidateUpdateTransactionDetailsDtoAsync(UpdateTransactionDetailsDto dto)
```

Parameters

## **dto** [UpdateTransactionDetailsDto](#)

The DTO containing updated transaction data.

Returns

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

An [InputValidationResult](#) containing any validation errors.

## ValidateUserId(string)

Validates that the user id respects the required format (GUID).

`InputValidationResult ValidateUserId(string id)`

Parameters

### **id** [string](#)

The user id to check

Returns

### [InputValidationResult](#)

An [InputValidationResult](#) containing any validation errors.

# Namespace Agora.API.Mapping

## Classes

[PostCategoryProfile](#)

[PostProfile](#)

[TransactionProfile](#)

[TransactionStatusProfile](#)

[UserProfile](#)

# Class PostCategoryProfile

Namespace: [Agora.API.Mapping](#)

Assembly: Agora.API.dll

```
public class PostCategoryProfile : Profile, IProfileExpression, IProfileConfiguration
```

## Inheritance

[object](#) ← Profile ← PostCategoryProfile

## Implements

IProfileExpression, IProfileConfiguration

## Inherited Members

Profile.DisableConstructorMapping() , Profile.CreateProjection<TSource, TDestination>() ,  
Profile.CreateProjection<TSource, TDestination>(MemberList) ,  
Profile.CreateMap<TSource, TDestination>() , Profile.CreateMap<TSource, TDestination>(MemberList) ,  
[Profile.CreateMap\(Type, Type\)](#) , [Profile.CreateMap\(Type, Type, MemberList\)](#) , Profile.ClearPrefixes() ,  
[Profile.ReplaceMemberName\(string, string\)](#) , [Profile.RecognizePrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizePostfixes\(params string\[\]\)](#) , [Profile.RecognizeDestinationPrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizeDestinationPostfixes\(params string\[\]\)](#) , [Profile.AddGlobalIgnore\(string\)](#) ,  
[Profile.IncludeSourceExtensionMethods\(Type\)](#) , Profile.ProfileName ,  
Profile.AllowNullDestinationValues , Profile.AllowNullCollections , Profile.ShouldMapProperty ,  
Profile.ShouldMapField , Profile.ShouldMapMethod , Profile.ShouldUseConstructor ,  
Profile.SourceMemberNamingConvention , Profile.DestinationMemberNamingConvention ,  
Profile.ValueTransformers , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### PostCategoryProfile()

```
public PostCategoryProfile()
```

# Class PostProfile

Namespace: [Agora.API.Mapping](#)

Assembly: Agora.API.dll

```
public class PostProfile : Profile, IProfileExpression, IProfileConfiguration
```

## Inheritance

[object](#) ← Profile ← PostProfile

## Implements

IProfileExpression, IProfileConfiguration

## Inherited Members

Profile.DisableConstructorMapping() , Profile.CreateProjection<TSource, TDestination>() ,  
Profile.CreateProjection<TSource, TDestination>(MemberList) ,  
Profile.CreateMap<TSource, TDestination>() , Profile.CreateMap<TSource, TDestination>(MemberList) ,  
[Profile.CreateMap\(Type, Type\)](#) , [Profile.CreateMap\(Type, Type, MemberList\)](#) , Profile.ClearPrefixes() ,  
[Profile.ReplaceMemberName\(string, string\)](#) , [Profile.RecognizePrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizePostfixes\(params string\[\]\)](#) , [Profile.RecognizeDestinationPrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizeDestinationPostfixes\(params string\[\]\)](#) , [Profile.AddGlobalIgnore\(string\)](#) ,  
[Profile.IncludeSourceExtensionMethods\(Type\)](#) , Profile.ProfileName ,  
Profile.AllowNullDestinationValues , Profile.AllowNullCollections , Profile.ShouldMapProperty ,  
Profile.ShouldMapField , Profile.ShouldMapMethod , Profile.ShouldUseConstructor ,  
Profile.SourceMemberNamingConvention , Profile.DestinationMemberNamingConvention ,  
Profile.ValueTransformers , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## PostProfile()

```
public PostProfile()
```

# Class TransactionProfile

Namespace: [Agora.API.Mapping](#)

Assembly: Agora.API.dll

```
public class TransactionProfile : Profile, IProfileExpression, IProfileConfiguration
```

## Inheritance

[object](#) ← Profile ← TransactionProfile

## Implements

IProfileExpression, IProfileConfiguration

## Inherited Members

Profile.DisableConstructorMapping() , Profile.CreateProjection<TSource, TDestination>() ,  
Profile.CreateProjection<TSource, TDestination>(MemberList) ,  
Profile.CreateMap<TSource, TDestination>() , Profile.CreateMap<TSource, TDestination>(MemberList) ,  
[Profile.CreateMap\(Type, Type\)](#) , [Profile.CreateMap\(Type, Type, MemberList\)](#) , Profile.ClearPrefixes() ,  
[Profile.ReplaceMemberName\(string, string\)](#) , [Profile.RecognizePrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizePostfixes\(params string\[\]\)](#) , [Profile.RecognizeDestinationPrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizeDestinationPostfixes\(params string\[\]\)](#) , [Profile.AddGlobalIgnore\(string\)](#) ,  
[Profile.IncludeSourceExtensionMethods\(Type\)](#) , Profile.ProfileName ,  
Profile.AllowNullDestinationValues , Profile.AllowNullCollections , Profile.ShouldMapProperty ,  
Profile.ShouldMapField , Profile.ShouldMapMethod , Profile.ShouldUseConstructor ,  
Profile.SourceMemberNamingConvention , Profile.DestinationMemberNamingConvention ,  
Profile.ValueTransformers , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### TransactionProfile()

```
public TransactionProfile()
```

# Class TransactionStatusProfile

Namespace: [Agora.API.Mapping](#)

Assembly: Agora.API.dll

```
public class TransactionStatusProfile : Profile, IProfileExpression, IProfileConfiguration
```

## Inheritance

[object](#) ← Profile ← TransactionStatusProfile

## Implements

IProfileExpression, IProfileConfiguration

## Inherited Members

Profile.DisableConstructorMapping() , Profile.CreateProjection<TSource, TDestination>() ,  
Profile.CreateProjection<TSource, TDestination>(MemberList) ,  
Profile.CreateMap<TSource, TDestination>() , Profile.CreateMap<TSource, TDestination>(MemberList) ,  
[Profile.CreateMap\(Type, Type\)](#) , [Profile.CreateMap\(Type, Type, MemberList\)](#) , Profile.ClearPrefixes() ,  
[Profile.ReplaceMemberName\(string, string\)](#) , [Profile.RecognizePrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizePostfixes\(params string\[\]\)](#) , [Profile.RecognizeDestinationPrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizeDestinationPostfixes\(params string\[\]\)](#) , [Profile.AddGlobalIgnore\(string\)](#) ,  
[Profile.IncludeSourceExtensionMethods\(Type\)](#) , Profile.ProfileName ,  
Profile.AllowNullDestinationValues , Profile.AllowNullCollections , Profile.ShouldMapProperty ,  
Profile.ShouldMapField , Profile.ShouldMapMethod , Profile.ShouldUseConstructor ,  
Profile.SourceMemberNamingConvention , Profile.DestinationMemberNamingConvention ,  
Profile.ValueTransformers , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### TransactionStatusProfile()

```
public TransactionStatusProfile()
```

# Class UserProfile

Namespace: [Agora.API.Mapping](#)

Assembly: Agora.API.dll

```
public class UserProfile : Profile, IProfileExpression, IProfileConfiguration
```

## Inheritance

[object](#) ← Profile ← UserProfile

## Implements

IProfileExpression, IProfileConfiguration

## Inherited Members

Profile.DisableConstructorMapping() , Profile.CreateProjection<TSource, TDestination>() ,  
Profile.CreateProjection<TSource, TDestination>(MemberList) ,  
Profile.CreateMap<TSource, TDestination>() , Profile.CreateMap<TSource, TDestination>(MemberList) ,  
[Profile.CreateMap\(Type, Type\)](#) , [Profile.CreateMap\(Type, Type, MemberList\)](#) , Profile.ClearPrefixes() ,  
[Profile.ReplaceMemberName\(string, string\)](#) , [Profile.RecognizePrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizePostfixes\(params string\[\]\)](#) , [Profile.RecognizeDestinationPrefixes\(params string\[\]\)](#) ,  
[Profile.RecognizeDestinationPostfixes\(params string\[\]\)](#) , [Profile.AddGlobalIgnore\(string\)](#) ,  
[Profile.IncludeSourceExtensionMethods\(Type\)](#) , Profile.ProfileName ,  
Profile.AllowNullDestinationValues , Profile.AllowNullCollections , Profile.ShouldMapProperty ,  
Profile.ShouldMapField , Profile.ShouldMapMethod , Profile.ShouldUseConstructor ,  
Profile.SourceMemberNamingConvention , Profile.DestinationMemberNamingConvention ,  
Profile.ValueTransformers , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Constructors

## UserProfile()

```
public UserProfile()
```

# Namespace Agora.API.QueryParams

## Classes

[PostCategoryQueryParameters](#)

[PostQueryParameters](#)

[TransactionQueryParameters](#)

[TransactionStatusQueryParameters](#)

[UserQueryParameters](#)

# Class PostCategoryQueryParameters

Namespace: [Agora.API.QueryParams](#)

Assembly: Agora.API.dll

```
public class PostCategoryQueryParameters : IPostCategoryFilter
```

## Inheritance

[object](#) ← PostCategoryQueryParameters

## Implements

[IPostCategoryFilter](#)

## Inherited Members

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

# Properties

## Name

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

## Property Value

[string](#)

## SortBy

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

## Property Value

[string](#)

## SortDesc

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

Property Value

[bool](#) ↗

# Class PostQueryParameters

Namespace: [Agora.API.QueryParams](#)

Assembly: Agora.API.dll

```
public class PostQueryParameters : IPostFilter
```

## Inheritance

[object](#) ← PostQueryParameters

## Implements

[IPostFilter](#)

## Inherited Members

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

# Properties

## MaxPrice

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

## Property Value

[int](#)?

## MinPrice

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

## Property Value

[int](#)?

## PostCategoryName

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

### Property Value

[string ↗](#)

## SortBy

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

### Property Value

[string ↗](#)

## SortDesc

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

### Property Value

[bool ↗](#)

## TitleOrDescription

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

### Property Value

[string ↗](#)

## TypeName

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

Property Value

[string](#) ↗

## UserName

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

Property Value

[string](#) ↗

# Class TransactionQueryParameters

Namespace: [Agora.API.QueryParams](#)

Assembly: Agora.API.dll

```
public class TransactionQueryParameters : ITransactionFilter
```

## Inheritance

[object](#) ← TransactionQueryParameters

## Implements

[ITransactionFilter](#)

## Inherited Members

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

# Properties

## MaxPrice

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

## Property Value

[int](#)?

## MinPrice

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

## Property Value

[int](#)?

## PostTitle

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

### Property Value

[string ↗](#)

## SortBy

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

### Property Value

[string ↗](#)

## SortDesc

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

### Property Value

[bool ↗](#)

## TransactionStatusName

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

### Property Value

[string ↗](#)

## UsersInvolvedUsername

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

Property Value

[string](#) ↗

# Class TransactionStatusQueryParameters

Namespace: [Agora.API.QueryParams](#)

Assembly: Agora.API.dll

```
public class TransactionStatusQueryParameters : ITransactionStatusFilter
```

## Inheritance

[object](#) ← TransactionStatusQueryParameters

## Implements

[ITransactionStatusFilter](#)

## Inherited Members

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

# Properties

## IsFinal

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

### Property Value

[bool](#)?

## IsSuccess

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

### Property Value

[bool](#)?

## NameOrDescription

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

Property Value

[string ↗](#)

## SortBy

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

Property Value

[string ↗](#)

## SortDesc

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

Property Value

[bool ↗](#)

# Class UserQueryParameters

Namespace: [Agora.API.QueryParams](#)

Assembly: Agora.API.dll

```
public class UserQueryParameters : IUserFilter
```

## Inheritance

[object](#) ← UserQueryParameters

## Implements

[IUserFilter](#)

## Inherited Members

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

# Properties

## Email

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

## Property Value

[string](#)

## MaxCreatedAt

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

## Property Value

[DateTime](#)?

## MaxCredit

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

### Property Value

[int](#)?

## MinCreatedAt

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

### Property Value

[DateTime](#)?

## MinCredit

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

### Property Value

[int](#)?

## SortBy

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

### Property Value

[string](#)?

## SortDesc

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

Property Value

[bool](#) ↗

Username

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

Property Value

[string](#) ↗

# Namespace Agora.API.Settings

## Classes

[UserSettings](#)

# Class UserSettings

Namespace: [Agora.API.Settings](#)

Assembly: Agora.Core.dll

```
public class UserSettings
```

## Inheritance

[object](#) ← UserSettings

## Inherited Members

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

# Properties

## InitialCredit

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

## Property Value

[int](#)

# Namespace Agora.Core.BusinessRules

## Classes

### [AuthorizationBusinessRules](#)

Determine if a certain user can see, create, modify or delete some entity.

### [BusinessRulesValidator](#)

# Class AuthorizationBusinessRules

Namespace: [Agora.Core.BusinessRules](#)

Assembly: Agora.Core.dll

Determine if a certain user can see, create, modify or delete some entity.

```
public class AuthorizationBusinessRules : IAuthorizationBusinessRules
```

Inheritance

[object](#) ← AuthorizationBusinessRules

Implements

[IAuthorizationBusinessRules](#)

Inherited Members

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

## Methods

CanManagePost(Post, UserContext)

```
public bool CanManagePost(Post post, UserContext userContext)
```

Parameters

post [Post](#)

userContext [UserContext](#)

Returns

[bool](#)

CanManageTransaction(Transaction, UserContext)

```
public bool CanManageTransaction(Transaction transaction, UserContext userContext)
```

Parameters

transaction [Transaction](#)

userContext [UserContext](#)

Returns

[bool](#) ↗

## CanViewPost(Post, UserContext)

```
public bool CanViewPost(Post post, UserContext userContext)
```

Parameters

post [Post](#)

userContext [UserContext](#)

Returns

[bool](#) ↗

## CanViewTransaction(Transaction, UserContext)

```
public bool CanViewTransaction(Transaction transaction, UserContext userContext)
```

Parameters

transaction [Transaction](#)

userContext [UserContext](#)

Returns

[bool](#)

## CanViewUser(string, UserContext)

```
public bool CanViewUser(string userId, UserContext userContext)
```

### Parameters

userId [string](#)

userContext [UserContext](#)

### Returns

[bool](#)

# Class BusinessRulesValidator

Namespace: [Agora.Core.BusinessRules](#)

Assembly: Agora.Core.dll

```
public class BusinessRulesValidator : IBusinessRulesValidator
```

## Inheritance

[object](#) ← BusinessRulesValidator

## Implements

[IBusinessRulesValidator](#)

## Inherited Members

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

## Constructors

BusinessRulesValidator(IPostRepository,  
IPostCategoryRepository, ITransactionStatusRepository,  
ITransactionRepository)

```
public BusinessRulesValidator(IPostRepository postRepo, IPostCategoryRepository  
postCategoryRepo, ITransactionStatusRepository transactionStatusRepo,  
ITransactionRepository transactionRepo)
```

## Parameters

postRepo [IPostRepository](#)

postCategoryRepo [IPostCategoryRepository](#)

transactionStatusRepo [ITransactionStatusRepository](#)

transactionRepo [ITransactionRepository](#)

# Methods

## ValidateNewPostAsync(Post, UserContext)

Validates whether a new post can be created by the current user based on business rules.

```
public Task<Result> ValidateNewPostAsync(Post newPost, UserContext userContext)
```

### Parameters

**newPost** [Post](#)

The new post to validate.

**userContext** [UserContext](#)

The context of the user attempting to create the post.

### Returns

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

A [Result](#) indicating success if the post is valid, or failure if a rule is violated (e.g., duplicate title).

### Remarks

The validation checks that the user does not already have another post (active or inactive) with the same title as the new post.

## ValidateNewPostCategoryAsync(PostCategory)

Validates the creation of a new post category by checking business rules, such as ensuring the category name does not already exist.

```
public Task<Result> ValidateNewPostCategoryAsync(PostCategory postCategory)
```

### Parameters

**postCategory** [PostCategory](#)

The post category entity to validate.

Returns

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

A [Result](#) indicating whether the post category passes validation. Returns a failure result if the name is already taken or violates other business rules.

## ValidatePostCategoryDeletionAsync(PostCategory)

Validates the deletion of an existing post category by verifying it is not used by any post.

```
public Task<Result> ValidatePostCategoryDeletionAsync(PostCategory postCategory)
```

Parameters

#### [postCategory](#) [PostCategory](#)

The post category to delete.

Returns

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

A [Result](#) indicating whether the deletion is valid. Returns a failure result if the category is used by one or several posts.

## ValidatePostCategoryUpdateAsync(PostCategory, string)

Validates the update of an existing post category by verifying business rules, such as ensuring the new name is different from the current one and does not already exist.

```
public Task<Result> ValidatePostCategoryUpdateAsync(PostCategory postCategory,  
string newName)
```

Parameters

#### [postCategory](#) [PostCategory](#)

The existing post category entity.

`newName` [string](#)

The new name proposed for the post category.

Returns

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

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

## ValidatePostDeletionAsync(Post, UserContext)

Validates whether a post can be deleted by a specific user, taking into account business rules such as ownership, status, and active transactions.

`public Task<Result> ValidatePostDeletionAsync(Post post, UserContext userContext)`

Parameters

`post` [Post](#)

The post that is requested to be deleted.

`userContext` [UserContext](#)

The context of the user attempting to delete the post.

Returns

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

A [Task<TResult>](#) representing the asynchronous validation result. The result indicates whether deletion is permitted and includes an error message if it is not.

## ValidatePostStatusChangeAsync(Post, PostStatus)

Validates whether a post is allowed to transition from its current status to a new specified status, based on business rules and the current state of the post.

```
public Result ValidatePostStatusChangeAsync(Post oldPost, PostStatus newStatus)
```

## Parameters

### oldPost [Post](#)

The existing post with its current status and related data.

### newStatus [PostStatus](#)

The new status to which the post is intended to transition.

## Returns

### [Result](#)

A [Result](#) indicating whether the status change is valid, including an error message if invalid.

## ValidatePostUpdateAsync(Post, string?, UserContext)

Validates the update of an existing post by verifying business rules, such as ensuring the new title is not duplicate with a title of another post of the user.

```
public Task<Result> ValidatePostUpdateAsync(Post oldPost, string? newTitle,  
UserContext userContext)
```

## Parameters

### oldPost [Post](#)

The existing post entity.

### newTitle [string](#) ↗

The new title proposed for the post.

### userContext [UserContext](#)

The context of the user attempting to update the post.

## Returns

## Task < Result>

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

## ValidateTransaction(Transaction)

Validates the internal consistency and business rules of a transaction (e.g., buyer/seller identity, ownership, credit).

```
public Result ValidateTransaction(Transaction transaction)
```

### Parameters

**transaction** [Transaction](#)

The transaction to validate.

### Returns

[Result](#)

A result indicating success if the transaction is valid; otherwise, a result containing the relevant validation errors.

## ValidateTransactionStatusChange(Transaction, TransactionStatusEnum, TransactionStatusEnum, UserContext)

Validates whether a user is authorized to change a transaction's status from the specified old status to the new status, based on business rules and the current user's context.

```
public Result ValidateTransactionStatusChange(Transaction transaction, TransactionStatusEnum oldStatus, TransactionStatusEnum newStatus, UserContext userContext)
```

### Parameters

**transaction** [Transaction](#)

The transaction being updated.

### **oldStatus** [TransactionStatusEnum](#)

The current status of the transaction.

### **newStatus** [TransactionStatusEnum](#)

The desired new status for the transaction.

### **userContext** [UserContext](#)

Information about the current user's identity and role.

Returns

### [Result](#)

A failure result if the transition is not permitted; otherwise, a success result.

## ValidateTransactionStatusUpdateAsync(TransactionStatus, TransactionStatusDetailsUpdate)

Validates the update of an existing transaction status by verifying business rules, such as ensuring the new name is not duplicate with a name of another transaction status.

```
public Task<Result> ValidateTransactionStatusUpdateAsync(TransactionStatus  
oldTransactionStatus, TransactionStatusDetailsUpdate newDetails)
```

Parameters

### **oldTransactionStatus** [TransactionStatus](#)

The existing transaction status entity.

### **newDetails** [TransactionStatusDetailsUpdate](#)

The new data proposed for the transaction status.

Returns

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

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

# Namespace Agora.Core.BusinessRules.Interfaces

## Interfaces

### [IAuthorizationBusinessRules](#)

Determine if a certain user can see, create, modify or delete some entity.

### [IBusinessRulesValidator](#)

# Interface IAuthorizationBusinessRules

Namespace: [Agora.Core.BusinessRules.Interfaces](#)

Assembly: Agora.Core.dll

Determine if a certain user can see, create, modify or delete some entity.

```
public interface IAuthorizationBusinessRules
```

## Methods

### CanManagePost(Post, UserContext)

```
bool CanManagePost(Post post, UserContext userContext)
```

Parameters

post [Post](#)

userContext [UserContext](#)

Returns

[bool](#) ↗

### CanManageTransaction(Transaction, UserContext)

```
bool CanManageTransaction(Transaction transaction, UserContext userContext)
```

Parameters

transaction [Transaction](#)

userContext [UserContext](#)

Returns

[bool](#)

## CanViewPost(Post, UserContext)

`bool CanViewPost(Post post, UserContext userContext)`

Parameters

`post` [Post](#)

`userContext` [UserContext](#)

Returns

[bool](#)

## CanViewTransaction(Transaction, UserContext)

`bool CanViewTransaction(Transaction transaction, UserContext userContext)`

Parameters

`transaction` [Transaction](#)

`userContext` [UserContext](#)

Returns

[bool](#)

## CanViewUser(string, UserContext)

`bool CanViewUser(string id, UserContext userContext)`

Parameters

**id** [string](#)

**userContext** [UserContext](#)

Returns

[bool](#)

# Interface IBusinessRulesValidator

Namespace: [Agora.Core.BusinessRules.Interfaces](#)

Assembly: Agora.Core.dll

```
public interface IBusinessRulesValidator
```

## Methods

### ValidateNewPostAsync(Post, UserContext)

Validates whether a new post can be created by the current user based on business rules.

```
Task<Result> ValidateNewPostAsync(Post newPost, UserContext userContext)
```

#### Parameters

**newPost** [Post](#)

The new post to validate.

**userContext** [UserContext](#)

The context of the user attempting to create the post.

#### Returns

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

A [Result](#) indicating success if the post is valid, or failure if a rule is violated (e.g., duplicate title).

#### Remarks

The validation checks that the user does not already have another post (active or inactive) with the same title as the new post.

### ValidateNewPostCategoryAsync(PostCategory)

Validates the creation of a new post category by checking business rules, such as ensuring the category name does not already exist.

`Task<Result> ValidateNewPostCategoryAsync(PostCategory postCategory)`

## Parameters

`postCategory` [PostCategory](#)

The post category entity to validate.

## Returns

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

A [Result](#) indicating whether the post category passes validation. Returns a failure result if the name is already taken or violates other business rules.

## ValidatePostCategoryDeletionAsync(PostCategory)

Validates the deletion of an existing post category by verifying it is not used by any post.

`Task<Result> ValidatePostCategoryDeletionAsync(PostCategory postCategory)`

## Parameters

`postCategory` [PostCategory](#)

The post category to delete.

## Returns

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

A [Result](#) indicating whether the deletion is valid. Returns a failure result if the category is used by one or several posts.

## ValidatePostCategoryUpdateAsync(PostCategory, string)

Validates the update of an existing post category by verifying business rules, such as ensuring the new name is different from the current one and does not already exist.

```
Task<Result> ValidatePostCategoryUpdateAsync(PostCategory postCategory, string newName)
```

## Parameters

**postCategory** [PostCategory](#)

The existing post category entity.

**newName** [string](#)

The new name proposed for the post category.

## Returns

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

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

## ValidatePostDeletionAsync(Post, UserContext)

Validates whether a post can be deleted by a specific user, taking into account business rules such as ownership, status, and active transactions.

```
Task<Result> ValidatePostDeletionAsync(Post post, UserContext userContext)
```

## Parameters

**post** [Post](#)

The post that is requested to be deleted.

**userContext** [UserContext](#)

The context of the user attempting to delete the post.

## Returns

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

A [Task<TResult>](#) representing the asynchronous validation result. The result indicates whether deletion is permitted and includes an error message if it is not.

## ValidatePostStatusChangeAsync(Post, PostStatus)

Validates whether a post is allowed to transition from its current status to a new specified status, based on business rules and the current state of the post.

Result **ValidatePostStatusChangeAsync**(Post oldPost, PostStatus newStatus)

### Parameters

**oldPost** [Post](#)

The existing post with its current status and related data.

**newStatus** [PostStatus](#)

The new status to which the post is intended to transition.

### Returns

[Result](#)

A [Result](#) indicating whether the status change is valid, including an error message if invalid.

## ValidatePostUpdateAsync(Post, string?, UserContext)

Validates the update of an existing post by verifying business rules, such as ensuring the new title is not duplicate with a title of another post of the user.

Task<Result> **ValidatePostUpdateAsync**(Post oldPost, string? newTitle, UserContext userContext)

### Parameters

**oldPost** [Post](#)

The existing post entity.

#### [newTitle](#) [string](#)

The new title proposed for the post.

#### [userContext](#) [UserContext](#)

The context of the user attempting to update the post.

Returns

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

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

## ValidateTransaction(Transaction)

Validates the internal consistency and business rules of a transaction (e.g., buyer/seller identity, ownership, credit).

Result [ValidateTransaction](#)([Transaction](#) [transaction](#))

Parameters

#### [transaction](#) [Transaction](#)

The transaction to validate.

Returns

#### [Result](#)

A result indicating success if the transaction is valid; otherwise, a result containing the relevant validation errors.

## ValidateTransactionStatusChange([Transaction](#), [TransactionStatusEnum](#), [TransactionStatusEnum](#), [UserContext](#))

Validates whether a user is authorized to change a transaction's status from the specified old status to the new status, based on business rules and the current user's context.

```
Result ValidateTransactionStatusChange(Transaction transaction, TransactionStatusEnum  
oldStatus, TransactionStatusEnum newStatus, UserContext userContext)
```

## Parameters

**transaction** [Transaction](#)

The transaction being updated.

**oldStatus** [TransactionStatusEnum](#)

The current status of the transaction.

**newStatus** [TransactionStatusEnum](#)

The desired new status for the transaction.

**userContext** [UserContext](#)

Information about the current user's identity and role.

## Returns

[Result](#)

A failure result if the transition is not permitted; otherwise, a success result.

## ValidateTransactionStatusUpdateAsync(TransactionStatus, TransactionStatusDetailsUpdate)

Validates the update of an existing transaction status by verifying business rules, such as ensuring the new name is not duplicate with a name of another transaction status.

```
Task<Result> ValidateTransactionStatusUpdateAsync(TransactionStatus oldTransactionStatus,  
TransactionStatusDetailsUpdate newDetails)
```

## Parameters

**oldTransactionStatus** [TransactionStatus](#)

The existing transaction status entity.

**newDetails** [TransactionStatusDetailsUpdate](#)

The new data proposed for the transaction status.

Returns

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

A [Result](#) indicating whether the update is valid. Returns a failure result if the new name is identical to the current name or already exists.

# Namespace Agora.Core.BusinessServices

## Classes

### [PostCategoryService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

### [PostService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

### [TransactionService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

### [TransactionStatusService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

### [UserService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

# Class PostCategoryService

Namespace: [Agora.Core.BusinessServices](#)

Assembly: Agora.Core.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class PostCategoryService : IPostCategoryService
```

Inheritance

[object](#) ← PostCategoryService

Implements

[IPostCategoryService](#)

Inherited Members

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

## Constructors

PostCategoryService(IPostCategoryRepository,  
IBusinessRulesValidator)

```
public PostCategoryService(IPostCategoryRepository postCategoryRepo,  
IBusinessRulesValidator businessRulesValidator)
```

Parameters

postCategoryRepo [IPostCategoryRepository](#)

businessRulesValidator [IBusinessRulesValidator](#)

## Methods

## CreatePostCategoryAsync(PostCategory)

Creates a new [PostCategory](#) after validating authorization.

```
public Task<Result<PostCategory>> CreatePostCategoryAsync(PostCategory postCategory)
```

Parameters

[postCategory](#) [PostCategory](#)

The [PostCategory](#) entity to create.

Returns

[Task](#) <[Result](#)<[PostCategory](#)>>

Success wrapping the created [PostCategory](#) if successful, or failure with relevant error details.

## DeletePostCategoryAsync(long)

Delete a [PostCategory](#) after validating authorization and business rules.

```
public Task<Result> DeletePostCategoryAsync(long postCategoryId)
```

Parameters

[postCategoryId](#) [long](#)

ID of the [PostCategory](#) to delete.

Returns

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

Success if the [PostCategory](#) was successfully deleted, or failure with relevant error details if invalid or unauthorized.

## GetAllPostCategoriesAsync(IPostCategoryFilter)

Retrieves all **PostCategory**, possibly filtered and sorted.

```
public Task<Result< IReadOnlyList<PostCategory>>>
GetAllPostCategoriesAsync(IPostCategoryFilter postCategoryFilter)
```

Parameters

**postCategoryFilter** [IPostCategoryFilter](#)

Filter criteria to apply on post categories.

Returns

[Task](#) <Result< IReadOnlyList <PostCategory>>>

A successful Result wrapping a list of **PostCategory**, or failure if an error occurs.

## GetPostCategoryByIdAsync(long)

Retrieves a single **PostCategory** by its ID.

```
public Task<Result<PostCategory>> GetPostCategoryByIdAsync(long postCategoryId)
```

Parameters

**postCategoryId** [long](#)

The ID of the **PostCategory** to retrieve.

Returns

[Task](#) <Result<PostCategory>>

Success wrapping the **PostCategory** if found, failure with NotFound if missing.

## UpdatePostCategoryNameAsync(long, string)

Updates name of an existing **PostCategory** after validating authorization and business rules.

```
public Task<Result> UpdatePostCategoryNameAsync(long postCategoryId, string newName)
```

## Parameters

**postCategoryId** [long](#)

ID of the [PostCategory](#) to update.

**newName** [string](#)

The new name of the [PostCategory](#).

## Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Class PostService

Namespace: [Agora.Core.BusinessServices](#)

Assembly: Agora.Core.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class PostService : IPostService
```

Inheritance

[object](#) ← PostService

Implements

[IPostService](#)

Inherited Members

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

## Constructors

PostService(IPostRepository, ITransactionRepository,  
IAuthorizationBusinessRules, IBusinessRulesValidator)

```
public PostService(IPostRepository postRepo, ITransactionRepository  
transactionRepo, IAuthorizationBusinessRules authorizationBusinessRules,  
IBusinessRulesValidator businessRulesValidator)
```

Parameters

postRepo [IPostRepository](#)

transactionRepo [ITransactionRepository](#)

authorizationBusinessRules [IAuthorizationBusinessRules](#)

businessRulesValidator [IBusinessRulesValidator](#)

# Methods

## ChangePostStatusAsync(long, UserContext, PostStatus)

Changes the status of an existing [Post](#) after authorization and validation of state transitions.

```
public Task<Result> ChangePostStatusAsync(long postId, UserContext userContext,  
PostStatus targetStatus)
```

### Parameters

**postId** [long](#)

ID of the [Post](#) to update status for.

**userContext** [UserContext](#)

Context of the user performing the status change.

**targetStatus** [PostStatus](#)

The new post status to apply.

### Returns

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

Success if the status was changed and saved successfully, or failure with relevant error details if invalid or unauthorized.

## CreatePostAsync(Post, UserContext)

Creates a new [Post](#) after validating authorization and business rules.

```
public Task<Result<Post>> CreatePostAsync(Post post, UserContext userContext)
```

### Parameters

**post** [Post](#)

The [Post](#) entity to create.

## `userContext` [UserContext](#)

Context of the user creating the post.

Returns

[Task](#) <[Result](#)<[Post](#)>>

Success wrapping the created [Post](#) if successful, or failure with relevant error details.

## `DeletePostAsync(long, UserContext)`

Delete a [Post](#) after validating authorization and business rules.

```
public Task<Result> DeletePostAsync(long postId, UserContext userContext)
```

Parameters

`postId` [long](#)

ID of the [Post](#) to delete.

`userContext` [UserContext](#)

Context of the user requesting the deletion.

Returns

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

Success if the [Post](#) was successfully deleted, or failure with relevant error details if invalid or unauthorized.

## `GetAllPostsAsync(PostVisibilityMode, PostFilter, UserContext?)`

Retrieves all [Post](#) visible to the user according to authorization rules and visibility mode.

```
public Task<Result<IReadOnlyList<Post>>> GetAllPostsAsync(PostVisibilityMode  
postVisibilityMode, PostFilter postFilter, UserContext? userContext)
```

## Parameters

### postVisibilityMode [PostVisibilityMode](#)

Specifies the scope of posts to retrieve (e.g., catalog only, user-owned, admin view).

### postFilter [PostFilter](#)

Filter criteria to apply when querying posts.

### userContext [UserContext](#)

Context of the current user making the request.

## Returns

### [Task](#) <[Result](#)<|[ReadOnlyList](#)<[Post](#)>>>

A successful Result wrapping a list of posts visible to user, or failure if an error occurs.

## GetPostByIdAsync(long, UserContext?)

Retrieves a single **Post** by its ID if visible to the user.

```
public Task<Result<Post>> GetPostByIdAsync(long postId, UserContext? userContext)
```

## Parameters

### postId [long](#)

The ID of the post to retrieve.

### userContext [UserContext](#)

Context of the current user requesting the post.

## Returns

### [Task](#) <[Result](#)<[Post](#)>>

Success wrapping the **Post** if found and authorized, failure with NotFound if missing, or failure with Unauthorized if the user cannot view it.

## UpdatePostDetailsAsync(long, PostDetailsUpdate, UserContext)

Updates specified details of an existing [Post](#) after validating authorization and business rules. Only non-null values in [newDetails](#) are applied.

```
public Task<Result> UpdatePostDetailsAsync(long postId, PostDetailsUpdate newDetails,  
UserContext userContext)
```

### Parameters

[postId](#) [long](#)

ID of the [Post](#) to update.

[newDetails](#) [PostDetailsUpdate](#)

The details to update.

[userContext](#) [UserContext](#)

Context of the user performing the update.

### Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Class TransactionService

Namespace: [Agora.Core.BusinessServices](#)

Assembly: Agora.Core.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class TransactionService : ITransactionService
```

Inheritance

[object](#) ← TransactionService

Implements

[ITransactionService](#)

Inherited Members

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

## Constructors

TransactionService(IPostRepository, ITransactionRepository,  
ITransactionStatusRepository, IUserRepository,  
IAuthorizationBusinessRules, IBusinessRulesValidator)

```
public TransactionService(IPostRepository postRepo, ITransactionRepository  
transactionRepo, ITransactionStatusRepository transactionStatusRepo,  
IUserRepository userRepo, IAuthorizationBusinessRules authorizationBusinessRules,  
IBusinessRulesValidator businessRulesValidator)
```

Parameters

postRepo [IPostRepository](#)

transactionRepo [ITransactionRepository](#)

transactionStatusRepo [ITransactionStatusRepository](#)

`userRepo` [IUserRepository](#)

`authorizationBusinessRules` [IAuthorizationBusinessRules](#)

`businessRulesValidator` [IBusinessRulesValidator](#)

## Methods

### ChangeTransactionStatusAsync(long, UserContext, TransactionStatusEnum)

Changes the status of an existing `Transaction` after authorization and validation of state transitions.

```
public Task<Result> ChangeTransactionStatusAsync(long transactionId, UserContext  
userContext, TransactionStatusEnum newStatus)
```

#### Parameters

`transactionId` [long](#)

ID of the `Transaction` to update status for.

`userContext` [UserContext](#)

Context of the user performing the status change.

`newStatus` [TransactionStatusEnum](#)

The new `Transaction` status to apply.

#### Returns

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

Success if the status was changed and saved successfully, or failure with relevant error details if invalid or unauthorized.

### CreateTransactionAsync(Transaction, UserContext)

Creates a new `Transaction` after validating authorization and business rules.

```
public Task<Result<Transaction>> CreateTransactionAsync(Transaction transaction,  
UserContext userContext)
```

## Parameters

### `transaction` [Transaction](#)

The [Transaction](#) entity to create.

### `userContext` [UserContext](#)

Context of the user creating the [Transaction](#).

## Returns

### [Task](#) <[Result](#)<[Transaction](#)>>

Success wrapping the created [Transaction](#) if successful, or failure with relevant error details.

## GetAllVisibleTransactionsAsync(ITransactionFilter, UserContext)

Retrieves all [Transaction](#) visible to the user based on authorization rules.

```
public Task<Result<IReadOnlyList<Transaction>>>  
GetAllVisibleTransactionsAsync(ITransactionFilter transactionFilter,  
UserContext userContext)
```

## Parameters

### `transactionFilter` [ITransactionFilter](#)

Filter criteria to apply on transactions.

### `userContext` [UserContext](#)

Context of the current user requesting transactions.

## Returns

### [Task](#) <[Result](#)<[IReadOnlyList](#)<[Transaction](#)>>>

A successful Result wrapping a list of visible [Transaction](#), or failure if an error occurs.

## GetVisibleTransactionByIdAsync(long, UserContext)

Retrieves a single [Transaction](#) by its ID if visible to the user.

```
public Task<Result<Transaction>> GetVisibleTransactionByIdAsync(long transactionId,  
UserContext userContext)
```

### Parameters

[transactionId](#) [long](#)

The ID of the [Transaction](#) to retrieve.

[userContext](#) [UserContext](#)

Context of the current user requesting the [Transaction](#).

### Returns

[Task](#) <[Result](#)<[Transaction](#)>>

Success wrapping the [Transaction](#) if found and authorized, failure with NotFound if missing, or failure with Unauthorized if the user cannot view it.

## UpdateTransactionDetailsAsync(long, TransactionDetailsUpdate, UserContext)

Updates specified details of an existing [Transaction](#) after validating authorization and business rules. Only non-null values in [newDetails](#) are applied.

```
public Task<Result> UpdateTransactionDetailsAsync(long transactionId,  
TransactionDetailsUpdate newDetails, UserContext userContext)
```

### Parameters

[transactionId](#) [long](#)

ID of the **Transaction** to update.

**newDetails** [TransactionDetailsUpdate](#)

The details to update.

**userContext** [UserContext](#)

Context of the user performing the update.

Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Class TransactionStatusService

Namespace: [Agora.Core.BusinessServices](#)

Assembly: Agora.Core.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class TransactionStatusService : ITransactionStatusService
```

Inheritance

[object](#) ← TransactionStatusService

Implements

[ITransactionStatusService](#)

Inherited Members

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

## Constructors

TransactionStatusService(ITransactionStatusRepository,  
IBusinessRulesValidator)

```
public TransactionStatusService(ITransactionStatusRepository transactionStatusRepo,  
IBusinessRulesValidator businessRulesValidator)
```

Parameters

transactionStatusRepo [ITransactionStatusRepository](#)

businessRulesValidator [IBusinessRulesValidator](#)

## Methods

## GetAllTransactionStatusAsync(ITransactionStatusFilter)

Retrieves all `TransactionStatus`, possibly filtered and sorted.

```
public Task<Result<IReadOnlyList<TransactionStatus>>>
GetAllTransactionStatusAsync(ITransactionStatusFilter transactionStatusFilter)
```

Parameters

`transactionStatusFilter` [ITransactionStatusFilter](#)

Filter criteria to apply on transaction status.

Returns

[Task](#) <Result<[IReadOnlyList](#)<[TransactionStatus](#)>>>

A successful Result wrapping a list of `TransactionStatus`, or failure if an error occurs.

## GetTransactionStatusByIdAsync(long)

Retrieves a single `TransactionStatus` by its ID.

```
public Task<Result<TransactionStatus>> GetTransactionStatusByIdAsync(long
transactionStatusId)
```

Parameters

`transactionStatusId` [long](#)

The ID of the `TransactionStatus` to retrieve.

Returns

[Task](#) <Result<[TransactionStatus](#)>>

Success wrapping the `TransactionStatus` if found, failure with `NotFound` if missing.

## UpdateTransactionStatusDetailsAsync(long, TransactionStatusDetailsUpdate)

Updates specified details of an existing [TransactionStatus](#) after validating authorization and business rules. Only non-null values in [newDetails](#) are applied.

```
public Task<Result> UpdateTransactionStatusDetailsAsync(long transactionStatusId,  
TransactionStatusDetailsUpdate newDetails)
```

### Parameters

[transactionStatusId](#) [long](#)

ID of the [TransactionStatus](#) to update.

[newDetails](#) [TransactionStatusDetailsUpdate](#)

The details to update.

### Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Class UserService

Namespace: [Agora.Core.BusinessServices](#)

Assembly: Agora.Core.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class UserService : IUserService
```

Inheritance

[object](#) ← UserService

Implements

[IUserService](#)

Inherited Members

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

## Constructors

### UserService(IUserRepository, IAuthorizationBusinessRules)

```
public UserService(IUserRepository userRepo, IAuthorizationBusinessRules  
authorizationBusinessRules)
```

Parameters

userRepo [IUserRepository](#)

authorizationBusinessRules [IAuthorizationBusinessRules](#)

## Methods

### GetAllUsersAsync(IUserFilter)

Retrieves all [User](#).

```
public Task<Result<IReadOnlyList<User>>> GetAllUsersAsync(IUserFilter userQueryParameters)
```

## Parameters

[userQueryParameters](#) [IUserFilter](#)

Filter criteria to apply when querying users.

## Returns

[Task](#)<[Result](#)<[IReadOnlyList](#)<[User](#)>>>

A successful Result wrapping a list of users, or failure if an error occurs.

## GetUserByIdAsync(string, UserContext)

Retrieves a single [User](#) by its ID.

```
public Task<Result<User>> GetUserByIdAsync(string userId, UserContext userContext)
```

## Parameters

[userId](#) [string](#)

The ID of the user to retrieve.

[userContext](#) [UserContext](#)

Context of the current user requesting user details.

## Returns

[Task](#)<[Result](#)<[User](#)>>

Success wrapping the [User](#) if found and authorized, failure with NotFound if missing.

# Namespace Agora.Core.Common

## Classes

[ErrorDetail](#)

[ErrorMessages](#)

[ErrorMessages.Post](#)

[ErrorMessages.PostCategory](#)

[ErrorMessages.Transaction](#)

[ErrorMessages.User](#)

[Result](#)

[Result<T>](#)

[UserContext](#)

# Class ErrorDetail

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public class ErrorDetail
```

## Inheritance

[object](#) ← ErrorDetail

## Inherited Members

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

# Constructors

## ErrorDetail(ErrorType, string)

```
public ErrorDetail(ErrorType type, string message)
```

## Parameters

type [ErrorType](#)

message [string](#)

# Properties

## Message

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

## Property Value

[string](#)

## Type

```
public ErrorType Type { get; }
```

Property Value

[ErrorType](#)

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class ErrorMessages

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public static class ErrorMessages
```

## Inheritance

[object](#) ← ErrorMessages

## Inherited Members

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

# Methods

## AlreadyDeleted(string)

```
public static string AlreadyDeleted(string objectName)
```

### Parameters

objectName [string](#)

### Returns

[string](#)

## AlreadyExists(string, string)

```
public static string AlreadyExists(string objectName, string objectValue)
```

### Parameters

objectName [string](#)

`objectValue` [string](#)

Returns

[string](#)

## ErrorWhenSavingToDb(string)

```
public static string ErrorWhenSavingToDb(string entityName)
```

Parameters

`entityName` [string](#)

Returns

[string](#)

## IsInvalid(string, string)

```
public static string IsInvalid(string objectName, string objectValue)
```

Parameters

`objectName` [string](#)

`objectValue` [string](#)

Returns

[string](#)

## NewMustBeDifferentFromCurrent(string)

```
public static string NewMustBeDifferentFromCurrent(string objectName)
```

Parameters

objectName [string](#)

Returns

[string](#)

## NotFound(string, string?)

```
public static string NotFound(string objectName, string? objectValue = null)
```

Parameters

objectName [string](#)

objectValue [string](#)

Returns

[string](#)

## RelatedEntityDoesNotExist(string, long?)

```
public static string RelatedEntityDoesNotExist(string entityName, long? entityId)
```

Parameters

entityName [string](#)

entityId [long](#)?

Returns

[string](#)

## SavedButNotRetrieved(string)

```
public static string SavedButNotRetrieved(string entityName)
```

Parameters

entityName [string](#)

Returns

[string](#)

## UnknownErrorDuringAction(string, string)

```
public static string UnknownErrorDuringAction(string entityName, string action)
```

Parameters

entityName [string](#)

action [string](#)

Returns

[string](#)

# Class ErrorMessages.Post

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public static class ErrorMessages.Post
```

## Inheritance

[object](#) ← ErrorMessages.Post

## Inherited Members

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

## Fields

### InvolvedInOngoingTransaction

```
public const string InvolvedInOngoingTransaction = "This post is involved in an ongoing  
transaction. It can't be modified or deleted."
```

#### Field Value

[string](#)

### SameTitle

```
public const string SameTitle = "User has already posted a post with same title."
```

#### Field Value

[string](#)

## Methods

## SameStatus(PostStatus)

```
public static string SameStatus(PostStatus status)
```

Parameters

status [PostStatus](#)

Returns

[string](#)

# Class ErrorMessages.PostCategory

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public static class ErrorMessages.PostCategory
```

## Inheritance

[object](#) ← ErrorMessages.PostCategory

## Inherited Members

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

# Fields

## InUse

```
public const string InUse = "Cannot delete a post category that is used by one or  
more posts."
```

## Field Value

[string](#)

# Class ErrorMessages.Transaction

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public static class ErrorMessages.Transaction
```

## Inheritance

[object](#) ← ErrorMessages.Transaction

## Inherited Members

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

## Fields

### AdminShouldResolveAConflict

```
public const string AdminShouldResolveAConflict = "Only an administrator can resolved a  
transaction in dispute."
```

#### Field Value

[string](#)

### BuyerEqualsSeller

```
public const string BuyerEqualsSeller = "Buyer and seller cannot be the same user."
```

#### Field Value

[string](#)

### CreditInsufficient

```
public const string CreditInsufficient = "Buyer does not have enough credit."
```

Field Value

[string](#)

## InitiatorCanCancelOwnTransaction

```
public const string InitiatorCanCancelOwnTransaction = "A user can cancel a transaction only  
if he/has initiated it."
```

Field Value

[string](#)

## InitiatorCantAcceptOrRefuseOwnTransaction

```
public const string InitiatorCantAcceptOrRefuseOwnTransaction = "A user can't accept or  
refuse the transaction he/she has initiated. This must be done by the other part."
```

Field Value

[string](#)

## NotInvolved

```
public const string NotInvolved = "Current user is not involved in the transaction."
```

Field Value

[string](#)

## NotPostOwner

```
public const string NotPostOwner = "Buyer or seller must be the owner of the post."
```

Field Value

[string](#)

## OtherPartShouldComplete

```
public const string OtherPartShouldComplete = "Only the other part is allowed to complete  
the transaction"
```

Field Value

[string](#)

## Methods

### InvalidTransactionStatusChange(string, string)

```
public static string InvalidTransactionStatusChange(string oldStatus, string newStatus)
```

Parameters

oldStatus [string](#)

newStatus [string](#)

Returns

[string](#)

# Class ErrorMessages.User

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public static class ErrorMessages.User
```

## Inheritance

[object](#) ← ErrorMessages.User

## Inherited Members

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

## Fields

### IdNotFoundInClaims

```
public const string IdNotFoundInClaims = "User ID not found in claims."
```

#### Field Value

[string](#)

### InvalidCredentials

```
public const string InvalidCredentials = "Invalid email or password."
```

#### Field Value

[string](#)

### NotAuthenticated

```
public const string NotAuthenticated = "User should be authenticated to perform  
this action."
```

Field Value

[string](#)

## NotAuthorized

```
public const string NotAuthorized = "User is not allowed to perform this action."
```

Field Value

[string](#)

## PasswordWithoutSpaceBeforeOrAfter

```
public const string PasswordWithoutSpaceBeforeOrAfter = "Password must not start or end with  
a space."
```

Field Value

[string](#)

## Methods

### BuyerOrSellerDoesNotExist(string, string)

```
public static string BuyerOrSellerDoesNotExist(string buyerOrSeller, string id)
```

Parameters

buyerOrSeller [string](#)

id [string](#)

Returns

[string](#)

## EmailAlreadyRegistered(string)

```
public static string EmailAlreadyRegistered(string email)
```

Parameters

[email](#) [string](#)

Returns

[string](#)

## InvalidIdFormat(string)

```
public static string InvalidIdFormat(string id)
```

Parameters

[id](#) [string](#)

Returns

[string](#)

# Class Result

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public class Result
```

## Inheritance

[object](#) ← Result

## Derived

[Result<T>](#)

## Inherited Members

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

# Constructors

## Result(bool, List<ErrorDetail>?)

```
protected Result(bool isSuccess, List<ErrorDetail>? errors)
```

## Parameters

isSuccess [bool](#)

errors [List](#)<[ErrorDetail](#)>

# Properties

## Errors

```
public List<ErrorDetail>? Errors { get; }
```

## Property Value

[List](#) <[ErrorDetail](#)>

## IsFailure

`public bool IsFailure { get; }`

Property Value

[bool](#)

## IsSuccess

`public bool IsSuccess { get; }`

Property Value

[bool](#)

## Methods

### Failure(ErrorType, string)

`public static Result Failure(ErrorType errorType, string errorMsg)`

Parameters

`errorType` [ErrorType](#)

`errorMsg` [string](#)

Returns

[Result](#)

## Failure(List<ErrorDetail>)

```
public static Result Failure(List<ErrorDetail> errors)
```

Parameters

errors [List](#)<[ErrorDetail](#)>

Returns

[Result](#)

## Success()

```
public static Result Success()
```

Returns

[Result](#)

# Class Result<T>

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public class Result<T> : Result where T : class
```

## Type Parameters

T

## Inheritance

[object](#) ← [Result](#) ← Result<T>

## Inherited Members

[Result.IsSuccess](#) , [Result.Errors](#) , [Result.IsFailure](#) , [Result.Success\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Properties

## Value

```
public T? Value { get; }
```

## Property Value

T

# Methods

## Failure(ErrorType, string)

```
public static Result<T> Failure(ErrorType errorType, string errorMsg)
```

Parameters

errorType [ErrorType](#)

errorMsg [string](#)

Returns

[Result](#)<T>

## Failure(ErrorType, string, T?)

```
public static Result<T> Failure(ErrorType errorType, string errorMsg, T? value)
```

Parameters

errorType [ErrorType](#)

errorMsg [string](#)

value T

Returns

[Result](#)<T>

## Failure(List<ErrorDetail>)

```
public static Result<T> Failure(List<ErrorDetail> errors)
```

Parameters

errors [List](#)<[ErrorDetail](#)>

Returns

[Result](#)<T>

## Failure(List<ErrorDetail>, T?)

```
public static Result<T> Failure(List<ErrorDetail> errors, T? value)
```

Parameters

errors [List](#)<[ErrorDetail](#)>

value T

Returns

[Result](#)<T>

## Success(T)

```
public static Result<T> Success(T value)
```

Parameters

value T

Returns

[Result](#)<T>

# Class UserContext

Namespace: [Agora.Core.Common](#)

Assembly: Agora.Core.dll

```
public class UserContext
```

## Inheritance

[object](#) ← UserContext

## Inherited Members

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

# Properties

## IsAdmin

```
public bool IsAdmin { get; init; }
```

### Property Value

[bool](#)

## UserId

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

### Property Value

[string](#)

# Namespace Agora.Core.Constants

## Classes

[Roles](#)

# Class Roles

Namespace: [Agora.Core.Constants](#)

Assembly: Agora.Core.dll

```
public class Roles
```

## Inheritance

[object](#) ← Roles

## Inherited Members

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

# Fields

## Admin

```
public const string Admin = "Admin"
```

## Field Value

[string](#)

# Namespace Agora.Core.Enums

## Enums

### [ErrorType](#)

Represents the category of an error for more precise handling and messaging.

### [PostStatus](#)

### [PostType](#)

### [PostVisibilityMode](#)

Specifies the visibility mode for retrieving posts.

### [TransactionStatusEnum](#)

# Enum ErrorType

Namespace: [Agora.Core.Enums](#)

Assembly: Agora.Core.dll

Represents the category of an error for more precise handling and messaging.

```
public enum ErrorType
```

## Fields

**Forbidden = 3**

The current user is authenticated but does not have permission to perform the requested action.

**Invalid = 4**

The request was invalid due to data validation or business rule violations.

**NotFound = 1**

The requested entity was not found in the database or source.

**Persistence = 5**

A database or persistence-related error occurred.

**Unauthorized = 2**

The current user is not authenticated, but the requested action requires authentication.

**Unknown = 0**

A generic or unknown error occurred.

# Enum PostStatus

Namespace: [Agora.Core.Enums](#)

Assembly: Agora.Core.dll

```
public enum PostStatus
```

## Fields

Active = 0

Deleted = 2

Inactive = 1

# Enum PostType

Namespace: [Agora.Core.Enums](#)

Assembly: Agora.Core.dll

```
public enum PostType
```

## Fields

Offer = 0

Request = 1

# Enum PostVisibilityMode

Namespace: [Agora.Core.Enums](#)

Assembly: Agora.Core.dll

Specifies the visibility mode for retrieving posts.

```
public enum PostVisibilityMode
```

## Fields

**AdminView** = 2

Shows all posts regardless of status, intended for administrators.

**CatalogOnly** = 0

Only active posts are shown, typically used for public catalog views.

**UserOwnPosts** = 1

Shows active and inactive posts belonging to the current user.

# Enum TransactionStatusEnum

Namespace: [Agora.Core.Enums](#)

Assembly: Agora.Core.dll

```
public enum TransactionStatusEnum
```

## Fields

Accepted = 3

Cancelled = 2

Completed = 7

Failed = 5

InDispute = 8

PartiallyValidated = 6

Pending = 1

Refused = 4

ResolvedAccepted = 9

ResolvedCancelled = 10

# Namespace Agora.Core.Extensions

## Classes

[ModelStateExtensions](#)

[StringExtensions](#)

# Class ModelStateExtensions

Namespace: [Agora.Core.Extensions](#)

Assembly: Agora.Core.dll

```
public static class ModelStateExtensions
```

## Inheritance

[object](#) ← ModelStateExtensions

## Inherited Members

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

# Methods

## ExtractErrors(ModelStateDictionary)

```
public static Dictionary<string, List<string>> ExtractErrors(ModelStateDictionary  
modelState)
```

### Parameters

modelState [ModelStateDictionary](#)

### Returns

[Dictionary](#) <[string](#), [List](#) <[string](#)>>

# Class StringExtensions

Namespace: [Agora.Core.Extensions](#)

Assembly: Agora.Core.dll

```
public static class StringExtensions
```

## Inheritance

[object](#) ← StringExtensions

## Inherited Members

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

# Methods

## CapitalizeFirstLetter(string)

```
public static string CapitalizeFirstLetter(this string input)
```

### Parameters

[input](#) [string](#)

### Returns

[string](#)

## IsGuid(string?)

```
public static bool IsGuid(this string? input)
```

### Parameters

[input](#) [string](#)

Returns

bool ↗

# Namespace Agora.Core.Interfaces

## Interfaces

### [IAuthService](#)

Defines authentication-related operations such as user registration, login, and logout.

### [IUserContextService](#)

# Interface IAuthService

Namespace: [Agora.Core.Interfaces](#)

Assembly: Agora.Core.dll

Defines authentication-related operations such as user registration, login, and logout.

```
public interface IAuthService
```

## Methods

### LoginAsync(UserSignInInfo)

Attempts to log in a user with the provided credentials.

```
Task<Result> LoginAsync(UserSignInInfo signInInfo)
```

#### Parameters

**signInInfo** [UserSignInInfo](#)

The user's credentials (email address and password).

#### Returns

[Task](#) <Result>

A [Result](#) indicating whether the login was successful or not.

### LogoutAsync()

Logs out the currently authenticated user.

```
Task LogoutAsync()
```

#### Returns

## RegisterAsync(UserRegistrationInfo)

Registers a new user with the specified registration information.

```
Task<Result<User>> RegisterAsync(UserRegistrationInfo registrationInfo)
```

### Parameters

**registrationInfo** [UserRegistrationInfo](#)

The user's registration data including email, username, and password.

### Returns

[Task](#) <[Result](#)<[User](#)>>

A [Result<T>](#) containing the created [User](#) on success, or an error on failure.

# Interface IUserContextService

Namespace: [Agora.Core.Interfaces](#)

Assembly: Agora.Core.dll

```
public interface IUserContextService
```

## Methods

### GetCurrentUserContext()

```
UserContext GetCurrentUserContext()
```

Returns

[UserContext](#)

### IsAuthenticated()

```
bool IsAuthenticated()
```

Returns

[bool](#)

# Namespace Agora.Core.Interfaces.Business Services

## Interfaces

### [IPostCategoryService](#)

Defines business operations related to post categories, including retrieval and management of post category details.

### [IPostService](#)

Defines business logic for managing user posts, including creation, updates, filtering, and status changes.

### [ITransactionService](#)

Encapsulates business rules for handling transactions between users, including creation, updates, and validation of transaction flows.

### [ITransactionStatusService](#)

Defines operations for managing transaction statuses, including creation, updates, and validation of status rules.

### [IUserService](#)

Provides business logic related to user management, including access control and profile retrieval.

# Interface IPostCategoryService

Namespace: [Agora.Core.Interfaces.BusinessServices](#)

Assembly: Agora.Core.dll

Defines business operations related to post categories, including retrieval and management of post category details.

```
public interface IPostCategoryService
```

## Methods

### CreatePostCategoryAsync(PostCategory)

Creates a new **PostCategory** after validating authorization.

```
Task<Result<PostCategory>> CreatePostCategoryAsync(PostCategory postCategory)
```

#### Parameters

**postCategory** [PostCategory](#)

The **PostCategory** entity to create.

#### Returns

[Task](#) <[Result](#)<[PostCategory](#)>>

Success wrapping the created **PostCategory** if successful, or failure with relevant error details.

### DeletePostCategoryAsync(long)

Delete a **PostCategory** after validating authorization and business rules.

```
Task<Result> DeletePostCategoryAsync(long postCategoryId)
```

## Parameters

`postCategoryId long`

ID of the `PostCategory` to delete.

## Returns

`Task<Result>`

Success if the `PostCategory` was successfully deleted, or failure with relevant error details if invalid or unauthorized.

## GetAllPostCategoriesAsync(IPostCategoryFilter)

Retrieves all `PostCategory`, possibly filtered and sorted.

```
Task<Result<IReadOnlyList<PostCategory>>> GetAllPostCategoriesAsync(IPostCategoryFilter postCategoryFilter)
```

## Parameters

`postCategoryFilter IPostCategoryFilter`

Filter criteria to apply on post categories.

## Returns

`Task<Result<IReadOnlyList<PostCategory>>>`

A successful Result wrapping a list of `PostCategory`, or failure if an error occurs.

## GetPostCategoryByIdAsync(long)

Retrieves a single `PostCategory` by its ID.

```
Task<Result<PostCategory>> GetPostCategoryByIdAsync(long postCategoryId)
```

## Parameters

`postCategoryId` [long](#)

The ID of the [PostCategory](#) to retrieve.

Returns

[Task](#) <[Result](#)<[PostCategory](#)>>

Success wrapping the [PostCategory](#) if found, failure with NotFound if missing.

## UpdatePostCategoryNameAsync(long, string)

Updates name of an existing [PostCategory](#) after validating authorization and business rules.

`Task<Result> UpdatePostCategoryNameAsync(long postCategoryId, string newName)`

Parameters

`postCategoryId` [long](#)

ID of the [PostCategory](#) to update.

`newName` [string](#)

The new name of the [PostCategory](#).

Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Interface IPostService

Namespace: [Agora.Core.Interfaces.BusinessServices](#)

Assembly: Agora.Core.dll

Defines business logic for managing user posts, including creation, updates, filtering, and status changes.

```
public interface IPostService
```

## Methods

### ChangePostStatusAsync(long, UserContext, PostStatus)

Changes the status of an existing [Post](#) after authorization and validation of state transitions.

```
Task<Result> ChangePostStatusAsync(long postId, UserContext userContext,  
PostStatus targetStatus)
```

#### Parameters

[postId](#) [long](#)

ID of the [Post](#) to update status for.

[userContext](#) [UserContext](#)

Context of the user performing the status change.

[targetStatus](#) [PostStatus](#)

The new post status to apply.

#### Returns

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

Success if the status was changed and saved successfully, or failure with relevant error details if invalid or unauthorized.

## CreatePostAsync(Post, UserContext)

Creates a new **Post** after validating authorization and business rules.

```
Task<Result<Post>> CreatePostAsync(Post post, UserContext userContext)
```

Parameters

**post** [Post](#)

The **Post** entity to create.

**userContext** [UserContext](#)

Context of the user creating the post.

Returns

[Task](#) <[Result](#)<[Post](#)>>

Success wrapping the created **Post** if successful, or failure with relevant error details.

## DeletePostAsync(long, UserContext)

Delete a **Post** after validating authorization and business rules.

```
Task<Result> DeletePostAsync(long postId, UserContext userContext)
```

Parameters

**postId** [long](#)

ID of the **Post** to delete.

**userContext** [UserContext](#)

Context of the user requesting the deletion.

Returns

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

Success if the [Post](#) was successfully deleted, or failure with relevant error details if invalid or unauthorized.

## GetAllPostsAsync(PostVisibilityMode, PostFilter, UserContext?)

Retrieves all [Post](#) visible to the user according to authorization rules and visibility mode.

```
Task<Result<IReadOnlyList<Post>>> GetAllPostsAsync(PostVisibilityMode postVisibilityMode,  
PostFilter postFilter, UserContext? userContext)
```

### Parameters

[postVisibilityMode](#) [PostVisibilityMode](#)

Specifies the scope of posts to retrieve (e.g., catalog only, user-owned, admin view).

[postFilter](#) [PostFilter](#)

Filter criteria to apply when querying posts.

[userContext](#) [UserContext](#)

Context of the current user making the request.

### Returns

[Task](#) <[Result](#)<[IReadOnlyList](#)<[Post](#)>>>

A successful Result wrapping a list of posts visible to user, or failure if an error occurs.

## GetPostByIdAsync(long, UserContext?)

Retrieves a single [Post](#) by its ID if visible to the user.

```
Task<Result<Post>> GetPostByIdAsync(long postId, UserContext? userContext)
```

### Parameters

[postId](#) [long](#)

The ID of the post to retrieve.

#### `userContext` [UserContext](#)

Context of the current user requesting the post.

Returns

#### `Task<Result<Post>>`

Success wrapping the `Post` if found and authorized, failure with `NotFound` if missing, or failure with `Unauthorized` if the user cannot view it.

## UpdatePostDetailsAsync(long, PostDetailsUpdate, UserContext)

Updates specified details of an existing `Post` after validating authorization and business rules. Only non-null values in `newDetails` are applied.

```
Task<Result> UpdatePostDetailsAsync(long postId, PostDetailsUpdate newDetails,  
UserContext userContext)
```

Parameters

#### `postId` [long](#)

ID of the `Post` to update.

#### `newDetails` [PostDetailsUpdate](#)

The details to update.

#### `userContext` [UserContext](#)

Context of the user performing the update.

Returns

#### `Task<Result>`

Success if update and save are successful, or failure with appropriate error messages.

# Interface ITransactionService

Namespace: [Agora.Core.Interfaces.BusinessServices](#)

Assembly: Agora.Core.dll

Encapsulates business rules for handling transactions between users, including creation, updates, and validation of transaction flows.

```
public interface ITransactionService
```

## Methods

### ChangeTransactionStatusAsync(long, UserContext, TransactionStatusEnum)

Changes the status of an existing [Transaction](#) after authorization and validation of state transitions.

```
Task<Result> ChangeTransactionStatusAsync(long transactionId, UserContext userContext, TransactionStatusEnum newStatus)
```

#### Parameters

[transactionId](#) [long](#)

ID of the [Transaction](#) to update status for.

[userContext](#) [UserContext](#)

Context of the user performing the status change.

[newStatus](#) [TransactionStatusEnum](#)

The new [Transaction](#) status to apply.

#### Returns

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

Success if the status was changed and saved successfully, or failure with relevant error details if invalid or unauthorized.

## CreateTransactionAsync(Transaction, UserContext)

Creates a new [Transaction](#) after validating authorization and business rules.

```
Task<Result<Transaction>> CreateTransactionAsync(Transaction transaction,  
UserContext userContext)
```

### Parameters

[transaction](#) [Transaction](#)

The [Transaction](#) entity to create.

[userContext](#) [UserContext](#)

Context of the user creating the [Transaction](#).

### Returns

[Task](#) <[Result](#)<[Transaction](#)>>

Success wrapping the created [Transaction](#) if successful, or failure with relevant error details.

## GetAllVisibleTransactionsAsync(ITransactionFilter, UserContext)

Retrieves all [Transaction](#) visible to the user based on authorization rules.

```
Task<Result<IReadOnlyList<Transaction>>> GetAllVisibleTransactionsAsync(ITransactionFilter  
transactionFilter, UserContext userContext)
```

### Parameters

[transactionFilter](#) [ITransactionFilter](#)

Filter criteria to apply on transactions.

[userContext](#) [UserContext](#)

Context of the current user requesting transactions.

Returns

[Task](#) <Result<|ReadOnlyList<Transaction>>>

A successful Result wrapping a list of visible [Transaction](#), or failure if an error occurs.

## GetVisibleTransactionByIdAsync(long, UserContext)

Retrieves a single [Transaction](#) by its ID if visible to the user.

```
Task<Result<Transaction>> GetVisibleTransactionByIdAsync(long transactionId,  
UserContext userContext)
```

Parameters

[transactionId](#) [long](#)

The ID of the [Transaction](#) to retrieve.

[userContext](#) [UserContext](#)

Context of the current user requesting the [Transaction](#).

Returns

[Task](#) <Result<Transaction>>

Success wrapping the [Transaction](#) if found and authorized, failure with NotFound if missing, or failure with Unauthorized if the user cannot view it.

## UpdateTransactionDetailsAsync(long, TransactionDetailsUpdate, UserContext)

Updates specified details of an existing [Transaction](#) after validating authorization and business rules. Only non-null values in [newDetails](#) are applied.

```
Task<Result> UpdateTransactionDetailsAsync(long transactionId, TransactionDetailsUpdate
```

```
newDetails, UserContext userContext)
```

## Parameters

**transactionId** [long](#)

ID of the **Transaction** to update.

**newDetails** [TransactionDetailsUpdate](#)

The details to update.

**userContext** [UserContext](#)

Context of the user performing the update.

## Returns

[Task](#) <Result>

Success if update and save are successful, or failure with appropriate error messages.

# Interface ITransactionStatusService

Namespace: [Agora.Core.Interfaces.BusinessServices](#)

Assembly: Agora.Core.dll

Defines operations for managing transaction statuses, including creation, updates, and validation of status rules.

```
public interface ITransactionStatusService
```

## Methods

### GetAllTransactionStatusAsync(ITransactionStatusFilter)

Retrieves all **TransactionStatus**, possibly filtered and sorted.

```
Task<Result<IReadOnlyList<TransactionStatus>>>
GetAllTransactionStatusAsync(ITransactionStatusFilter transactionStatusFilter)
```

#### Parameters

**transactionStatusFilter** [ITransactionStatusFilter](#)

Filter criteria to apply on transaction status.

#### Returns

[Task](#)<[Result](#)<[IReadOnlyList](#)<[TransactionStatus](#)>>>

A successful Result wrapping a list of **TransactionStatus**, or failure if an error occurs.

### GetTransactionStatusByIdAsync(long)

Retrieves a single **TransactionStatus** by its ID.

```
Task<Result<TransactionStatus>> GetTransactionStatusByIdAsync(long transactionStatusId)
```

## Parameters

`transactionStatusId` [long](#)

The ID of the `TransactionStatus` to retrieve.

## Returns

[Task](#) <[Result](#)<[TransactionStatus](#)>>

Success wrapping the `TransactionStatus` if found, failure with `NotFound` if missing.

## UpdateTransactionStatusDetailsAsync(long, TransactionStatusDetailsUpdate)

Updates specified details of an existing `TransactionStatus` after validating authorization and business rules. Only non-null values in `newDetails` are applied.

```
Task<Result> UpdateTransactionStatusDetailsAsync(long transactionStatusId,  
TransactionStatusDetailsUpdate newDetails)
```

## Parameters

`transactionStatusId` [long](#)

ID of the `TransactionStatus` to update.

`newDetails` [TransactionStatusDetailsUpdate](#)

The details to update.

## Returns

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

Success if update and save are successful, or failure with appropriate error messages.

# Interface IUserService

Namespace: [Agora.Core.Interfaces.BusinessServices](#)

Assembly: Agora.Core.dll

Provides business logic related to user management, including access control and profile retrieval.

```
public interface IUserService
```

## Methods

### GetAllUsersAsync(IUserFilter)

Retrieves all [User](#).

```
Task<Result<IReadOnlyList<User>>> GetAllUsersAsync(IUserFilter userQueryParameters)
```

#### Parameters

[userQueryParameters](#) [IUserFilter](#)

Filter criteria to apply when querying users.

#### Returns

[Task](#) <[Result](#)<[IReadOnlyList](#) <[User](#)>>>

A successful Result wrapping a list of users, or failure if an error occurs.

### GetUserByIdAsync(string, UserContext)

Retrieves a single [User](#) by its ID.

```
Task<Result<User>> GetUserByIdAsync(string userId, UserContext userContext)
```

#### Parameters

`userId` [string](#)

The ID of the user to retrieve.

`userContext` [UserContext](#)

Context of the current user requesting user details.

Returns

[Task](#) <[Result](#)<[User](#)>>

Success wrapping the `User` if found and authorized, failure with `NotFound` if missing.

# Namespace Agora.Core.Interfaces.Filters Interfaces

[IPostCategoryFilter](#)

[IPostFilter](#)

[ITransactionFilter](#)

[ITransactionStatusFilter](#)

[IUserFilter](#)

# Interface IPostCategoryFilter

Namespace: [Agora.Core.Interfaces.Filters](#)

Assembly: Agora.Core.dll

```
public interface IPostCategoryFilter
```

## Properties

### Name

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

### Property Value

[string](#) ↗

### SortBy

```
string? SortBy { get; set; }
```

### Property Value

[string](#) ↗

### SortDesc

```
bool SortDesc { get; set; }
```

### Property Value

[bool](#) ↗

# Interface IPostFilter

Namespace: [Agora.Core.Interfaces.Filters](#)

Assembly: Agora.Core.dll

```
public interface IPostFilter
```

## Properties

### MaxPrice

```
int? MaxPrice { get; set; }
```

#### Property Value

[int](#)?

### MinPrice

```
int? MinPrice { get; set; }
```

#### Property Value

[int](#)?

### PostCategoryName

```
string? PostCategoryName { get; set; }
```

#### Property Value

[string](#)?

## SortBy

```
string? SortBy { get; set; }
```

Property Value

[string](#) ↗

## SortDesc

```
bool SortDesc { get; set; }
```

Property Value

[bool](#) ↗

## TitleOrDescription

```
string? TitleOrDescription { get; set; }
```

Property Value

[string](#) ↗

## TypeName

```
string? TypeName { get; set; }
```

Property Value

[string](#) ↗

## UserName

```
string? UserName { get; set; }
```

Property Value

[string](#) ↗

# Interface ITransactionFilter

Namespace: [Agora.Core.Interfaces.Filters](#)

Assembly: Agora.Core.dll

```
public interface ITransactionFilter
```

## Properties

### MaxPrice

```
int? MaxPrice { get; set; }
```

Property Value

[int](#)?

### MinPrice

```
int? MinPrice { get; set; }
```

Property Value

[int](#)?

### PostTitle

```
string? PostTitle { get; set; }
```

Property Value

[string](#)?

## SortBy

```
string? SortBy { get; set; }
```

Property Value

[string](#) ↗

## SortDesc

```
bool SortDesc { get; set; }
```

Property Value

[bool](#) ↗

## TransactionStatusName

```
string? TransactionStatusName { get; set; }
```

Property Value

[string](#) ↗

## UsersInvolvedUsername

```
string? UsersInvolvedUsername { get; set; }
```

Property Value

[string](#) ↗

# Interface ITransactionStatusFilter

Namespace: [Agora.Core.Interfaces.Filters](#)

Assembly: Agora.Core.dll

```
public interface ITransactionStatusFilter
```

## Properties

### IsFinal

```
bool? IsFinal { get; set; }
```

Property Value

[bool](#)?

### IsSuccess

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

Property Value

[bool](#)?

### NameOrDescription

```
string? NameOrDescription { get; set; }
```

Property Value

[string](#)?

## SortBy

```
string? SortBy { get; set; }
```

Property Value

[string](#) ↗

## SortDesc

```
bool SortDesc { get; set; }
```

Property Value

[bool](#) ↗

# Interface IUserFilter

Namespace: [Agora.Core.Interfaces.Filters](#)

Assembly: Agora.Core.dll

```
public interface IUserFilter
```

## Properties

### Email

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

### Property Value

[string](#)?

### MaxCreatedAt

```
DateTime? MaxCreatedAt { get; set; }
```

### Property Value

[DateTime](#)?

### MaxCredit

```
int? MaxCredit { get; set; }
```

### Property Value

[int](#)?

## MinCreatedAt

`DateTime? MinCreatedAt { get; set; }`

### Property Value

[DateTime](#)?

## MinCredit

`int? MinCredit { get; set; }`

### Property Value

[int](#)?

## SortBy

`string? SortBy { get; set; }`

### Property Value

[string](#)?

## SortDesc

`bool SortDesc { get; set; }`

### Property Value

[bool](#)?

## Username

```
string? Username { get; set; }
```

Property Value

[string](#) ↗

# Namespace Agora.Core.Interfaces.Repositories Interfaces

[IPostCategoryRepository](#)

[IPostRepository](#)

[ITransactionRepository](#)

[ITransactionStatusRepository](#)

[IUserRepository](#)

# Interface IPostCategoryRepository

Namespace: [Agora.Core.Interfaces.Repositories](#)

Assembly: Agora.Core.dll

```
public interface IPostCategoryRepository
```

## Methods

### AddPostCategory(PostCategory)

```
void AddPostCategory(PostCategory postCategory)
```

Parameters

postCategory [PostCategory](#)

### ApplySorting(IQueryable<PostCategory>, IPostCategoryFilter)

```
IQueryable<PostCategory> ApplySorting(IQueryable<PostCategory> query,  
IPostCategoryFilter queryParams)
```

Parameters

query [IQueryable](#)<[PostCategory](#)>

queryParams [IPostCategoryFilter](#)

Returns

[IQueryable](#)<[PostCategory](#)>

### DeletePostCategory(PostCategory)

```
void DeletePostCategory(PostCategory postCategory)
```

Parameters

postCategory [PostCategory](#)

## GetAllPostCategoriesAsync(IPostCategoryFilter)

```
Task<IReadOnlyList<PostCategory>> GetAllPostCategoriesAsync(IPostCategoryFilter filter)
```

Parameters

filter [IPostCategoryFilter](#)

Returns

[Task](#) <[IReadOnlyList](#) <[PostCategory](#)>>

## GetPostCategoryByIdAsync(long)

```
Task<PostCategory?> GetPostCategoryByIdAsync(long id)
```

Parameters

id [long](#)

Returns

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

## NameExistsAsync(string)

```
Task<bool> NameExistsAsync(string name)
```

Parameters

`name` [string](#)

Returns

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

## PostCategoryExistsAsync(long)

`Task<bool> PostCategoryExistsAsync(long id)`

Parameters

`id` [long](#)

Returns

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

## SaveChangesAsync()

`Task<bool> SaveChangesAsync()`

Returns

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

# Interface IPostRepository

Namespace: [Agora.Core.Interfaces.Repositories](#)

Assembly: Agora.Core.dll

```
public interface IPostRepository
```

## Methods

### AddPost(Post)

```
void AddPost(Post post)
```

Parameters

post [Post](#)

### ApplySorting(IQueryable<Post>, IPostFilter)

```
IQueryable<Post> ApplySorting(IQueryable<Post> query, IPostFilter queryParams)
```

Parameters

query [IQueryable](#) <[Post](#)>

queryParams [IPostFilter](#)

Returns

[IQueryable](#) <[Post](#)>

### DeletePost(Post)

```
void DeletePost(Post post)
```

Parameters

post [Post](#)

## GetAllPostsAsync(PostFilter)

```
Task<IReadOnlyList<Post>> GetAllPostsAsync(PostFilter filter)
```

Parameters

filter [PostFilter](#)

Returns

[Task](#) <[IReadOnlyList](#) <[Post](#)>>

## GetPostByIdAsync(long)

```
Task<Post?> GetPostByIdAsync(long id)
```

Parameters

id [long](#)

Returns

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

## IsCategoryInUseAsync(long)

```
Task<bool> IsCategoryInUseAsync(long postCategoryId)
```

Parameters

`postCategoryId long`

Returns

`Task<bool>`

## PostExistsAsync(long)

`Task<bool> PostExistsAsync(long id)`

Parameters

`id long`

Returns

`Task<bool>`

## SaveChangesAsync()

`Task<bool> SaveChangesAsync()`

Returns

`Task<bool>`

# Interface ITransactionRepository

Namespace: [Agora.Core.Interfaces.Repositories](#)

Assembly: Agora.Core.dll

```
public interface ITransactionRepository
```

## Methods

### AddTransaction(Transaction)

```
void AddTransaction(Transaction transaction)
```

Parameters

`transaction` [Transaction](#)

### ApplySorting(IQueryable<Transaction>, ITransactionFilter)

```
IQueryable<Transaction> ApplySorting(IQueryable<Transaction> query,  
ITransactionFilter queryParams)
```

Parameters

`query` [IQueryable](#)<[Transaction](#)>

`queryParams` [ITransactionFilter](#)

Returns

[IQueryable](#)<[Transaction](#)>

### DeleteTransaction(Transaction)

```
void DeleteTransaction(Transaction transaction)
```

Parameters

**transaction** [Transaction](#)

## GetAllTransactionsAsync(ITransactionFilter)

```
Task<IReadOnlyList<Transaction>> GetAllTransactionsAsync(ITransactionFilter filter)
```

Parameters

**filter** [ITransactionFilter](#)

Returns

[Task](#) <[IReadOnlyList](#) <[Transaction](#)>>

## GetTransactionByIdAsync(long)

```
Task<Transaction?> GetTransactionByIdAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

## IsPostInOnGoingTransactionAsync(long)

```
Task<bool> IsPostInOnGoingTransactionAsync(long postId)
```

Parameters

`postId` [long](#)

Returns

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

## IsPostInTransactionAsync(long)

`Task<bool> IsPostInTransactionAsync(long postId)`

Parameters

`postId` [long](#)

Returns

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

## SaveChangesAsync()

`Task<bool> SaveChangesAsync()`

Returns

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

# Interface ITransactionStatusRepository

Namespace: [Agora.Core.Interfaces.Repositories](#)

Assembly: Agora.Core.dll

```
public interface ITransactionStatusRepository
```

## Methods

ApplySorting(IQueryable<TransactionStatus>,  
ITransactionStatusFilter)

```
IQueryable<TransactionStatus> ApplySorting(IQueryable<TransactionStatus> query,  
ITransactionStatusFilter queryParams)
```

Parameters

query [IQueryable](#)<TransactionStatus>

queryParams [ITransactionStatusFilter](#)

Returns

[IQueryable](#)<TransactionStatus>

GetAllTransactionStatusAsync(ITransactionStatusFilter)

```
Task<IReadOnlyList<TransactionStatus>> GetAllTransactionStatusAsync(ITransactionStatusFilter  
filter)
```

Parameters

filter [ITransactionStatusFilter](#)

Returns

[Task](#) <[IReadOnlyList](#)<[TransactionStatus](#)>>

## GetIdByEnumAsync(TransactionStatusEnum)

Task<[long](#)> [GetIdByEnumAsync](#)([TransactionStatusEnum](#) [statusEnum](#))

Parameters

[statusEnum](#) [TransactionStatusEnum](#)

Returns

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

## GetTransactionStatusByIdAsync([long](#))

Task<[TransactionStatus](#)?> [GetTransactionStatusByIdAsync](#)([long](#) [id](#))

Parameters

[id](#) [long](#)

Returns

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

## NameExistsAsync([string](#))

Task<[bool](#)> [NameExistsAsync](#)([string](#) [name](#))

Parameters

[name](#) [string](#)

Returns

[Task](#) <bool>

## SaveChangesAsync()

Task<bool> **SaveChangesAsync()**

Returns

[Task](#) <bool>

## TransactionStatusExistsAsync(long)

Task<bool> **TransactionStatusExistsAsync(long id)**

Parameters

**id** [long](#)

Returns

[Task](#) <bool>

# Interface IUserRepository

Namespace: [Agora.Core.Interfaces.Repositories](#)

Assembly: Agora.Core.dll

```
public interface IUserRepository
```

## Methods

### AddUserAsync(User, string)

```
Task<IdentityResult> AddUserAsync(User user, string password)
```

Parameters

user [User](#)

password [string](#)

Returns

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

### GetAllUsersAsync(IUserFilter)

```
Task<IReadOnlyList<User>> GetAllUsersAsync(IUserFilter filter)
```

Parameters

filter [IUserFilter](#)

Returns

[Task](#) <[IReadOnlyList](#) <[User](#)>>

## GetUserByEmailAsync(string)

Task<User?> GetUserByEmailAsync([string](#) email)

Parameters

[email](#) [string](#)

Returns

[Task](#) <User>

## GetUserByIdAsync(string)

Task<User?> GetUserByIdAsync([string](#) id)

Parameters

[id](#) [string](#)

Returns

[Task](#) <User>

## GetUserByUsernameAsync(string)

Task<User?> GetUserByUsernameAsync([string](#) username)

Parameters

[username](#) [string](#)

Returns

[Task](#) <User>

## UserExistsAsync(string)

Task<bool> UserExistsAsync(string id)

### Parameters

**id** [string](#)

### Returns

[Task](#)<bool>

# Namespace Agora.Core.Models

## Classes

[BaseEntity](#)

[Post](#)

[PostCategory](#)

[Transaction](#)

[TransactionStatus](#)

[User](#)

# Class BaseEntity

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class BaseEntity
```

## Inheritance

[object](#) ← BaseEntity

## Derived

[Post](#), [PostCategory](#), [Transaction](#), [TransactionStatus](#)

## Inherited Members

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

# Properties

## Id

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

## Property Value

[long](#)

# Class Post

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class Post : BaseEntity
```

## Inheritance

[object](#) ← [BaseEntity](#) ← Post

## Inherited Members

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

# Properties

## CreatedAt

```
public required DateTime CreatedAt { get; set; }
```

### Property Value

[DateTime](#)

## Description

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

### Property Value

[string](#)

## Owner

```
public User Owner { get; set; }
```

Property Value

[User](#)

OwnerUserId

```
public required string OwnerUserId { get; set; }
```

Property Value

[string](#)

PostCategory

```
public PostCategory PostCategory { get; set; }
```

Property Value

[PostCategory](#)

PostCategoryId

```
public required long PostCategoryId { get; set; }
```

Property Value

[long](#)

Price

```
public required int Price { get; set; }
```

Property Value

[int](#)

Status

```
public required PostStatus Status { get; set; }
```

Property Value

[PostStatus](#)

Title

```
public required string Title { get; set; }
```

Property Value

[string](#)

Transactions

```
public ICollection<Transaction> Transactions { get; set; }
```

Property Value

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

Type

```
public required PostType Type { get; set; }
```

Property Value

[PostType](#)

UpdatedAt

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

Property Value

[DateTime](#)?

# Class PostCategory

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class PostCategory : BaseEntity
```

## Inheritance

[object](#) ← [BaseEntity](#) ← PostCategory

## Inherited Members

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

## Properties

### Name

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

### Property Value

[string](#)

### Posts

```
public ICollection<Post> Posts { get; set; }
```

### Property Value

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

# Class Transaction

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class Transaction : BaseEntity
```

## Inheritance

[object](#) ← [BaseEntity](#) ← Transaction

## Inherited Members

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

# Properties

## Buyer

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

### Property Value

[User](#)

## BuyerConfirmed

```
public required bool BuyerConfirmed { get; set; }
```

### Property Value

[bool](#)

## BuyerId

```
public required string BuyerId { get; set; }
```

Property Value

[string](#) ↗

CompletedAt

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

Property Value

[DateTime](#) ↗?

CreatedAt

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

Property Value

[DateTime](#) ↗

InitiatorId

```
public required string InitiatorId { get; set; }
```

Property Value

[string](#) ↗

Post

```
public Post? Post { get; set; }
```

Property Value

[Post](#)

PostId

```
public required long? PostId { get; set; }
```

Property Value

[long](#)?

Price

```
public required int Price { get; set; }
```

Property Value

[int](#)

Seller

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

Property Value

[User](#)

SellerConfirmed

```
public required bool SellerConfirmed { get; set; }
```

Property Value

[bool](#)

SellerId

```
public required string SellerId { get; set; }
```

Property Value

[string](#)

Title

```
public required string Title { get; set; }
```

Property Value

[string](#)

TransactionDate

```
public DateOnly? TransactionDate { get; set; }
```

Property Value

[DateOnly](#)?

TransactionStatus

```
public TransactionStatus? TransactionStatus { get; set; }
```

Property Value

[TransactionStatus](#)

## TransactionStatusId

```
public required long TransactionStatusId { get; set; }
```

Property Value

[long](#)

## UpdatedAt

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

Property Value

[DateTime](#)?

## Methods

### Deconstruct(out int, out Post?, out string, out User?, out string)

```
public void Deconstruct(out int price, out Post? post, out string buyerId, out User? buyer,  
out string sellerId)
```

Parameters

price [int](#)

post [Post](#)

`buyerId` [string](#) ↗

`buyer` [User](#)

`sellerId` [string](#) ↗

# Class TransactionStatus

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class TransactionStatus : BaseEntity
```

## Inheritance

[object](#) ← [BaseEntity](#) ← TransactionStatus

## Inherited Members

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

## Properties

### Description

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

#### Property Value

[string](#)

#### EnumValue

```
public required TransactionStatusEnum EnumValue { get; set; }
```

#### Property Value

[TransactionStatusEnum](#)

## IsFinal

```
public required bool IsFinal { get; set; }
```

Property Value

[bool](#)

## IsSuccess

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

Property Value

[bool](#)

## Name

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

Property Value

[string](#)

## Transactions

```
public ICollection<Transaction> Transactions { get; set; }
```

Property Value

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

# Class User

Namespace: [Agora.Core.Models](#)

Assembly: Agora.Core.dll

```
public class User : IdentityUser
```

## Inheritance

[object](#) ← [IdentityUser<string>](#) ← [IdentityUser](#) ← User

## Inherited Members

[IdentityUser<string>.ToString\(\)](#) , [IdentityUser<string>.Id](#) , [IdentityUser<string>.UserName](#) ,  
[IdentityUser<string>.NormalizedUserName](#) , [IdentityUser<string>.Email](#) ,  
[IdentityUser<string>.NormalizedEmail](#) , [IdentityUser<string>.EmailConfirmed](#) ,  
[IdentityUser<string>.PasswordHash](#) , [IdentityUser<string>.SecurityStamp](#) ,  
[IdentityUser<string>.ConcurrencyStamp](#) , [IdentityUser<string>.PhoneNumber](#) ,  
[IdentityUser<string>.PhoneNumberConfirmed](#) , [IdentityUser<string>.TwoFactorEnabled](#) ,  
[IdentityUser<string>.LockoutEnd](#) , [IdentityUser<string>.LockoutEnabled](#) ,  
[IdentityUser<string>.AccessFailedCount](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#)

# Properties

## CreatedAt

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

## Property Value

[DateTime](#)

## Credit

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

Property Value

[int](#)

LastLoginAt

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

Property Value

[DateTime](#)?

Posts

```
public ICollection<Post> Posts { get; set; }
```

Property Value

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

TransactionsAsBuyer

```
public ICollection<Transaction> TransactionsAsBuyer { get; set; }
```

Property Value

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

TransactionsAsSeller

```
public ICollection<Transaction> TransactionsAsSeller { get; set; }
```

Property Value

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

# Namespace Agora.Core.Models.Filters

## Classes

[PostFilter](#)

# Class PostFilter

Namespace: [Agora.Core.Models.Filters](#)

Assembly: Agora.Core.dll

```
public class PostFilter : IPostFilter
```

## Inheritance

[object](#) ← PostFilter

## Implements

[IPostFilter](#)

## Inherited Members

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

# Properties

## MaxPrice

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

## Property Value

[int](#)?

## MinPrice

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

## Property Value

[int](#)?

## PostCategoryName

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

### Property Value

[string](#) ↗

## SortBy

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

### Property Value

[string](#) ↗

## SortDesc

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

### Property Value

[bool](#) ↗

## StatusNames

```
public List<string> StatusNames { get; set; }
```

### Property Value

[List](#) ↗ <[string](#) ↗>

## TitleOrDescription

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

Property Value

[string](#)

TypeName

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

Property Value

[string](#)

UserId

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

Property Value

[string](#)

UserName

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

Property Value

[string](#)

# Namespace Agora.Core.Models.Requests

## Classes

[PostDetailsUpdate](#)

[TransactionDetailsUpdate](#)

[TransactionStatusDetailsUpdate](#)

[UserRegistrationInfo](#)

[UserSignInInfo](#)

# Class PostDetailsUpdate

Namespace: [Agora.Core.Models.Requests](#)

Assembly: Agora.Core.dll

```
public class PostDetailsUpdate
```

## Inheritance

[object](#) ← PostDetailsUpdate

## Inherited Members

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

# Properties

## Description

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

### Property Value

[string](#)?

### PostCategoryId

```
public long? PostCategoryId { get; set; }
```

### Property Value

[long](#)?

### Price

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

Property Value

int?

Title

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

Property Value

string?

Type

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

Property Value

string?

# Class TransactionDetailsUpdate

Namespace: [Agora.Core.Models.Requests](#)

Assembly: Agora.Core.dll

```
public class TransactionDetailsUpdate
```

## Inheritance

[object](#) ← TransactionDetailsUpdate

## Inherited Members

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

# Properties

## PostId

```
public long? PostId { get; set; }
```

### Property Value

[long](#)?

## Price

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

### Property Value

[int](#)?

## Title

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

Property Value

[string](#) ↗

TransactionDate

```
public DateOnly? TransactionDate { get; set; }
```

Property Value

[DateOnly](#) ↗?

# Class TransactionStatusDetailsUpdate

Namespace: [Agora.Core.Models.Requests](#)

Assembly: Agora.Core.dll

```
public class TransactionStatusDetailsUpdate
```

## Inheritance

[object](#) ← TransactionStatusDetailsUpdate

## Inherited Members

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

## Properties

### Description

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

#### Property Value

[string](#)

### Name

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

#### Property Value

[string](#)

# Class UserRegistrationInfo

Namespace: [Agora.Core.Models.Requests](#)

Assembly: Agora.Core.dll

```
public class UserRegistrationInfo
```

## Inheritance

[object](#) ← UserRegistrationInfo

## Inherited Members

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

# Properties

## Email

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

## Property Value

[string](#)

## Password

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

## Property Value

[string](#)

## UserName

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

Property Value

[string](#) ↗

# Class UserSignInInfo

Namespace: [Agora.Core.Models.Requests](#)

Assembly: Agora.Core.dll

```
public class UserSignInInfo
```

## Inheritance

[object](#) ← UserSignInInfo

## Inherited Members

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

# Properties

## Email

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

## Property Value

[string](#)

## Password

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

## Property Value

[string](#)

# Namespace Agora.Core.Validation

## Classes

[NotEmptyOrWhitespaceAttribute](#)

[ValidationRules](#)

[ValidationRules.AppUser](#)

[ValidationRules.Post](#)

[ValidationRules.PostCategory](#)

[ValidationRules.Transaction](#)

[ValidationRules.TransactionStatus](#)

# Class NotEmptyOrWhitespaceAttribute

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
[AttributeUsage(AttributeTargets.Property | AttributeTargets.Field, AllowMultiple = false)]
public class NotEmptyOrWhitespaceAttribute : ValidationAttribute
```

## Inheritance

[object](#) ← [Attribute](#) ← [ValidationAttribute](#) ← NotEmptyOrWhitespaceAttribute

## Inherited Members

[ValidationAttribute.FormatErrorMessage\(string\)](#) ,  
[ValidationAttribute.GetValidationResult\(object, ValidationContext\)](#) ,  
[ValidationAttribute.IsValid\(object, ValidationContext\)](#) ,  
[ValidationAttribute.Validate\(object, ValidationContext\)](#) , [ValidationAttribute.Validate\(object, string\)](#) ,  
[ValidationAttribute.ErrorMessage](#) , [ValidationAttribute\(ErrorMessageResourceName\)](#) ,  
[ValidationAttribute\(ErrorMessageResourceType\)](#) , [ValidationAttribute.ErrorMessageString](#) ,  
[ValidationAttribute.RequiresValidationContext](#) , [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\(\)](#)

## Constructors

### NotEmptyOrWhitespaceAttribute()

```
public NotEmptyOrWhitespaceAttribute()
```

## Methods

### IsValid(object?)

Determines whether the specified value of the object is valid.

```
public override bool IsValid(object? value)
```

#### Parameters

**value** [object](#)

The value of the object to validate.

#### Returns

[bool](#)

[true](#) if the specified value is valid; otherwise, [false](#).

#### Exceptions

[InvalidOperationException](#)

The current attribute is malformed.

[NotImplementedException](#)

Neither overload of [IsValid](#) has been implemented by a derived class.

# Class ValidationRules

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public class ValidationRules
```

## Inheritance

[object](#) ← ValidationRules

## Inherited Members

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

# Class ValidationRules.AppUser

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public static class ValidationRules.AppUser
```

## Inheritance

[object](#) ← ValidationRules.AppUser

## Inherited Members

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

## Fields

### EmailMaxLength

```
public const int EmailMaxLength = 255
```

#### Field Value

[int](#)

### EmailMinLength

```
public const int EmailMinLength = 5
```

#### Field Value

[int](#)

### PasswordMaxLength

```
public const int PasswordMaxLength = 255
```

Field Value

[int ↗](#)

## PasswordMinLength

```
public const int PasswordMinLength = 8
```

Field Value

[int ↗](#)

## UsernameMaxLength

```
public const int UsernameMaxLength = 50
```

Field Value

[int ↗](#)

## UsernameMinLength

```
public const int UsernameMinLength = 5
```

Field Value

[int ↗](#)

# Class ValidationRules.Post

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public static class ValidationRules.Post
```

## Inheritance

[object](#) ← ValidationRules.Post

## Inherited Members

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

## Fields

### DescriptionMaxLength

```
public const int DescriptionMaxLength = 2000
```

Field Value

[int](#)

### DescriptionMinLength

```
public const int DescriptionMinLength = 5
```

Field Value

[int](#)

### PriceMax

```
public const int PriceMax = 100000
```

Field Value

[int ↗](#)

## PriceMin

```
public const int PriceMin = 0
```

Field Value

[int ↗](#)

## TitleMaxLength

```
public const int TitleMaxLength = 100
```

Field Value

[int ↗](#)

## TitleMinLength

```
public const int TitleMinLength = 3
```

Field Value

[int ↗](#)

# Class ValidationRules.PostCategory

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public static class ValidationRules.PostCategory
```

## Inheritance

[object](#) ← ValidationRules.PostCategory

## Inherited Members

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

## Fields

### NameMaxLength

```
public const int NameMaxLength = 100
```

Field Value

[int](#)

### NameMinLength

```
public const int NameMinLength = 3
```

Field Value

[int](#)

# Class ValidationRules.Transaction

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public static class ValidationRules.Transaction
```

## Inheritance

[object](#) ← ValidationRules.Transaction

## Inherited Members

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

## Fields

### PriceMax

```
public const int PriceMax = 100000
```

#### Field Value

[int](#)

### PriceMin

```
public const int PriceMin = 0
```

#### Field Value

[int](#)

### TitleMaxLength

```
public const int TitleMaxLength = 200
```

Field Value

[int ↗](#)

## TitleMinLength

```
public const int TitleMinLength = 10
```

Field Value

[int ↗](#)

# Class ValidationRules.TransactionStatus

Namespace: [Agora.Core.Validation](#)

Assembly: Agora.Core.dll

```
public static class ValidationRules.TransactionStatus
```

## Inheritance

[object](#) ← ValidationRules.TransactionStatus

## Inherited Members

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

## Fields

### DescriptionMaxLength

```
public const int DescriptionMaxLength = 500
```

Field Value

[int](#)

### DescriptionMinLength

```
public const int DescriptionMinLength = 3
```

Field Value

[int](#)

### NameMaxLength

```
public const int NameMaxLength = 50
```

Field Value

[int ↗](#)

## NameMinLength

```
public const int NameMinLength = 3
```

Field Value

[int ↗](#)

# Namespace Agora.Infrastructure.Configuration

## Classes

[PostCategoryConfiguration](#)

[PostConfiguration](#)

[TransactionConfiguration](#)

[TransactionStatusConfiguration](#)

[UserConfiguration](#)

# Class PostCategoryConfiguration

Namespace: [Agora.Infrastructure.Configuration](#)

Assembly: Agora.Infrastructure.dll

```
public class PostCategoryConfiguration : IEntityConfiguration<PostCategory>
```

## Inheritance

[object](#) ← PostCategoryConfiguration

## Implements

[IEntityTypeConfiguration](#) <[PostCategory](#)>

## Inherited Members

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

## Methods

### Configure(EntityTypeBuilder<PostCategory>)

Configures the entity of type [PostCategory](#).

```
public void Configure(EntityTypeBuilder<PostCategory> builder)
```

## Parameters

builder [EntityTypeBuilder](#) <[PostCategory](#)>

The builder to be used to configure the entity type.

# Class PostConfiguration

Namespace: [Agora.Infrastructure.Configuration](#)

Assembly: Agora.Infrastructure.dll

```
public class PostConfiguration : IEntityTypeConfiguration<Post>
```

## Inheritance

[object](#) ← PostConfiguration

## Implements

[IEntityTypeConfiguration](#) <[Post](#)>

## Inherited Members

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

## Methods

### Configure(EntityTypeBuilder<Post>)

Configures the entity of type [Post](#).

```
public void Configure(EntityTypeBuilder<Post> builder)
```

## Parameters

**builder** [EntityTypeBuilder](#) <[Post](#)>

The builder to be used to configure the entity type.

# Class TransactionConfiguration

Namespace: [Agora.Infrastructure.Configuration](#)

Assembly: Agora.Infrastructure.dll

```
public class TransactionConfiguration : IEntityConfiguration<Transaction>
```

## Inheritance

[object](#) ← TransactionConfiguration

## Implements

[IEntityTypeConfiguration](#)<[Transaction](#)>

## Inherited Members

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

## Methods

### Configure(EntityTypeBuilder<Transaction>)

Configures the entity of type [Transaction](#).

```
public void Configure(EntityTypeBuilder<Transaction> builder)
```

## Parameters

builder [EntityTypeBuilder](#)<[Transaction](#)>

The builder to be used to configure the entity type.

# Class TransactionStatusConfiguration

Namespace: [Agora.Infrastructure.Configuration](#)

Assembly: Agora.Infrastructure.dll

```
public class TransactionStatusConfiguration : IEntityConfiguration<TransactionStatus>
```

## Inheritance

[object](#) ← TransactionStatusConfiguration

## Implements

[IEntityTypeConfiguration](#)<[TransactionStatus](#)>

## Inherited Members

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

## Methods

### Configure(EntityTypeBuilder<TransactionStatus>)

Configures the entity of type [TransactionStatus](#).

```
public void Configure(EntityTypeBuilder<TransactionStatus> builder)
```

## Parameters

builder [EntityTypeBuilder](#)<[TransactionStatus](#)>

The builder to be used to configure the entity type.

# Class UserConfiguration

Namespace: [Agora.Infrastructure.Configuration](#)

Assembly: Agora.Infrastructure.dll

```
public class UserConfiguration : IEntityConfiguration<User>
```

## Inheritance

[object](#) ← UserConfiguration

## Implements

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

## Inherited Members

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

## Methods

### Configure(EntityTypeBuilder<User>)

Configures the entity of type [User](#).

```
public void Configure(EntityTypeBuilder<User> builder)
```

## Parameters

**builder** [EntityTypeBuilder](#)<[User](#)>

The builder to be used to configure the entity type.

# Namespace Agora.Infrastructure.Data

## Classes

[AgoraDbContext](#)

[AgoraDbContextSeed](#)

# Class AgoraDbContext

Namespace: [Agora.Infrastructure.Data](#)

Assembly: Agora.Infrastructure.dll

```
public class AgoraDbContext : IdentityDbContext<User>, IInfrastructure<IServiceProvider>,
IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,
IDisposable, IAsyncDisposable
```

## Inheritance

```
object ↵ ← DbContext ↵ ←
IdentityUserContext<User, string>, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>>> ←
IdentityDbContext<User, IdentityRole, string>, IdentityUserClaim<string>, IdentityUserRole<
<string>, IdentityUserLogin<string>, IdentityRoleClaim<string>, IdentityUserToken<string>
>> ←
IdentityDbContext<User, IdentityRole, string> ← IdentityDbContext<User> ← AgoraDbContext
```

## Implements

```
IInfrastructure<IServiceProvider>, IDbContextDependencies, IDbSetCache, IDbContextPoolable,
IResettableService, IDisposable, IAsyncDisposable
```

## Inherited Members

```
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>,
IdentityUserLogin<string>, IdentityRoleClaim<string>,
IdentityUserToken<string>.OnModelCreating(ModelBuilder) ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>,
IdentityUserLogin<string>, IdentityRoleClaim<string>, IdentityUserToken<string>.UserRoles ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>,
IdentityUserLogin<string>, IdentityRoleClaim<string>, IdentityUserToken<string>.Roles ,
IdentityDbContext<User, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>,
IdentityUserLogin<string>, IdentityRoleClaim<string>, IdentityUserToken<string>.RoleClaims ,
IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>.OnModelCreating(ModelBuilder) ,
IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>.Users ,
IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>,
IdentityUserToken<string>.UserClaims ,
```

[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.UserLogins](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.UserTokens](#) ,  
[IdentityUserContext<User, string, IdentityUserClaim<string>, IdentityUserLogin<string>, IdentityUserToken<string>>.SchemaVersion](#) ,  
[DbContext.Set< TEntity >\(\)](#) , [DbContext.Set< TEntity >\(string\)](#) ,  
[DbContext.OnConfiguring\(DbContextOptionsBuilder\)](#) ,  
[DbContext.ConfigureConventions\(ModelConfigurationBuilder\)](#) , [DbContext.SaveChanges\(\)](#) ,  
[DbContext.SaveChanges\(bool\)](#) , [DbContext.SaveChangesAsync\(CancellationToken\)](#) ,  
[DbContext.SaveChangesAsync\(bool, CancellationToken\)](#) , [DbContext.Dispose\(\)](#) ,  
[DbContext.DisposeAsync\(\)](#) , [DbContext.Entry< TEntity >\(TEntity\)](#) , [DbContext.Entry\(object\)](#) ,  
[DbContext.Add< TEntity >\(TEntity\)](#) , [DbContext.AddAsync< TEntity >\(TEntity, CancellationToken\)](#) ,  
[DbContext.Attach< TEntity >\(TEntity\)](#) , [DbContext.Update< TEntity >\(TEntity\)](#) ,  
[DbContext.Remove< TEntity >\(TEntity\)](#) , [DbContext.Add\(object\)](#) ,  
[DbContext.AddAsync\(object, CancellationToken\)](#) , [DbContext.Attach\(object\)](#) ,  
[DbContext.Update\(object\)](#) , [DbContext.Remove\(object\)](#) , [DbContext.AddRange\(params object\[\]\)](#) ,  
[DbContext.AddRangeAsync\(params object\[\]\)](#) , [DbContext.AttachRange\(params object\[\]\)](#) ,  
[DbContext.UpdateRange\(params object\[\]\)](#) , [DbContext.RemoveRange\(params object\[\]\)](#) ,  
[DbContext.AddRange\(IEnumerable< object >\)](#) ,  
[DbContext.AddRangeAsync\(IEnumerable< object >, CancellationToken\)](#) ,  
[DbContext.AttachRange\(IEnumerable< object >\)](#) , [DbContext.UpdateRange\(IEnumerable< object >\)](#) ,  
[DbContext.RemoveRange\(IEnumerable< object >\)](#) , [DbContext.Find\(Type, params object\[\]\)](#) ,  
[DbContext.FindAsync\(Type, params object\[\]\)](#) ,  
[DbContext.FindAsync\(Type, object\[\], CancellationToken\)](#) , [DbContext.Find< TEntity >\(params object\[\]\)](#) ,  
[DbContext.FindAsync< TEntity >\(params object\[\]\)](#) ,  
[DbContext.FindAsync< TEntity >\(object\[\], CancellationToken\)](#) ,  
[DbContext.FromExpression< TResult >\(Expression< Func< IQueryable< TResult > > \)](#) ,  
[DbContext.Database](#) , [DbContext.ChangeTracker](#) , [DbContext.Model](#) , [DbContext.ContextId](#) ,  
[DbContext.SavingChanges](#) , [DbContext.SavedChanges](#) , [DbContext.SaveChangesFailed](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### AgoraDbContext(DbContextOptions)

```
public AgoraDbContext(DbContextOptions options)
```

## Parameters

options [DbContextOptions](#)

## Properties

### PostCategories

```
public DbSet<PostCategory> PostCategories { get; set; }
```

### Property Value

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

### Posts

```
public DbSet<Post> Posts { get; set; }
```

### Property Value

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

### TransactionStatus

```
public DbSet<TransactionStatus> TransactionStatus { get; set; }
```

### Property Value

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

### Transactions

```
public DbSet<Transaction> Transactions { get; set; }
```

Property Value

[DbSet](#) <Transaction>

## Methods

### OnModelCreating(ModelBuilder)

Configures the schema needed for the identity framework.

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

Parameters

modelBuilder [ModelBuilder](#)

# Class AgoraDbContextSeed

Namespace: [Agora.Infrastructure.Data](#)

Assembly: Agora.Infrastructure.dll

```
public static class AgoraDbContextSeed
```

## Inheritance

[object](#) ← AgoraDbContextSeed

## Inherited Members

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

## Methods

SeedDevelopmentDataAsync(AgoraDbContext,  
UserManager<User>)

```
public static Task SeedDevelopmentDataAsync(AgoraDbContext context,  
UserManager<User> userManager)
```

### Parameters

context [AgoraDbContext](#)

userManager [UserManager](#)<[User](#)>

### Returns

[Task](#)

SeedRolesAsync(RoleManager<IdentityRole>)

```
public static Task SeedRolesAsync(RoleManager<IdentityRole> roleManager)
```

Parameters

roleManager [RoleManager](#) <[IdentityRole](#)>

Returns

[Task](#)

# Namespace Agora.Infrastructure.Migrations

## Classes

### [AddEnumValueToTransactionStatus](#)

A base class inherited by each EF Core migration.

### [AddEnumValueToTransactionStatusStringValue](#)

A base class inherited by each EF Core migration.

### [AddInfoForTransactionStatusChangesToTransactionEntity](#)

A base class inherited by each EF Core migration.

### [AddTitleToTransaction](#)

A base class inherited by each EF Core migration.

### [AddTransactionDate](#)

A base class inherited by each EF Core migration.

### [AddUpdatedAtToTransactionEntityAndRenameAppUserToUser](#)

A base class inherited by each EF Core migration.

### [InitialCreate](#)

A base class inherited by each EF Core migration.

### [RenameUserToOwnerInPost](#)

A base class inherited by each EF Core migration.

### [RestrictPostCategoryDeletion](#)

A base class inherited by each EF Core migration.

# Class AddEnumValueToTransactionStatus

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250613165206_AddEnumValueToTransactionStatus")]
public class AddEnumValueToTransactionStatus : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddEnumValueToTransactionStatus

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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.

# Class

# AddEnumValueToTransactionStatusStringValue

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250613165420_AddEnumValueToTransactionStatusStringValue")]
public class AddEnumValueToTransactionStatusStringValue : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddEnumValueToTransactionStatusStringValue

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

## Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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'.

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

## Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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.

# Class

# AddInfoForTransactionStatusChangesToTransactionEntity

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250613172104_AddInfoForTransactionStatusChangesToTransactionEntity")]
public class AddInfoForTransactionStatusChangesToTransactionEntity : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddInfoForTransactionStatusChangesToTransactionEntity

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

## Parameters

**migrationBuilder** [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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'.

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

## Parameters

**migrationBuilder** [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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.

# Class AddTitleToTransaction

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250610133742_AddTitleToTransaction")]
public class AddTitleToTransaction : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddTitleToTransaction

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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.

# Class AddTransactionDate

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250606132933_AddTransactionDate")]
public class AddTransactionDate : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← AddTransactionDate

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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.

# Class

## AddUpdatedAtToTransactionEntityAndRenameAppUserToUser

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250616193033_AddUpdatedAtToTransactionEntityAndRenameAppUserToUser")]
public class AddUpdatedAtToTransactionEntityAndRenameAppUserToUser : Migration
```

### Inheritance

[object](#) ← [Migration](#) ← AddUpdatedAtToTransactionEntityAndRenameAppUserToUser

### 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

### Parameters

[modelBuilder](#) [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

## Parameters

**migrationBuilder** [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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'.

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

## Parameters

**migrationBuilder** [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

## 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.

# Class InitialCreate

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250606080627_InitialCreate")]
public class InitialCreate : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← InitialCreate

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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.

# Class RenameUserToOwnerInPost

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250606142239_RenameUserToOwnerInPost")]
public class RenameUserToOwnerInPost : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← RenameUserToOwnerInPost

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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.

# Class RestrictPostCategoryDeletion

Namespace: [Agora.Infrastructure.Migrations](#)

Assembly: Agora.Infrastructure.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(AgoraDbContext))]
[Migration("20250613111631_RestrictPostCategoryDeletion")]
public class RestrictPostCategoryDeletion : Migration
```

## Inheritance

[object](#) ← [Migration](#) ← RestrictPostCategoryDeletion

## 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\(\)](#)

## Remarks

See [Database migrations](#) for more information and examples.

## Methods

### BuildTargetModel(ModelBuilder)

Implemented to build the [TargetModel](#).

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

## Parameters

`modelBuilder` [ModelBuilder](#)

The [ModelBuilder](#) to use to build the model.

## Remarks

See [Database migrations](#) for more information and examples.

## Down(MigrationBuilder)

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

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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'.

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

### Parameters

`migrationBuilder` [MigrationBuilder](#)

The [MigrationBuilder](#) that will build the operations.

### 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 Agora.Infrastructure.Repositories

## Classes

[PostCategoryRepository](#)

[PostRepository](#)

[TransactionRepository](#)

[TransactionStatusRepository](#)

[UserRepository](#)

# Class PostCategoryRepository

Namespace: [Agora.Infrastructure.Repositories](#)

Assembly: Agora.Infrastructure.dll

```
public class PostCategoryRepository : IPostCategoryRepository
```

## Inheritance

[object](#) ← PostCategoryRepository

## Implements

[IPostCategoryRepository](#)

## Inherited Members

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

## Constructors

### PostCategoryRepository(AgoraDbContext)

```
public PostCategoryRepository(AgoraDbContext context)
```

## Parameters

context [AgoraDbContext](#)

## Methods

### AddPostCategory(PostCategory)

```
public void AddPostCategory(PostCategory postCategory)
```

## Parameters

postCategory [PostCategory](#)

## ApplySorting(IQueryable<PostCategory>, IPostCategoryFilter)

```
public IQueryable<PostCategory> ApplySorting(IQueryable<PostCategory> query,  
IPostCategoryFilter queryParams)
```

Parameters

query [IQueryable](#)<[PostCategory](#)>

queryParams [IPostCategoryFilter](#)

Returns

[IQueryable](#)<[PostCategory](#)>

## DeletePostCategory(PostCategory)

```
public void DeletePostCategory(PostCategory postCategory)
```

Parameters

postCategory [PostCategory](#)

## GetAllPostCategoriesAsync(IPostCategoryFilter)

```
public Task<IReadOnlyList<PostCategory>> GetAllPostCategoriesAsync(IPostCategoryFilter  
filter)
```

Parameters

filter [IPostCategoryFilter](#)

Returns

[Task](#)<[IReadOnlyList](#)<[PostCategory](#)>>

## GetPostCategoryByIdAsync(long)

```
public Task<PostCategory?> GetPostCategoryByIdAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

## NameExistsAsync(string)

```
public Task<bool> NameExistsAsync(string name)
```

Parameters

**name** [string](#)

Returns

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

## PostCategoryExistsAsync(long)

```
public Task<bool> PostCategoryExistsAsync(long id)
```

Parameters

**id** [long](#)

Returns

[Task](#) <bool>

## SaveChangesAsync()

```
public Task<bool> SaveChangesAsync()
```

Returns

[Task](#) <bool>

# Class PostRepository

Namespace: [Agora.Infrastructure.Repositories](#)

Assembly: Agora.Infrastructure.dll

```
public class PostRepository : IPostRepository
```

## Inheritance

[object](#) ← PostRepository

## Implements

[IPostRepository](#)

## Inherited Members

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

## Constructors

### PostRepository(AgoraDbContext)

```
public PostRepository(AgoraDbContext context)
```

## Parameters

context [AgoraDbContext](#)

## Methods

### AddPost(Post)

```
public void AddPost(Post post)
```

## Parameters

post [Post](#)

## ApplySorting(IQueryable<Post>, IPostFilter)

```
public IQueryable<Post> ApplySorting(IQueryable<Post> query, IPostFilter queryParams)
```

### Parameters

query [IQueryable](#)<Post>

queryParams [IPostFilter](#)

### Returns

[IQueryable](#)<Post>

## DeletePost(Post)

```
public void DeletePost(Post post)
```

### Parameters

post [Post](#)

## GetAllPostsAsync(PostFilter)

```
public Task<IReadOnlyList<Post>> GetAllPostsAsync(PostFilter filter)
```

### Parameters

filter [PostFilter](#)

### Returns

[Task](#)<[IReadOnlyList](#)<Post>>

## GetPostByIdAsync(long)

```
public Task<Post?> GetPostByIdAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

## IsCategoryInUseAsync(long)

```
public Task<bool> IsCategoryInUseAsync(long postCategoryId)
```

Parameters

**postCategoryId** [long](#)

Returns

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

## PostExistsAsync(long)

```
public Task<bool> PostExistsAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

## SaveChangesAsync()

```
public Task<bool> SaveChangesAsync()
```

Returns

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

# Class TransactionRepository

Namespace: [Agora.Infrastructure.Repositories](#)

Assembly: Agora.Infrastructure.dll

```
public class TransactionRepository : ITransactionRepository
```

## Inheritance

[object](#) ← TransactionRepository

## Implements

[ITransactionRepository](#)

## Inherited Members

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

## Constructors

### TransactionRepository(AgoraDbContext)

```
public TransactionRepository(AgoraDbContext context)
```

## Parameters

context [AgoraDbContext](#)

## Methods

### AddTransaction(Transaction)

```
public void AddTransaction(Transaction transaction)
```

## Parameters

transaction [Transaction](#)

## ApplySorting(IQueryable<Transaction>, ITransactionFilter)

```
public IQueryable<Transaction> ApplySorting(IQueryable<Transaction> query,  
ITransactionFilter queryParams)
```

### Parameters

query [IQueryable](#)<[Transaction](#)>

queryParams [ITransactionFilter](#)

### Returns

[IQueryable](#)<[Transaction](#)>

## DeleteTransaction(Transaction)

```
public void DeleteTransaction(Transaction transaction)
```

### Parameters

transaction [Transaction](#)

## GetAllTransactionsAsync(ITransactionFilter)

```
public Task<IReadOnlyList<Transaction>> GetAllTransactionsAsync(ITransactionFilter filter)
```

### Parameters

filter [ITransactionFilter](#)

### Returns

[Task](#)<[IReadOnlyList](#)<[Transaction](#)>>

## GetTransactionByIdAsync(long)

```
public Task<Transaction?> GetTransactionByIdAsync(long id)
```

Parameters

[id](#) [long](#)

Returns

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

## IsPostInOnGoingTransactionAsync(long)

```
public Task<bool> IsPostInOnGoingTransactionAsync(long postId)
```

Parameters

[postId](#) [long](#)

Returns

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

## IsPostInTransactionAsync(long)

```
public Task<bool> IsPostInTransactionAsync(long postId)
```

Parameters

[postId](#) [long](#)

Returns

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

## SaveChangesAsync()

```
public Task<bool> SaveChangesAsync()
```

Returns

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

# Class TransactionStatusRepository

Namespace: [Agora.Infrastructure.Repositories](#)

Assembly: Agora.Infrastructure.dll

```
public class TransactionStatusRepository : ITransactionStatusRepository
```

## Inheritance

[object](#) ← TransactionStatusRepository

## Implements

[ITransactionStatusRepository](#)

## Inherited Members

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

## Constructors

### TransactionStatusRepository(AgoraDbContext)

```
public TransactionStatusRepository(AgoraDbContext context)
```

## Parameters

context [AgoraDbContext](#)

## Methods

### ApplySorting(IQueryable<TransactionStatus>, ITransactionStatusFilter)

```
public IQueryable<TransactionStatus> ApplySorting(IQueryable<TransactionStatus> query,  
ITransactionStatusFilter queryParams)
```

Parameters

query [IQueryable](#)<[TransactionStatus](#)>

queryParams [ITransactionStatusFilter](#)

Returns

[IQueryable](#)<[TransactionStatus](#)>

## GetAllTransactionStatusAsync(ITransactionStatusFilter)

```
public Task<IReadOnlyList<TransactionStatus>>
GetAllTransactionStatusAsync(ITransactionStatusFilter filter)
```

Parameters

filter [ITransactionStatusFilter](#)

Returns

[Task](#)<[IReadOnlyList](#)<[TransactionStatus](#)>>

## GetIdByEnumAsync(TransactionStatusEnum)

```
public Task<long> GetIdByEnumAsync(TransactionStatusEnum statusEnum)
```

Parameters

statusEnum [TransactionStatusEnum](#)

Returns

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

## GetTransactionStatusByIdAsync(long)

```
public Task<TransactionStatus?> GetTransactionStatusByIdAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

## NameExistsAsync(string)

```
public Task<bool> NameExistsAsync(string name)
```

Parameters

**name** [string](#)

Returns

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

## SaveChangesAsync()

```
public Task<bool> SaveChangesAsync()
```

Returns

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

## TransactionStatusExistsAsync(long)

```
public Task<bool> TransactionStatusExistsAsync(long id)
```

Parameters

**id** [long](#)

Returns

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

# Class UserRepository

Namespace: [Agora.Infrastructure.Repositories](#)

Assembly: Agora.Infrastructure.dll

```
public class UserRepository : IUserRepository
```

## Inheritance

[object](#) ← UserRepository

## Implements

[IUserRepository](#)

## Inherited Members

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

## Constructors

UserRepository(UserManager<User>, SignInManager<User>)

```
public UserRepository(UserManager<User> userManager, SignInManager<User> signInManager)
```

## Parameters

userManager [UserManager](#)<User>

signInManager [SignInManager](#)<User>

## Methods

AddUserAsync(User, string)

```
public Task<IdentityResult> AddUserAsync(User user, string password)
```

## Parameters

user [User](#)

password [string](#)

Returns

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

## GetAllUsersAsync(IUserFilter)

```
public Task<IReadOnlyList<User>> GetAllUsersAsync(IUserFilter filter)
```

Parameters

filter [IUserFilter](#)

Returns

[Task](#) <[IReadOnlyList](#) <[User](#)>>

## GetUserByEmailAsync(string)

```
public Task<User?> GetUserByEmailAsync(string email)
```

Parameters

email [string](#)

Returns

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

## GetUserByIdAsync(string)

```
public Task<User?> GetUserByIdAsync(string id)
```

Parameters

**id** [string](#)

Returns

[Task](#) <User>

## GetUserByUsernameAsync(string)

```
public Task<User?> GetUserByUsernameAsync(string username)
```

Parameters

**username** [string](#)

Returns

[Task](#) <User>

## UserExistsAsync(string)

```
public Task<bool> UserExistsAsync(string id)
```

Parameters

**id** [string](#)

Returns

[Task](#) <bool>

# Namespace Agora.Infrastructure.Services

## Classes

### [AuthService](#)

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

### [UserContextService](#)

# Class AuthService

Namespace: [Agora.Infrastructure.Services](#)

Assembly: Agora.Infrastructure.dll

Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

```
public class AuthService : IAuthService
```

Inheritance

[object](#) ← AuthService

Implements

[IAuthService](#)

Inherited Members

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

## Constructors

AuthService(SignInManager<User>, UserManager<User>,  
IUserRepository, IOptions<UserSettings>)

```
public AuthService(SignInManager<User> signInManager, UserManager<User> userManager,  
IUserRepository userRepo, IOptions<UserSettings> userSettings)
```

Parameters

signInManager [SignInManager](#)<User>

userManager [UserManager](#)<User>

userRepo [IUserRepository](#)

userSettings [IOptions](#)<UserSettings>

# Methods

## LoginAsync(UserSignInInfo)

Attempts to log in a user with the provided credentials.

```
public Task<Result> LoginAsync(UserSignInInfo signInInfo)
```

Parameters

`signInInfo` [UserSignInInfo](#)

The user's credentials (email address and password).

Returns

[Task](#) <Result>

A [Result](#) indicating whether the login was successful or not.

## LogoutAsync()

Logs out the currently authenticated user.

```
public Task LogoutAsync()
```

Returns

[Task](#)

## RegisterAsync(UserRegistrationInfo)

Registers a new user with the specified registration information.

```
public Task<Result<User>> RegisterAsync(UserRegistrationInfo registrationInfo)
```

Parameters

## **registrationInfo** [UserRegistrationInfo](#)

The user's registration data including email, username, and password.

Returns

[Task](#) <[Result](#)<[User](#)>>

A [Result<T>](#) containing the created [User](#) on success, or an error on failure.

# Class UserContextService

Namespace: [Agora.Infrastructure.Services](#)

Assembly: Agora.Infrastructure.dll

```
public class UserContextService : IUserContextService
```

## Inheritance

[object](#) ← UserContextService

## Implements

[IUserContextService](#)

## Inherited Members

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

## Constructors

### UserContextService(IHttpContextAccessor)

```
public UserContextService(IHttpContextAccessor httpContextAccessor)
```

## Parameters

httpContextAccessor  [IHttpContextAccessor](#)

## Methods

### GetCurrentUserContext()

```
public UserContext GetCurrentUserContext()
```

## Returns

[UserContext](#)

## IsAuthenticated()

```
public bool IsAuthenticated()
```

Returns

[bool](#)