

## My Project

Generated by Doxygen 1.9.8



|   |          |
|---|----------|
| <b>1 Namespace Index</b>  | <b>1</b> |
| 1.1 Namespace List  | 1        |
| <b>2 Hierarchical Index</b>   | <b>3</b> |
| 2.1 Class Hierarchy   | 3        |
| <b>3 Class Index</b>  | <b>5</b> |
| 3.1 Class List  | 5        |
| <b>4 Namespace Documentation</b>                                    | <b>7</b> |
| 4.1 WsCrud.Namespace Reference                                      | 7        |
| 4.2 WsCrud.Controllers.Namespace Reference                          | 7        |
| 4.3 WsCrud.Interfaces.Namespace Reference                           | 7        |
| 4.4 WsCrud.Models.Namespace Reference                               | 7        |
| 4.5 WsCrud.Repositories.Namespace Reference                         | 7        |
| 4.6 WsCrud.Services.Namespace Reference                             | 8        |
| <b>5 Class Documentation</b>  | <b>9</b> |
| 5.1 WsCrud.Services.BasicAuthenticationHandler Class Reference      | 9        |
| 5.1.1 Detailed Description  | 9        |
| 5.2 WsCrud.Services.FileSystemStorage Class Reference               | 9        |
| 5.2.1 Detailed Description  | 10       |
| 5.2.2 Member Function Documentation                                 | 10       |
| 5.2.2.1 Exists()  | 10       |
| 5.2.2.2 Read()  | 10       |
| 5.2.2.3 Write()   | 11       |
| 5.3 WsCrud.Interfaces.IFileStorage Interface Reference              | 11       |
| 5.3.1 Detailed Description  | 12       |
| 5.3.2 Member Function Documentation                                 | 12       |
| 5.3.2.1 Exists()  | 12       |
| 5.3.2.2 Read()  | 12       |
| 5.3.2.3 Write()   | 12       |
| 5.4 WsCrud.Repositories.InMemoryPersonRepository Class Reference    | 13       |
| 5.4.1 Detailed Description  | 14       |
| 5.4.2 Member Function Documentation                                 | 14       |
| 5.4.2.1 AddAsync()  | 14       |
| 5.4.2.2 DeleteAsync()   | 14       |
| 5.4.2.3 GetAllAsync()   | 14       |
| 5.4.2.4 GetByIdAsync()  | 15       |
| 5.4.2.5 UpdateAsync()   | 15       |
| 5.5 WsCrud.Interfaces.IRepository< T > Interface Template Reference | 15       |
| 5.5.1 Detailed Description  | 16       |
| 5.6 WsCrud.Repositories.JsonPersonRepository Class Reference        | 16       |
| 5.6.1 Detailed Description  | 17       |

|  |           |
|--|-----------|
| 5.6.2 Constructor & Destructor Documentation . . . . .             | 17        |
| 5.6.2.1 JsonPersonRepository() . . . . .                           | 17        |
| 5.6.3 Member Function Documentation . . . . .                      | 17        |
| 5.6.3.1 AddAsync() . . . . .                                       | 17        |
| 5.6.3.2 DeleteAsync() . . . . .                                    | 17        |
| 5.6.3.3 GetAllAsync() . . . . .                                    | 18        |
| 5.6.3.4 GetByIdAsync() . . . . .                                   | 18        |
| 5.6.3.5 UpdateAsync() . . . . .                                    | 18        |
| 5.7 WsCrud.Models.Person Class Reference . . . . .                 | 19        |
| 5.7.1 Detailed Description . . . . .                               | 19        |
| 5.8 WsCrud.Controllers.PersonsController Class Reference . . . . . | 19        |
| 5.9 Program Class Reference . . . . .                              | 19        |
| <b>Index</b>   | <b>21</b> |

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

|                                     |   |
|-------------------------------------|---|
| <a href="#">WsCrud</a>              | 7 |
| <a href="#">WsCrud.Controllers</a>  | 7 |
| <a href="#">WsCrud.Interfaces</a>   | 7 |
| <a href="#">WsCrud.Models</a>       | 7 |
| <a href="#">WsCrud.Repositories</a> | 7 |
| <a href="#">WsCrud.Services</a>     | 8 |



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

|  |    |
|--|----|
| AuthenticationHandler                                  |    |
| WsCrud.Services.BasicAuthenticationHandler . . . . .   | 9  |
| ControllerBase   |    |
| WsCrud.Controllers.PersonsController . . . . .         | 19 |
| WsCrud.Interfaces.IFileStorage . . . . .               | 11 |
| WsCrud.Services.FileSystemStorage . . . . .            | 9  |
| WsCrud.Interfaces.IRepository< T > . . . . .           | 15 |
| WsCrud.Interfaces.IRepository< Person > . . . . .      | 15 |
| WsCrud.Repositories.InMemoryPersonRepository . . . . . | 13 |
| WsCrud.Repositories.JsonPersonRepository . . . . .     | 16 |
| WsCrud.Models.Person . . . . .                         | 19 |
| Program . . . . .                                      | 19 |





## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

|   |    |
|---|----|
| <a href="#">WsCrud.Services.BasicAuthenticationHandler</a>  |    |
| Handles HTTP Basic authentication using credentials from environment variables . . . . .  | 9  |
| <a href="#">WsCrud.Services.FileSystemStorage</a>   |    |
| Provides file I/O operations using the actual file system . . . . .   | 9  |
| <a href="#">WsCrud.Interfaces.IFileStorage</a>  |    |
| Defines low-level file operations used by repositories and services . . . . .   | 11 |
| <a href="#">WsCrud.Repositories.InMemoryPersonRepository</a>  |    |
| An in-memory repository storing Person objects in a thread-safe list. Suitable for testing and development purposes . . . . .   | 13 |
| <a href="#">WsCrud.Interfaces.IRepository&lt; T &gt;</a>  |    |
| Generic repository interface for basic CRUD operations . . . . .  | 15 |
| <a href="#">WsCrud.Repositories.JsonPersonRepository</a>  |    |
| JSON-backed implementation of the <a href="#">IRepository&lt;Person&gt;</a> interface. Person records are stored in a single file and serialized/deserialized as JSON . . . . . | 16 |
| <a href="#">WsCrud.Models.Person</a>  |    |
| Represents a person with an Id, Name, and Age . . . . .   | 19 |
| <a href="#">WsCrud.Controllers.PersonsController</a>  |    |
| Handles all CRUD operations for Person resources. Requires Basic Authentication for access .  | 19 |
| <a href="#">Program</a> . . . . .   | 19 |



## Chapter 4

# Namespace Documentation

### 4.1 WsCrud Namespace Reference

### 4.2 WsCrud.Controllers Namespace Reference

#### Classes

- class [PersonsController](#)  
*Handles all CRUD operations for Person resources. Requires Basic Authentication for access.*

### 4.3 WsCrud.Interfaces Namespace Reference

#### Classes

- interface [IFileStorage](#)  
*Defines low-level file operations used by repositories and services.*
- interface [IRepository](#)  
*Generic repository interface for basic CRUD operations.*

### 4.4 WsCrud.Models Namespace Reference

#### Classes

- class [Person](#)  
*Represents a person with an Id, Name, and Age.*

### 4.5 WsCrud.Repositories Namespace Reference

#### Classes

- class [InMemoryPersonRepository](#)  
*An in-memory repository storing Person objects in a thread-safe list. Suitable for testing and development purposes.*
- class [JsonPersonRepository](#)  
*JSON-backed implementation of the [IRepository<Person>](#) interface. Person records are stored in a single file and serialized/deserialized as JSON.*

## 4.6 WsCrud.Services Namespace Reference

### Classes

- class [BasicAuthenticationHandler](#)  
*Handles HTTP Basic authentication using credentials from environment variables.*
- class [FileSystemStorage](#)  
*Provides file I/O operations using the actual file system.*

## Chapter 5

# Class Documentation

### 5.1 WsCrud.Services.BasicAuthenticationHandler Class Reference

Handles HTTP Basic authentication using credentials from environment variables.

Inheritance diagram for WsCrud.Services.BasicAuthenticationHandler:

Collaboration diagram for WsCrud.Services.BasicAuthenticationHandler:

#### Public Member Functions

- **BasicAuthenticationHandler** (IOptionMonitor< AuthenticationSchemeOptions > options, ILoggerFactory logger, System.Text.Encodings.Web.UrlEncoder encoder, ISystemClock clock)  
*Constructs the BasicAuthenticationHandler with required dependencies.*

#### Protected Member Functions

- override Task< AuthenticateResult > **HandleAuthenticateAsync** ()  
*Handles authentication for each incoming request. Validates the "Authorization" header against username and password loaded from environment variables.*

#### 5.1.1 Detailed Description

Handles HTTP Basic authentication using credentials from environment variables.

The documentation for this class was generated from the following file:

- WsCrud/Services/BasicAuthenticationHandler.cs

### 5.2 WsCrud.Services.FileSystemStorage Class Reference

Provides file I/O operations using the actual file system.

Inheritance diagram for WsCrud.Services.FileSystemStorage:

Collaboration diagram for WsCrud.Services.FileSystemStorage:

#### Public Member Functions

- bool **Exists** (string path)  
*Checks if a file exists at the specified path.*

**Parameters**

|      |            |
|------|------------|
| path | File path. |
|------|------------|

**Returns**

*True if the file exists; otherwise, false.*

- string [Read](#) (string path)

*Reads all text from the specified file path.*

**Parameters**

|      |            |
|------|------------|
| path | File path. |
|------|------------|

**Returns**

*Contents of the file as string.*

- void [Write](#) (string path, string content)

*Writes text content to the specified file path.*

**Parameters**

|         |                        |
|---------|------------------------|
| path    | Destination file path. |
| content | Text content to write. |

## 5.2.1 Detailed Description

Provides file I/O operations using the actual file system.

## 5.2.2 Member Function Documentation

### 5.2.2.1 Exists()

```
bool WsCrud.Services.FileSystemStorage.Exists (
    string path )
```

Checks if a file exists at the specified path.

**Parameters**

|             |            |
|-------------|------------|
| <i>path</i> | File path. |
|-------------|------------|

**Returns**

True if the file exists; otherwise, false.

Implements [WsCrud.Interfaces.IFileStorage](#).

### 5.2.2.2 Read()

```
string WsCrud.Services.FileSystemStorage.Read (
    string path )
```

Reads all text from the specified file path.

#### Parameters

|             |            |
|-------------|------------|
| <i>path</i> | File path. |
|-------------|------------|

#### Returns

Contents of the file as string.

Implements [WsCrud.Interfaces.IFileStorage](#).

#### 5.2.2.3 Write()

```
void WsCrud.Services.FileSystemStorage.Write (
    string path,
    string content )
```

Writes text content to the specified file path.

#### Parameters

|                |                        |
|----------------|------------------------|
| <i>path</i>    | Destination file path. |
| <i>content</i> | Text content to write. |

Implements [WsCrud.Interfaces.IFileStorage](#).

The documentation for this class was generated from the following file:

- WsCrud/Services/FileSystemStorage.cs

## 5.3 WsCrud.Interfaces.IFileStorage Interface Reference

Defines low-level file operations used by repositories and services.

Inheritance diagram for WsCrud.Interfaces.IFileStorage:

#### Public Member Functions

- bool [Exists](#) (string path)  
*Checks if a file exists at the specified path.*
- string [Read](#) (string path)  
*Reads all text from the specified file path.*
- void [Write](#) (string path, string content)  
*Writes text content to the specified file path.*

### 5.3.1 Detailed Description

Defines low-level file operations used by repositories and services.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 Exists()

```
bool WsCrud.Interfaces.IFileStorage.Exists (
    string path )
```

Checks if a file exists at the specified path.

##### Parameters

|             |            |
|-------------|------------|
| <i>path</i> | File path. |
|-------------|------------|

##### Returns

True if the file exists; otherwise, false.

Implemented in [WsCrud.Services.FileSystemStorage](#).

#### 5.3.2.2 Read()

```
string WsCrud.Interfaces.IFileStorage.Read (
    string path )
```

Reads all text from the specified file path.

##### Parameters

|             |            |
|-------------|------------|
| <i>path</i> | File path. |
|-------------|------------|

##### Returns

Contents of the file as string.

Implemented in [WsCrud.Services.FileSystemStorage](#).

#### 5.3.2.3 Write()

```
void WsCrud.Interfaces.IFileStorage.Write (
    string path,
    string content )
```

Writes text content to the specified file path.



## Parameters

|                |                        |
|----------------|------------------------|
| <i>path</i>    | Destination file path. |
| <i>content</i> | Text content to write. |

Implemented in [WsCrud.Services.FileSystemStorage](#).

The documentation for this interface was generated from the following file:

- WsCrud/Interfaces/IFileStorage.cs

## 5.4 WsCrud.Repositories.InMemoryPersonRepository Class Reference

An in-memory repository storing Person objects in a thread-safe list. Suitable for testing and development purposes.

Inheritance diagram for WsCrud.Repositories.InMemoryPersonRepository:

Collaboration diagram for WsCrud.Repositories.InMemoryPersonRepository:

### Public Member Functions

- Task< List< [Person](#) > > [GetAllAsync](#) ()  
*Retrieves all people from the repository.*
- Task< [Person](#)?> [GetByIdAsync](#) (int id)  
*Retrieves a person by their ID.*
- Task [AddAsync](#) ([Person](#) person)  
*Adds a new person to the repository and assigns a unique ID.*
- Task [UpdateAsync](#) ([Person](#) person)  
*Updates an existing person in the repository.*
- Task [DeleteAsync](#) (int id)  
*Deletes a person from the repository by ID.*

### Public Member Functions inherited from [WsCrud.Interfaces.IRepository](#)< [Person](#) >

- Task< List< T > > [GetAllAsync](#) ()  
*Gets all entities.*
- Task< T?> [GetByIdAsync](#) (int id)  
*Gets a specific entity by ID.*
- Task [AddAsync](#) (T entity)  
*Adds a new entity.*
- Task [UpdateAsync](#) (T entity)  
*Updates an existing entity.*
- Task [DeleteAsync](#) (int id)  
*Deletes an entity by ID.*

### 5.4.1 Detailed Description

An in-memory repository storing Person objects in a thread-safe list. Suitable for testing and development purposes.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 AddAsync()

```
Task WsCrud.Repositories.InMemoryPersonRepository.AddAsync (
    Person person ) [inline]
```

Adds a new person to the repository and assigns a unique ID.

##### Parameters

|               |   |
|---------------|---|
| <i>person</i> | The person object to add. The ID will be overwritten. |
|---------------|---|

##### Returns

A completed task.

#### 5.4.2.2 DeleteAsync()

```
Task WsCrud.Repositories.InMemoryPersonRepository.DeleteAsync (
    int id ) [inline]
```

Deletes a person from the repository by ID.

##### Parameters

|           |  |
|-----------|--|
| <i>id</i> | The unique identifier of the person to remove. |
|-----------|--|

##### Returns

A completed task.

#### 5.4.2.3 GetAllAsync()

```
Task< List< Person > > WsCrud.Repositories.InMemoryPersonRepository.GetAllAsync ( ) [inline]
```

Retrieves all people from the repository.

##### Returns

A task that returns a new list of all Person objects.

#### 5.4.2.4 GetByIdAsync()

```
Task< Person?> WsCrud.Repositories.InMemoryPersonRepository.GetByIdAsync (
    int id ) [inline]
```

Retrieves a person by their ID.

##### Parameters

|           |                                      |
|-----------|--------------------------------------|
| <i>id</i> | The unique identifier of the person. |
|-----------|--------------------------------------|

##### Returns

A task that returns the matching Person, or null if not found.

#### 5.4.2.5 UpdateAsync()

```
Task WsCrud.Repositories.InMemoryPersonRepository.UpdateAsync (
    Person person ) [inline]
```

Updates an existing person in the repository.

##### Parameters

|               |  |
|---------------|--|
| <i>person</i> | The updated person. The person is matched by ID. |
|---------------|--|

##### Returns

A completed task.

The documentation for this class was generated from the following file:

- WsCrud/Repositories/InMemoryPersonRepository.cs

## 5.5 WsCrud.Interfaces.IRepository< T > Interface Template Reference

Generic repository interface for basic CRUD operations.

### Public Member Functions

- Task< List< T > > **GetAllAsync** ()  
*Gets all entities.*
- Task< T?> **GetByIdAsync** (int id)  
*Gets a specific entity by ID.*
- Task **AddAsync** (T entity)  
*Adds a new entity.*
- Task **UpdateAsync** (T entity)  
*Updates an existing entity.*
- Task **DeleteAsync** (int id)  
*Deletes an entity by ID.*

### 5.5.1 Detailed Description

Generic repository interface for basic CRUD operations.

#### Template Parameters

|          |              |
|----------|--------------|
| <i>T</i> | Entity type. |
|----------|--------------|

The documentation for this interface was generated from the following file:

- WsCrud/Interfaces/IRepository.cs

## 5.6 WsCrud.Repositories.JsonPersonRepository Class Reference

JSON-backed implementation of the [IRepository<Person>](#) interface. Person records are stored in a single file and serialized/deserialized as JSON.

Inheritance diagram for WsCrud.Repositories.JsonPersonRepository:

Collaboration diagram for WsCrud.Repositories.JsonPersonRepository:

#### Public Member Functions

- [JsonPersonRepository](#) (string filePath, [IFileStorage](#) storage)  
*Initializes a new instance of the JsonPersonRepository class. Loads existing data from the JSON file if it exists.*
- Task< List< [Person](#) > > [GetAllAsync](#) ()  
*Retrieves all persons from memory.*
- Task< [Person](#)?> [GetByIdAsync](#) (int id)  
*Retrieves a person with the specified ID.*
- Task [AddAsync](#) ([Person](#) person)  
*Adds a new person to the list, assigning them a unique ID. Also persists the updated list to disk.*
- Task [UpdateAsync](#) ([Person](#) person)  
*Updates an existing person by ID. Overwrites the stored object.*
- Task [DeleteAsync](#) (int id)  
*Deletes all persons with the specified ID and saves the result.*

#### Public Member Functions inherited from [WsCrud.Interfaces.IRepository< Person >](#)

- Task< List< T > > [GetAllAsync](#) ()  
*Gets all entities.*
- Task< T?> [GetByIdAsync](#) (int id)  
*Gets a specific entity by ID.*
- Task [AddAsync](#) (T entity)  
*Adds a new entity.*
- Task [UpdateAsync](#) (T entity)  
*Updates an existing entity.*
- Task [DeleteAsync](#) (int id)  
*Deletes an entity by ID.*

### 5.6.1 Detailed Description

JSON-backed implementation of the [IRepository<Person>](#) interface. Person records are stored in a single file and serialized/deserialized as JSON.

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 JsonPersonRepository()

```
WsCrud.Repositories.JsonPersonRepository.JsonPersonRepository (
    string filePath,
    IFileStorage storage ) [inline]
```

Initializes a new instance of the JsonPersonRepository class. Loads existing data from the JSON file if it exists.

##### Parameters

|                 |                                    |
|-----------------|------------------------------------|
| <i>filePath</i> | The file path to the JSON storage. |
| <i>storage</i>  | An injected file I/O abstraction.  |

### 5.6.3 Member Function Documentation

#### 5.6.3.1 AddAsync()

```
Task WsCrud.Repositories.JsonPersonRepository.AddAsync (
    Person person ) [inline]
```

Adds a new person to the list, assigning them a unique ID. Also persists the updated list to disk.

##### Parameters

|               |                        |
|---------------|------------------------|
| <i>person</i> | The new person to add. |
|---------------|------------------------|

##### Returns

A completed task.

#### 5.6.3.2 DeleteAsync()

```
Task WsCrud.Repositories.JsonPersonRepository.DeleteAsync (
    int id ) [inline]
```

Deletes all persons with the specified ID and saves the result.

##### Parameters

|           |                                    |
|-----------|------------------------------------|
| <i>id</i> | The ID of the person(s) to delete. |
|-----------|------------------------------------|

**Returns**

A completed task.

**5.6.3.3 GetAllAsync()**

```
Task< List< Person > > WsCrud.Repositories.JsonPersonRepository.GetAllAsync ( ) [inline]
```

Retrieves all persons from memory.

**Returns**

A task that returns a new copy of the person list.

**5.6.3.4 GetByIdAsync()**

```
Task< Person?> WsCrud.Repositories.JsonPersonRepository.GetByIdAsync (
    int id ) [inline]
```

Retrieves a person with the specified ID.

**Parameters**

|           |                              |
|-----------|------------------------------|
| <i>id</i> | The unique ID of the person. |
|-----------|------------------------------|

**Returns**

A task that returns the matching person if found; otherwise, null.

**5.6.3.5 UpdateAsync()**

```
Task WsCrud.Repositories.JsonPersonRepository.UpdateAsync (
    Person person ) [inline]
```

Updates an existing person by ID. Overwrites the stored object.

**Parameters**

|               |                            |
|---------------|----------------------------|
| <i>person</i> | The updated person object. |
|---------------|----------------------------|

**Returns**

A completed task.

The documentation for this class was generated from the following file:

- WsCrud/Repositories/JsonPersonRepository.cs

## 5.7 WsCrud.Models.Person Class Reference

Represents a person with an Id, Name, and Age.

### Properties

- **int Id** [get, set]  
*Unique identifier (auto-assigned).*
- **string Name** = string.Empty [get, set]  
*Full name of the person.*
- **int Age** [get, set]  
*Age in years.*

### 5.7.1 Detailed Description

Represents a person with an Id, Name, and Age.

The documentation for this class was generated from the following file:

- WsCrud/Models/Person.cs

## 5.8 WsCrud.Controllers.PersonsController Class Reference

Handles all CRUD operations for Person resources. Requires Basic Authentication for access.

Inheritance diagram for WsCrud.Controllers.PersonsController:

## 5.9 Program Class Reference

The documentation for this class was generated from the following file:

- WsCrud/Program.cs





# Index

- AddAsync
  - WsCrud.Repositories.InMemoryPersonRepository, [14](#)
  - WsCrud.Repositories.JsonPersonRepository, [17](#)
- DeleteAsync
  - WsCrud.Repositories.InMemoryPersonRepository, [14](#)
  - WsCrud.Repositories.JsonPersonRepository, [17](#)
- Exists
  - WsCrud.Interfaces.IFileStorage, [12](#)
  - WsCrud.Services.FileSystemStorage, [10](#)
- GetAllAsync
  - WsCrud.Repositories.InMemoryPersonRepository, [14](#)
  - WsCrud.Repositories.JsonPersonRepository, [18](#)
- GetByIdAsync
  - WsCrud.Repositories.InMemoryPersonRepository, [14](#)
  - WsCrud.Repositories.JsonPersonRepository, [18](#)
- JsonPersonRepository
  - WsCrud.Repositories.JsonPersonRepository, [17](#)
- Program, [19](#)
- Read
  - WsCrud.Interfaces.IFileStorage, [12](#)
  - WsCrud.Services.FileSystemStorage, [10](#)
- UpdateAsync
  - WsCrud.Repositories.InMemoryPersonRepository, [15](#)
  - WsCrud.Repositories.JsonPersonRepository, [18](#)
- Write
  - WsCrud.Interfaces.IFileStorage, [12](#)
  - WsCrud.Services.FileSystemStorage, [11](#)
- WsCrud, [7](#)
- WsCrud.Controllers, [7](#)
- WsCrud.Controllers.PersonsController, [19](#)
- WsCrud.Interfaces, [7](#)
- WsCrud.Interfaces.IFileStorage, [11](#)
  - Exists, [12](#)
  - Read, [12](#)
  - Write, [12](#)
- WsCrud.Interfaces.IRepository< T >, [15](#)
- WsCrud.Models, [7](#)
- WsCrud.Models.Person, [19](#)
- WsCrud.Repositories, [7](#)
- WsCrud.Repositories.InMemoryPersonRepository, [13](#)
  - AddAsync, [14](#)
  - DeleteAsync, [14](#)
  - GetAllAsync, [14](#)
  - GetByIdAsync, [14](#)
  - UpdateAsync, [15](#)
- WsCrud.Repositories.JsonPersonRepository, [16](#)
  - AddAsync, [17](#)
  - DeleteAsync, [17](#)
  - GetAllAsync, [18](#)
  - GetByIdAsync, [18](#)
  - JsonPersonRepository, [17](#)
  - UpdateAsync, [18](#)
- WsCrud.Services, [8](#)
- WsCrud.Services.BasicAuthenticationHandler, [9](#)
- WsCrud.Services.FileSystemStorage, [9](#)
  - Exists, [10](#)
  - Read, [10](#)
  - Write, [11](#)