**Archive Package Documentation**

**Overview**

**ArchiveService:** is a service that interacts with a remote API for archiving purposes. It provides two primary methods:

1. **SaveFiles**: Uploads files and associated metadata to the archive.
2. **AuthorizedUrl**: Retrieves an authorized URL for a file in the archive.

**Enums**

namespace ArchivePackage.Enums;

public enum EnvironmentType

{

Release = 0,

Debug = 1

}

**DTOs**

**SaveFileDTO :**

namespace ArchivePackage.DTOs;

public class SaveFileDTO

{

public int UserId { get; set; }

public int ProjectId { get; set; }

public string RequestId { get; set; }

public List<IFormFile> Files { get; set; }

public EnvironmentType State { get; set; }

}

**GetTokenDTO :**

namespace ArchivePackage.DTOs;

public class GetTokenDTO

{

public string UserName { get; set; } = string.Empty;

public string Password { get; set; } = string.Empty;

public int Minutes { get; set; }

}

#### Public Methods

##### 1. **SaveFiles (SaveFileDTO saveFile)**

Uploads files to the archive service with the provided metadata.

* **Parameters:**
  + **saveFile**: An object of type **SaveFileDTO** containing:
    - **UserId (**int**)**: The ID of the user initiating the request.
    - **ProjectId (**int**)**: The ID of the project associated with the request.
    - **RequestId (**string**)**: A string id for the request.
    - **Files (**List<IFormFile>**)**: A list of files to be uploaded.
    - **State (**EnvironmentType**)**: An Enum which represent if we in debug or release environment.
* **Returns:**
  + A **GenericResponse<List<string>>** indicating the result of the operation, which includes:
    - A list of strings if successful.
    - A failure message if the request was unsuccessful.
* **Example** :

using ArchivePackage.Services;

namespace ArchivePackage;

public class TestArchive

{

public void TestSaveFiles ()

{

var SaveFile = new SaveFileDTO()

{

UserId = 1,

RequestId = "11553",

ProjectId = 1,

State = EnvironmentType.Debug,

Files = []

};

var result = ArchiveService.SaveFiles(SaveFile);

}

}

##### 2. **AuthorizedUrl (string fileURL, EnvironmentType State)**

Fetches an authorized URL for a file in the archive.

* **Parameters:**
  + **fileURL**: The URL of the file for which the authorized URL is to be retrieved.
  + **State**: An enum value (Debug or Release) indicating the environment type.
* **Returns:**
  + A **GenericResponse<string>** containing the authorized URL if successful.
  + A failure message if the request was unsuccessful.
* **Example:**

using ArchivePackage.Services;

namespace ArchivePackage;

public class TestArchive

{

public void TestAuthorizedUrl()

{

string fileURL = "https://example.com/sample.pdf";

EnvironmentType state = EnvironmentType.Debug;

Task<GenericResponse<string>>? response = ArchiveService.AuthorizedUrl(fileURL, state);

}

}

#### Dependencies

* Microsoft.AspNetCore.Http;
* ArchivePackage.Enums;
* PremiumMainCS.GenericResponse;

#### Environment

* **Debug API URL**: <https://iistest.premiumcard.net:7000>
* **Release API URL**: <https://iis.premiumcard.net:7000>