

TAAS - Feature #494

Backup DB

01/28/2025 05:45 PM - Arathy PS

Status:	Ready for Coding	Start date:	01/28/2025
Priority:	Normal	Due date:	
Assignee:	dheeraja premdas	% Done:	20%
Category:		Estimated time:	0:00 hour
Target version:	22.12	Spent time:	0:00 hour
Owner(Agency):	Travvise	Tested By:	
Time Taken(HH):		Code Reviewed By:	
Module:	TAAS-General		
Description			

History

#1 - 03/29/2025 02:09 PM - Anil KV

- Status changed from New to Ready for Coding
- % Done changed from 0 to 20

Ref Sample Code

```
using System;
using System.Diagnostics;
using Ionic.Zip; // Install DotNetZip from NuGet: Install-Package DotNetZip

class Program
{
    static void Main(string[] args)
    {
        // PostgreSQL Configuration
        string pgDumpPath = @"C:\Program Files\PostgreSQL\15\bin\pg_dump.exe"; // Path to pg_dump
        string host = "localhost";
        string database = "your_database_name";
        string username = "your_username";
        string password = "your_password"; // Use environment variables for better security
        string dumpFilePath = @"C:\backup\database_backup.sql"; // Backup file path
        string zipFilePath = @"C:\backup\database_backup.zip"; // ZIP file path
        string zipPassword = "your_secure_password"; // Password for the ZIP file

        // Step 1: Generate PostgreSQL Backup
        ProcessStartInfo startInfo = new ProcessStartInfo
        {
            FileName = pgDumpPath,
            Arguments = $"-h {host} -U {username} -d {database} -F c -f \"{dumpFilePath}\"",
            RedirectStandardOutput = true,
            RedirectStandardError = true,
            UseShellExecute = false,
            CreateNoWindow = true
        };

        startInfo.Environment["PGPASSWORD"] = password; // Set PostgreSQL password securely

        try
        {
            using (Process process = Process.Start(startInfo))
            {
                process.WaitForExit();

                if (process.ExitCode == 0)
                {
                    Console.WriteLine($"Database backup successfully created: {dumpFilePath}");
                }
            }
        }
    }
}
```

```

        // Step 2: Compress Backup into ZIP with Password
        using (ZipFile zip = new ZipFile())
        {
            zip.Password = zipPassword; // Add password to the ZIP file
            zip.AddFile(dumpFilePath, ""); // Add the SQL dump file to the ZIP
            zip.Save(zipFilePath); // Save ZIP file to disk
        }

        Console.WriteLine($"Backup successfully compressed and password-protected: {zipFilePath}")
;
        }
        else
        {
            Console.WriteLine($"Error during backup: {process.StandardError.ReadToEnd()}");
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine($"An error occurred: {ex.Message}");
}
}
}

```

or

```

using System;
using System.Diagnostics;

class Program
{
    static void Main(string[] args)
    {
        // PostgreSQL Configuration
        string pgDumpPath = @"C:\Program Files\PostgreSQL\15\bin\pg_dump.exe"; // Path to pg_dump
        string host = "localhost"; // PostgreSQL host
        string database = "your_database_name"; // Database to backup
        string username = "your_username"; // PostgreSQL user
        string password = "your_password"; // Secure password, consider using environment variables
        string backupFilePath = @"C:\backup\database_backup.sql"; // File path for backup

        // Command construction for pg_dump
        string command = $"-h {host} -U {username} -d {database} -F c -f \"{backupFilePath}\"";

        // Create process to execute command
        ProcessStartInfo processStartInfo = new ProcessStartInfo
        {
            FileName = pgDumpPath, // Path to pg_dump executable
            Arguments = command, // pg_dump command arguments
            RedirectStandardOutput = true,
            RedirectStandardError = true,
            UseShellExecute = false,
            CreateNoWindow = true
        };

        // Set the PGPASSWORD environment variable
        processStartInfo.Environment["PGPASSWORD"] = password;

        try
        {
            using (Process process = Process.Start(processStartInfo))
            {
                // Wait for process to complete
                process.WaitForExit();

                if (process.ExitCode == 0)
                {
                    Console.WriteLine("Database backup completed successfully.");
                }
                else
                {
                    Console.WriteLine($"Backup failed: {process.StandardError.ReadToEnd()}");
                }
            }
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        Console.WriteLine($"An error occurred: {ex.Message}");
    }
}
}

```

#2 - 03/29/2025 02:15 PM - Anil KV

File return from Controller

```

using Microsoft.AspNetCore.Mvc;
using System.IO;

[ApiController]
[Route("api/[controller]")]
public class FileController : ControllerBase
{
    [HttpGet("download-zip")]
    public IActionResult DownloadZip()
    {
        string zipFilePath = @"C:\backup\database_backup.zip"; // Path to the ZIP file

        if (!System.IO.File.Exists(zipFilePath))
        {
            return NotFound("File not found.");
        }

        var fileBytes = System.IO.File.ReadAllBytes(zipFilePath);

        return File(fileBytes, "application/zip", "database_backup.zip");
    }
}

```

#3 - 03/29/2025 02:37 PM - Anil KV

=>Menu:-

->More->Tools->DB Backup

->GUI Name -> Database Backup

->Permission only admin and system admin users

=>Location:-

->Base

=>GUI:-

->Left Side

-Data Base - Selectbox -> Main DB(Default), Log DB

-File Type - Selectbox -> Zip File

-Password - Checkbox(Tick)->System Password + Inputbox -> Disable the 2 comp

*Use Database password for zip

->Right Side(Label table)

Last Backup Date : Date & Time

Last Backup User :

Last Backup DB :

#4 - 04/05/2025 12:00 PM - Anil KV

- Assignee changed from travvise Admin to dheeraja premdas