

Technical Documentation

Task Management API

ASP.NET Core (.NET 10) with SQL
Server 2025

Platform & Database

Platform

ASP.NET Core (.NET 10)

Database

SQL Server 2025

Tools

SSMS 20, Visual Studio Insider

Architecture: **Layered Architecture** (Controller → Service → Repository → Database)

1) Project Purpose

This project is a **Task Management API** providing the following features:

- User Registration & Login with JWT
- Task Management (CRUD)
- Task list & advanced search
- Full Task Change History (Audit Trail)
- User Administration
- Database operations via Stored Procedures
- SQL Server 2025 integration using Dapper
- Enterprise-grade layered architecture

2) Technologies

Technology	Description
ASP.NET Core (.NET 10 Web API)	Core API development
C# 12	Main programming language
SQL Server 2025	Primary database
SSMS 20	SQL Server management
Dapper	Executing stored procedures
JWT Authentication	Authentication & access control
Stored Procedures	Insert / Update / Delete / Search
AppDbContext (EF Core minimal)	Model binding only – no migrations
Postman	API testing

3) Database Structure

Database Name: **InternalProcessDb**

dbo.Users

Column	Type	Description
Id	int	Primary Key
Username	nvarchar	Unique username
PasswordHash	nvarchar	Encrypted password hash
Role	nvarchar	User role (Admin / Researcher)

dbo.Tasks

Column	Type	Description
Id	int	Primary Key
Title	nvarchar	Task title
Description	nvarchar	Task description
Status	nvarchar	Task status (Pending / Done / ...)
AssignedTo	int	Assigned user
CreatedAt	datetime	Creation timestamp
UpdatedAt	datetime	Last updated timestamp

dbo.TaskHistory

Records the complete change history of each task.

Column	Type	Description
Id	int	Primary Key
TaskId	int	Modified task ID
Action	nvarchar	Create / Update / Delete
OldValues	nvarchar(max)	Old values
NewValues	nvarchar(max)	New values
ChangedAt	datetime	Change timestamp
ChangedBy	int	User who made the change

Actual Tables in Your Database

dbo.Users

dbo.Roles

dbo.Tasks

dbo.TaskHistory

dbo._EFMigrationsHistory (system)

Stored Procedures

Location: **Programmability → Stored Procedures → dbo**

- dbo.Task_Insert
- dbo.Task_Update
- dbo.Task_Delete
- dbo.Task_GetById
- dbo.Task_Search
- dbo.Task_UpdateStatus

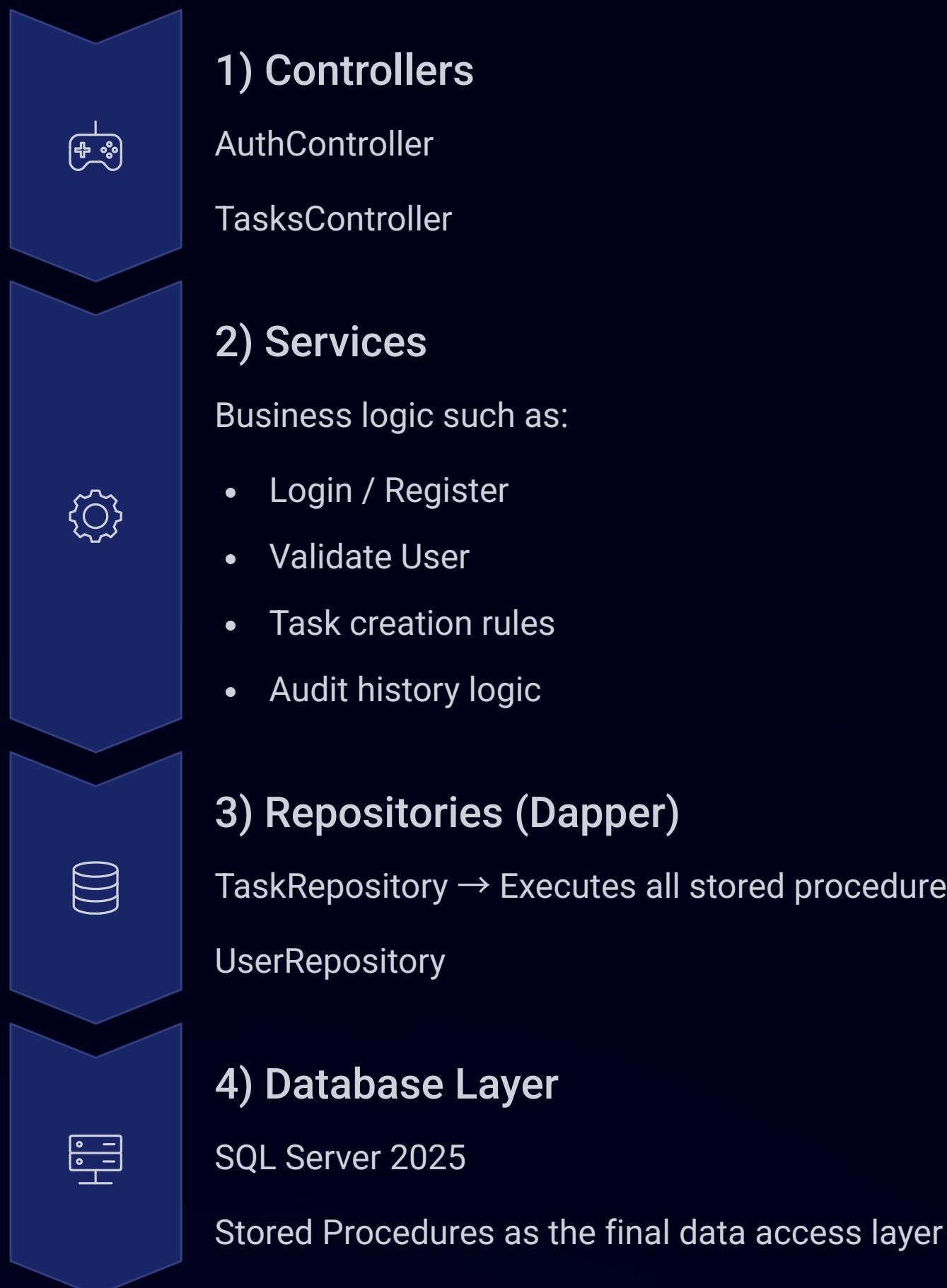
4) Project Architecture

Updated according to your real project structure.

Solution Structure (from Visual Studio Insider)

```
Connected Services
Dependencies
Properties
Controllers
AuthController.cs
TasksController.cs
WeatherForecastController.cs
Data
AppDbContext.cs
DbInitializer.cs
DTOs
TaskCreateDto.cs
TaskQueryDto.cs
TaskUpdateDto.cs
UserLoginDto.cs
UserRegisterDto.cs
Models
Role.cs
TaskHistory.cs
TaskItem.cs
User.cs
Repositories
ITaskRepository.cs
IUserRepository.cs
TaskRepository.cs
UserRepository.cs
Services
appsettings.json
Internal ProcessMgmt.http
Program.cs
WeatherForecast.cs
```

Layered Architecture



DTOs, Models & Data Layer

5) DTOs

- TaskCreateDto
- TaskUpdateDto
- TaskQueryDto
- UserLoginDto
- UserRegisterDto

6) Models

- User
- TaskItem
- TaskHistory
- Role (enum)

7) Data Layer

AppDbContext (used
only for model binding
– no migrations)

DbInitializer (optional
seed roles or test
data)

5) Project Screenshots

1-5) Postman – Register User

The screenshot shows the Postman interface for a 'Register' API endpoint. The request is a POST method to `http://localhost:5000/api/auth/register`. The body is set to raw JSON with the following content:

```
1  {
2    "username": "user2",
3    "email": "user2@example.com",
4    "password": "123456",
5    "role": "Researcher"
6 }
```

The response status is 200 OK, with a duration of 946 ms and a size of 265 B. A note indicates 'Response not stored' because the 'Save response setting' was off.

2-5) Postman – Login + Token

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'naghmeh etaate's Workspace' containing collections, environments, history, flows, and files. The main area displays a POST request to 'https://localhost:5001/api/auth/login'. The request body is set to raw JSON:

```
1 {
2   "username": "user2",
3   "password": "123456"
4 }
```

The response status is 200 OK, with a duration of 699 ms and a size of 484 B. A message at the bottom says 'Response not stored'.

3-5) Create Task (POST /tasks)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'naghmeh etaate's Workspace' containing collections, environments, history, flows, and files. The main area displays a POST request to 'https://localhost:5001/api/tasks'. The request headers are set to:

Key	Value	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWlOIj1c2...	
Content-Type	application/json	

The response status is 201 Created, with a duration of 328 ms and a size of 448 B. A message at the bottom says 'Response not stored'.

Task Operations

The screenshot shows the Postman application interface. On the left, the sidebar displays 'naghmeh etaate's Workspace' with a collection named 'InternalProcessMgmt' containing various API endpoints like 'Get All Tasks', 'Get Task By ID', 'Create Task', etc. The main panel shows a POST request to 'https://localhost:5001/api/tasks'. The request body contains the following JSON:

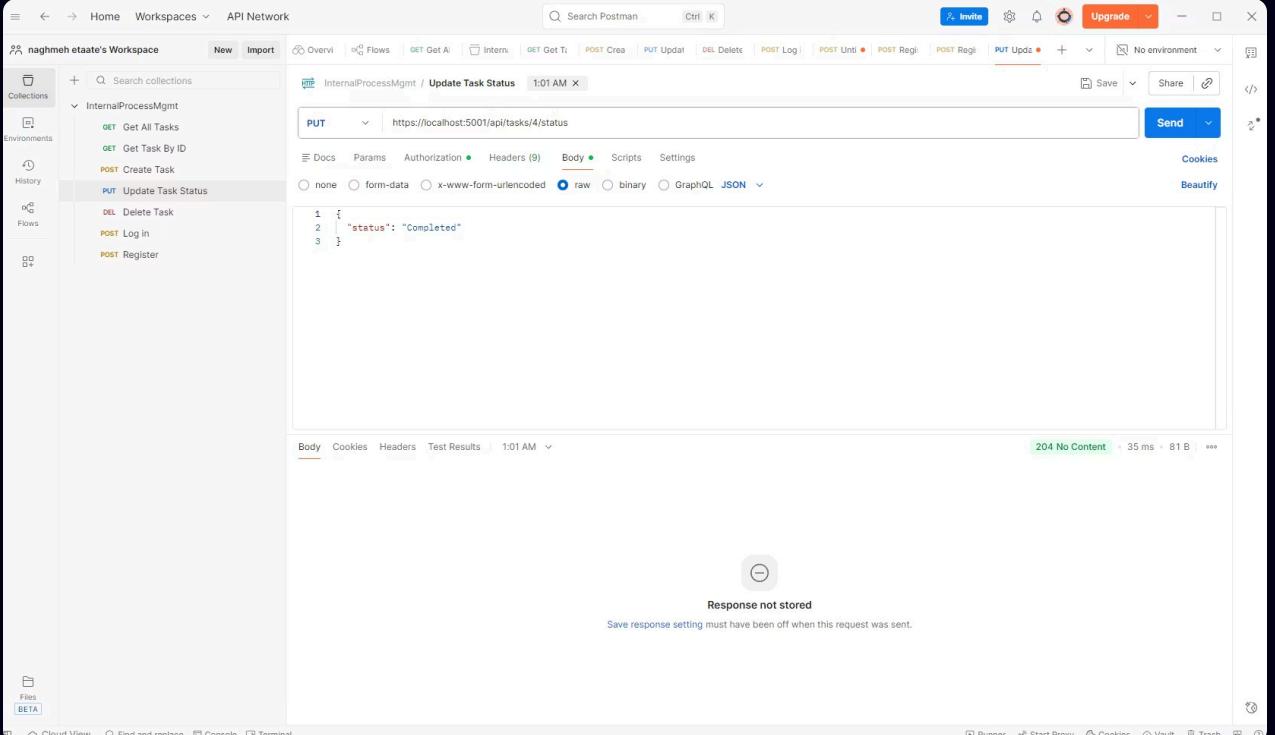
```
1 {  
2   "title": "Second Task",  
3   "description": "Details for second task",  
4   "status": "Pending",  
5   "priority": "High",  
6   "assignedToUserId": 5  
7 }  
8
```

The response status is '201 Created' with a duration of '328 ms' and a size of '448 B'. A message at the bottom states 'Response not stored'.

4-5) Update Task

The screenshot shows the Postman application interface. On the left, the sidebar displays 'naghmeh etaate's Workspace' with a collection named 'InternalProcessMgmt' containing various API endpoints like 'Get All Tasks', 'Get Task By ID', 'Create Task', etc. The main panel shows a PUT request to 'https://localhost:5001/api/tasks/4/status'. The request headers section includes an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWlrcjI...'. The response status is '204 No Content' with a duration of '35 ms' and a size of '81 B'. A message at the bottom states 'Response not stored'.

Delete Task & Solution Structure

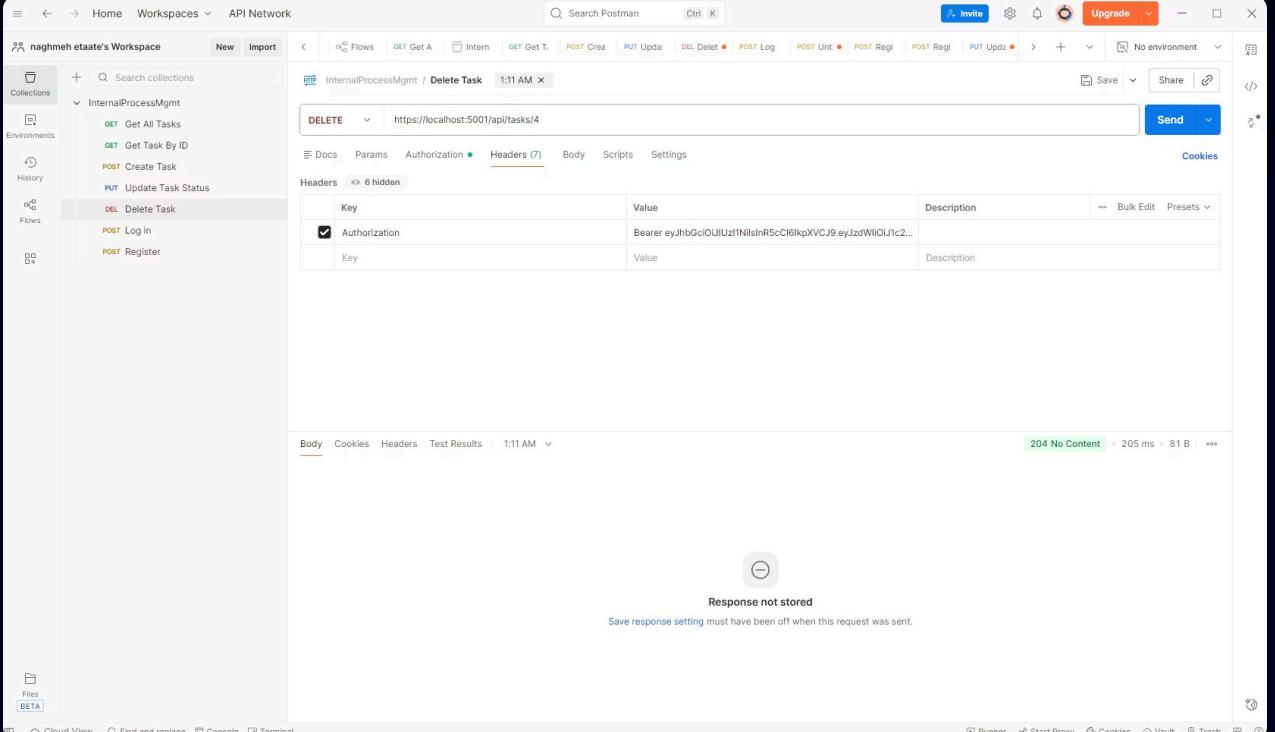


The screenshot shows the Postman application interface. On the left, the 'Collections' sidebar is open, displaying a collection named 'InternalProcessMgmt' with several requests: 'Get All Tasks', 'Get Task By ID', 'Create Task', 'Update Task Status', 'Delete Task', 'Log In', and 'Register'. The 'Update Task Status' request is selected and expanded. The main workspace shows a 'PUT https://localhost:5001/api/tasks/4/status' request. The 'Body' tab contains the following JSON payload:

```
1: { "status": "Completed" }
```

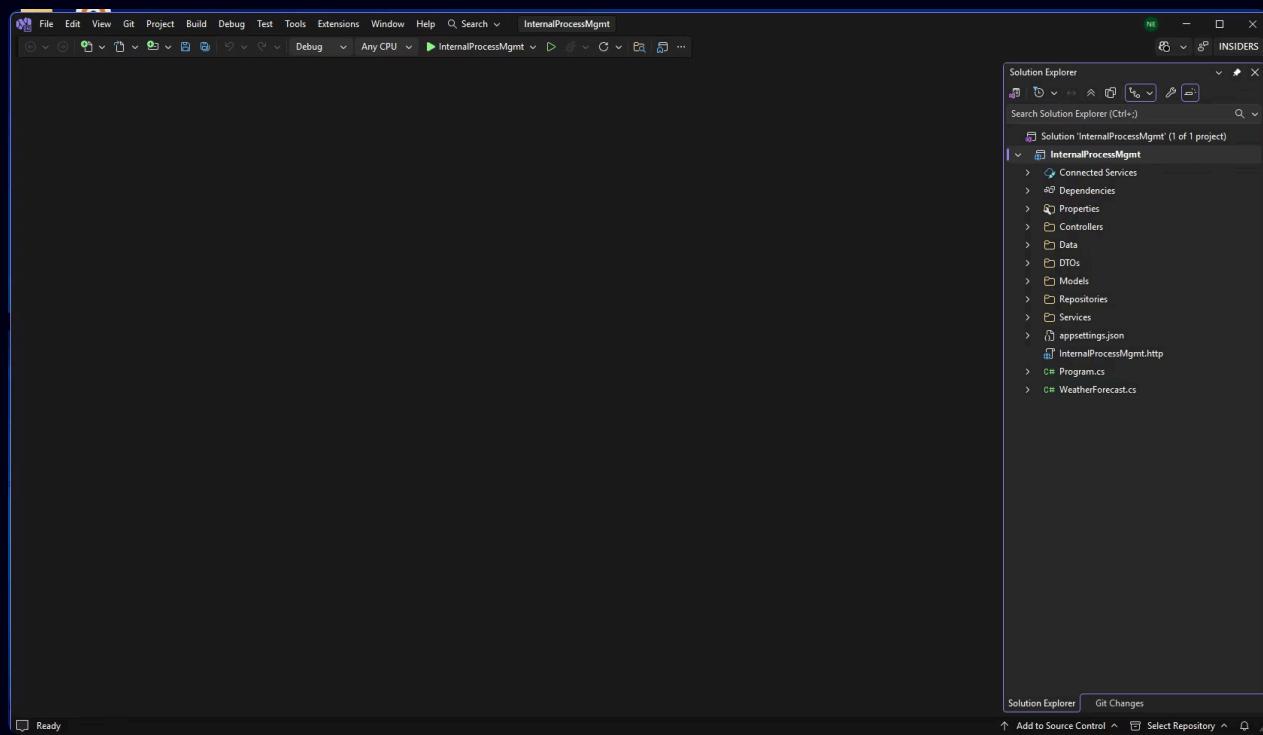
The response section shows a 204 No Content status with a 35 ms duration and 81 B size. A message at the bottom says 'Response not stored'.

5-5) Delete Task



The screenshot shows the Postman application interface. The 'Collections' sidebar is open, displaying the same 'InternalProcessMgmt' collection. The 'Delete Task' request is selected and expanded. The main workspace shows a 'DELETE https://localhost:5001/api/tasks/4' request. The 'Headers' tab shows an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJtC2...'. The response section shows a 204 No Content status with a 205 ms duration and 81 B size. A message at the bottom says 'Response not stored'.

6-5) Solution Structure – Visual Studio



7-5) TaskHistory Table (SSMS)

A screenshot of the SQL Server Management Studio (SSMS) interface. The query window title is "SQLQuery1.sql - loc...rocessDb (sa (106))". The query is: "SELECT * FROM TaskHistory ORDER BY ChangedAt DESC;". The results pane shows a single row of data in a table format. The columns are: HistoryId, TaskId, ChangedByUserId, ChangeType, OldValue, NewValue, and ChangedAt. The data is: HistoryId 1, TaskId 1, ChangedByUserId 1, ChangeType Create, OldValue NULL, NewValue Initial Task, and ChangedAt 2025-12-03 13:53:42.8542589. Below the results, the status bar says "Query executed successfully." and "localhost (17.0 RTM) | sa (106) | InternalProcessDb | 00:00:00 | 1 rows".