

BLOQS

BLOQS IS A HYPERMODERN LOW-CODE
PLATFORM DESIGNED FOR DEVELOPERS
WHO LOVE C#.



PLATFORM OVERVIEW

Bloqs is a powerful low-code platform specifically designed for developers who love C#. Unlike traditional low-code solutions that force you into proprietary languages, Bloqs allows you to create modern applications both visually and programmatically while staying within the familiar C# and .NET ecosystem.

HIGHLIGHTS

One powerful programming language: C#

Modeling: both in code and in browser

Just a NuGet package

Based on commodity tech

Deploy anywhere



TECH STACK



BLAZOR WASM



MINIMAL WEB APIS



MONGODB
(DEFAULT
DATABASE)



.NET ASPIRE
(DEPLOYMENT)

CORE PHILOSOPHY



Stay in C#: Whether building UI or API, you never need to switch languages or tools - everything stays in the C# ecosystem you know and love.



Flexible Development: Choose between visual modeling for rapid results or code-first development for full control and type-safety.



No Runtime Lock-in: Bloqs is distributed as a simple NuGet package, making it easy to integrate and deploy anywhere.

TWO WAYS TO MODEL

In-App (Runtime) ⚡
Model live in the browser—
fast, visual, and intuitive.

In Visual Studio 🛠️
Model with full compile-time
safety and code quality.



VISUAL DEVELOPMENT (IN-BROWSER MODELING)

In-browser visual development for rapid prototyping with live modeling, real-time style editing, and Scratch-like logic design, allowing instant feedback and seamless export to maintainable C# production code.



Home



Cards



Manage



Charts



Access control



default

Activity Cards

DISCARD CHANGES

SAVE CHANGES

B Box



C Card



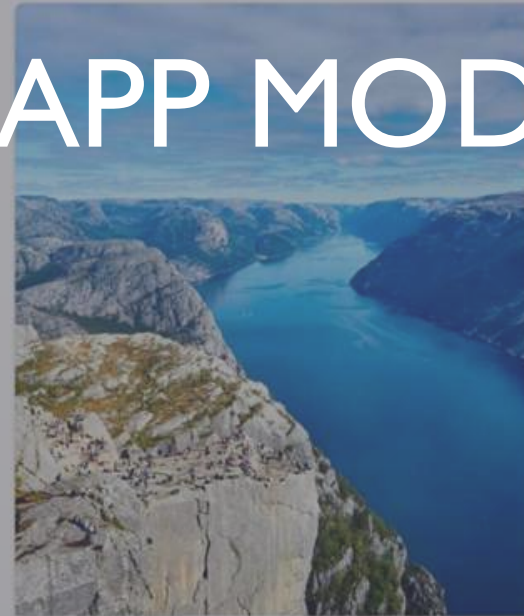
C Card



C Card



IN-APP MODELING



B Box



B Box



B Box





Logic block

Name*

AddActivityComment

Kind*

Blockly

- Logic
- Loops
- Math
- Text
- Lists
- Variables
- Functions
- Messaging
- User Interface
- Common
- Data
- User



IN-APP LOGIC BLOCKS,
SCRATCH-LIKE PROGRAMMING



Activity Cards

EDIT PAGE



Edit

Title*

Activity Tracker

Primary Color*

#1fc198



MODIFY THE APPLICATION'S STYLE
DIRECTLY FROM WITHIN THE APP

CANCEL

✓ UPDATE

CODE-FIRST DEVELOPMENT (VISUAL STUDIO)

A professional development setup with full compile-time type safety, IntelliSense, advanced refactoring, debugging, and profiling, using clean architecture patterns, modular feature-based design, dependency injection, and seamless Blazor WASM/API integration.

```

var builder = new AppBuilder(AppConstants.App.Name, "Activity Tracker", "#1fc198", "AA");
builder
    .AddEntityStorage(globalStorage)
    .AddEntityStorage(defaultStorage)

    .AddEntityClass<Organization>(globalStorage)
    .AddSimpleCrudForClass<Organization>(accessControl)
    .AddBasicAccessControl<UserAccount, OrganizationTenant>(globalStorage, accessControl)

    .AddEntityClass<Activity>(defaultStorage)
    .AddEntityClass<Category>(defaultStorage)

    .AddLogicBlock(new LogicBlock() { .. })
    .AddLogicBlock(new LogicBlock() { .. })
    .AddLogicBlock(new LogicBlock() { .. })
    .AddPage(new Page()
    {
        Id = AppConstants.Pages.HomePage,
        Title = "Home",
        Name = "home",
        PageArea = new BoxComponent()
        {
            Items = [
                new BoxItem() {
                    Component = new HtmlComponent() {

```

MODELING USING VISUAL STUDIO, DIRECTLY IN C#

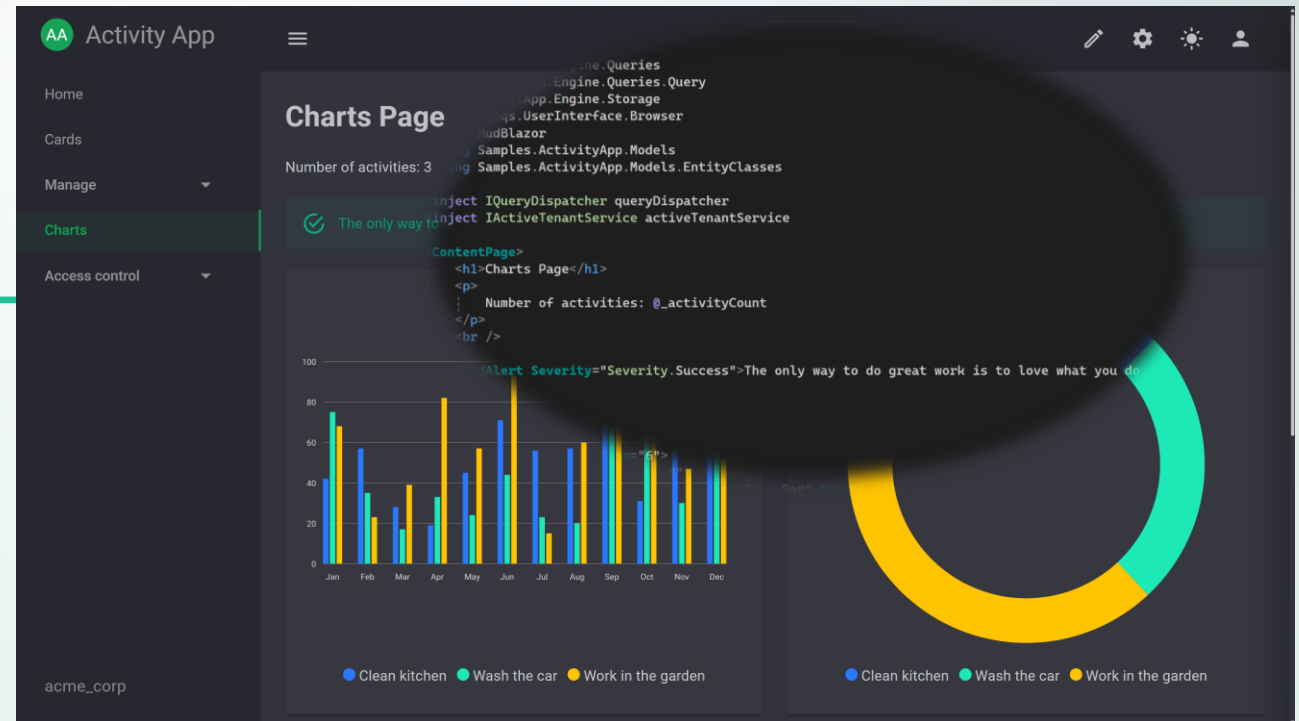
PROGRAMMING LOGIC USING COMMAND & QUERY DISPATCHER PATTERN (CQS)

- Feature-based architecture.
- Works with Blazor WASM & API.
- Leverages powerful Dependency Injection.

```
[EntityCommandHandler<Activity>]
0 references
public class NewActivity : ICommandHandler<NewEntityCommand, NewEntityCommandResult>
{
    0 references
    public Task<NewEntityCommandResult> HandleAsync(
        NewEntityCommand command,
        CancellationToken cancellationToken = default
    )
    {
        var newActivity = new Activity()
        {
            Name = "New Activity",
            Description = "New Activity Description",
        };

        return Task.FromResult(NewEntityCommandResult.CreateSuccess(newActivity));
    }
}
```


ADD CUSTOM COMPONENTS, PAGES AND API ENDPOINTS WITH EASE





BUILT-IN SECURITY

Bloqs provides robust security with OAuth authentication, JWT-based access, role management, and secure API communication, ensuring safe, maintainable, and scalable applications.

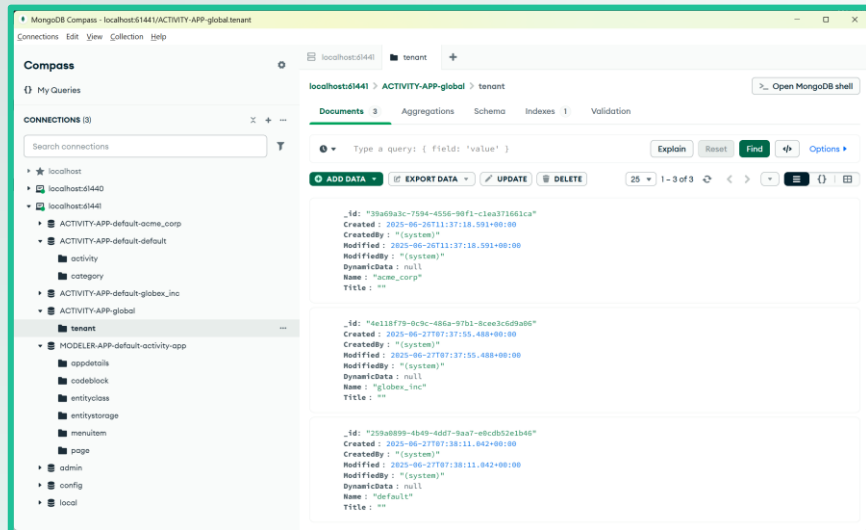
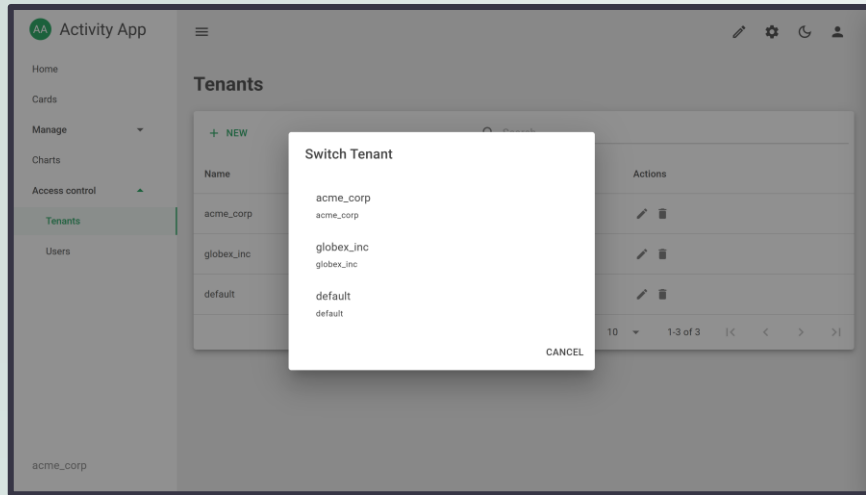
ACCESS CONTROL

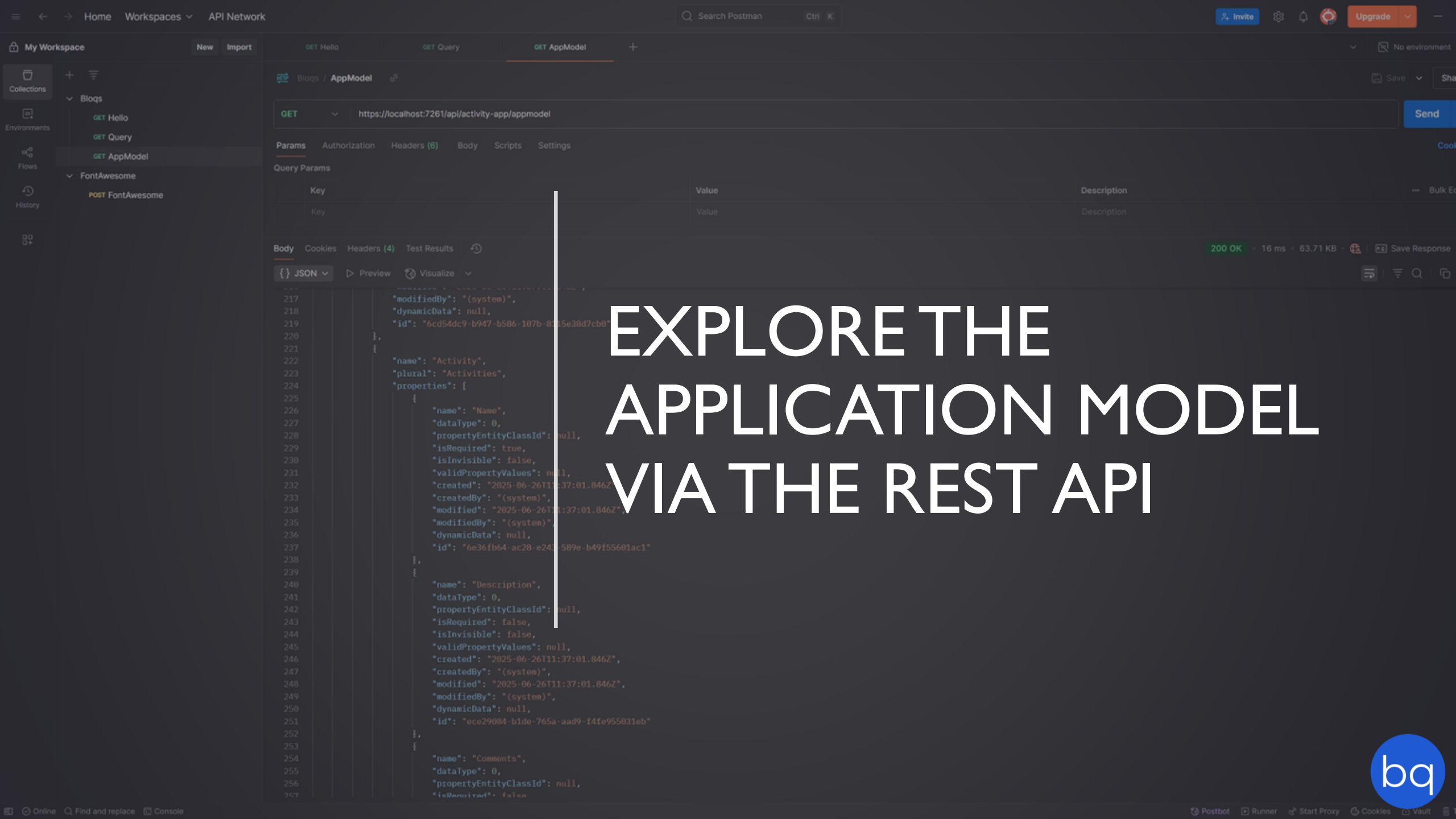
Authentication using JWT & OIDC (OpenID Connect).

Multi-tenancy support.

Users, permissions, roles.

Automatic database creation.





EXPLORE THE APPLICATION MODEL VIA THE REST API

```
217     "modifiedBy": "(system)",
218     "dynamicData": null,
219     "id": "6cd54dc9-b947-b586-107b-8f1a5e38d7cb0"
220   },
221   {
222     "name": "Activity",
223     "plural": "Activities",
224     "properties": [
225       {
226         "name": "Name",
227         "dataType": 0,
228         "propertyEntityClassId": null,
229         "isRequired": true,
230         "isInvisible": false,
231         "validPropertyValues": null,
232         "created": "2025-06-26T11:37:01.846Z",
233         "createdBy": "(system)",
234         "modified": "2025-06-26T11:37:01.846Z",
235         "modifiedBy": "(system)",
236         "dynamicData": null,
237         "id": "6e36fb64-ac28-e243-589e-b49f55601ac1"
238       },
239       {
240         "name": "Description",
241         "dataType": 0,
242         "propertyEntityClassId": null,
243         "isRequired": false,
244         "isInvisible": false,
245         "validPropertyValues": null,
246         "created": "2025-06-26T11:37:01.846Z",
247         "createdBy": "(system)",
248         "modified": "2025-06-26T11:37:01.846Z",
249         "modifiedBy": "(system)",
250         "dynamicData": null,
251         "id": "ece29084-b1de-765a-aad9-f4fe955031eb"
252       }
253     ],
254     "name": "Comments",
255     "dataType": 0,
256     "propertyEntityClassId": null,
257     "isRequired": false
```



DEFINE MULTIPLE DATA STORES ON GLOBAL OR TENANT LEVEL

The screenshot shows the 'Entity storage' configuration page in the 'Activity Tracker' application. The left sidebar contains a navigation menu with items: Home, Cards, Manage (highlighted with a green arrow), Activities, Categories, Charts, and Access control. The main content area is titled 'Entity storage' and features a form with two fields: 'Name*' with the value 'global' and 'Entity Storage Type*' with a dropdown menu. The dropdown menu is open, showing two options: 'Global' (highlighted in green) and 'Tenant'. The bottom of the sidebar shows the word 'default'.

Activity Tracker

Home

Cards

Manage

Activities

Categories

Charts

Access control

default

Entity storage

Name*

global

Entity Storage Type*

Global

Global

Tenant

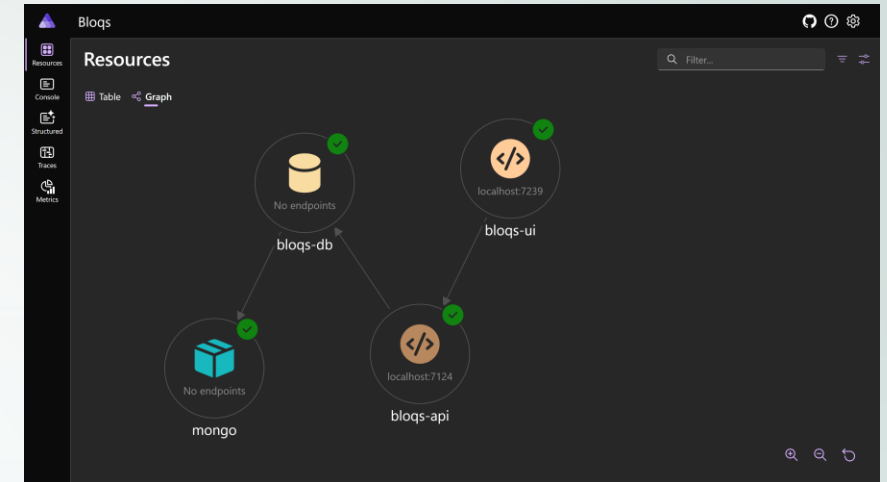


EXPOSE A ROBUST REST API, PROTECTED WITH JWT- AUTHENTICATION



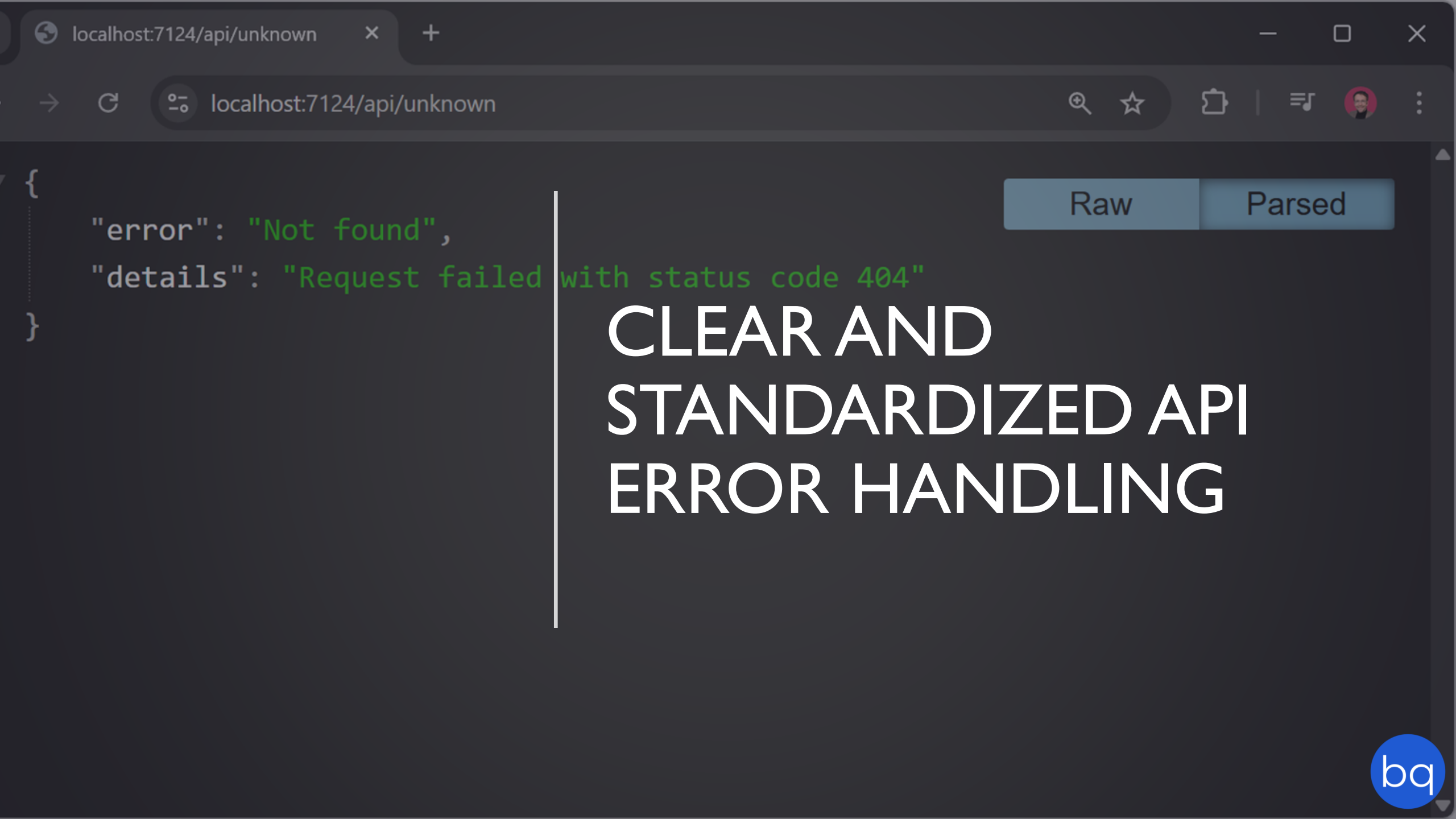
EFFORTLESS DEPLOYMENT WITH ASPIRE

Deployment with Aspire offers a streamlined, automated process for delivering applications quickly and reliably. It handles environment setup, builds, and releases with minimal configuration, ensuring consistent deployments across development, staging, and production while reducing errors and downtime.



The screenshot shows the 'Console logs' view in the Aspire IDE for the 'bloqs-api' resource. The logs display the following sequence of events:

```
1 Waiting for resource 'mongo' to enter the 'Running' state.
2 Waiting for resource 'bloqs-db' to enter the 'Running' state.
3 Waiting for resource 'mongo' to become healthy.
4 Waiting for resource ready to execute for 'bloqs-db'.
5 Finished waiting for resource 'bloqs-db'.
6 Waiting for resource ready to execute for 'mongo'.
7 Finished waiting for resource 'mongo'.
8 Info: Microsoft.Hosting.Lifetime[14]
9   Now listening on: https://localhost:61895
10 Info: Microsoft.Hosting.Lifetime[14]
11   Now listening on: http://localhost:61896
12 Info: Microsoft.Hosting.Lifetime[0]
13   Application started. Press Ctrl+C to shut down.
14 Info: Microsoft.Hosting.Lifetime[0]
15   Hosting environment: Development
16 Info: Microsoft.Hosting.Lifetime[0]
17   Content root path: C:\Estiom\git\Bloqs\Bloqs\Bloqs.Api
```



```
{
  "error": "Not found",
  "details": "Request failed with status code 404"
}
```

Raw Parsed

```
"error": "Not found",
"details": "Request failed with status code 404"
```

CLEAR AND STANDARDIZED API ERROR HANDLING

CURRENT STATUS



Closed Beta - Currently onboarding select teams.



Contact the Bloqs team to request early access and help shape the future of low-code in .NET.

