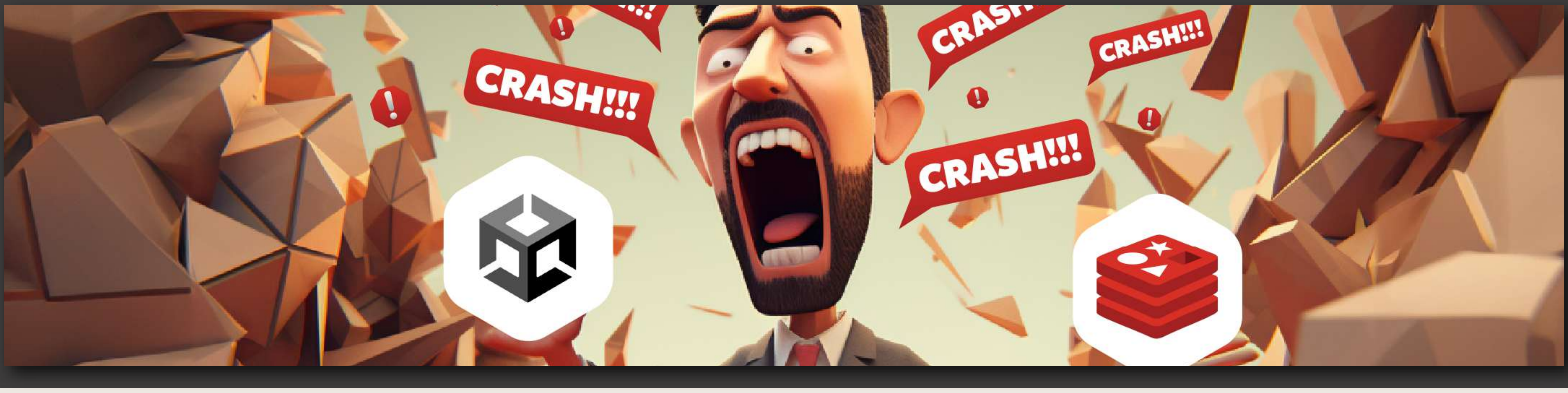


INTRODUCTION

Nesco - CrashReporter is a simple yet effective tool designed for Unity game developers. This tool is employed to swiftly detect errors occurring within Unity projects and to save detailed reports of these errors to a remote database. It facilitates the quick collection of errors when testing your game on independent devices or when releasing a demo, ensuring a prompt and efficient error-gathering process.

HOW DOES IT WORK

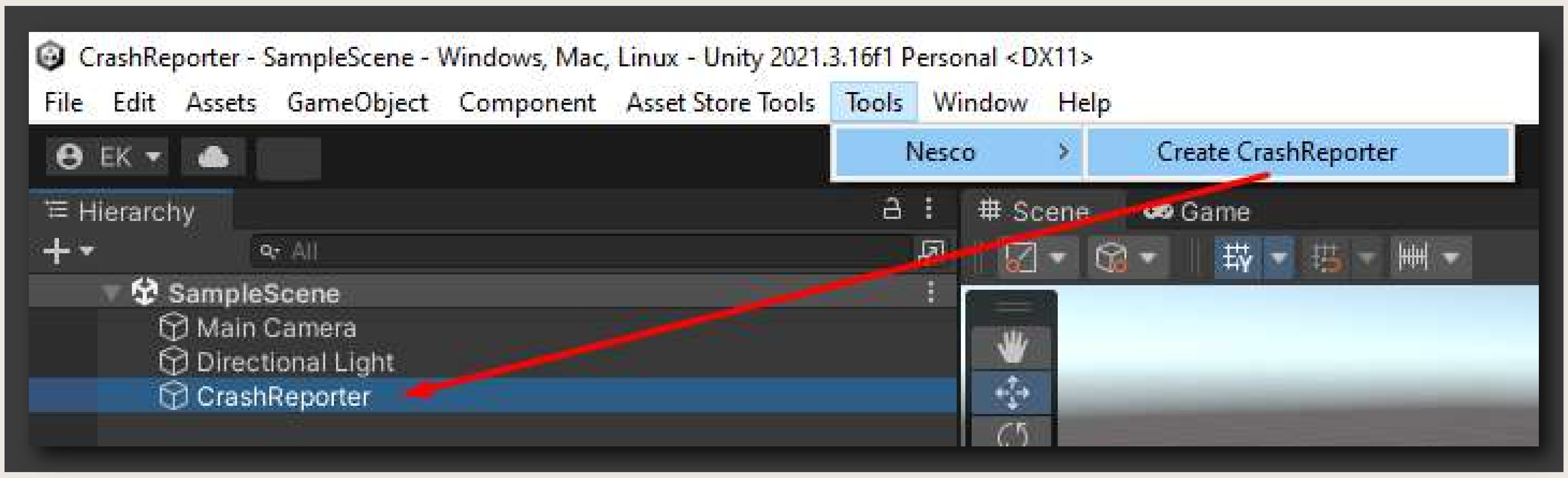
Nesco - CrashReporter automatically comes into play when an unhandled error occurs in your application. It swiftly gathers the details of the error, generates a comprehensive report, and, within milliseconds of the error's occurrence, transmits this information to a database via a specified REST URL, providing developers with immediate access to the incident.

SETUP

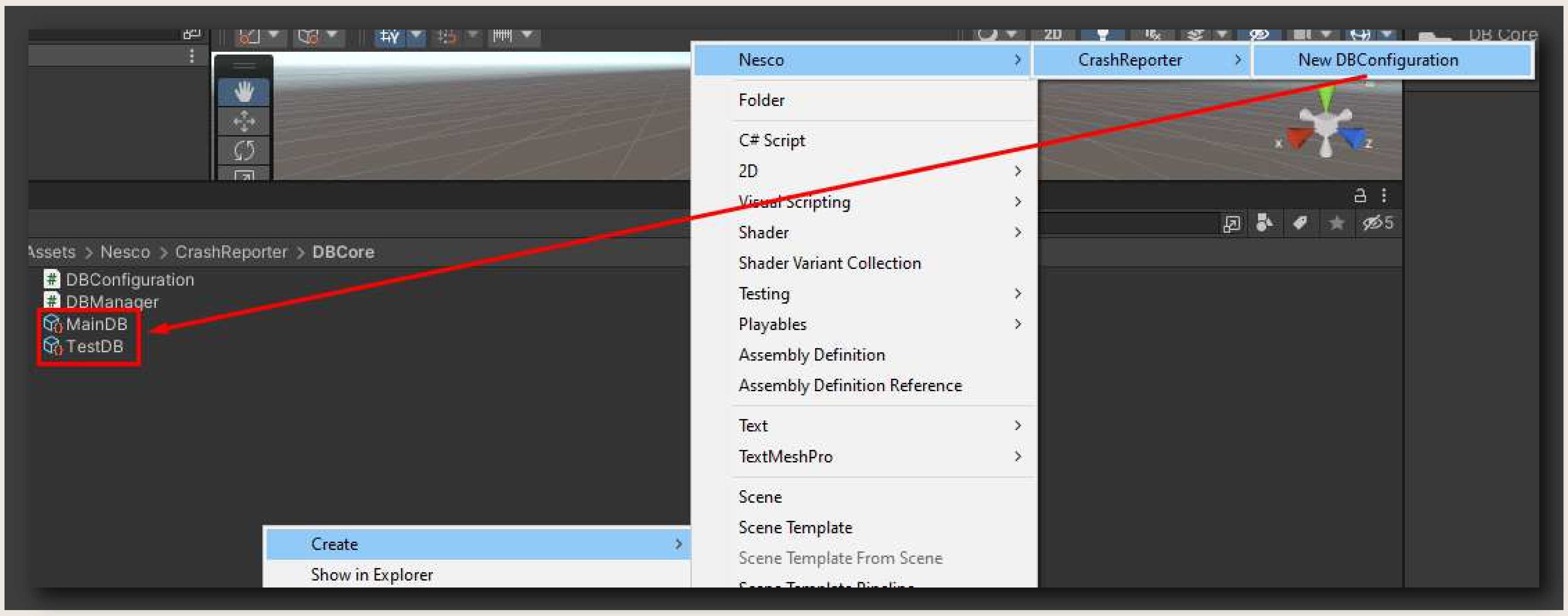
To be able to use Nesco - CrashReporter in Unity, you don't need to install an external package. However, since it writes crash reports to a remote database, the setup is a two-side process.

→ Unity Editor Side:

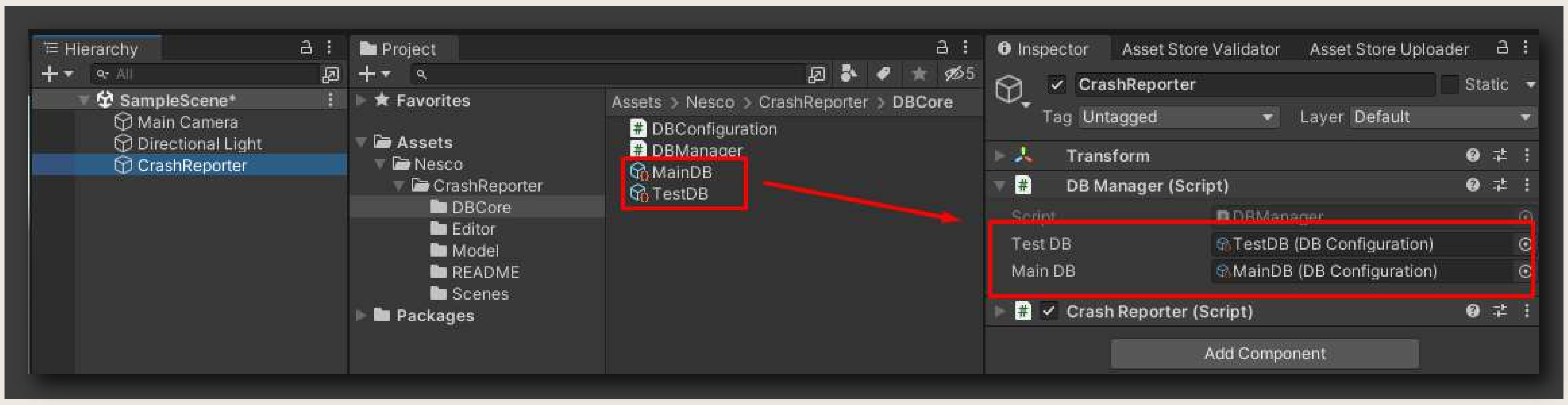
- Click on the "Tools/Nesco/Create CrashReporter" tab from the main menu above. (This process creates the necessary object and adds the required components.)



- Navigate to Assets/Nesco/CrashReporter/DBCore in the Project tab, and find the DBConfiguration scriptable object instances of MainDB and TestDB . (If you can't find them or want to create your own, go to the desired path, right-click, select Create/Nesco/CrashReporter/New DBConfiguration to create a new instance.)

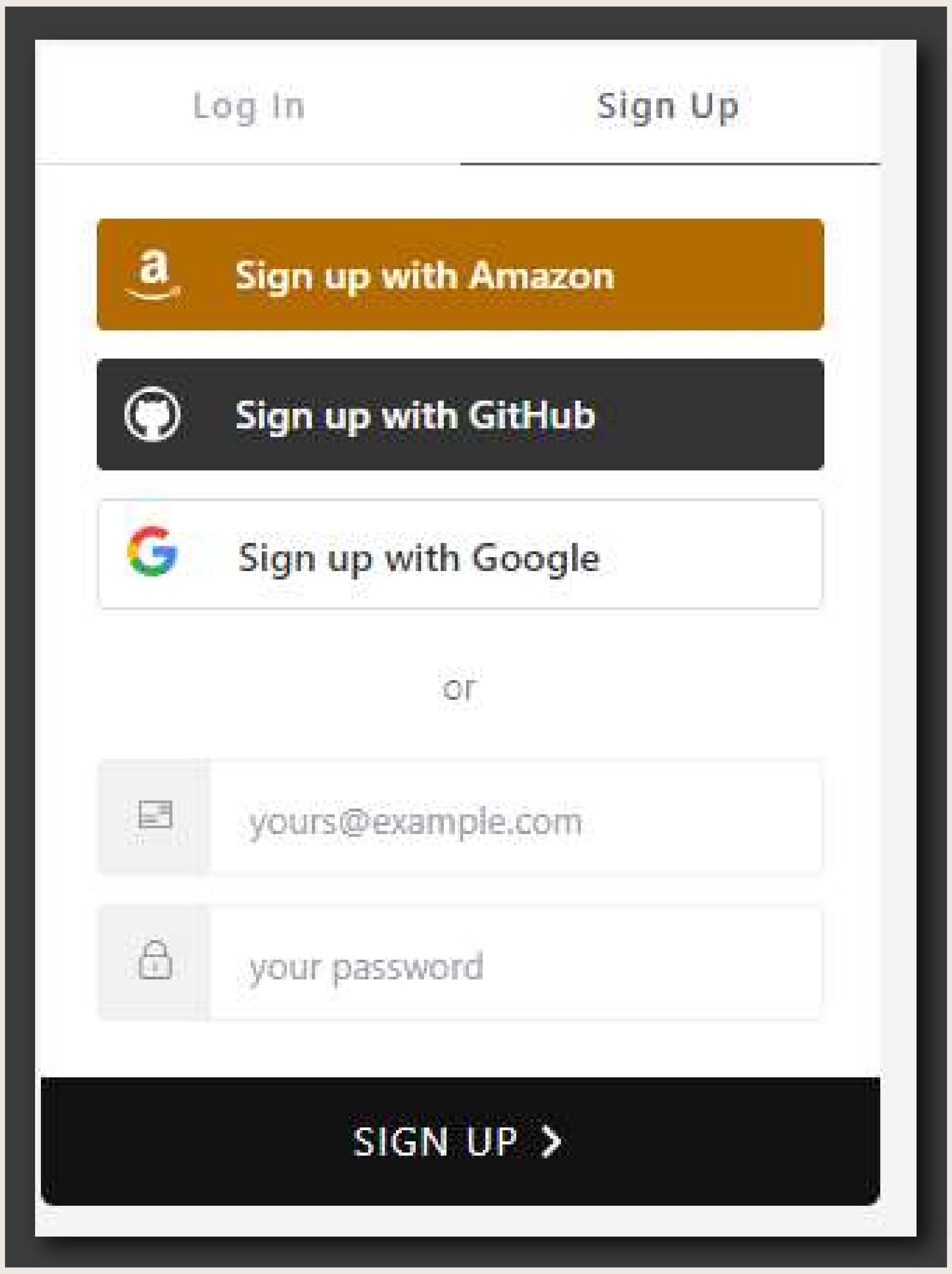


- Drag the DBConfiguration instance you created for testing into the TestDB field of the DBManager component inside the CrashReporter object in your scene. Repeat the same process for your Main DB configuration. (If you don't have a CrashReporter object in your scene, follow step 1.)

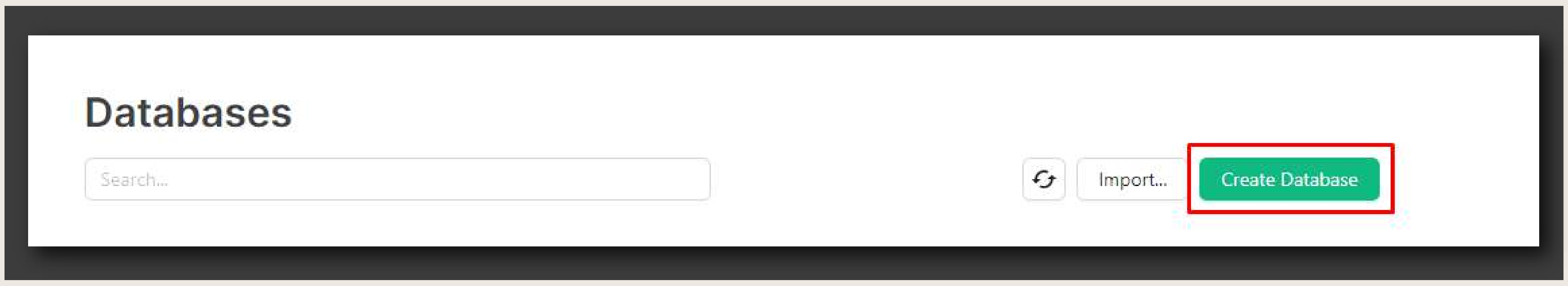


→ Database Side:

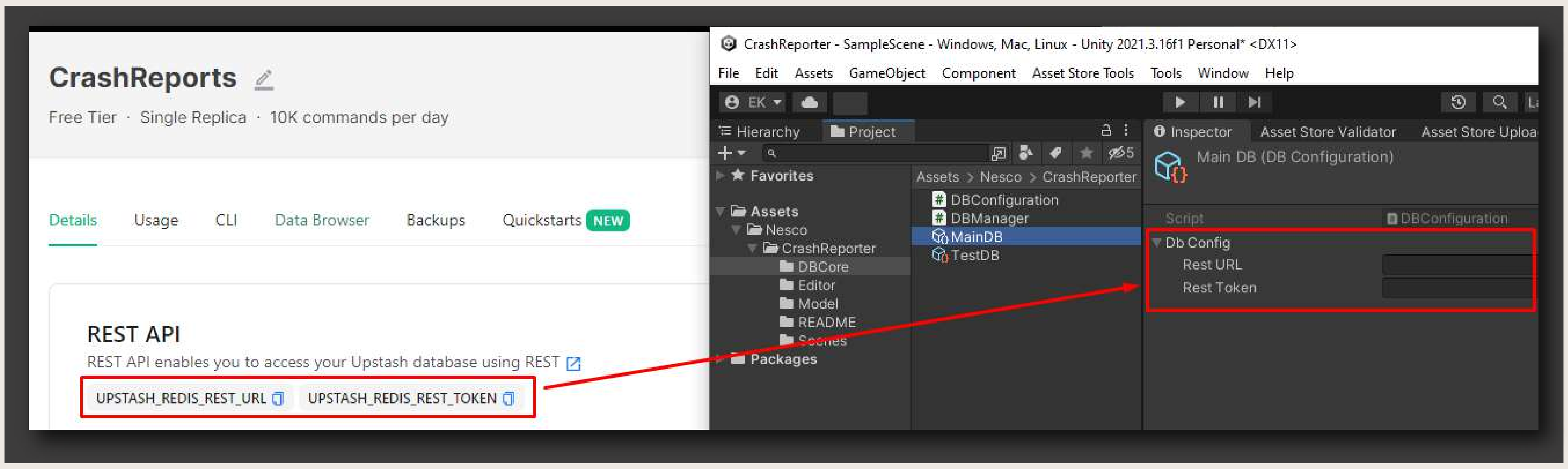
- Visit upstash.com and register using one of the various sign-up options provided. (Refer to the Why Use Upstash/Redis Section to understand why Upstash Redis is used in the project.)



- Create a database for yourself.



- Assign the REST_URL and REST_TOKEN values from the Details section of your database to the corresponding fields in the DBConfiguration instances you previously created in order to access your database from Unity.



WHY USE UPSTASH / REDIS

- Developer-friendly: For your applications in the testing and development stages, they offer a generously sufficient free tier plan, and if you wish to upgrade your database, there is a guaranteed fixed pricing limit (https://upstash.com/pricing)

- Easy access, use and setup

- Fast communication with Database

- Unlike other Redis databases, it provides durable storage capability. (https://upstash.com/docs/redis/features/durability)

CRASH REPORTER INCLUDES

- public string? PlayerID
- public string Message
- public LogType LogType
- public string StackTrace
- public int SceneIndex
- public string DateAndTime
- public RuntimePlatform? Platform
- public string DeviceModel
- public string OS
- public OperatingSystemFamily? OperatingSystemFamily
- public string DeviceName
- public DeviceType? DeviceType
- public string GraphicsDeviceName
- public GraphicsDeviceType? GraphicsDeviceType
- public string GraphicsDeviceVersion
- public int GraphicsMemorySize
- public string ProcessorType
- public int SystemMemorySize
- public int ProcessorCount
- public int ProcessorFrequency
- public bool? SupportsGyroscope
- public bool? SupportsInstancing
- public bool? SupportsLocationService
- public bool? SupportsRayTracing
- public bool? SupportsVibration