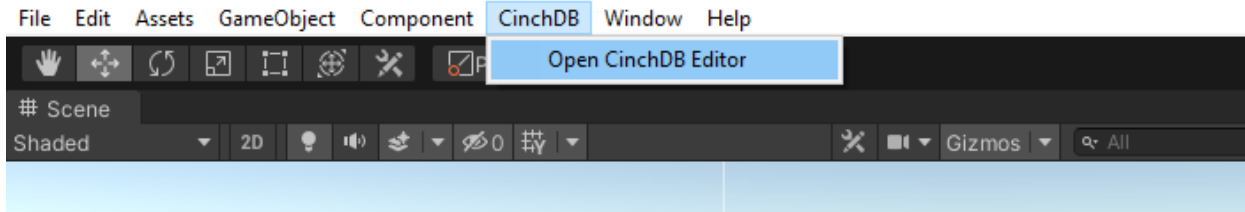


CinchDB Documentation

CinchDB is a lightweight database designed for storing non-sensitive content, such as leaderboard scores, promo-codes, user feedback, bug reports etc. It is quick to set up and integrate into a Unity project and is designed to help facilitate rapid prototyping.

1. Opening the Editor

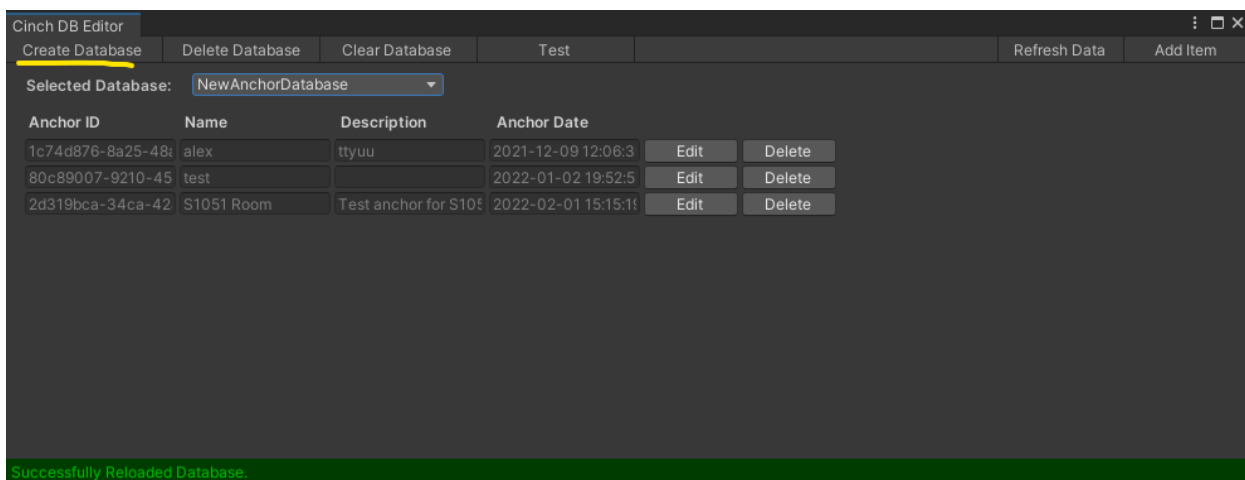
To open the editor, go to CinchDB > Open CinchDB Editor.



2. Creating a Database

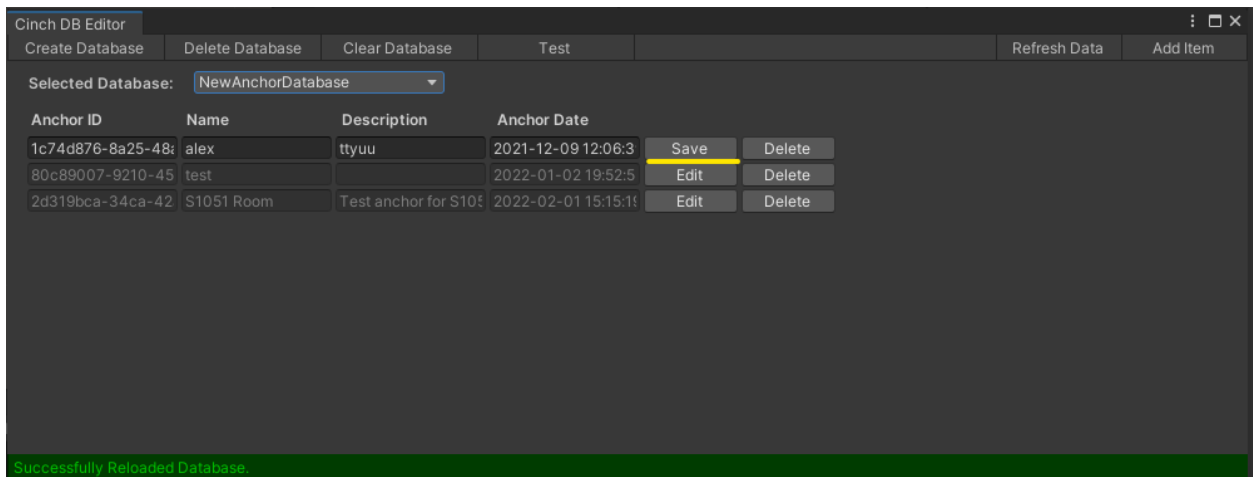
To create a database, open the editor and click on the 'Create Database' button. This will create a database and give it a unique string key. The database key is stored in a ScriptableObject. For the database to work, it must be contained in a folder labeled 'Resources'. By default, it should be created in the resources folder containing your existing databases, so you can move the CinchDB package in the hierarchy without any issues. If you want to copy an existing database into another Unity project, you can either copy the ScriptableObject into the new project.

*Note: By default, all new projects come with a single unique database already included. If you are copying the CinchDB folder from another project though, the database(s) will have already been assigned IDs and will therefore the databases in both projects will now link to the **SAME** database content. This may or may not be desirable, depending on your intended use case. If it is not, remember to create a new database and delete the database from the project (Clicking the 'Delete Database' button or removing the CinchDBDatabase ScriptableObject from your project doesn't actually affect the stored content, it just removes it from the current project).*



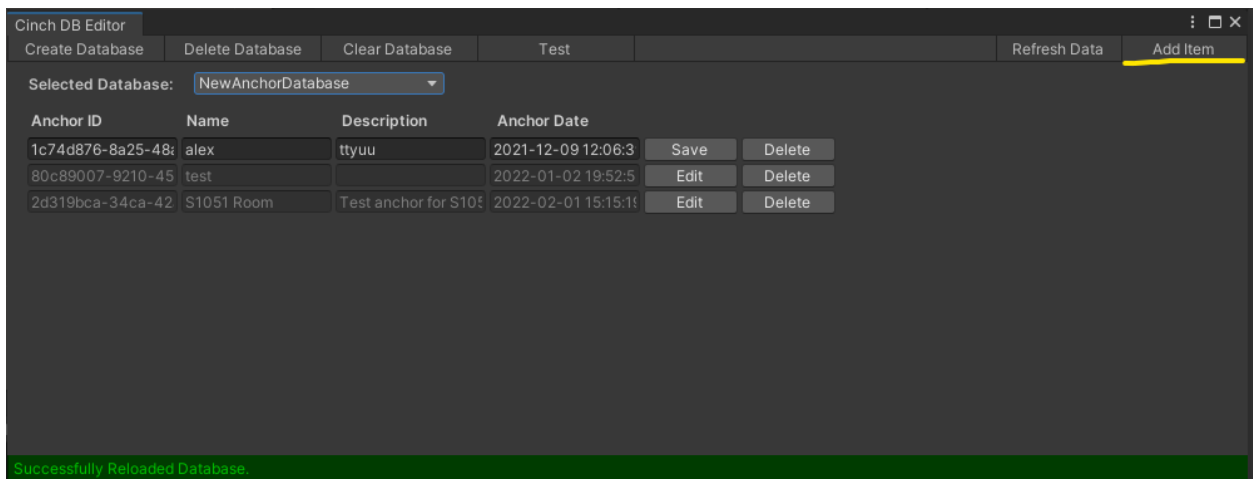
3. Editing the Database

To edit the content in a database, click on the ‘Edit’ button. This will allow the content for that record to be updated. When you have made the necessary changes, press the ‘Save’ button to save those changes.



4. Adding Records to the Database

To add a record to the database, click on the ‘Add Item’ button in the top-right corner of the window. This will give you an empty record. Add the data to the record and then press the ‘Save’ button to confirm the new record.



5. Retrieving Records via Code

You can add a record via code by using a reference to an existing CinchDB database. E.g.:

```
public CinchDBDatabase myDatabase;

private async void ProcessMyRecords()
{
    List<CinchDBRecord> records = await myDatabase.GetAllRecords();
    foreach (CinchDBRecord record in records)
    {
        Debug.Log (record.Columns[0]);
    }
}
```

5. Adding a Record via Code

You can add a record via code by using a reference to an existing CinchDB database. E.g.:

```
public CinchDBDatabase myDatabase;

void Start()
{
    myDatabase.AddRecord("Col 1 Data", "Col 2 Data", ...);
}
```

5. Editing a Record via Code

```
public CinchDBDatabase myDatabase;

private async void ProcessMyRecords()
{
    List<CinchDBRecord> records = await myDatabase.GetAllRecords();
    foreach (CinchDBRecord record in records)
    {
        // Do something.
        if (record.Columns[1] == "FOO")
        {
            record.Columns[0] = "BAR";
            await myDatabase.UpdateRecords(record);
        }
    }
}
```