# OneSqlite

OneSqlite is a database solution under Unity. Use open source Sqlite as the database.
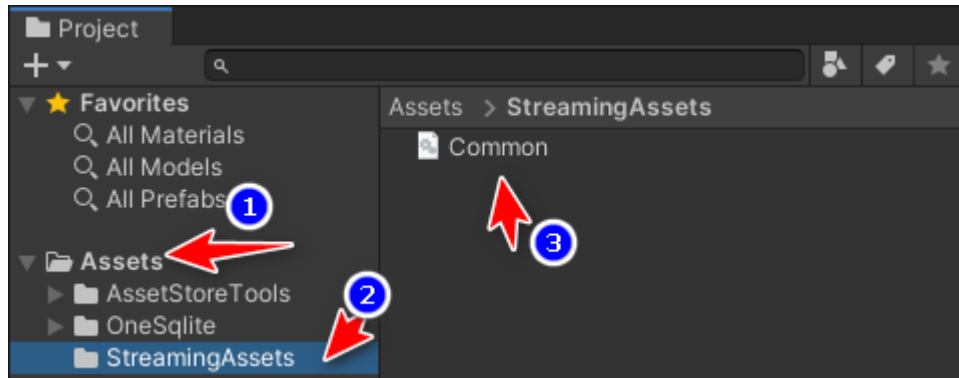
# Directory

# Features

- Built-in a function to change the PlayerPrefs that comes with Unity to the Key and Value form of database storage.
- The database supports int, long, float, bool, string and other types.
- Supports the creation of custom databases, custom tables, database additions, deletions, changes, and other functions.
- Cross-platform: currently supports Windows, Mac, Android, IOS.
- Support expansion and modification.
- Contains all source code and sample scenes.

# Install

The folder "Assets/OneSqlite" include all OneSqlite scripts. You can put OneSqlite anywhere in the Assets folder.
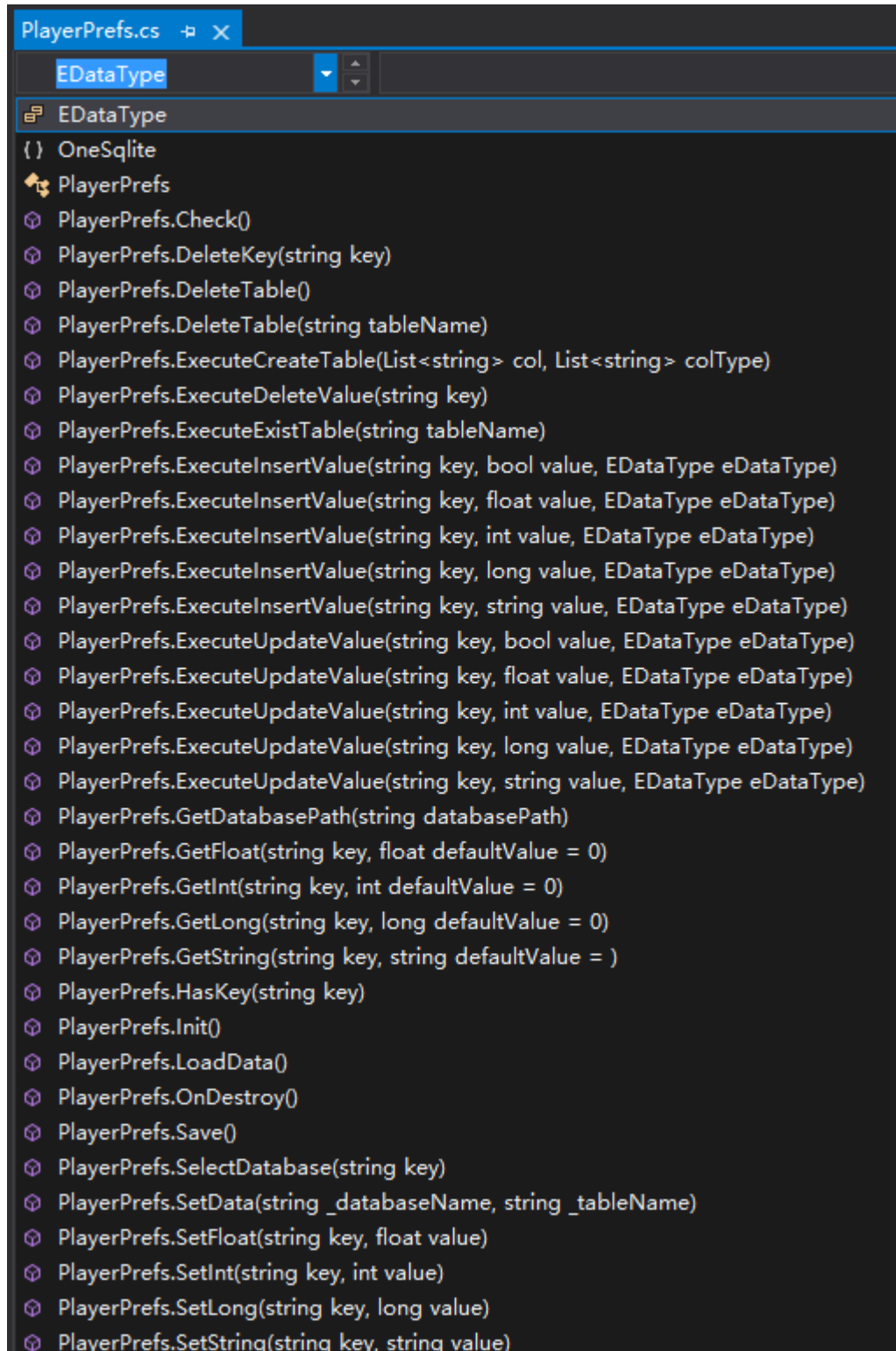
# How to use

You need to create "StreamingAssets" folder in then "Assets" folder. Then the database will create in the "StreamingAssets".

## PlayerPrefs

PlayerPrefs.cs : Built-in a function to change the PlayerPrefs that comes with Unity to the Key and Value form of database storage.

PlayerPrefs.cs  ⊹ ✕

EDataType ▾ ⬍

⬚ EDataType
{} OneSqlite
⚡ PlayerPrefs
⬡ PlayerPrefs.Check()
⬡ PlayerPrefs.DeleteKey(string key)
⬡ PlayerPrefs.DeleteTable()
⬡ PlayerPrefs.DeleteTable(string tableName)
⬡ PlayerPrefs.ExecuteCreateTable(List<string> col, List<string> colType)
⬡ PlayerPrefs.ExecuteDeleteValue(string key)
⬡ PlayerPrefs.ExecuteExistTable(string tableName)
⬡ PlayerPrefs.ExecuteInsertValue(string key, bool value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteInsertValue(string key, float value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteInsertValue(string key, int value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteInsertValue(string key, long value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteInsertValue(string key, string value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteUpdateValue(string key, bool value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteUpdateValue(string key, float value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteUpdateValue(string key, int value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteUpdateValue(string key, long value, EDataType eDataType)
⬡ PlayerPrefs.ExecuteUpdateValue(string key, string value, EDataType eDataType)
⬡ PlayerPrefs.GetDatabasePath(string databasePath)
⬡ PlayerPrefs.GetFloat(string key, float defaultValue = 0)
⬡ PlayerPrefs.GetInt(string key, int defaultValue = 0)
⬡ PlayerPrefs.GetLong(string key, long defaultValue = 0)
⬡ PlayerPrefs.GetString(string key, string defaultValue = )
⬡ PlayerPrefs.HasKey(string key)
⬡ PlayerPrefs.Init()
⬡ PlayerPrefs.LoadData()
⬡ PlayerPrefs.OnDestroy()
⬡ PlayerPrefs.Save()
⬡ PlayerPrefs.SelectDatabase(string key)
⬡ PlayerPrefs.SetData(string _databaseName, string _tableName)
⬡ PlayerPrefs.SetFloat(string key, float value)
⬡ PlayerPrefs.SetInt(string key, int value)
⬡ PlayerPrefs.SetLong(string key, long value)
⬡ PlayerPrefs.SetString(string key, string value)

```csharp
1 个引用
private void OnClickAdd()
{
    string username = Input_UserName.text;
    string password = Input_Password.text;

    if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))
    {
        Debug("username or password IsNullOrEmpty.");
        return;
    }

    PlayerPrefs.SetString(USERNAME, username);
    PlayerPrefs.SetString(PASSWORD, password);

    Debug(OneSqliteString.Format("username is {0}, password is {1}.", username, password));
}

1 个引用
private void OnClickDelete()
{
    PlayerPrefs.DeleteKey(USERNAME);
    PlayerPrefs.DeleteKey(PASSWORD);
    Debug("username and password is delete.");
}
```

Just use

PlayerPrefs.SetString(USERNAME, username);

PlayerPrefs.DeleteKey(USERNAME);

But before use the PlayerPrefs, you must init, like bellow.

```csharp
@Unity 消息|0 个引用
private void Awake()
{
    PlayerPrefs.Init();
}

@Unity 消息|0 个引用
private void OnDestroy()
{
    PlayerPrefs.OnDestroy();
}
```

## SqliteConnection

Use the SqliteConnection for connect the database.

## SqliteCommand

Use the SqliteCommand for query the database.

### SqliteDataReader

Use the SqliteDataReader for read from the database.

### Mono or IL2CPP

If you want to support the mono and IL2CPP ，must use the SqliteConnection、SqliteCommand、SqliteDataReader.

```
/// <summary>
/// SqliteConnection
/// </summary>
private SqliteConnection SqlConnection;
/// <summary>
/// SqliteCommand
/// </summary>
private SqliteCommand SqlCommand;
/// <summary>
/// SqliteDataReader
/// </summary>
private SqliteDataReader SqlDataReader;
```

# Contact

For more detail you can go to the web:    https://github.com/onelei/OneSqlite
Email: 936496193@qq.com