Welcome

Thank you for choosing Save System

For more up-to-date information, please check out the Official Documentation

Introduction

This package includes everything to help you manage your game or app data with lots of useful features, let's take a look at features:

- Cross platform
- Unified API
- Storage features such as Backup, Meta data, catalog, ...
- Full-featured JSON Serialization
- Security with encryption
- Third-party integrations such as PlayMaker, Firebase, PlayFab, ...
- So if you're a programmer head into the Getting Started as Programmer, otherwise if you're a designer or a beginner head into the Getting Started as Designer

Requirements

- Unity 2018.1+ or higher
- .NET 4.x or higher

Cross Platform

The package is tested with IL2CPP and Mono, and is also tested on Standalone, Mobile, WebGL and UWP, but it would work on all other platforms that Unity supports, just maybe some platforms doesn't support File operations or may have a limited file access, so you might need to adjust the settings to make it work with your own requirements, also you can use Cloud storage on platforms which don't have a file storage.

Learn more

Unified API

SaveSystemAPI class provides static methods for saving, loading, deleting, ... storage items and data which also allows you to use custom settings. Using these methods you can manage data on a specific storage:

- Save
- Load
- LoadInto
- Exists
- Delete
- Move
- Copy
- List
- ListAll
- Clear
- SaveCatalog
- LoadCatalog
- HasMetaData
- SaveMetaData
- LoadMetaData
- WriteAllText
- WriteAllBytes
- ReadAllText
- RealAllBytes
- SaveImagePNG
- SaveImageJPG

- SaveImageEXR
- SaveImageTGA
- LoadImage
- CreateBackup
- GetLatestBackup
- GetBackups
- RestoreLatestBackup
- RestoreBackup
- DeleteBackup
- DeleteBackups

Learn more

Storages

There are different types of storages built-in already, such as:

- File Storage: This is the traditional file storage implementation
- PlayerPrefs Storage: This is the Unity PlayerPrefs storage which is mostly uesd for saving game config and works on all platforms that unity supports

And some Third-party storages such as:

- Firebase Realtime Database Storage: This uses the Firebase Realtime Database service for saving and loading data
- Firebase Cloud Storage: This uses the Firebase Storage service for saving and loading the files through uploading files
- PlayFab Entity Objects Storage: This uses the new PlayFab entity objects storage for storing the data
- PlayFab User Data Storage: This uses the PlayFab UserData API for storing the data

You can also easily implement your own Storage by extending the StorageBase class. There are Storage factories included for instantiating different kind of storage using a connection string, this enables modularity of storages for the developer and user, you can also create your own storage factory.

Each storage can be initialized using a connection string, or through the programming interface with a new keyword, but here we prefer to use connection string for simplicity of use and flexibility.

Learn more

Serialization

This uses Bayat.Json as the underlying serialization library that is a modified version of Newtonsoft Json.Net, that means it includes all the great features from that popular library also includes a few new features and capabilities, plus a few adjustments to make it more functional with saving and loading data.

This allows you to serialize all kinds of data, from Unity objects to C# objects:

- Unity objects (GameObject, Materials, Component, MonoBehaviour, ...)
- C# objects (Unity C# objects such as Vector3, Vector2 and so on or custom objects or any c# built-in object ...)

Learn more

Security

Symmetric encryption algortihms are implemented in this package and AES is the built-in and default encryption option whcih allows you to secure your data from cheating and further modifications by user, but you can also implement your own encryption as always.

Learn more

Final Notes

As always you can modify every part of this package to your desire and requirements, it already includes the flexibility for implementing your own Storage and Encryption Algorithm, but you can modify everything or add your own things on top of it to make the most out of it.

Keep reading the documentation for more information and exciting stuff!