

Save Data: AES Encrypted

Table Of Contents

- StorageManager
 - **Introduction**
 - **Methods**
 - AESCryptographicSystem
 - **Introduction**
 - **Methods**
-

StorageManager

Introduction

StorageManager class is used for saving and retrieving data in plain and encrypted form. Use SetLocation() method at least once before starting to save and retrieve data.

Methods

SetLocation()

Set path for StorageManager where further operations will be done

```
public static void SetLocation(string storageLocation)
```

Parameters

storageLocation (string): Path for StorageManager where further operations will be done

Example:

- StorageManager.SetLocation(StorageManager.ExternalLocation)
 - StorageManager.SetLocation(StorageManager.StreamingAssetsLocation)
 - StorageManager.SetLocation(StorageManager.ProjectLocation)
-

CheckFileExist()

Check whether file exists or no.

```
public static bool CheckFileExist(string path)
```

Parameters

path (string): File path for checking it's existence

Write()

Save data to file

```
public static void Write(string file, string data)
```

Parameters

- file (string): Name of file
- data (string): Data to be stored

```
public static void Write(string file, byte[] data)
```

Parameters

- file (string): Name of file
 - data (byte[]): Data to be stored
-

EncryptAndWrite()

Encrypt and Save data to file

```
public static void EncryptAndWrite(string file, string data, string password = null, byte[] salt = null)
```

Parameters

- file (string): Name of file
 - data (string): Data to be stored
 - password (string): Password to be used to encrypt the data
 - salt (byte[]): Salt to be used while encrypting the data
-

```
public static void EncryptAndWrite(string file, byte[] data, string password = null, byte[] salt = null)
```

Parameters

- file (string): Name of file
 - data (byte[]): Data to be stored
 - password (string): Password to be used to encrypt the data
 - salt (byte[]): Salt to be used while encrypting the data
-

ReadNow()

```
public static string ReadNow(string file, string defaultData = null)
```

Parameters

- file (string): Name of file to be read
 - defaultData (string): Default data to be return when given file is not found.
-

ReadEncryptedNow()

```
public static string ReadEncryptedNow(string file, string defaultData = null, string password = null, byte[] salt = null)
```

Parameters

- file(string): Name of file to be read
 - defaultData (string): Default data to be return when given file is not found.
 - password (string): Password used to decrypt the data
 - salt (byte[]): Salt used to decrypt the data
-

ReadBytesNow()

```
public static byte[] ReadBytesNow(string file)
```

Parameters

- file (string): Name of file to be read

ReadEncryptedBytesNow()

```
public static byte[] ReadEncryptedBytesNow(string file, string password = null, byte[] salt = null)
```

Parameters

- file(string): Name of file to be read
- password (string): Password used to decrypt the data
- salt (byte[]): Salt used to decrypt the data

EraseEverything()

```
public static void EraseEverything()
```

Clear all saved data inside currently selected location

Delete()

```
public static bool Delete(string filePath)
```

Deletes file at specified location if it exists

Parameters

- filePath (string): Name [with path] of file to be deleted

ObjectToByteArray()

```
public static byte[] ObjectToByteArray(Object obj)
```

Convert Object's instance to Byte array

Parameters

obj (Object) : Instance of Object class to be converted to Byte array

ByteArrayToObject()

```
public static Object ByteArrayToObject(byte[] arrBytes)
```

Convert Byte array to Object Instance

Parameters

arrBytes (byte[]) : Array bytes to be converted to Object class's Instance

AESCryptographicSystem

Introduction

AESCryptographicSystem class is used for encrypting and decrypting data

Methods

Encrypt()

Encrypts input data with AES and returns it

```
public static byte[] Encrypt(byte[] input, string password = null, byte[] salt = null)
```

Parameters

- input (byte[]): Byte array to be encrypted.
- password (string): Password for encryption. If not set [NOT RECOMMENDED], default 20 chars Password will be used.
- salt (byte[]): Salt used for encryption. If not set [NOT RECOMMENDED], default 8 byte SALT will be used.

```
public static string Encrypt(string input, string password = null, byte[] salt = null)
```

Parameters

- input (string): Text to be encrypted.

- password (string): Password for encryption. If not set [NOT RECOMMENDED], default 20 chars Password will be used.
 - salt (byte[]): Salt used for encryption. If not set [NOT RECOMMENDED], default 8 byte SALT will be used.
-

Decrypt()

Decrypts input data and returns it

```
public static byte[] Decrypt(byte[] input, string password = null, byte[] salt = null)
```

Parameters

- input (byte[]): Byte array to be decrypted.
- password (string): Password for decryption. If not set [NOT RECOMMENDED], default 20 chars Password will be used.
- salt (byte[]): Salt for decryption. If not set [NOT RECOMMENDED], default 8 byte SALT will be used.

```
public static string Decrypt(string input, string password = null, byte[] salt = null)
```

- input (string): Text to be decrypted.
- password (string): Password decryption. If not set [NOT RECOMMENDED], default 20 chars Password will be used.
- salt (byte[]): Salt for decryption. If not set [NOT RECOMMENDED], default 8 byte SALT will be used.