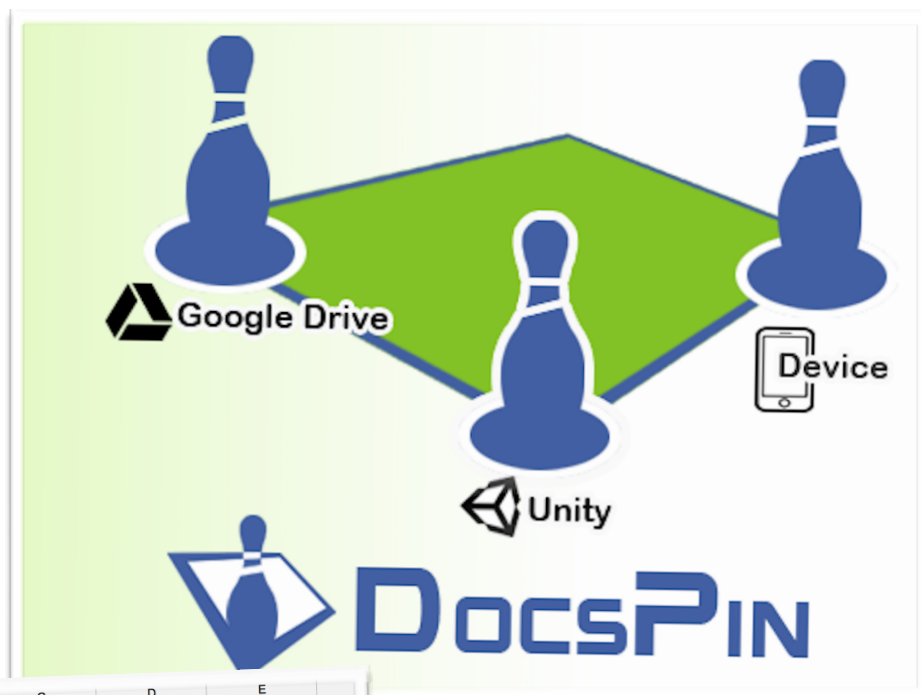
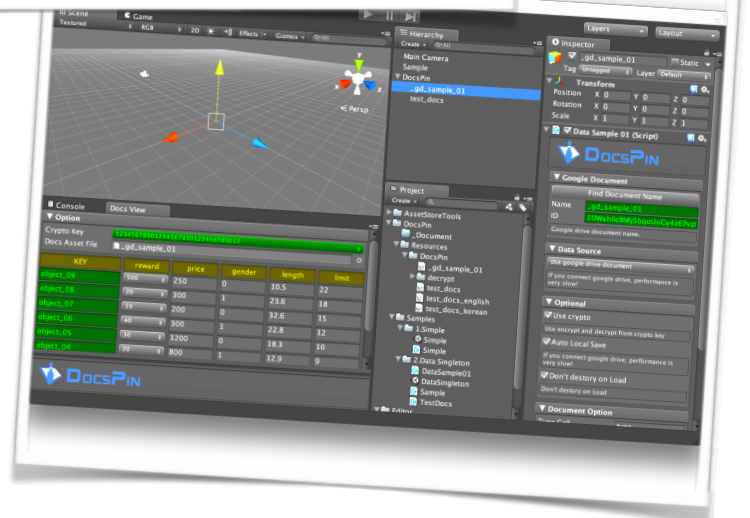


# DocsPin

## Unity Localize Script Service



A	B	C	D	E	
	int	int	int	list	double
	gender	price	limit	reward	length
sct_01	0	100	3	10,100,200	
sct_02	1	200	4	10,100,200	
sct_03	1	200	4	10,100,200	
sct_04	1	200	4	10,100,200	
sct_05	1	200	4	10,100,200	
sct_06	1	200	4	10,100,200	
sct_07	1	200	4	10,100,200	
sct_08	1	200	4	10,100,200	
sct_09	1	200	4	10,100,200	
sct_10	1	200	4	10,100,200	
sct_11	1	200	4	10,100,200	



# DocsPin

## Unity Localize Script Service

### Introduction

Data for text and balancing is used within an application or game. DocsPin is designed to easily apply these text and balancing data in Unity. Text and balancing data is created in Google Drive spreadsheet and it can be easily used in Unity with some simple settings. Efficient development process can be achieved by separating text and balancing data from other development.

Many will consider the following questions when developing an application or game.

- **How can we reduce hard coding? (text and balancing data)**
- **How can we easily apply multiple languages?**
- **How can we increase efficiency between engineers and designers?**

DocsPin is a solution to solve all of the issues above. DocsPin will increase resource efficiency.

### Preparation

DocsPin uses Google Drive spreadsheet. Google Drive is used as an online document editing tool. It is especially used when there are multiple members participating in a project and also for storing and managing text and balancing data. The following steps are required prior to using DocsPin.

- Google account (existing accounts can be used)
- Google Drive manual (reference: <https://www.google.co.kr/intl/en/drive/index.html>)
- Google Drive spreadsheet (similar to MS Excel)

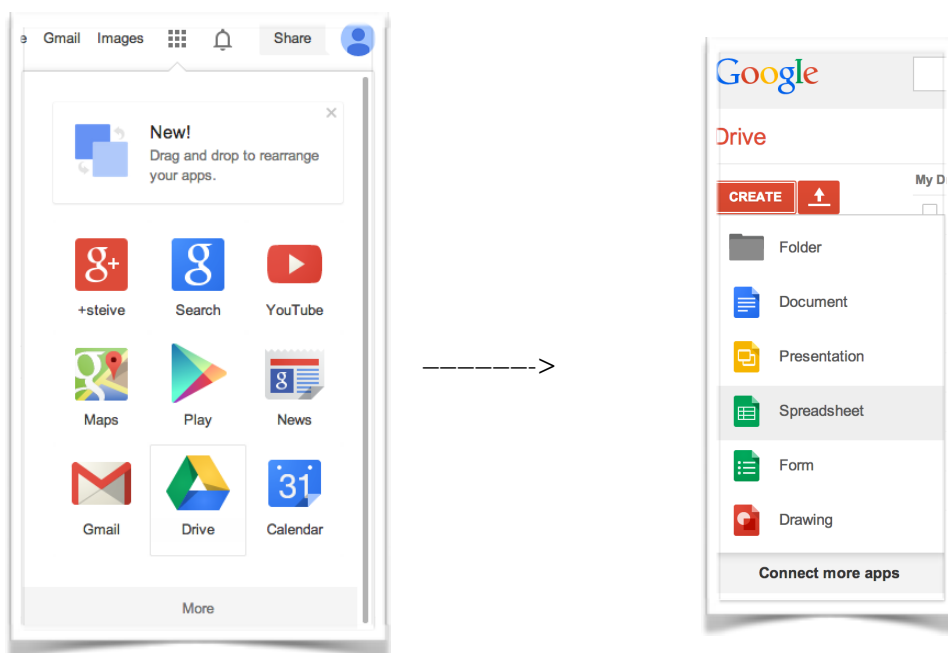
## Functions of DocsPin

Following are the features of DocsPin.

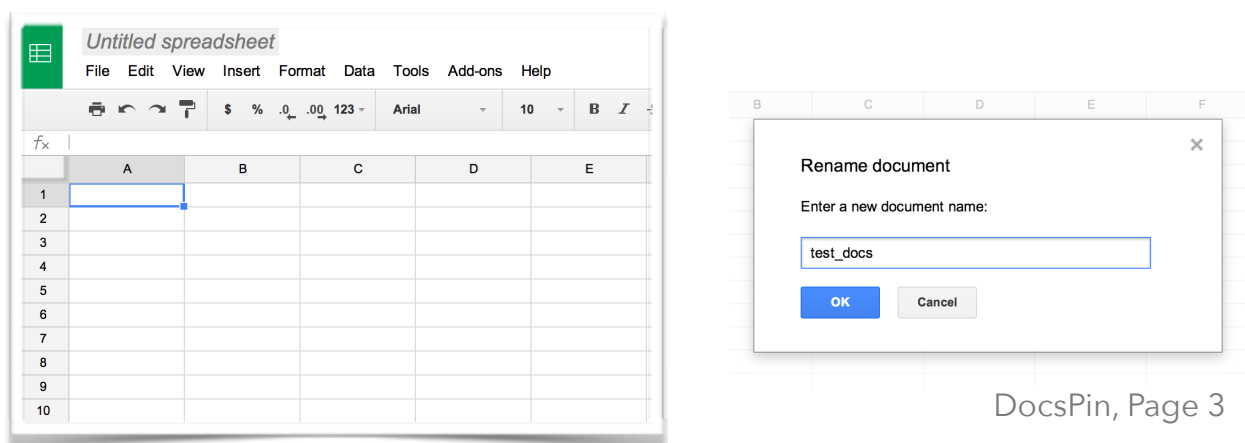
- Retrieve data from Google Drive
- Save and upload in local file all of Google Drive data
- Protect and encode data
- Manage multi-language text data

## Google Drive

1. Create Google account or login.
2. Create Spreadsheet in Google Drive.



3. Create document name.



#### 4. Create "type".

	A	B	C	D	E
1	type	int	string	double	list
2					

- First row is used to set "type". It is data type used in the program.
- DocsPin supports 4 types.
  - int : int data format
  - string : string data format
  - double : double data format
  - list : ArrayList data format in Unity must be inputted as "," unit.

#### 5. Set Column name.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3					

- Second row is used to set name of column.
- Column name is used to retrieve data from program.
- Duplicate column names in one spreadsheet is not allowed.

#### 6. Input data.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					

- Data must be inputted as defined in type.
- Duplicate IDs are not accepted. They must be unique. (i.e. object\_01, object\_02, etc)
- ID and column names are used to retrieve data from program.

## 7. Multi-language Service (only if required)

Below steps are required if data management is required in multiple languages.

- Create new sheet.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					
6					

\* Select "+" to create.

	A	B	C	D
1				
2				
3				
4				
5				
6				

- Change name of sheet.

	A	B	C	D
1				
2				
3				
4				
5				
6				

\* Use language name by Unity

\* **Application.systemLanguage**

\* Reference : <http://docs.unity3d.com/ScriptReference/SystemLanguage.html>

\* English, Chinese, Korean, German, Japanese etc

- Input data as described in steps 4-6.

	A	B	C	D	E
1	type	string	string		
2	id	title	content		
3	object_01	KNIFE	A knife (plural knives) is a cutting tool with a cutting edge or blade, hand-held or otherwise, with		
4	object_02	GUN	A gun is a normally tubular weapon or other device designed to discharge projectiles or other m		
5					
6					

data English

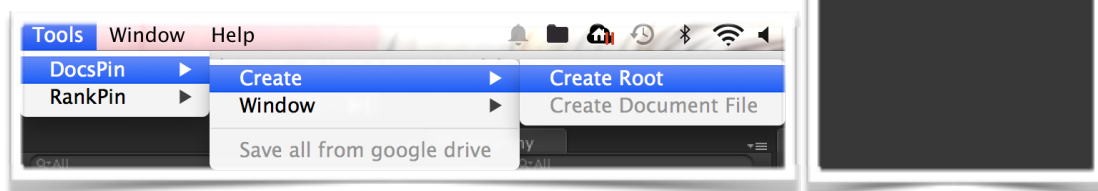
- Repeat steps to create new sheet for additional languages.

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	칼	칼은 날카로운 면을 가진 도구로 어떤 것을 자를 때 쓴다
4	object_02	총	총(銃)은 일반적으로 개인이 휴대할 수 있는 구경이 작은 화기를 말한다.
5			
6			

data English Korean

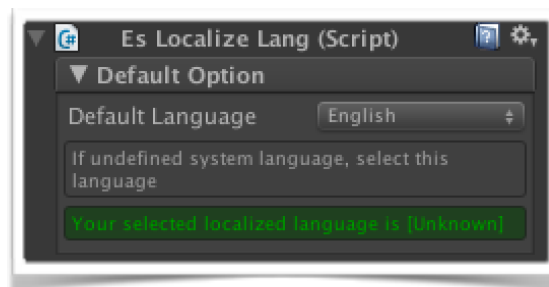
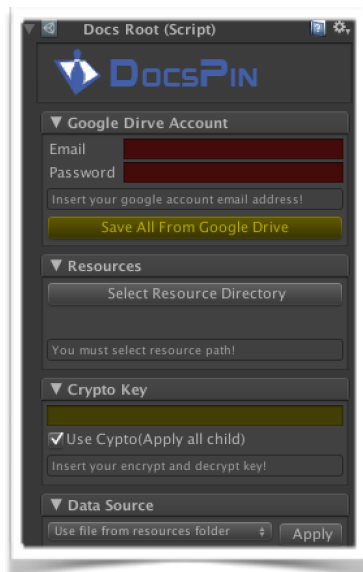
## Unity

1. Select Create Root option from DocsPin's menu.



- GameObject for DocsPin and file is created when Create Root is selected.

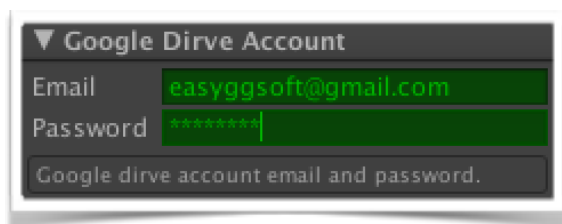
## 2. Select DocsPin GameObject from Hierarchy.



- DocsPin Root has two Script.
- Localize Lang Script is the Script used to set default language. Device language setting is used for DocsPin's default language. If device language is set to Korean, default will be Korean but this is an option used if Korean is not supported by Google Drive. Default language set above must be provided in Google Drive.

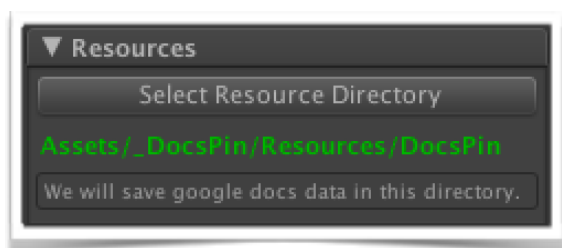
- Docs Root Script is the setting for Google account information setting and Google Drive spreadsheet data management.

## 3. Input Google Drive account in inspector window.



- Input email and password of Google account.
- Retrieve spreadsheet from account.

## 4. Set resources folder.

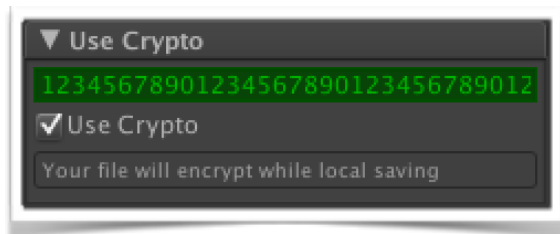


- Set resources folder in Unity.
- DocsPin saves Google Drive data in local and uses the saved data in program.
- Google Drive data can be retrieved without being saved, but this method (saving data in Resources folder) is recommended to reduce

time.

- Resources folder must be set as it is used when exporting data files in iOS and Android.

#### 5. Set encoding key (skip if not required).



- File can be encoded as DocsPin saves files locally.

- To encode, set a 32 byte encoding key.

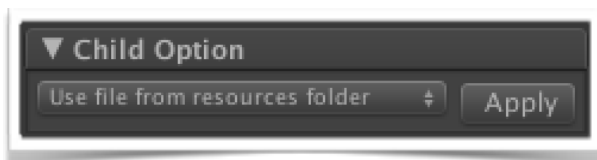
- Skip if not required.

- Use Crypto option is used to apply to all

GameObject below. Individual GameObject can be set and this feature allows all below GameObject to be turned On or Off.

- Encoding key is saved in Inspector and it is recommended to be noted in a separate place as it can be lost if an Unity error occurs.
- Program encrypt key : 12345678901234567890123456789012

#### 6. Set other options.

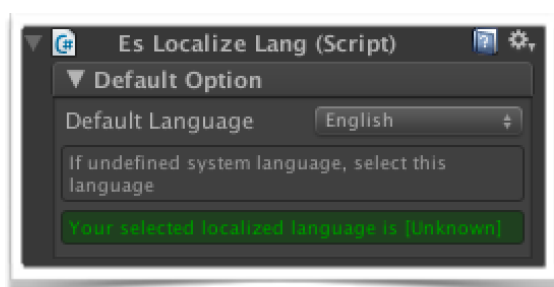


- This option applies to all GameObject below.

- Use file from resources folder is a method to use saved local files.

- Use Google Drive document is a method to use data by connecting to Google Drive. This method connects to Google Drive everytime and has the advantage of applying adjusted data in realtime but it is slow.

#### 7. Set localize information.

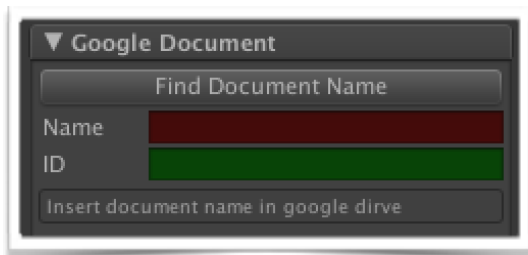


- Set default language.

- Setting must be equivalent to name on Google Drive spreadsheet.

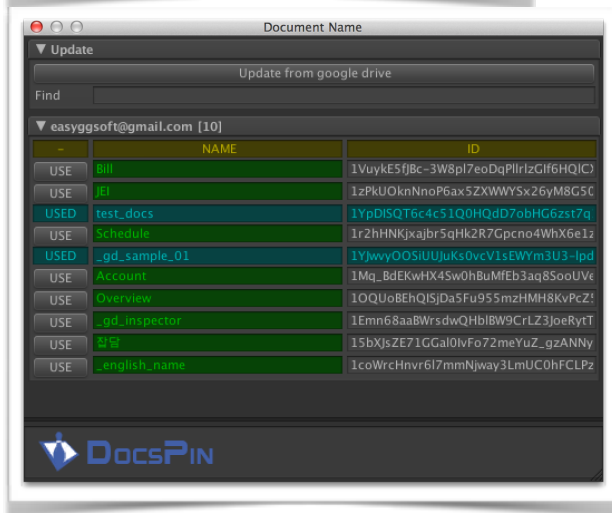


## 8. Select GameObject of DocsPin GameObject.



- Set Document Name and ID.

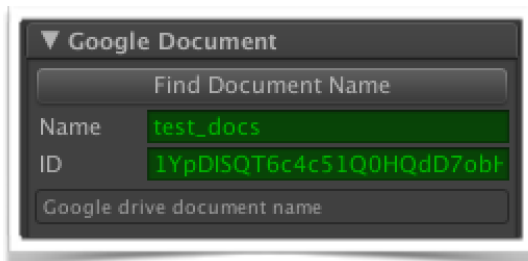
- This menu will appear when Find Document Name is selected.



- Contents of Google Drive spreadsheet will be retrieved when Update from Google Drive is selected.

- Select USE to choose spreadsheet.

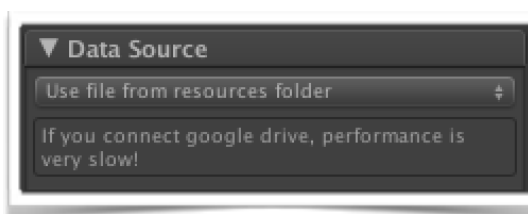
- Name and ID will be set when selected.



- GameObject can be found in program with Name and ID.

- `DocsPin.DocsRoot.findData("test_docs");`

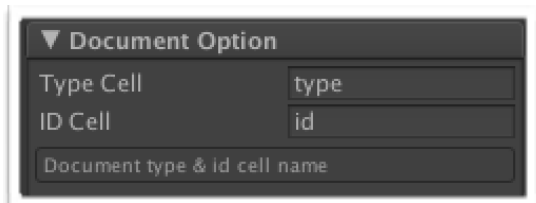
## 9. Set other options.



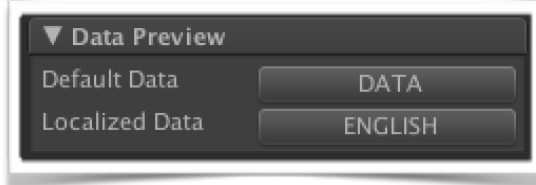
- Select whether to use individual GameObject from local file or to retrieve real-time data from Google Drive.



- Use Crypto: Option to set password.
- Auto Local Save: Automatically save retrieved data from Google Drive.
- Don't destroy on load: Use Object without deleting when switching Scene.

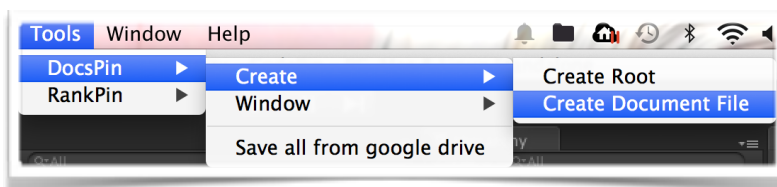


- Used when ID of spreadsheet and name of Type's cell is changed.



- Can only be checked when run on Unity.
- Check current data in use.

10. Select Create File from top menu when adding files from other spreadsheet.



- Use many different spreadsheet data.

## Samples

### 1. Simple

Example of retrieving data with spreadsheet name

#### TestDocs

```
DocsPin.DocsData data = DocsPin.DocsRoot.findData("test_docs");
if(data != null)
{
    int price = data.get<int>("object_01", "price");
    int limit = data.get<int>("object_01", "limit");
    string name = data.get<string>("object_01", "name");
    string title = data.get<string>("object_01", "title");
    string content = data.get<string>("object_01", "content");
    Debug.Log(string.Format("Price:{0}, Limit:{1}, Name:{2}, Title:{3}, Content:{4}",
        price, limit, name, title, content));
}
```

#### \_gd\_sampe\_01

```
DocsPin.DocsData data = DocsPin.DocsRoot.findData("_gd_sample_01");
if(data != null)
{
    int gender = data.get<int>("object_01", "gender");
    int price = data.get<int>("object_01", "price");
    int limit = data.get<int>("object_01", "limit");
    ArrayList reward = data.get<ArrayList>("object_01", "reward");
    double length = data.get<int>("object_01", "length");
    string rewardString = "";
    foreach(object r in reward)
    {
        rewardString += string.Format("{0} ", r);
    }
    Debug.Log(string.Format("Gender:{0},Price:{1},Limit:{2},Reward:[{3}],Length:{4}",
        gender, price, limit, rewardString, length));
}
```

### 2. Data Singleton

Example of using GameObject into Singleton in selected spreadsheet

#### TestDocs

```
int price = TestDocs.price("price");
int limit = TestDocs.limit("limit");
string name = TestDocs.names("name");
string title = TestDocs.title("title");
string content = TestDocs.content("content");
Debug.Log(string.Format("Price:{0}, Limit:{1}, Name:{2}, Title:{3}, Content:{4}",
    price, limit, name, title, content));
```

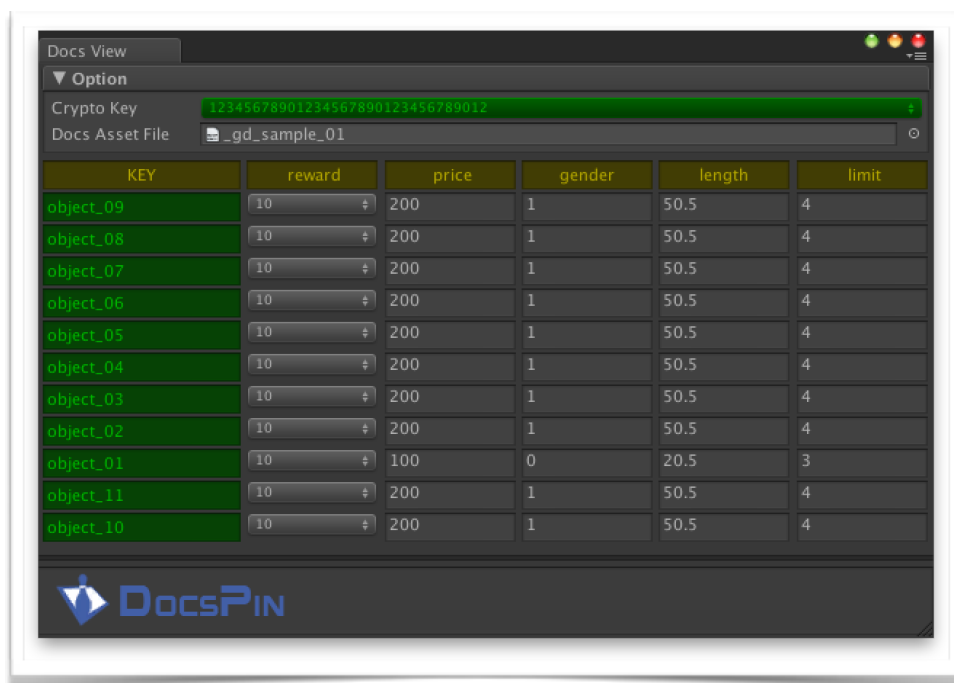
#### \_gd\_sampe\_01

```
int gender = DataSample01.gender("object_01");
int price = DataSample01.price("object_01");
int limit = DataSample01.limit("object_01");
ArrayList reward = DataSample01.reward("object_01");
double length = DataSample01.length("object_01");
string rewardString = "";
foreach(object r in reward)
{
    rewardString += string.Format("{0} ", r);
}
Debug.Log(string.Format("Gender:{0},Price:{1},Limit:{2},Reward:[{3}],Length:{4}",
    gender, price, limit, rewardString, length));
```

## Other Window

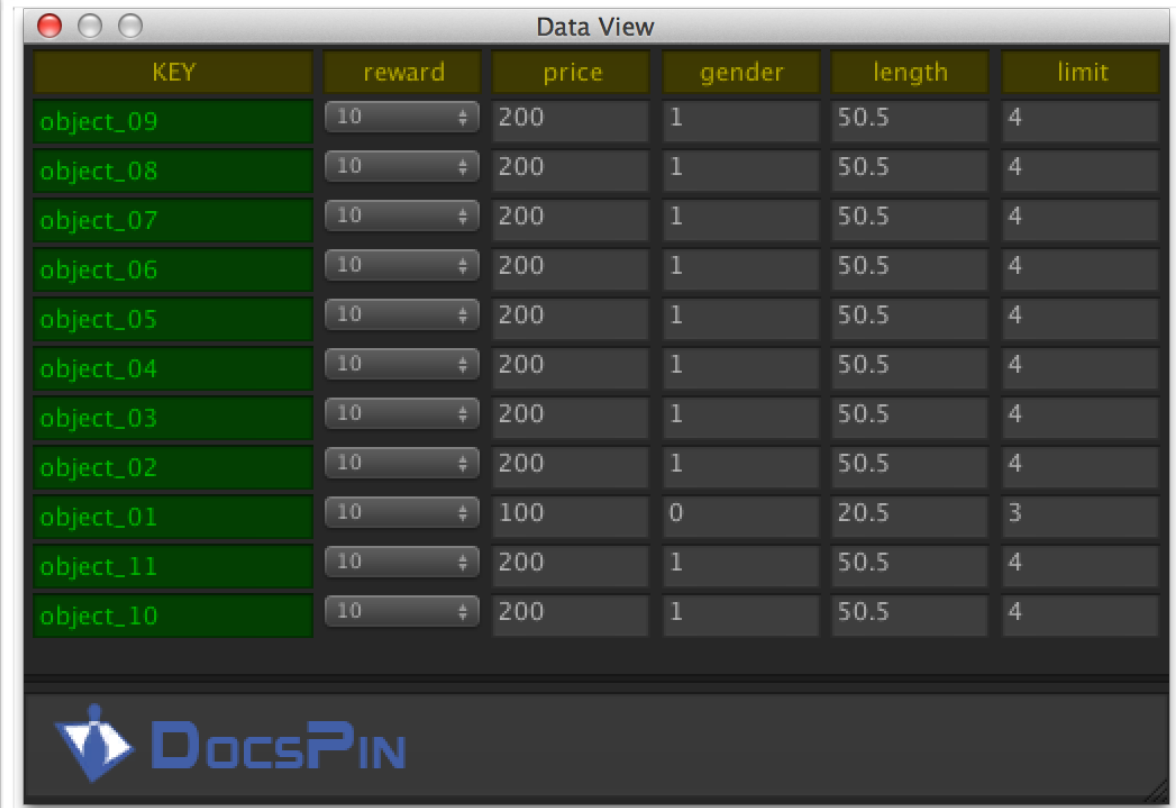
### 1. Docs View Window

Window displaying locally saved spreadsheet files.




## 2. Data View Window

Window displaying currently used spreadsheet when Unity is in operation.



KEY	reward	price	gender	length	limit
object_09	10	200	1	50.5	4
object_08	10	200	1	50.5	4
object_07	10	200	1	50.5	4
object_06	10	200	1	50.5	4
object_05	10	200	1	50.5	4
object_04	10	200	1	50.5	4
object_03	10	200	1	50.5	4
object_02	10	200	1	50.5	4
object_01	10	100	0	20.5	3
object_11	10	200	1	50.5	4
object_10	10	200	1	50.5	4

 DocsPin

## Sample of Google Drive spreadsheet

### 1. Test Docs

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					
6					

+ 
≡ 
data 
English 
Korean 
✓

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	KNIFE	A knife (plural knives) is a cutting tool with a cutting edge or blade, hand-held or otherwise, with or without a handle
4	object_02	GUN	A gun is a normally tubular weapon or other device designed to discharge projectiles or other material
5			

+ 
≡ 
data 
English 
Korean 
✓

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	칼	칼은 날카로운 면을 가진 도구로 어떤 것을 자를 때 쓴다
4	object_02	총	총(銃)은 일반적으로 개인이 휴대할 수 있는 구경이 작은 화기를 말한다.
5			
6			
7			

+ 
≡ 
data 
English 
Korean 
✓

## 2. \_gd\_sample\_01

	A	B	C	D	E	F
1	type	int	int	int	list	double
2	id	gender	price	limit	reward	length
3	object_01	0	100	3	10,100,200	20.5
4	object_02	1	200	4	10,100,200	50.5
5	object_03	1	200	4	10,100,200	50.5
6	object_04	1	200	4	10,100,200	50.5
7	object_05	1	200	4	10,100,200	50.5
8	object_06	1	200	4	10,100,200	50.5
9	object_07	1	200	4	10,100,200	50.5
10	object_08	1	200	4	10,100,200	50.5
11	object_09	1	200	4	10,100,200	50.5
12	object_10	1	200	4	10,100,200	50.5
13	object_11	1	200	4	10,100,200	50.5
14						
15						

+ 
≡ 
시트1 
✓