

Key-Value Online Storage

By Hippo Games

Welcome! Hope you will enjoy working with my asset!

## About:

Key-Value Online Storage (KVOS) is the simplest way to share small data between your users. It is based on Google App Scripts and each developer can create his own storage for free.

You can use KVOS to share user content. Here are some use cases:

* Sending gifts, credits, resources
* Sharing user content: characters, maps and other as tiny data (usually JSON)

You cannot use KVOS for:

* Saving game progress
* Long-term data storing as data lifetime is not guaranteed (old data is erased when the storage becomes full)
* Large data storing like screenshots

If you need long-term data storage without size limitations, I would recommend you to use Google Firebase.

## Features:

* Sharing user content
* User and developer friendly
* Dedicated storage instance with Google App Script
* Clean and simple C# code
* Unity 5, 2017, 2018 support
* Any platform (Standalone, Android, iOS, WebGL, UWP)

## Limitations:

* All limitations are made by Google
* Total storage size limit is 500kB
* Single key-value pair size is limited to 9kB

## Storage maintenance:

* Automated storage cleanup when no free space (deleting old data)

## Usage:

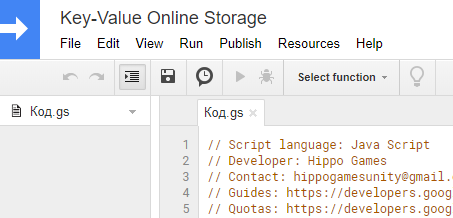
* Open Example.scene in Editor and run it
* Press SET and GET to test how it works

## Service response:

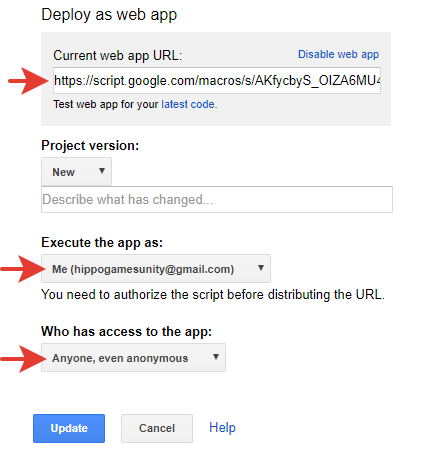
* Service returns ServiceResponse as JSON for Get and Set requests
* ServiceResponse contain *int* ErrorCode and *string* Response

## Storage setup:

* You need Google account
* Visit <https://script.google.com/home> and create a new project
* Start editing your script, you should see something like this:



* Name it as Key-Value Online Storage for example
* Find Storage.gs file from the asset folder
* Copy script source code to your Google App
* Press Publish/Deploy as web app
* Select Execute the app as: Me
* Select anonymous access



* Copy web app URL to clipboard
* Press Publish/Update button
* Go to Unity, open KeyValueService.cs and replace ServiceUrl
* Test it with Example.scene

## Storage maintenance:

* Open web app script and check its’ global settings (valiables)
* Change adminPassword
* Navigate to [URL]?set=[VALUE] to set a value and output a key
* Navigate to [URL]?get=[KEY] to get and output a value
* Navigate to [URL]?output&password=[PASSWORD] to output all storage
* Navigate to [URL]?count&password=[PASSWORD] to output records count
* Navigate to [URL]?size&password=[PASSWORD] to output total storage size in bytes
* Navigate to [URL]?clear&password=[PASSWORD] to clear your storage
* Navigate to [URL]?testfill&password=[PASSWORD] to fill your storage with random data

## References:

* Help: <https://developers.google.com/apps-script/>
* Guides: <https://developers.google.com/apps-script/guides/properties>
* Quotas: <https://developers.google.com/apps-script/guides/services/quotas>

## Contact me:

* hippogamesunity@mail.com