

Readme – JSON Tool And API Helpers 1.1.4

This package provides the Newtonsoft.JSON plugin that works on mobile(Android, iOS) and desktop(PC, Mac, Linux). An essential and high-performance tool for handling JSON data between client and server. Optimized for smaller package size, better performance and compatible with Unity environment.

One another point that worthy of your attention... is that our JSON plugin (DLL) supports running on multi-platforms without separating DLL files for the platforms. Just one file, for all supported platforms!

Plus API helper classes for using some popular, valuable 3rd-party services, such as Weather API, Time Zone API, and Giphy API. These helper classes (or manager), are carefully designed, work out of the box, would be very helpful for creating related features in your app, as well as for learning purposes! Detailed examples included.

JSON Tool

Newtonsoft Json .NET, is a standard JSON library in .NET ecosystem. Lightweight, high performance, easy to use, and it is free for commercial use. So, why you should use this package instead of the origin package? Because this JSON package is further optimized for smaller file size, added support for AOT environment like iOS(as well as Xbox, PS, Switch*), and workaround for the differences between Microsoft.NET & Unity3D Mono .NET. Work on Android, iOS, Windows, Mac, Linux, and Unity Editor.

* The included JSON library theoretically works on console platforms, but we may not test on all of these platforms as much more resources required for those testings. If you have a result of using this JSON library on Xbox, PS, Switch, or any else platforms, welcome to contact us (we just want to confirm that), and we will be happy to give you some special offers on our other assets.

API Helpers

(1) Weather & TimeZone API - Allows you to access current, past and future weather data for use in your apps. Including time zone, local weather, ski and mountain weather and marine, sailing, surfing and astronomy data.

WorldWeatherOnline docs : <https://www.worldweatheronline.com/developer/api/docs>

(2) GIF API - With our GIF helper classes, you can easily call the GIF APIs and receive response in Json format. Includes Upload API, GIF APIs, and GIF Sticker APIs. Fully customizable (limit, offset, rating, language and so many other parameters).

GIPHY docs : <https://developers.giphy.com/docs/api>

Support Platform

Unity Editor, Android, iOS, Windows, Mac, Linux

Unity 5 and newer

Reminders

Build iOS: **.NET 2.0 or greater Api Compatibility Level** is required for Newtonsoft.Json to work properly on iOS, please select **.NET 2.0 or greater (Not 2.0 subset)** in Player Settings before building XCode project. (**File > Build Settings > Player Settings > Other Settings > Optimization > Api Compatibility Level**)

*** Please noted that all the availability of the APIs helper classes provided with this asset may depend on related 3rd party services. We can't guarantee all the 3rd party services' availability, quality, as time goes. However, we will do our best to maintain the asset!

(1) Weather & TimeZone API Helper

To use World Weather Online API, please apply your own API keys here:

<https://developer.worldweatheronline.com/api>

How to USE? Run the demo scene for details!

Demo scene included: **WVO-ApiDemo.unity**

Example:

```
WVOApiHelper api = new WVOApiHelper(m_PremiumApiKey);
api.GetTimeZone(timeZoneQueryObj, (success, timeZone) => {
    if(success)
    {
        // Your code to use the timezone result data...
    }
});
```

APIs:

```
public void GetTimeZone(TimeZoneAPI.TimeZoneQuery query,
    Action<bool, TimezoneAPI> onComplete);
```

```
public void GetLocalWeather(LocalWeatherAPI.LocalWeatherQuery query,
    Action<bool, LocalWeatherAPI> onComplete);
```

```
public void GetLocalHistory(LocalHistoryAPI.LocalHistoryQuery query,
    Action<bool, LocalHistoryAPI> onComplete);
```

```
public void GetMarineWeather(MarineWeatherAPI.MarineWeatherQuery query,
    Action<bool, MarineWeatherAPI> onComplete);
```

```
public void GetMarineHistory(MarineHistoryAPI.MarineHistoryQuery query,
    Action<bool, MarineHistoryAPI> onComplete);
```

```
public void SearchLocation(SearchLocationAPI.SearchLocationQuery query,
    Action<bool, SearchLocationAPI> onComplete);
```

```
public void GetSkiWeather(SkiWeatherAPI.SkiWeatherQuery query,
    Action<bool, SkiWeatherAPI> onComplete);
```

```
public void GetAstronomy(AstronomyAPI.AstronomyQuery query,
    Action<bool, AstronomyAPI> onComplete);
```

*** Note that the detailed comments for each API and the parameters are included in the API Helper scripts.**

(2) Giphy API Helper

To use Giphy API, it requires a Giphy account to create API KEY and Upload API Key. You need to request a production key for the Upload API as well.

APPLY HERE: <https://developers.giphy.com/dashboard>

How to USE? Run the demo scene for details!
Demo scene included: **GifApiDemo.unity**

Example:

```
GiphyManager.Instance.Trending((result)=>{
    if (result == null || result.data.Count == 0)
    {
        // OnFail
    }
    else
    {
        // Success: Your code to use the result...
    }
});
```

GIF APIs:

Search GIFs by keyword(s):

```
GiphyManager.Instance.Search(keyWordList, onCompleteAction, onFailAction);
```

Get a single GIF by the GIF Id:

```
GiphyManager.Instance.GetById(giphyGifId, onCompleteAction, onFailAction);
```

Get multiple GIFs by the GIF Ids:

```
GiphyManager.Instance.GetByIds(giphyGifIdList, onCompleteAction, onFailAction);
```

Random a single GIF:

```
GiphyManager.Instance.Random(onCompleteAction, onFailAction);
```

Random a single GIF by tag:

```
GiphyManager.Instance.Random(tag, onCompleteAction, onFailAction);
```

Translate a keyword to GIF:

```
GiphyManager.Instance.Translate(keyWord, onCompleteAction, onFailAction);
```

Get the popular GIFs on the Giphy network:

```
GiphyManager.Instance.Trending(onCompleteAction, onFailAction);
```

Search GIF Stickers by keyword(s):

```
GiphyManager.Instance.Search_Sticker(keyWordList, onCompleteAction, onFailAction);
```

Random a single GIF Sticker:

```
GiphyManager.Instance.Random_Sticker(onCompleteAction, onFailAction);
```

Random a single GIF Sticker by tag:

```
GiphyManager.Instance.Random_Sticker(tag, onCompleteAction, onFailAction);
```

Translate a keyword to GIF Sticker:

```
GiphyManager.Instance.Translate_Sticker(keyWord, onCompleteAction, onFailAction);
```

Get the popular GIF Stickers on the Giphy network:

```
GiphyManager.Instance.Trending_Sticker(onCompleteAction, onFailAction);
```

Upload API:

Upload a local GIF file to the Giphy network and return the GIF Id:

```
GiphyManager.Instance.Upload(gifPath, tagList, onCompleteAction, onProgressAction, onFailAction);
```

Other APIs:

Random Id (Giphy user Id):

```
GiphyManager.Instance.GenerateRandomIdForUser(onCompleteAction, onFailAction);
```

Get the GIF categories on the GIPHY network:

```
GiphyManager.Instance.GetGifCategories(onCompleteAction, onFailAction);
```

Get a list of valid terms that completes the given tag:

```
GiphyManager.Instance.GetAutocompleteTerms(tag, limit, offset, onCompleteAction, onFailAction);
```

Get the Giphy channels matching the keyword:

```
GiphyManager.Instance.SearchChannels(keyword, limit, offset, onCompleteAction, onFailAction);
```

Get a list of tag terms related to the given tag:

```
GiphyManager.Instance.SearchSuggestions(tag, onCompleteAction, onFailAction);
```

Get a list of the most popular trending search terms on the GIPHY network:

```
GiphyManager.Instance.GetTrendingSearchTerms(onCompleteAction, onFailAction);
```

*** Note that the detailed comments for each API and the parameters are included in the API Helper scripts.**

(4) Social Share

Share GIF/image Url(s) return by the Giphy APIs. GIF preview and playback depends on the social platform. Support up to 15 social platforms (Facebook, Twitter, Tumblr, VK, Pinterest, LinkedIn, Odnoklassniki, Reddit, QQZone, Weibo, Baidu, MySpace, LineMe, Skype).

Share GIF and/or text message:

```
GifSocialShare gifShare = new GifSocialShare();
```

```
gifShare.ShareTo(Social: socialPlatformType, string: title, string: description, string: url1, string: url2);
```

THANK YOU

Thank you for using our assets!

For any question and bug report please contact us at swan.ob2@gmail.com.

Remember to rate this asset on the Asset Store. Your review is always appreciated, and very important to the development of this asset!

[Review And Rating](#)

Visit our asset page to find out more!

<https://www.swanob2.com/assets>

SWAN DEV