

Forum:[Hack Forums](#)

Thread: [Kleenscan API C#](#)

SIM-4704770 - May 30, 2022, 6:55 pm

An unofficial .Net wrapper for the kleenscan api

This was made on .NET 4.7.2, it references Newtonsoft.Json 13.0.1

```
Code using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net.Http;
using System.Text;
```

```
namespace Kleenscan
{
    public class Config
    {
        /// <summary>
        /// Your X-Auth-Token, Log in to your account, go to Profile > API > Generate, in order to
create a fresh X-Auth-Token
        /// </summary>
        public static string AuthenticationToken;
    }

    public class AVResult
    {
        public string Name;
        public string Status;
        public string Flagname;
        public string Lastupdate;
    }

    public class AVList
    {
        /// <summary>
        /// Retrieves a list of runtime scanners
        /// </summary>
        /// <returns>File antivirus list</returns>
        public static List<string> GetFileScanners()
        {
            using (var httpClient = new HttpClient())
            {
                using (var request = new HttpRequestMessage(new HttpMethod("GET"),
"https://kleenscan.com/api/v1/get/avlist"))
                {
                    request.Headers.Add("X-Auth-Token", Config.AuthenticationToken);
                }
            }
        }
    }
}
```

```

        dynamic json =
JsonConvert.DeserializeObject(httpClient.SendAsync(request).Result.Content.ReadAsStringAsync()
.Result);

        if (json.success == "true")
        {
            List<string> avList = new List<string>();

            foreach (string av in json.data.file)
                avList.Add(av);

            return avList;
        }
        else throw new InvalidOperationException((string)json.message);
    }
}

```

```

/// <summary>
/// Retrieves a list of runtime scanners
/// </summary>
/// <returns>URL antivirus list</returns>
public static List<string> GetURLScanners()
{
    using (var httpClient = new HttpClient())
    {
        using (var request = new HttpRequestMessage(new HttpMethod("GET"),
"https://kleenscan.com/api/v1/get/avlist"))
        {
            request.Headers.Add("X-Auth-Token", Config.AuthenticationToken);

```

```

        dynamic json =
JsonConvert.DeserializeObject(httpClient.SendAsync(request).Result.Content.ReadAsStringAsync()
.Result);

        if (json.success == "true")
        {
            List<string> avList = new List<string>();

            foreach (string av in json.data.url)
                avList.Add(av);

            return avList;
        }
        else throw new InvalidOperationException((string)json.message);
    }
}

```

```

/// <summary>
/// Retrieves a list of runtime scanners
/// </summary>

```



```

        {
            AVResult aVResult = new AVResult
            {
                Name = av.avname,
                Status = av.status,
                Flagname = av.flagname,
                Lastupdate = av.lastupdate
            };
            avResultList.Add(aVResult);
        }
        return avResultList;
    }
    else throw new InvalidOperationException((string)json.message);
}
}
}

```

```

/// <summary>
/// Used for scanning files, with all antiviruses, uploads from current computer.
/// </summary>
/// <param name="filePath"></param>
/// <returns>Scan token</returns>
public static string ScanFile(string filePath)
{
    using (var httpClient = new HttpClient())
    {
        using (var request = new HttpRequestMessage(new HttpMethod("POST"),
"https://kleenscan.com/api/v1/file/scan"))
        {
            request.Headers.Add("X-Auth-Token", Config.AuthenticationToken);

            var multipartContent = new MultipartFormDataContent
            {
                {
                    new ByteArrayContent(File.ReadAllBytes(filePath)),
                    "path",
                    Path.GetFileName(filePath)
                },
                {
                    new StringContent("all"),
                    "avList"
                }
            };
            request.Content = multipartContent;

            dynamic json =
JsonConvert.DeserializeObject(httpClient.SendAsync(request).Result.Content.ReadAsStringAsync()
.Result);

            if (json.success == "true")
            {
                return (string)json.data.scan_token;
            }

```

```

        else throw new InvalidOperationException((string)json.message);
    }
}

/// <summary>
/// Used for scanning files, with a custom set of antiviruses, uploads from current computer.
/// </summary>
/// <param name="filePath"></param>
/// <returns>Scan token</returns>
public static string ScanFile(string filePath, string avlist)
{
    using (var httpClient = new HttpClient())
    {
        using (var request = new HttpRequestMessage(new HttpMethod("POST"),
"https://kleenscan.com/api/v1/file/scan"))
        {
            request.Headers.Add("X-Auth-Token", Config.AuthenticationToken);

            var multipartContent = new MultipartFormDataContent
            {
                {
                    new ByteArrayContent(File.ReadAllBytes(filePath)),
                    "path",
                    Path.GetFileName(filePath)
                },
                {
                    new StringContent(avlist),
                    "avList"
                }
            };
            request.Content = multipartContent;

            dynamic json =
JsonConvert.DeserializeObject(httpClient.SendAsync(request).Result.Content.ReadAsStringAsync()
.Result);

            if (json.success == "true")
            {
                return (string)json.data.scan_token;
            }
            else throw new InvalidOperationException((string)json.message);
        }
    }
}
}

```

Examples of how to use the wrapper.

```
Code List<string> avList = Kleenscan.AVList.GetFilesScanners();
```

```
foreach (string av in avList)
{
    Console.WriteLine(av);
}
```

```
}
```

Gets a list of a file scanners and then prints them to console.

```
Code string scan_token = Kleenscan.FileScanner.ScanFile("C:\\example.exe");
List<Kleenscan.AVResult> scanResults = Kleenscan.FileScanner.GetScanResults(scan_token);
foreach (Kleenscan.AVResult result in scanResults)
{
    Console.WriteLine(result.Name);
    Console.WriteLine(result.Status);
    Console.WriteLine(result.Flagname);
}
```

Gets scan token from uploading example.exe and scanning it. Then it gets scan results from the token.

I'll be adding url scanning soon, if anyone has other sites they would like a c# wrapper for, leave suggestions.

Immortal - May 31, 2022, 2:15 pm

Looks good, code is clean as I don't have much experience with Kleenscan about how long does the operation take to upload and get results back?

SIM-4704770 - May 31, 2022, 10:48 pm

(May 31, 2022 02:15 PM)Immortal Wrote: [\[CITE\]](#) Looks good, code is clean as I don't have much experience with Kleenscan about how long does the operation take to upload and get results back? Like 15-30 seconds

jmark7 - June 1, 2022, 2:26 pm

Very good release,
Thanks for considering my request.

SIM-4704770 - June 2, 2022, 1:54 am

(Jun 1, 2022 02:26 PM)jmark7 Wrote: [\[CITE\]](#) Very good release,
Thanks for considering my request.

No problem, thanks for making the request, haven't worked with c# in a while so it was a nice refresher