ADVANTAGE CSP
WEB API

advantagecsp.com

Overview

Advantage CSP introduces a uniquely wrapped RESTful Web Services API, which allows you to create and consume content and data from Advantage CSP.

The main benefits of using these services are the following:

- Support for multiple RESTful API services
- User-friendly UI for configuration of the services
- Control over the set of content types that are exposed
- Option to allow anonymous access per profile or type
- Easy access to related data
- Auto-generated API reference
- Support for saved queries and calculated fields
- Ability to specify select returned fields, plus master ID and version

The Advantage API allows developers to easily setup and access any Defined Business Object quickly and securely. Granular control can be configured from the Advantage CSP admin console.

Client-side consumers can easily access the data using standard web ajax calls or may use the Advantage Wrapper that makes retrieving data simple.

Using Advantage CSP's UI, you can include but not limited in your web service the following types of content:

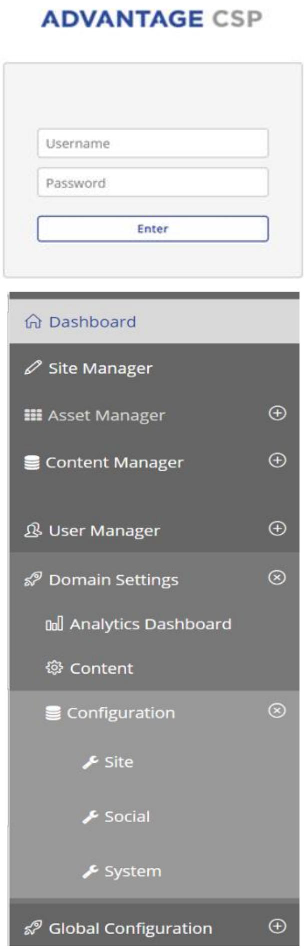
- News items
- Blogs and Blog posts
- Events and Calendars
- Images and Image libraries
- Videos and Video libraries
- Documents and Document libraries
- Lists and List items
- Comments
- Dynamic content
- Shared content blocks
- Flat and Hierarchical taxonomies
- Folders
- Pages
- Detailed API reference

Advantage CSP Server API Setup

Configuring any object for API Access can easily be done by creating an API entry in the configuration section.

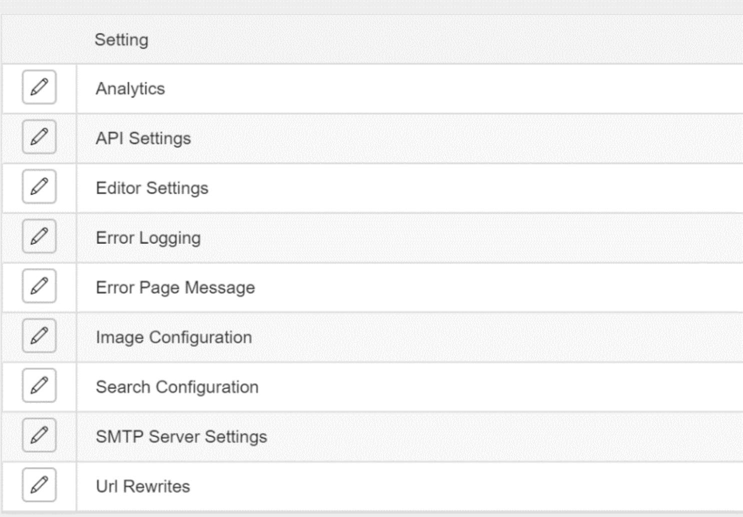
You may configure the same object under different API names in order to provide different clients with different levels of authentication or workflow.

You may configure the same object under different API names in order to provide different clients with different levels of authentication or workflow



To configure any object Login to the Advantage CSP admin.

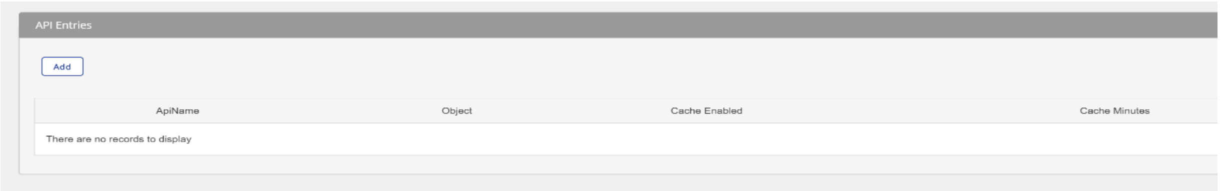
1. Type the following URL (mywebsite.com/admin
2. Once logged in, select configuration from the main admin navigation pane.
3. The node will expand to reveal three options; site, social and system.
4. Select System, the main admin workspace will now show a grid, as per below.
5. Select API Settings, by clicking the pencil.



	Setting
✎	Analytics
✎	API Settings
✎	Editor Settings
✎	Error Logging
✎	Error Page Message
✎	Image Configuration
✎	Search Configuration
✎	SMTP Server Settings
✎	Url Rewrites

Advantage Server API Setup

7. Configuring any object for API Access by selecting “add” to create a new entry.

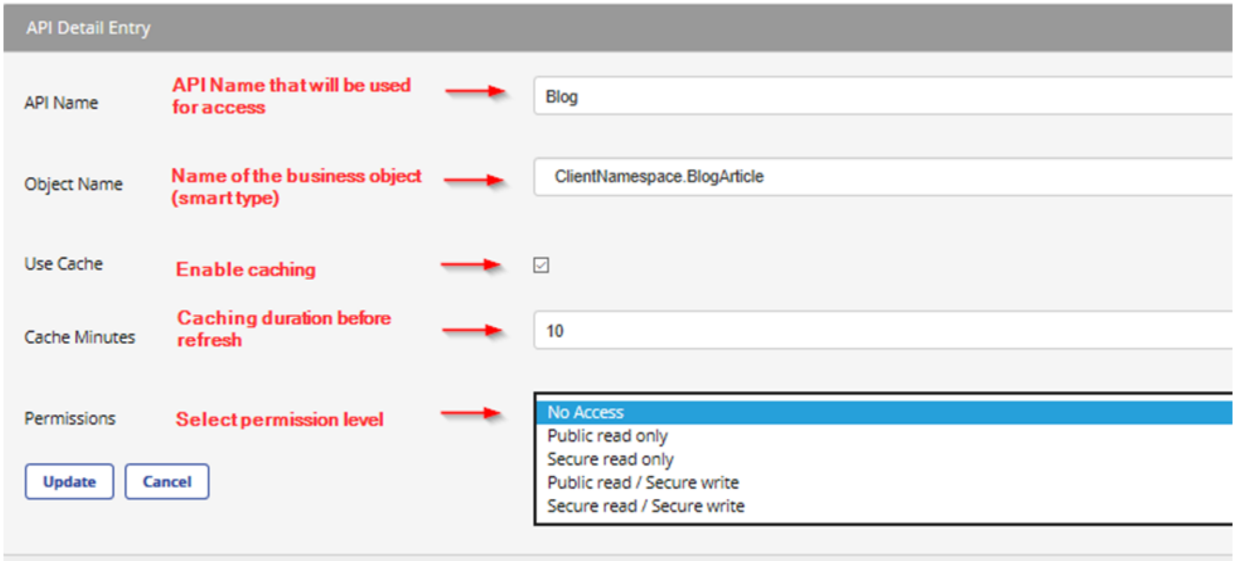


API Entries

Add

ApiName	Object	Cache Enabled	Cache Minutes
There are no records to display			

8. Enter all the required information as indicated below.



API Detail Entry

API Name — API Name that will be used for access → Blog

Object Name — Name of the business object (smart type) → ClientNamespace.BlogArticle

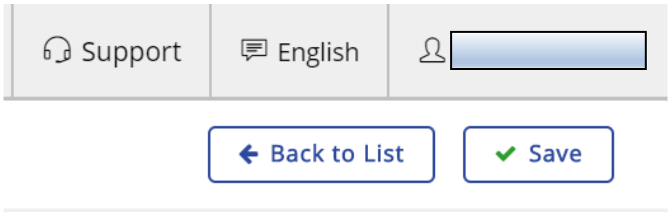
Use Cache — Enable caching → ☑

Cache Minutes — Caching duration before refresh → 10

Permissions — Select permission level → No Access / Public read only / Secure read only / Public read / Secure write / Secure read / Secure write

9. Press update

10. Then save



Advantage Server API Security

In order to access BusinessObjects that have been defined as “Secure” in the configuration a **“AdvantageSecureContentControllerHandler”** must be defined and added to the **AdvantageCSP.API.WebApiConfig”.**

Sample Implementation

A sample implementation is provided with Advantage for ease of implementation. Security can be customized.

AdvantageSecureContentControllerHandler

```
namespace AdvantageCSP.WebAPI
{
    //Sample class to show how you can authenticate requests.
    /// <summary>
    /// This is a code sample on how to implement the AdvantageSecureController.
This example looks at a response header.
    /// </summary>
    public class SecureContentControllerHandler :
AdvantageSecureContentControllerHandler
    {
        private static string myKey = "d32b2c8c-dc57-4978-bf98-7c82db7c027c";
        private class AuthResponse
        {
            public string UserName { get; set; }
            public string Password { get; set; }
            public DateTime TimeStamp { get; set; }
        }

        public override AdvantageAuthResponse
AuthenticateRequest(HttpRequestMessage request)
        {
            var credentials = ParseAuthorizationHeader(request);
            //check for credentials and if they expired?
            if (credentials != null)//&& credentials.TimeStamp >
DateTime.Now.AddDays(-1)
                return new AdvantageAuthResponse(){Success =true};
            else
                return new AdvantageAuthResponse() { Success = false };
        }

        private AuthResponse ParseAuthorizationHeader(HttpRequestMessage
request)
        {
            try
            {
                AuthResponse result = new AuthResponse();
                string authHeader = null;
                var auth = request.Headers.Authorization;
                if (auth != null && auth.Scheme == "Basic")  authHeader =
auth.Parameter;
                if (string.IsNullOrEmpty(authHeader))  return null;
                authHeader =
Encryption.Decrypt(Encoding.Default.GetString(Convert.FromBase64Stri
ng(authHeader)), myKey);
                var tokens = authHeader.Split(':');
                if (tokens.Length < 3) return null;
                result.UserName = tokens[0];
                result.Password = tokens[1];
                result.TimeStamp =
DateTimeOffset.FromUnixTimeSeconds(Convert.ToInt64(tokens[2])).Date
Time;

                return result;
            }
            catch{ return null; }



    //Sample of some authorization header for use in secure api.
        public static string GenerateAuthHeader()
        {
            string username = "SomeName";
            string password = "SomePassword";
            string datetime =
DateTimeOffset.UtcNow.ToUnixTimeSeconds().ToString();
            byte[] data =
ASCIIEncoding.ASCII.GetBytes(Encryption.Encrypt(string.Format("{0}:{1}:
{2}", username, password, datetime), myKey));
            return @"{ ""Authorization"": ""Basic "+
System.Convert.ToBase64String(data).ToString() + @"""}";
        }

    }
}
```

Global.asax

Register the handler defined for access

```
<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {

AdvantageCSP.API.WebApiConfig.Register(GlobalConfiguration.
Configuration, new
AdvantageCSP.WebAPI.SecureContentControllerHandler());

    }
</script>
```

Advantage Client API

This method will return a list of objects that have been defined as API accessible.

Usage

- Implement the routing (and optional security handler in Global.asax)
- Using secure methods requires you to return a successfull 'AdvantageAuthResponse' from the method "AuthenticateRequest" implemented in the handler. Optional indexing and filtering parameters can be sent via the 'data' parameter.
- Index (optional index defined on the object)
  - {"Key":"indexname","DataType":"index data type","Comparison":"comparitor","Value":"value to compare"}
- Filter (optional array of 'where' criteria)
  - [{"Key":"field name","DataType":"filter data type","Comparison":"comparitor","Value":"value to compare"}]
- SorlList (optional array of sorting criteria)
  - [{"field name","direction"}]
- FieldList (optional array of fields to populate)
  - [{"field name"}]
- MaxRecords (optional maximum records to return '0' is all)
- SkipRecords (optional records to skip '0' is none)

Authentication

- Authentication is used via the implementation of the **AdvantageSecureContentControllerHandle**r. This is a messaging handler that needs to be registered when you register the routing for the API in the Global.asax.

Error Codes

- 401 secure methods must implement the **'AdvantageSecureContentControllerHandler'** and return successful security request.
- 500 unable to access the API

POST Get All Objects Example

#Basic list request sample usage of object "article" for language "en" using secure method

`//yourserver.com/advantageapi/secure/content/get/article/en`

Headers

Content-Type	application/json
Accept	application/json
Authorization	Basic enRBUVNiVlVHdVZwTVNRbGZ1SXUwNWl5dE1zQWZac2JOK3pDbnBUUFpVckFmcWd6MHN5YnZiQlRNQjJQbVFXVEh0eTdndmZ2VGk1QzdsKzltaGc2T2paeUk3eEtCRy9iSEhYVnZVSTlmMXM9 Sample authentication header (encrypted with a username, password, datetime created)

POST Get Filtered Objects Example

```
//yourserver.com/advantageapi/secure/content/get/article/en
```

#Advanced list request Sample usage of object "article" for language "en"

Headers

Content-Type	application/json
Accept	application/json
Authorization	Basic enRBUVNiVlVHdVZwTVNRbGZ1SXUwNWl5dE1zQWZac2JOK3pDbnBUUFpVckFmcWd6MHN5YnZiQlRNQjJQbVFXVEh0eTdndmZ2VGk1QzdsKzltaGc2T2paeUk3eEtCRy9iSEhYVnZVSTlmMXM9 Sample authentication header (encrypted with a username, password, datetime created)

Body

raw (application/json)

```
{           "Index":{"Key":"Author","DataType":"string","Comparison":"equals","Value":"The Enginess Team"},
            "FieldList": ["Author", "ArticleDate", "ArticleImage", "Title"],
            "SortList": [{"FieldName":"ArticleDate","Direction":"asc"}, {"FieldName":"Title", "Direction":"desc"}],
            "Filter":[{"Key":"Title","DataType":"string","Comparison":"like","Value":"https"}]
}
```

POST Get Filtered Objects Example

```
//yourserver.com/advantageapi/secure/content/get/article/en
```

#Advanced list request Sample usage of object "article" for language "en"

Headers

Content-Type	application/json
Accept	application/json
Authorization	Basic enRBUVNiVlVHdVZwTVNRbGZ1SXUwNWl5dE1zQWZac2JOK3pDbnBUUFpVckFmcWd6MHN5YnZiQlRNQjJQbVFXVEh0eTdndmZ2VGk1QzdsKzltaGc2T2paeUk3eEtCRy9iSEhYVnZVSTlmMXM9 Sample authentication header (encrypted with a username, password, datetime created)

Body

raw (application/json)

```
{		"Index":{"Key":"Author","DataType":"string","Comparison":"equals","Value":"The Enginess Team"},
		"FieldList": ["Author", "ArticleDate", "ArticleImage", "Title"],
		"SortList": [{"FieldName":"ArticleDate","Direction":"asc"}, {"FieldName":"Title", "Direction":"desc"}],
		"Filter":[{"Key":"Title","DataType":"string","Comparison":"like","Value":"https"}]
}
```

Advantage Client API Wrapper

Advantage CSP offers a client wrapper that you can add to your client pages to simplify retrieving data.

Reference the API wrapper JS file from the advantage server

```
<script src="//yourserver.com/AdvantageCMS.Resource.WebResource.axd?d=AdvantageAPI.js"
type="text/javascript"></script>
```

The AdvantageAPI.js file objects to assist in defining method calls.

```
// Defined objects:
advDataType = Object.freeze({ String: 'string', Date: 'date', Boolean: 'boolean', Decimal: 'decimal', Integer: 'integer', UniqueIdentifier:
'uniqueidentifier' });

advCompare = Object.freeze({ Equals: 'equals', NotEquals: 'notequals', Like: 'like', NotLike: 'notlike', GreaterThan: 'greaterthan',
GreaterOrEquals: 'greaterorequals', LessThan: 'lessthan', LessOrEquals: 'lessorequals', All: 'all' });
```

Sample implementation

```
  <script type="text/javascript">

    function Sample() {
      // Create the request context API Aricle, language english
      var tmp = new AdvContentRequest("Article","en");

            // specify secure API access
      tmp.Authenticate = true;
      // specify the security header (custom)
      tmp.Header = 'encrypted security context';
            // OPTIONAL: specify an index
            // Parameters: Field Name, Value, Data Type, Compare Method
      tmp.set_Index("Author", 'Ernest Hemingway', advDataType.String, advCompare.Like);
            // OPTIONAL: specify a sort order
      tmp.AddSort(advDirection.Asc,’ 'Title' );
            // OPTIONAL: specify field list (reduces payload from server)
      tmp.AddField("Title");
      tmp.AddField("ArticleDate");
      tmp.AddField("Author");
      tmp.AddField("ArticleImage");
            // call function to return data.
            // Parameters: AdvContentRequest, Success Method, Failure Method

      advGetContent(tmp, showResult, showResult); //Get the data.
    }

    function showResult(resp) {
       // Show the pretty result.
       var pretty = JSON.stringify(resp, undefined, 2);
       alert( pretty);

    }
  </script>
```

Contact Us

Team is always available to help you get started or answer any questions you may have. Reach out to them at:

support@advantagecsp.com

or

Advantage CSP Portal

US & Canada: 1.844.901.8951