Consentua API Integration Guide

Version 0.2

Author

Tom Ashcroft

Document Control

Version	Status	Date	Author
0.1	*Initial Draft*	*5th September 2017*	*Tom Ashcroft*
0.2	*Revised Edition for Beluga*	*3rd June 2018*	*Tom Ashcroft*

Contents

Introduction & Purpose

Consentua is a consent management system. It provides a series of APIs that allow for easy integration with 3rd party applications.

This document aims to give the systems integrator and developer an insight into how the Consentua API's work. It also demonstrates the workflows involved in interacting with the Consentua service.

The purpose is not to document the Consentua API as this is done extensively via the Swagger interface (https://swagger.io). For those not familiar with Swagger API Documentation an explanation is provided.

Terms & Definitions

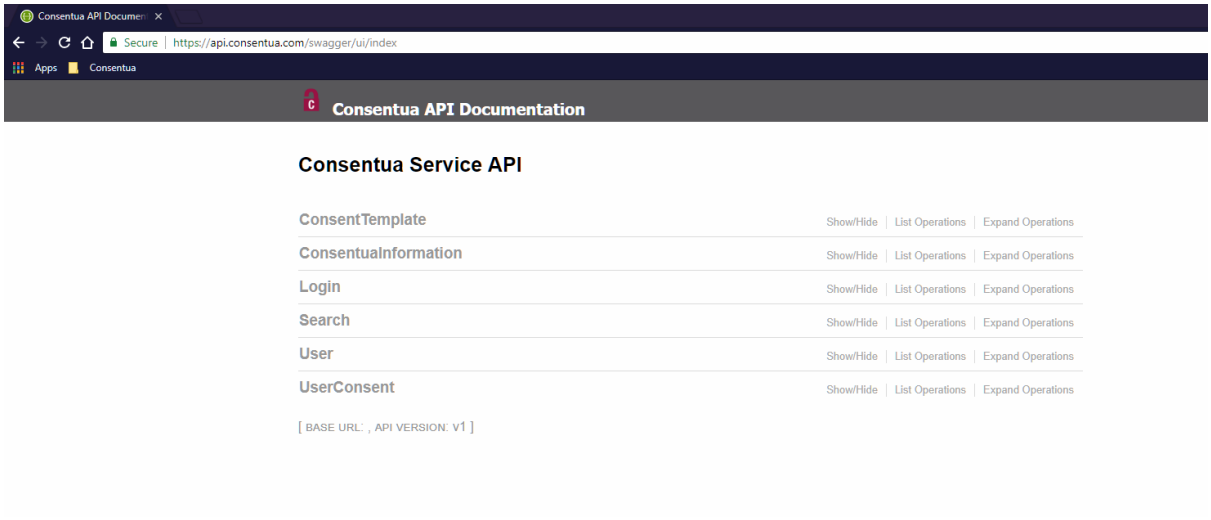
Term	Definition
Consentua	The Consentua Service
API	Application Programming Interface
REST	**RE**presentational **S**tate **T**ransfer
Consent	Permission for something to happen or agreement to do something.
Client	The company using the Consentua service
Service	The service provided by the client to their users
Client User	Users of the Consentua Service registered by a Client
Role	Roles within consentua that allow access to functionality
User	A User of the Client's Service
Consent Template	The template of consents that Users interact with
Purpose Group	A Group of Purposes
Purpose	The Purpose of the consent, what data will be used and what that data will be used for
MD5	A common hashing algorithm
Consent Registrar	A trusted third party that provides consent management according to the GDPR regulations
Consent Receipt	A form of exchanging consent records between consent registrar's
GUID	Globally Unique Identifier
GDPR	General Data Protection Regulation – a European personal data privacy law that will be enacted on May 25th 2018

Swagger Documentation

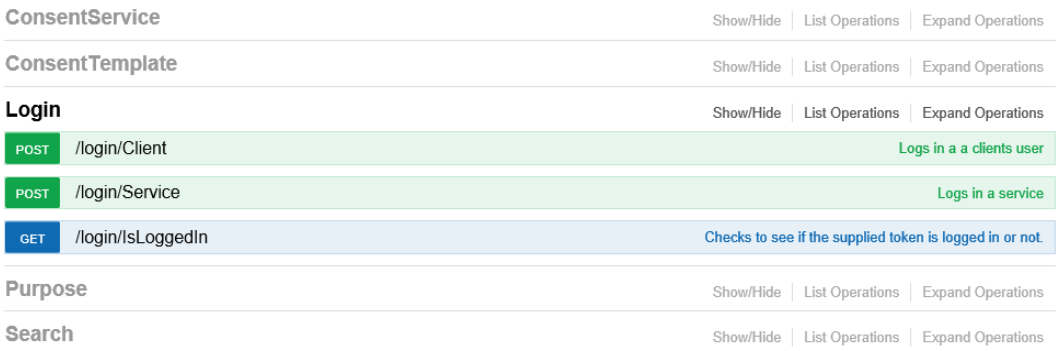
The documentation for the consentua API is provided via Swagger (https://swagger.io). This allows us to take the inline code comments and provide them to the user as a comprehensive form of documentation for each API call.

Documentation for Consentua can be found at api.consentua.com/swagger.

The Main Layout of the swagger documentation is shown below.



Each area of the documentation represents a REST endpoint with the various operations available. Selecting the endpoint will expand its operations.



The use of an operation is described opposite each operation's interface.

Selecting the operation will expand that endpoint to show the inputs and outputs of the operation at hand.

Full swagger documentation can be found at https://swagger.io/ .

Consentua Environments

There are 2 publicly available environments for Consentua both of which are comprised of several components: -

- The API Itself
- The Administrative Dashboard
- The Web SDK Server

In all cases backwards compatibility will be maintained but endpoints may become depreciated and may require superseding.

Test Environment

This is the consentua test environment where all integration and testing should be done it will also be the release testing point for all future versions and upgrades.

API – https://test.consentua.com

Dashboard – https://testdashboard.consentua.com

Web SDK – https://websdk-test.consentua.com

Live Environment

This is the consentua environment used for production environments only.

This environment will undergo fully managed upgrades and is only available to paying customers.

Upon migration from the test to the live environments all consentua Id's will change.

API – https://test.consentua.com

Dashboard – https://testdashboard.consentua.com

Web SDK – https://websdk-test.consentua.com

Consentua Integration

This integration guide represents the current state of the Consentua service. The guide is also laid out so that the integration points are documented in logical order. You only need to read the parts that are relevant to the stage of development you are at.

Considerations

As Consentua usage increases we will seek to rationalise our data models. The data models may therefore change over time. This will be notified to clients and integration work can be done on our public test server.

As previously backward compatibility will be maintained always but depreciated API calls will need to be reengineered as depreciated API calls will have a finite lifespan.

The root URL of the service should be a configurable variable to enable changing from test to live environments.

The roadmap for future development is available: -

https://trello.com/b/AktQXrOh/consentua-roadmap

Integration Areas

There are several areas of integration with consentua: -

Test API – Basic connectivity testing and token validation check.

Service Validation – Validating the service against the Consentua API.

User Management – Associating users with a Consentua service.

Consent Display – Displaying the service consent requirements to the user.

Consent Management – Recording and displaying the current user consents.

Consent Query – Checking that your service has consent to perform an action.

Provided Information

Consentua will provide the following to you: -

Client Id – a number denoting your client Id within the consentua system.

Client Key – A Key (GUID) that is associated with your client id. This can be changed via the Consentua management dashboard.

Endpoint – The endpoint that the identification information allows you access to.

This information should be configurable as it will change between consentua environments and potentially instances too.

Time

Time and tide wait for no man.

To prevent any time and date issues with consentua all times and dates in consentua are in UTC.

Connectivity Testing

This is a preparation task for integration with Consentua.

The ConsentuaInformation API is provided for base connectivity testing, it is documented here at https://test.consentua.com/swagger/ui/index#!/ConsentuaInformation/GetSystemInformation

Essentially it provides a set of version information useful in the initial stages of integration.

It also provides a way to check your token hashing results with the expectations of the Consentua token system.

{endpoint}/test/key

This is further documented under Generating a LoginToken

Service Validation

This is the first integration task, without a token the rest of consentua cannot be accessed.

To validate a service against consentua it will need to obtain a token. Consentua issues tokens for authorisation purposes not authentication. To that end tokens are valid for a given period.

Static Identifiers

ClientId [Required] The Client Id supplied by consentua to you.

ServiceId [Required] The Id of the service being logged in. This can be found on the Administrative Dashboard.

Obtaining a Token

To obtain a token to access Consentua as a service your own service will have to login. The endpoint for that is: -

{endpoint}/login/Service

This is a POST API the JSON body is: -

```
{
  " ClientId ": 0,
  " ServiceId ": 0,
  "LoginToken": "string", [See below on how to generate]
  " DeviceId ": "string" [Optional]
}
```

The DeviceId is a parameter for app developers to audit the device used to manage consent.

Generating a LoginToken

To generate the LoginToken, you will need to use the Service key.

The service key can be found via the Administrative Dashboard

All service keys are UUID's

To generate the LoginToken take the Service Key and append _Day_Month_Year in string format to the end of the key and then generate an MD5 hash of that string.

REMINDER: ALL CONSENTUA TIMES ARE UTC!

Please note that the day and month are not padded, and the year is the full year.

Given a Service Key of: - B0AFC018-0895-4603-9B2B-8FB2FBC6E501

The string to hash for September 19, 2017 would be: -

"B0AFC018-0895-4603-9B2B-8FB2FBC6E501_19_9_2017"

The string to hash for January 9, 2018 would be: -

"B0AFC018-0895-4603-9B2B-8FB2FBC6E501_9_1_2018"

The string to hash for November 21, 2027 would be: -

"B0AFC018-0895-4603-9B2B-8FB2FBC6E501_21_11_2027"

Testing

We have included a tester for you to check your hash against to ensure that we are comparing like with like. It is available at {endpoint}/test/key.

When you enter your service key it will append the date and generate a hash (Useful for using swagger to test the API).

Login Token Lifetime

As noted before the consentua token is used for authentication not authorisation. Therefore by default consentua tokens roll over at midnight UTC.

User Management

To check if the user is registered with the service and to register the user if not, we use the user controller.

The user controller has 3 methods: -
{endpoint}/user/GetServiceUser
{endpoint}/user/AddUserToService
{endpoint}/user/AddUserToServiceEx

The workflow is to call GetServiceUser to check if the user is registered with your service and if not then to call AddUserToServiceEx to create and add the user to your service and provide the user identifier.

GetServiceUser requires: -

- Token - Your Login Token.
- ServiceId - Your ServiceId.
- Identifier - The Identifier returned via AddUserToServiceEx.

It returns a ServiceUserModel for the User.
ServiceUserModel
{
UserId (integer): The Id of the user in consentua ,
EmailAddress (string, optional), the email of the user must be a valid email if provided
Identifier (string, optional): The service identifier for this user. Allocated by consentua ,
ServiceId (integer, optional): The consentua Id of the service
}

The AddUserToService is depreciated and maintained for backward compatibility

AddUserToServiceEx requires: -

```
{
  "Token": [Login token],
  "User": {
    "UserId": [leave blank],
    "EmailAddress": [email address] [optional] [must be valid email if supplied] ,
    "Identifier": [leave blank],
    "ServiceId": [Your ServiceId]
  }
}
```

It returns a ServiceUserModel the same as GetServiceUser consentua will have filled in the UserId and the Identifier, these should be stored for retrieval and query elsewhere in consentua.

Consent Display

To populate the UI to display the consent purposes to the user we provide widgets for Android and IOS. These widgets automatically display the consent templates to the user and record their consent.

For web-based UI there is a toolkit that performs the same functionality as the widgets for Android and IOS.

Consent templates purpose groups and purposes are managed via the Consentua Administration Dashboard

The access point to retrieve the templates and purposes for display is: -

{endpoint}/template/get

This API Requires: -

- ClientId - Your ClientId
- ServiceId - Your ServiceId
- AccessToken - Your Login Token
- Language - The 2-letter language code for the language you want the template displayed in.

This returns an array of the consent template with purpose groups and purposes for display. If there is only one purpose group within a template, then the display wording will be contained in the consent template. If the template has more than one group, then the wording for the display of each group will be held in the purpose group.

Consent Management

Once the consent profile for the client service is displayed to the user, Consentua provides APIs to set and retrieve the users current consent profile for the client service.

This is the UserConsent Controller and provides the following methods: -

{endpoint}/userconsent/SetConsents

{endpoint}/userconsent/SetConsentsEx

{endpoint}/userconsent/GetConsents

Note: SetConsents is being depreciated

The workflow for consent management can vary but generally it starts with GetConsents to retrieve the users current consent profile to display it to the user. Then when the user updates their consents the client system should call SetConsents for an entire Template.

This call requires an array for the purposes and the state of the consent. When setting consent in the Consentua service we suggest that you set consent for each template individually to avoid time outs for services with multiple consent templates.

When setting consent, you should set consents for all purposes within a template whether the purpose is consented to or not.

Consentua then manages all your user consent interactions and performs all necessary audit functions for regulatory compliance.

The user will then have access to a consent receipt via their endpoint.

A system user can access consent receipts if granted access.

Consent Query

At the point a service wants to use consented information you can query that consent has been given.

Consentua offers users different forms of querying client consent services.

Specific Consent

Specific consent involves querying consent for a user of a service. It can be accessed at: -

{endpoint}/search/QueryUser

This is for use when a specific query of consent is needed before data processing can commence.

Purpose Consent

Purpose consent is querying a list of consenting users for a purpose. For example, a consent template may send email in which case the purpose query is best as it gives a list of consenting users for the purpose.

The endpoint is documented: -

{endpoint}/search/QueryPurpose

Consent Receipt

The consent receipt uses a standard specified by the Kantara Initiative. Full information can be found here at https://kantarainitiative.org/confluence/display/infosharing/Consent+Receipt+Specification.

Feedback

Integration Guide and API Documentation

This Integration guide and the API documentation within Swagger are living documents. We invite feedback to improve the experience for other developers.

You can provide feedback on the documentation at documentation@consentua.com

Bug Reports

We are working hard to ensure that Consentua is stable, consistent and bug free.

We are also aware that the brilliant developer community will uncover issues and bugs from time to time. We apologise if you are affected by any of these issues and would really like to hear from you when they occur.

Please report all software issues with Consentua to support@consentua.com.