

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 | Corrections | 23rd September 2017 | David Patterson |
|  |  |  |  |

# Contents

[Document Control 2](#_Toc492570072)

[Contents 3](#_Toc492570073)

[Introduction & Purpose 4](#_Toc492570074)

[Terms & Definitions 5](#_Toc492570075)

[Swagger Documentation 6](#_Toc492570076)

[Consentua Integration 7](#_Toc492570077)

[Considerations 7](#_Toc492570078)

[Integration Areas 7](#_Toc492570079)

[Provided Information 7](#_Toc492570080)

[Test API 9](#_Toc492570081)

[Service Validation 10](#_Toc492570082)

[Obtaining a Token 10](#_Toc492570083)

[User Management 11](#_Toc492570084)

[Consent Display 12](#_Toc492570085)

[Consent Management 13](#_Toc492570086)

[Consent Query 14](#_Toc492570087)

# 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

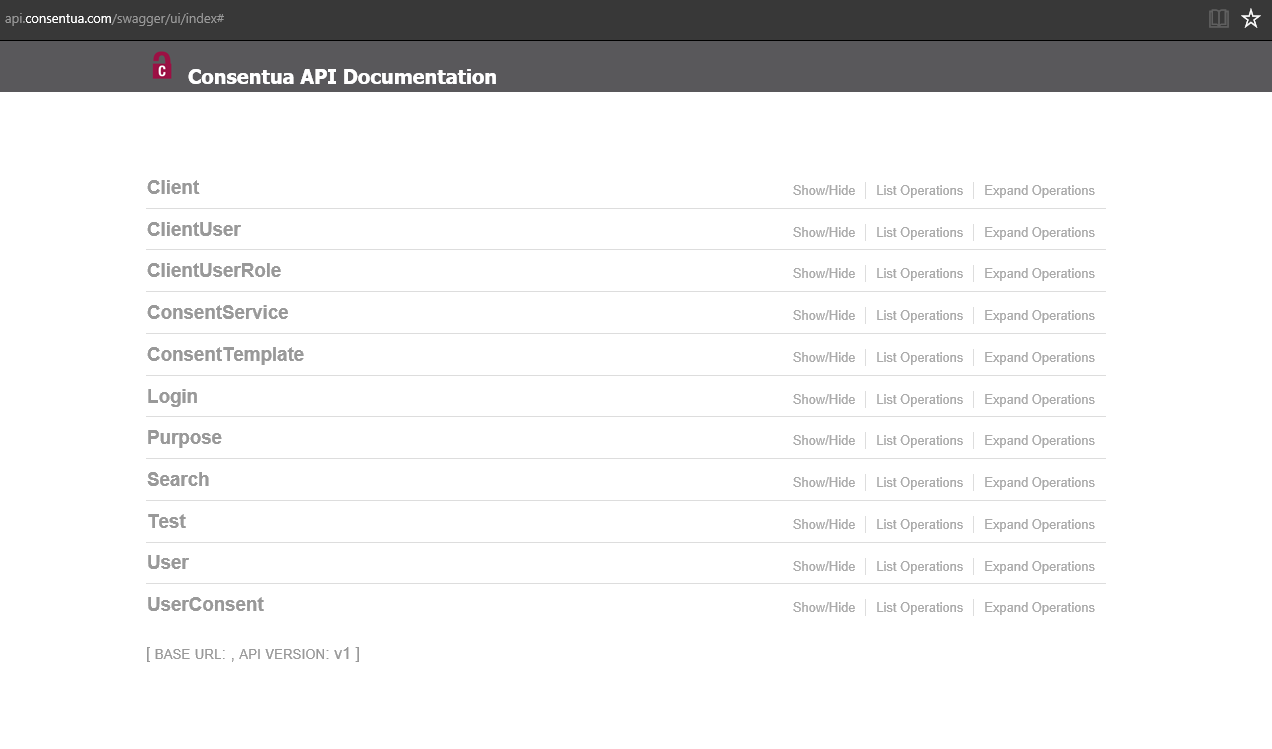
|  |  |
| --- | --- |
| Term | Definition |
| Consentua | The Consentua Service |
| API | Application Programming Interface |
| REST | **RE**presentational **S**tate **T**ransfer |
| Consent | Permission for something to happen or agreement to do something. |
| Client | The company using the Consentua 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

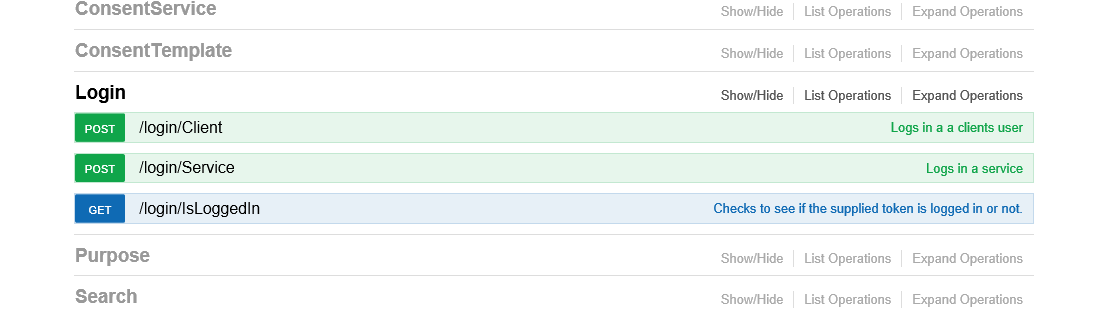
The documentation for the consentua API is provided via Swagger (https://swagger.io). This allows us to take the inline code comments and provide them to the user as a comprehensive form of documentation for each API call.

Documentation for Consentua can be found at [api.consentua.com/swagger](http://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 Integration

This integration guide represents the current state of the Consentua service. This includes all of the APIs for management of the service which are provided via the Consentua dashboard (<app.consentua.com>). 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

The future roadmap includes separation of dashboard and service API into different root URL’s. It also includes a redesign of the Service login API. The old API will be maintained until all clients have migrated

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.

The root URL of the service should be a configurable variable to enable changing from test to live environments. Also, there are multiple instances of consentua so the final live endpoint may differ from our generally published API endpoint.

The roadmap for future development is available from [contact@consentua.com](mailto:contact@consentua.com). You can find notifications for future changes at Consentua.com.

## 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 for either the testing or live Consentua environment.

# Test API

This is a preparation task for integration with Consentua.

The test API is provided for base connectivity testing, it is documented [here](http://api.consentua.com/swagger/ui/index#!/Test) at <http://api.consentua.com/swagger/ui/index#!/Test>

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

[{endpoint}/test/key](http://api.consentua.com/swagger/ui/index#!/Test/Key)

# 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. These tokens are single use tokens for a service so a new token must be obtained before every call to the Consentua service API.

## Obtaining a Token

In order 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](http://api.consentua.com/swagger/ui/index#!/Login/Service)

Your ClientId, ServiceId and DeviceId are required as well as a login token.

The ServiceId can be retrieved from the Consentua Dashboard as can the Service Key.

The DeviceId is a parameter for App Developers to Audit the Device used to manage consent.

To generate the login token, you will need to use the Service key and append \_Day\_Month\_Year to the end of the key and then generate an MD5 hash of that string.

Please note that the day and month are not padded and the year is the full year, so the string to hash for September 19, 2017 would be “{Service Key}\_19\_9\_2017”

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](http://api.consentua.com/swagger/ui/index#!/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).

This service is subject to changes that will be published via consentua.com. Do not rely on it to generate login tokens.

# 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 2 methods: -

[{endpoint}[/user/GetServiceUser](http://api.consentua.com/swagger/ui/index#!/User/GetServiceUser)](http://api.consentua.com/swagger/ui/index#!/User/GetServiceUser)

[{endpoint}/user/AddUserToService](http://api.consentua.com/swagger/ui/index#!/User/AddUserToService)

The workflow is to call GetServiceUser to check if the user is registered with your service within Consentua.

If the user is not registered then AddUserToService and record their Consentua ID within your service, indexed to their email.

Each call will need a new access token.

# 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.

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

[{endpoint}/template/get](http://api.consentua.com/swagger/ui/index#!/ConsentTemplate/Get)

This provides 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.

To set user consents the endpoint is: -

[{endpoint}/userconsent/setconsents](http://api.consentua.com/swagger/ui/index#!/UserConsent/SetConsents)

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](http://api.consentua.com/swagger/ui/index#!/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](http://api.consentua.com/swagger/ui/index#!/Search/QueryPurpose)

## Consent Receipt

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