

|  |  |
| --- | --- |
| **Project:** | **Tesis - Web services for Blue portal** |

**Documentation Web Services**

Index

[Revisions 3](#_bookmark0)

[References 4](#_bookmark1)

[Premise 5](#_bookmark2)

[Architecture 6](#_bookmark3)

[Access to services 7](#_bookmark4)

[General rules 7](#_bookmark5)

[Services available 8](#_bookmark6)

[Authentication 8](#_bookmark7)

[Ticket 9](#_bookmark9)

[Contracts 11](#_bookmark11)

[System tables 14](#_bookmark12)

[Examples (code) 15](#_bookmark13)

[Authentication c# 15](#_bookmark14)

[XML 16](#_bookmark15)

[Xml example of authentication 16](#_bookmark16)

[Xml Numticketaperti example 16](#_bookmark17)

[Example Xml Numcontratti 17](#_bookmark18)

# Revisions

|  |  |  |
| --- | --- | --- |
| Index of revisions | | |
| Overhaul | Description of the revisions | Date |
| 1.0 | First issue | 12/10/2018 |
| 1.1 | Production release of the first documented services. New mode of authentication.  Examples | 20/11/2018 |
| 1.2 | Added i services List of contracts and Elencocontrattina | 23/11/2018 |

# References

|  |  |  |
| --- | --- | --- |
| Index of references | | |
| Document or other reference | Author | Date |
|  |  |  |
|  |  |  |

# Premise

Through the specially developed integration services it is possible to have the Blue portal dialogue with Tesis to obtain data in real time.

This document will describe how to interface to Tesis.

# Architecture

Tesis software was developed in Web architecture.

Tesis interoperability with other software/computers is supported by SOAP Web Services. Tesis Web Services are software interfaces described through WSDL documents.

Other systems can interact with Web Services by activating the operations described in the interface with a Message system.

These messages are in XML format and transported via http/https protocol. Web services comply with WS-I 1.1 specifications

Invoke Web Service

Xml

Answer

**TESIS**

Portale Blue

Xml

Web Service

# Access to services

Services are accessible via Internet with http protocol

The access address to the production environment [is: hiips://servicedesk.sedoc.it/Tesissi/Sedoc/serviziblue.asmx](https://servicedesk.sedoc.it/TesisSI/Sedoc/serviziblue.asmx)

The access address to the test environment shall be :

(not implemented)

# General rules

Every object passed in the service request message requires property setting

*Authenticationtoken. This property is obtained by invoking the authentication service.*

Every object obtained as a reply has as first *Failed field*, through which it is possible to identify if the transaction has been successful.

If the transaction has not been successful will be valued also the second field

*Failuremessage* with the detail of the problem found.

# Services available

The following is a description of the services available and their methods (operations). The list of services is also available at: hiips:[//servicedesk.sedoc.it/Tesissi/Sedoc/serviziblue.asmx](https://servicedesk.sedoc.it/TesisSI/Sedoc/serviziblue.asmx)

Each service is described by the relative WSDL directly visualizable to the address hiips://servicedesk.sedoc.it/Tesissi/Sedoc/serviziblue.asmx? WSDL

**Authentication**

Operations :

* Authenticateuser

The service manages user authentication. It must be used at the beginning of each transaction as it returns the necessary Token string for all other services.

Requires name and password and returns GUID authentication string or error message

Path data requested by the service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Userlogin | String | Mandatory field (Username) |
| Password | String | Required field (user password) |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| **Authenticationtoken** | Guid | **Identifier from utilize in all calls to the other WS.** |
| Idutente | Long | Unique id assigned to Tesis user |

* Validateauthenticationtoken

The service allows verification of the token.

It requires authentication string Guid and returns if it is valid or not. Path data requested by the service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Authenticationtoken | Guid | Mandatory field (Identifier) |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| ValidateAuthenticationTokenResult | Boolean | True if Guid is valid, False if Guid is invalid or expired |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Project: | Web services for Blue portal | | Date: | 23/11/2018 |
| Drawn up by: | Enrico Cantarelli |  | Version | 1.2 |
| Document: | Documentation web services | | Page | 8 of 17 |

**Ticket**

Operation:

* Number

Returns the number of open tickets. It’s all the tickets that were opened. Through the Typoopening parameter *it is* possible to have all contracts opened automatically by monitoring systems or opened directly by the customer. With the parameter *Typomanagement you* can know those that are still to be managed or those under management. The tickets still to be handled are all those that have not had any intervention or treatment to take charge (see configuration of Tesis states) while the tickets that are handled are all the remaining ones still open.

Path data requested by the service

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Obb.** | **Notes** |
| Authenticationtoken | Guid | YES | Guid returned from authenticateuser service |
| Customer | Long | YES | Unique customer ID PK Tesis customers. |
| Typeopening | String | YES | If 'Automatic' then it returns  all tickets that have been opened from the monitoring system  If 'Customer' then it returns  all tickets that have been opened  directly from the customer  If 'All' it counts all the tickets  open |
| Typomanagement | String | NO | If 'To be handled then it returns  all the tickets they have undergone  a management.  If 'Managed' then it returns everyone  tickets that are still from  handle |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| Numeroticket | String | Ticket number |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Project: | Web services for Blue portal | | Date: | 23/11/2018 |
| Drawn up by: | Enrico Cantarelli |  | Version | 1.2 |
| Document: | Documentation web services | | Page | 9 of 17 |

* Lists

The list of open tickets returns. The parameters are the same as for the NumTicketAperti operation. The response is a table that returns: Call code or ticket code, ticket opening date, ticket taking charge date, if the ticket is billable, ticket notes, ticket subject, restore time provided, expected restore time, response time provided, expected response time, if the response time was met and if the recovery time was met

Path data requested by the service

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Obb.** | **Notes** |
| Authenticationtoken | Guid | YES | Guid returned from Authenticateuser service |
| Customer | Long | YES | Unique PK customer identifier Tesis customers. |
| Typeopening | String | YES | If 'Automatic' then it returns all tickets  that have been opened by the system  monitoring  If 'Customer' then it returns all the tickets that  were opened directly by the customer  If 'All' it counts all open tickets |
| Typomanagement | String | NO | If 'To be managed then it returns all tickets  which have undergone management.  If 'Managed' then it returns all tickets that  are still to be managed |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| Dsticketaperts | Datatable | Ticket list |

Dsticketaperti. Zz\_sedoc\_blue\_vw\_ticketaperts

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Dim.** | **Description** |
| Coded | String | 20 | Call code (alphanumeric unique) |
| Opening date | Datetime |  | Date Opening call |
| Date | Datetime |  | Date of taking charge of  ticket. It’s the date that coincides with the first state of taking charge. |
| Billable | Boolean |  | If Ticket Billable |
| Notes | String |  | Ticket notes, request notes |
| Requesting | String | 200 | Subject of the ticket, request |
| Temporisposta | Numeric | (18,2) | Response time provided |
| Temporispostaprevisto | Numeric | (18,2) | Expected response time |
| Remarried | Boolean |  | If the response time has been respected |

**Contracts**

Operation :

* Number of contracts

Returns the number of open contracts. It’s all the contracts that are up to date. The service provides a list of all valid contract types defined in Tesis. The types of contracts are contained in a configuration table and allow to characterize the contracts entered. The service will list all the types contracted valid in table with next the count of how many contracts for type are in existence to the date of the request. If the count is zero, it means that the contract type is valid but at the moment there are no assets for the customer.

Path data requested by the service

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Obb.** | **Notes** |
| Authenticationtoken | Guid | YES | Guid returned from Authenticateuser service |
| Customer | Long | YES | Unique PK customer identifier Tesis customers. |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| Dscontratti | Dataset | Datatable with contract type count |

Dscontratti. Zz\_sedoc\_blue\_vw\_numcontratti

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Dim.** | **Description** |
| Typeface | String | 6 | Code of type of contract |
| Descriptiontipocontratto | String | 500 | Description of the type of contract from Tesis table |
| Counttipocontratto | Int | 6 | Number of contracts outstanding by type of contract |

* List of contracts

The service allows to have a list of all the assets of the customer contained in the respective contracts that have a deadline and that on the date have not already expired. Assets that are tacitly renewed, that is, have no expiry date, are not counted and visible within the data table. For each asset (Articlesmerged) the days remaining before its expiry are counted.

Path data requested by the service

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Obb.** | **Notes** |
| Authenticationtoken | Guid | YES | Guid returned from Authenticateuser service |
| Customer | Long | YES | Unique PK customer identifier Tesis customers. |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| Dscontratti | Dataset | Datatable with asset list and maturity |

Dscontratti. ZZ\_SEDOC\_BLUE\_Vw\_ElencoContrattiScadenza

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Dim.** | **Description** |
| Typeface | String | 6 | Code of type of contract |
| Descriptiontipocontratto | String | 200 | Description of the type of contract from Tesis table |
| Article | String | 50 | Article number (Tesis product type field) |
| Descriptioninarticlencontr | String | 200 | Product Description (Tesis Product Description Field) |
| Countgiorniscadenza | Int |  | Number of days to maturity |

* Call sign

The service allows you to have a list of all types of service that are not active for the customer but are available in the catalog services Sedoc and Tesis.

Path data requested by the service

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Obb.** | **Notes** |
| Authenticationtoken | Guid | YES | Guid returned from Authenticateuser service |
| Customer | Long | YES | Unique PK customer identifier Tesis customers. |

Path data returned from service

|  |  |  |
| --- | --- | --- |
| **Nomecampo** | **Kind** | **Notes** |
| Failed | Boolean | Whether the transaction failed |
| Failuremessage | String | If failed transaction reports error |
| Dscontratti | Dataset | Datatable with asset list and maturity |

Dscontratti. ZZ\_SEDOC\_BLUE\_Vw\_ElencoContrattiScadenza

|  |  |  |  |
| --- | --- | --- | --- |
| **Nomecampo** | **Kind** | **Dim.** | **Description** |
| Article | String | 50 | Article number (Tesis product type field) |
| Descriptionarticle | String | 200 | Product Description (Tesis Product Description Field) |

# System tables

System tables referenced in paths

# Examples (code)

Below are some examples of c# code for using Tesis Webservices. The examples were developed in the Microsoft Visual Studio environment.

**Authentication c#**

The following c# code is extracted from a form (project windows form) which authenticates the user and gets the string Token and assigns it to the local variable authenticationToken.

Blueservices web service reference was created in the project

The instructions :

private Guid authenticationToken;

private void btnLogin\_Click(object sender, Eventargs e)

{

// We request the web refernce

Blueservices.Serviziblue ws = new Blueservices.Serviziblue(); Blueservices.Userauthenticationreqmsg req = new

BlueServices.UserAuthenticationReqMsg(); ws. Url = "[http://addressowebservice](http://indirizzowebservice/)"; req.Userlogin = "Myusername"; req. Password = "Miapassword";

Blueservices.Userauthenticationrespmsg resp = ws.Authenticateuser(req);

if (resp.Failed)

{

// login failed, I show the received error message in txtFailureMessage txtFailureMessage.Text = resp.Failuremessage;

txtFailed.Text = resp.Failed.Tostring();

}

else

{

// Login successfully txtFailureMessage.Text = ""; txtFailed.Text = resp.Failed.Tostring();

txtToken.Text = resp.AuthenticationToken.Tostring();

// except in local variable "authenticationToken" that will serve for invoking the subsequent services authenticationToken = resp.Authenticationtoken;

}

}

**XML**

It is possible to consume soap services with specific test applications such as Soapui ([hiips://www.soapui.org/](https://www.soapui.org/) ) or Boomerang ([hiips://chrome.google.com/webstore/detail/boomerang -](https://chrome.google.com/webstore/detail/boomerang-soap-rest-clien/eipdnjedkpcnlmmdfdkgfpljanehloah) [soap-rest-](https://chrome.google.com/webstore/detail/boomerang-soap-rest-clien/eipdnjedkpcnlmmdfdkgfpljanehloah)clien/eipdnjedkpcnlmmdfdkgfpljanehloah)

**Xml example of authentication**

<x:Envelope xmlns:x="hiip://Schem as.xmlsoap.org/soap/envelope/" xmlns:tem="hiip://tempuri.org/">

<x:Header/>

<x:Body>

<tem:Authenticateuser>

<tem:req>

<tem:Userlogin>Miologin</tem:Userlogin>

<tem:Password>Miapassword</tem:Password>

</tem:req>

</tem:Authenticateuser>

</x:Body>

</x:Envelope>

**Xml Numticketaperti example**

<x:Envelope xmlns:x="hiip://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="hiip://tempuri.org/">

<x:Header/>

<x:Body>

<tem:Numbersrules>

<tem:req>

<tem:Authenticationtoken>501b09be-620f-47b7-A605- d539df34ba23</tem:Authenticationtoken>

<tem:Idcliente>19303</tem:Idcliente>

<tem:Typeopening>CLIENT</tem:Typeopening>

<tem:Typemanagement></tem:Typemanagement>

</tem:req>

</tem:Numticketaperts>

</x:Body>

</x:Envelope>

**Example Xml Numcontratti**

<x:Envelope xmlns:x="<http://schemas.xmlsoap.org/soap/envelope/>" xmlns:tem="hiip://tempu ri.org/">

<x:Header/>

<x:Body>

<tem:Numcontratti>

<tem:req>

<tem:Authenticationtoken>262f96fb-df0d-4b91-9ea3-c943241d6441</tem:Authenticationtoken>

<tem:Idcliente>19303</tem:Idcliente>

</tem:req>

</tem:Numcontratti>

</x:Body>

</x:Envelope>