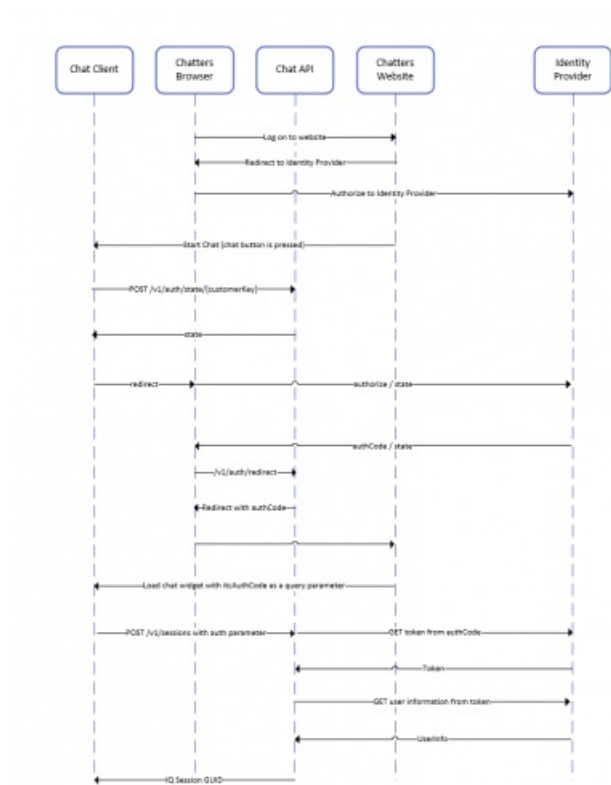


Configuring secure chat (SSO)

This article describes the steps you need to follow in order to set up secure chat based on single sign on (SSO). For some chat solutions, it is essential that the end user is authenticated and securely identified for the communication to take place. Puzzel's secure chat solution will enable an authenticated users e.g. signed in to a "my page" or similar using an Open ID Connect (OIDC) based authentication (Signicat, Azure AD B2B etc.), to use the same authentication when starting a chat. In this way agents will know the authentication details e.g. the end user's identity, instead of the end user stating their identity manually.

Below is the general flow chart for Puzzel's Chat SSO authentication using Open ID Connect:



Steps to set up secure chat

1. In the Administration Portal, go to "Admin → Users → Products → Secure Chat"

Secure Chat ?

Quick find users/user groups

Company

Braathe dev

User Group

agent

User

Select User

	Inherit	Value	Inherit	Value	Inherit	Value
Claims to be masked (Semicolon separated like: sub;nationalId)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>			
OIDC ACR Values	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>			
OIDC Client Id	<input type="checkbox"/>	https://euwa-dev.e	<input checked="" type="checkbox"/>	https://euwa-dev.e		
OIDC Client Secret	<input type="checkbox"/>	*****	<input checked="" type="checkbox"/>	*****		
OIDC endpoint for authorization	<input type="checkbox"/>	https://login.micro	<input checked="" type="checkbox"/>	https://login.micro		
OIDC endpoint for token	<input type="checkbox"/>	https://login.micro	<input checked="" type="checkbox"/>	https://login.micro		
OIDC endpoint for token authentication method (client_secret_basic or client_secret_post)	<input type="checkbox"/>	client_secret_pos	<input checked="" type="checkbox"/>	client_secret_pos		
OIDC endpoint for userinfo	<input type="checkbox"/>	https://graph.micr	<input checked="" type="checkbox"/>	https://graph.micr		
OIDC Scope	<input type="checkbox"/>	openid profile use	<input checked="" type="checkbox"/>	openid profile use		

Claims to be masked - semicolon separated list; every claim present in the list will be masked by the Chat API. Neither the user or the agent will be able to see the full value of the claim.

OIDC ACR Values - Authentication Context Class Reference Values (see https://openid.net/specs/openid-connect-eap-acr-values-1_0.html)

OIDC Client ID - Your client id

OIDC Client Secret - Your client secret

OIDC endpoint for authorization - Authorization endpoint

OIDC endpoint for token - Token retrieval endpoint

OIDC endpoint for token authentication method:

- *client_secret_post* - the client secret will be present in the POST data.
- *client_secret_basic* - the client secret will be present as Authorization header.

OIDC endpoint for userinfo - User info endpoint

OIDC Scope - Limitation of what user data can be retrieved (see <https://oauth.net/2/scope/>)

2. In the Administration Portal, go to "Admin → Services → Services → {Customer Service Number} → Queues"

ID (queue_key)	Queue Name	Qualities	SLA (sec)	Alternative SLA (sec)	SLA Overflow 1 (score)	SLA Overflow 2 (score)	Wrap-up (sec)	Call Recording	Reserved Agent (s)	Autostore reserved agent (d)	Maximum in Queue	Auth. Name	Refuse if All Logged Out	Refuse if All In Pause	Sync/Add to Statistics	Require Skill on queue for Reserved	Require First in queue for Reserved
q_chat_o	Cha	1.5 A	0	0	0	0		No	0	0		OIDX	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

See the **Auth. Name** field. If you want the queue to accept only secure chat requests, you should fill it in with the connection name (**OIDC** in our case)

3. In the Administration Portal, go to "Admin → Services → Chat → {Configuration} → Secure Chat"

Secure Chat			
PARAMETER		INHERITED	VALUE
Authentication - Type Name [authConnectionName] ?		<input type="checkbox"/>	oidc
Authentication - Mapping [authMapping] ?		<input type="checkbox"/>	+
Key name	Map Type NickName	Description Name	
Key email	Map Type ChatId	Description E-mail	
Key photo	Map Type Variable	Description Avatar	

authConnectionName - connection name, should be the same as in step 2 above.

authMapping - data mapping. The key is the first-level properties coming from the user info endpoint (described in step 1).