

## Secure Chat

Secure chat enables you to optionally set up your chat with a secure identity method through our partner Signicat - <https://www.signicat.com/>. Relevant methods are usually related to national IDs (Norwegian BankID, Swedish BankID, Danish NemID etc.), and a typical case scenario is having a web portal that requires authentication with National ID, and after you have been verified, a chat is started form within the site. Puzzel can then utilise the National ID verification that already has been issued and pass the information in the chat dialogue towards the contact centre agent that will handle the chat enquiry.

## Prerequisites

In order to use Puzzel Secure Chat, your identity method must be handled thru our partner Signicat - <https://www.signicat.com/identity-methods>.

## How to set up Secure Chat

To enable secure chat with your country's national ID method through Signicat, the following configuration must be set:

```
authConnectionName: 'Signicat',  
authClientId: 'puzzelchatssodev',  
authEndpoint: 'https://preprod.signicat.com/oidc',  
authMapping: { }
```

authClientId is a client Id given to you from Signicat  
authEndpoint is the endpoint to your authClientId open ID Connect

authMapping is a mapping between claims coming from Signicat and what is displayed towards the Agent i.e.:

```
authMapping: {  
  "AuthProvider": {  
    "MapType": "Variable",  
    "Description": "Authentication"  
  },  
  "AuthUserId": {  
    "MapType": "ChatId",  
    "Description": "Name"  
  },  
  "national_id": {  
    "MapType": "Variable",  
    "Description": "Personnummer"  
  },  
  "given_name": {  
    "MapType": "NickName"  
  },  
  "name": {  
    "MapType": "Variable",  
    "Description": "Name"  
  },  
  "family_name": {  
    "MapType": "Variable",  
    "Description": "Family name"  
  },  
  "birthdate": {  
    "MapType": "Variable",  
    "Description": "Fødselsdato"  
  },  
  "sub": {  
    "MapType": "Variable",  
    "Description": "Sub"  
  },  
  "locale": {  
    "MapType": "Variable",  
    "Description": "Locale"  
  },  
  "picture": {  
    "MapType": "Variable",
```

```
"Description": "Picture"  
},  
"address": {  
  "MapType": "Variable",  
  "Description": "Address"  
},  
"phone_number": {  
  "MapType": "Variable",  
  "Description": "phone number"  
},  
"email": {  
  "MapType": "Variable",  
  "Description": "email"  
}  
}
```

See <http://demo.puzzel.com/chat/signicat/index.html> for code example and to simulate a customer login.

#### Note

The relevant queue must be defined as secure in Puzzel Admin, under “Services -> Services -> {service} -> Queue -> Auth0 Connection”. A Puzzel administrator should assist you with this.

Also, a “Signicat Client Secret” must be set to access the claims given on your client id from Signicat. This is an encrypted property found in Puzzel Admin under “Users -> Products -> Chat -> Signicat Client Secret”. It is not possible to see this encrypted property through Puzzel Admin, only set it.

## Flow Chart of Signicat Web Server State (BankID used as example)

- User logs into customer website with BankID through Signicat and surfs around on the customer internal pages.
- When he starts a chat, the browser window with the chat is redirected towards Signicat (on a federated client ID with Puzzel and Customer) for verification.
- Since the user is already authenticated, he is immediately redirected further to Puzzel and then again back to the web page he came from with an authentication code. This will be like blinks for the user, but the page will be redirected a couple of times.
- If the redirect is a problem for the customer page, the chat should be started in its own window.
- When the user tries to enter a secure queue for the first time, Puzzel uses the authentication code and “Signicat Client Secret” stored in Puzzel Admin to get the claims from Signicat.
- The claims are added to the chat as variables, chatId or nickname based on configuration of authMapping. NationalId will be masked, last 5 digits will be replaced by \*.

