

Teamcenter & OneLogin MFA

Briefly: Here, we will try to integrate two-factor authentication based on OneLogin with Teamcenter.

Contents

Introduction.....	2
Prerequisites.....	3
Steps on the Teamcenter Infrastructure Side.....	4
Steps on the OneLogin service side	15
How can I check all this?!	22

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

Introduction

Here we will try to implement two-factor authentication based on OneLogin for Teamcenter. Guys, be prepared that this document might not work in your specific conditions, as there are too many uncertainties. However, I hope that it will provide the right directions for your research.

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

Prerequisites

- Teamcenter 13.3 (TcSS 13.3) + AWC 6
- Tomcat 9.0+
- Single server for all of TC components
- Hostname: *TC133AWC6ZQL
- All software installed in C:\Siemens
- OneLogin (onelogin.com) dev account:
 - *sdistcrus-dev.onelogin.com
- Will be used OpenID (OIDC) as technology for provide authorize data
- We will not use LDAP catalog on-premise!

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

Steps on the Teamcenter Infrastructure Side

1. Generating self-signed certificates - <https://www.selfsignedcertificate.com/>

Self-Signed Certificate Generator

Development Tips

About

Self-Signed Certificate Generator

Self-signed ssl certificates can be used to set up temporary ssl servers. You can use it for test and development servers where security is not a big concern. Use the form below to generate a self-signed ssl certificate and key.

Server name: [Generate >>](#)

1

About SSL Certificates

SSL certificates are required in order to run web sites using the HTTPS protocol. For professional web sites, you usually buy such a certificate from Verisign, Thawte or any other ssl certificate vendor. SSL certificates use a chain of trust, where each certificate is signed

2

MySQL Performance problems?
Download [Jet Profiler for MySQL](#) and speed up your database.

Certificate generated

Your self-signed certificate has been generated. Download the files below and store in a folder reachable by the web server, for example /etc/apache2/ssl.

» [tc133awc6zql.key](#)

» [tc133awc6zql.cert](#)

Download that

Apache Configuration

You need the following ssl configuration in your VirtualHost:

```
<VirtualHost tc133awc6zql:443>
    ServerName tc133awc6zql

    SSLEngine on
    SSLCertificateKeyFile /etc/apache2/ssl/tc133awc6zql.key
    SSLCertificateFile /etc/apache2/ssl/tc133awc6zql.cert
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown

```

MySQL Performance problems?

Download [Jet Profiler for MySQL](#) and speed up your database.

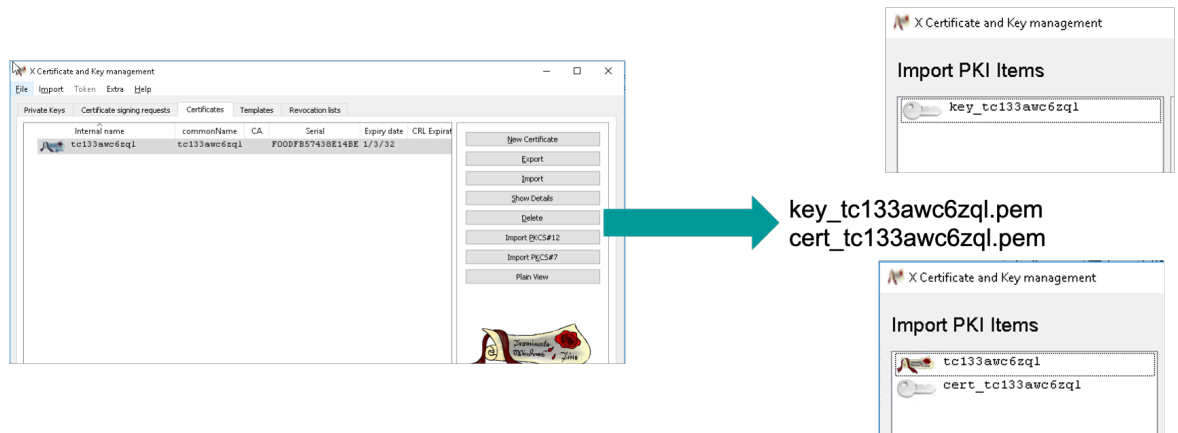
Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

2. Convert key and cert to PEM format

- Download <https://hohnstaedt.de/xca>
- Via XCA application, convert cert and key to PEM files



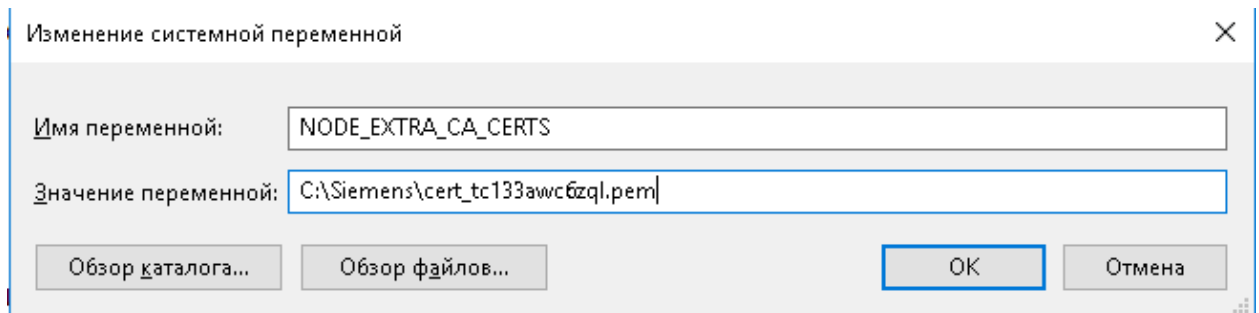
3. Move the:

key_tc133awc6zql.pem

cert_tc133awc6zql.pem

files to C:\Siemens and c:\Siemens\Tomcat9SSO\conf\

Create the next system variable (it needs for AWC)



Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

4. Configuring AWC for access via HTTPS

Open **config.json** (c:\Siemens\Teamcenter13\microservices\gateway-1.6.0\config.json)

```
},
"sso": {
  "tcSSOAppID": "AWCSSO",
  "tcSSOURL": "https://TC133AWC6ZQL:8001/LoginService",
  "tcSSORedirectMethod": "GET",
  "tcSSOLogoutURL": "",
  "proxyServerUrl": "",
  "queryParams": ""
},
"fms": {
  "bootstrapFSCURLs": [  ],

},
"csrf": {
  "cookie": true
},
"csrfExclusions": [
  "/AWSSOLogin",
  "/graphql",
  "/LoginService/sso_login_callback",
  "siemensdisrus.okta.com",
  "/siemensdisrus.okta.com"
],
"userAgentWhiteList": [
  "Apache-HttpClient",
```

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

5. Configuring Tomcat for deploying TcSS components

Open server.xml (c:\Siemens\Tomcat9SSO\conf\server.xml)

```
<Connector port="8002" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

```
<Connector
  protocol="org.apache.coyote.http11.Http11NioProtocol"
  port="8001" maxThreads="200"
  scheme="https" secure="true" SSLEnabled="true"
  SSLCertificateFile="conf/cert tc133awc6zql.pem"
  SSLCertificateKeyFile="conf/key tc133awc6zql.pem"
  SSLVerifyClient="optional" SSLProtocol="TLSv1+TLSv1.1+TLSv1.2" useIPVHosts="true" address="0.0.0.0" />
```

Authorship: <https://www.linkedin.com/in/sedoykin>

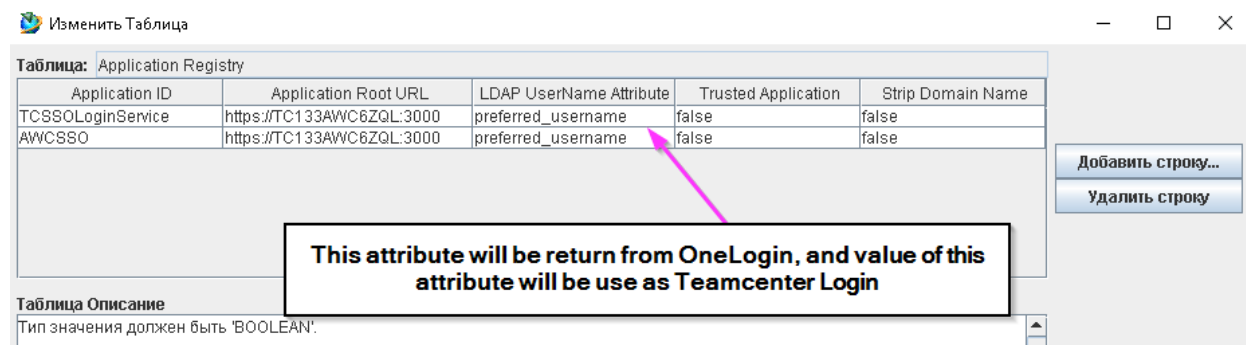
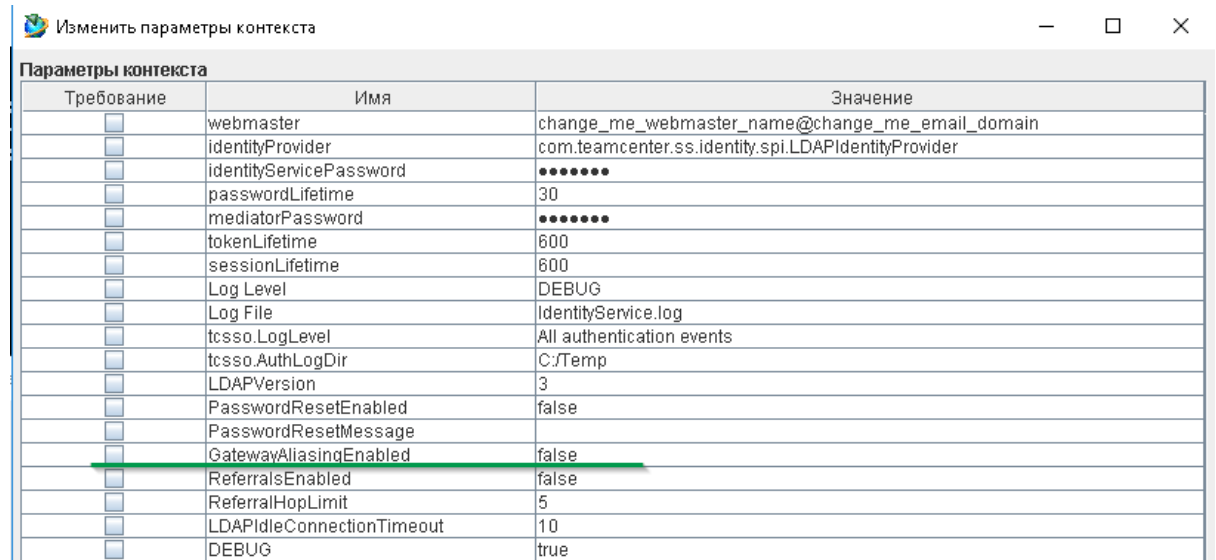
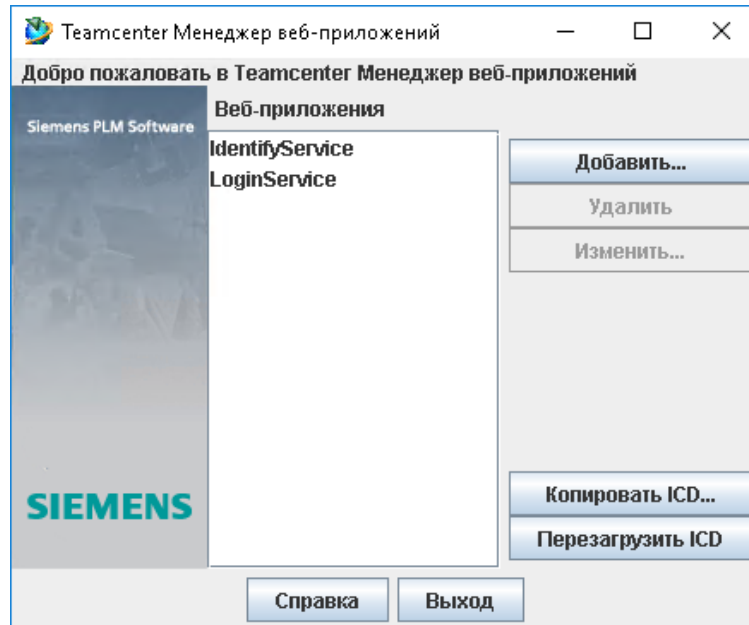
Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

6. Creating Login and Identify Services

Ohhh, for details go to: [https://docs.sw.siemens.com/en-](https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20210421143201885.tss00001/html/xid373993)

[US/product/282219420/doc/PL20210421143201885.tss00001/html/xid373993](https://docs.sw.siemens.com/en-US/product/282219420/doc/PL20210421143201885.tss00001/html/xid373993)

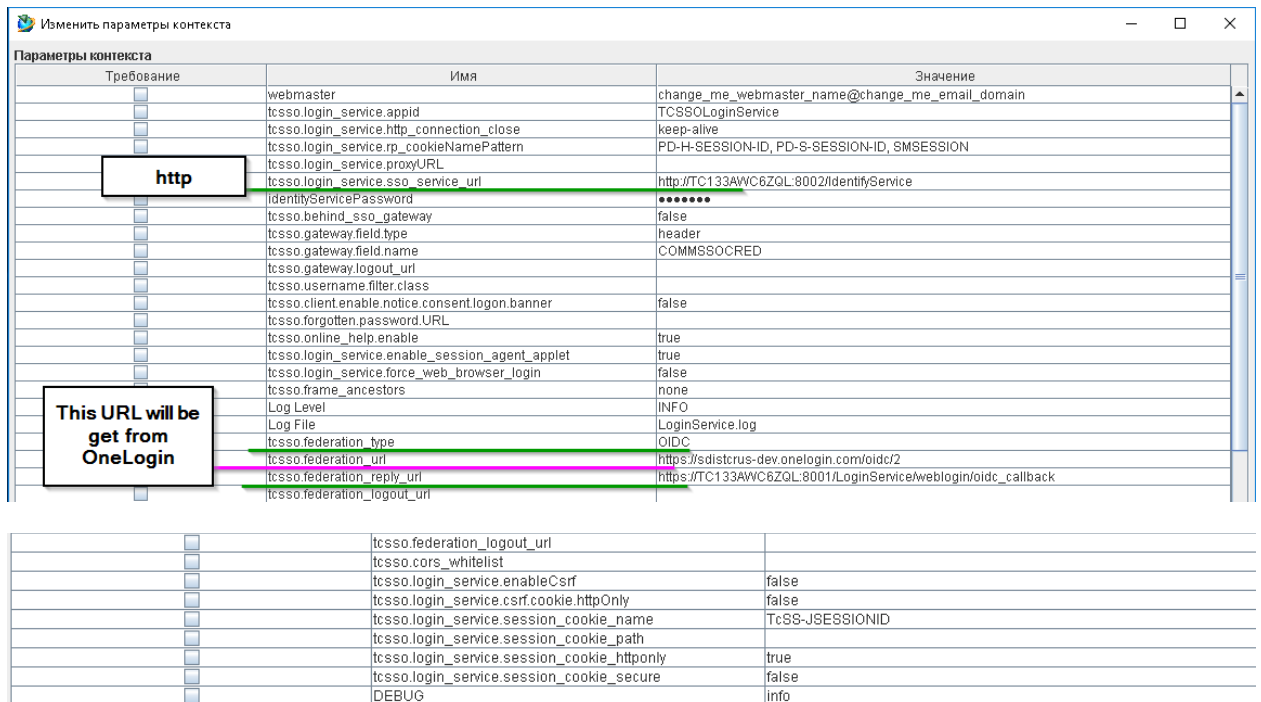
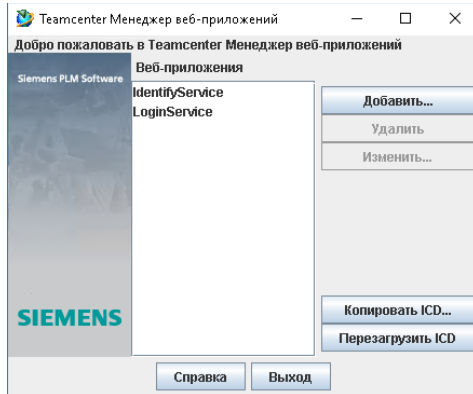


Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

7. Creating Login and Identify Services



Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

8. Configuring federation.properties (1)

Open **federation.properties**(c:\Siemens\SSOSrv\LoginService\webapp_root\WEB-INF\classes**federation.properties**)

```
88 # OIDC properties
89 #
90
91 # This value is obtained from the OpenID Provider when TcSS is registered
92 # as an OIDC Client
93 tcso.oidc.client_id=b3[REDACTED]7
94
95 # Only required if tcso.oidc.client_auth_method=client_secret_basic
96 # This value is obtained from the OpenID Provider when TcSS is registered
97 # as an OIDC Client
98 # WARNING: Sensitive value
99 tcso.oidc.client_secret=6bc[REDACTED]ff487
100
101 # The value of the authorization_endpoint from the OpenID Provider
102 tcso.oidc.auth_endpoint=https://sdistcrus-dev.onelogin.com/oidc/2/auth
103 # The value of the token_endpoint from the OpenID Provider
104 tcso.oidc.token_endpoint=https://sdistcrus-dev.onelogin.com/oidc/2/token
105 # The value of the jwks_uri from the OpenID Provider
106 tcso.oidc.jwks_endpoint=https://sdistcrus-dev.onelogin.com/oidc/2/certs
107
108 # Specifies the name of the OIDC ID Token claim that will be used as the
109 # Teamcenter SSO User ID.
110 # Note: The value of the tcso.oidc.scope above will affect which claims are
111 # returned in the ID Token. Make sure the correct scope(s) are configured to
112 # include this desired claim
113 tcso.oidc.userid_claim=preferred_username
114
115 # The value of the scope that will be included in the OIDC Authentication Request
116 tcso.oidc.scope=openid profile
117
118 # Specifies the algorithm that will be used to verify the signature of the OIDC ID Token
119 tcso.oidc.token_sig_alg=RS256
120
121 # Specifies the value of the OIDC display parameter sent in the Authentication Request - page default
122 tcso.oidc.display=page
```

These data will be get from OneLogin

This attribute will be return from OneLogin, and it uses in IdentifyService (Application Registry)

openid - is mandatory scope, profile - scope has preferred_username attribute

9. Configuring federation.properties (2)

```
125 # OIDC properties - Advanced Configurations
126 #
127
128 # The Authentication method that will be used when sending a Token Request to the
129 # OpenID Provider
130 # Currently supported methods:
131 # - client_secret_basic
132 # - private_key_jwt (This method requires that the following tcso.oidc.signing..
133 #   properties be set and that TcSS is registered at the OpenID Provider to use
134 #   private_key_jwt for Token Endpoint authentication)
135 # Note that the OpenID Provider may not support all the above methods
136 tcso.oidc.client_auth_method=client_secret_basic
137
138 # The following four tcso.oidc.signing.. properties are only required if the
139 # private_key_jwt Token Endpoint authentication method is selected above.
140 #
141 # A JKS formatted keystore containing the public key that should be passed to the
142 # OpenID Provider to validate the signed JWT sent with a Token Request as well as
143 # the corresponding private key that we will use to sign the JWT.
144 tcso.oidc.signing_jks_file=</secure/path/oidc_signing_key.jks>
145 # WARNING: Sensitive value
146 tcso.oidc.signing_jks_file_pwd=<password>
147 # The name of the private key in the keystore used to sign the JWT
148 tcso.oidc.signing_private_key_name=<key name>
149 # WARNING: Sensitive value
150 tcso.oidc.signing_private_key_pwd=<password>
151
152 # If private_key_jwt Client Authentication is selected, specifies the signing
153 # algorithm used to sign the jwt sent to the OpenID Provider
154 tcso.oidc.jwt.sig_alg=RS256
155 # If private_key_jwt Client Authentication is selected, specifies the expiration
156 # time (in seconds) of the jwt sent to the OpenID Provider
157 tcso.oidc.jwt.expiration=60
```

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

10. Configuring federation.properties (3)

```
158
159 #
160 # The following four tcsso.oidc.encryption.. properties are only required if ID
161 # Token encryption is supported and enabled at the OpenID Provider.
162 #
163 # A JKS formatted keystore containing the public key that should be passed to the
164 # OpenID Provider to encrypt an OIDC ID Token, and the corresponding private key
165 # that we will use to decrypt the OIDC ID Token.
166 tcsso.oidc.encryption_jks_file=</secure/path/oidc_encrypt_key.jks>
167 # WARNING: Sensitive value
168 tcsso.oidc.encryption_jks_file_pwd=<password>
169 # The name of the private key in the keystore used to decrypt the OIDC ID Token
170 tcsso.oidc.encryption_private_key_name=<key name>
171 # WARNING: Sensitive value
172 tcsso.oidc.encryption_private_key_pwd=<password>
173
174 # If true, a jwks_uri will be exposed at the <LoginService>/certs endpoint.
175 # Public keys that are configured above will be converted to JWK format and
176 # returned from that endpoint.
177 tcsso.oidc.enable_jwks_uri=false
```

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

11. Configuring Teamcenter core

Open `tc_profilevars.bat` (C:\Siemens\tcdata\tc_profilevars.bat)

```
24 @rem TC_LOG_VOLUME_DIR=
25 @rem TC_LOG_VOLUME_NAME=
26 @rem set TCI=
27 @rem set TC_ZEUS_ACCESS_KEY_ID=
28 @rem set TC_ZEUS_ACCESS_PASSWORD_FILE=
29 set TC_SSO_LOGIN_URL=http://TC133AWC6ZQL:8002/LoginService
30 set TC_SSO_SERVICE=http://TC133AWC6ZQL:8002/IdentifyService
31 set TC_SSO_APP_ID=AWCSSO
32
33 if not defined TEMINSTALL goto :SKIP_TEM_SETTINGS
34
35 set TC_BIN=%TC_ROOT%\bin
36 set TC_PERL=%TC_ROOT%\perl\bin\perl.exe
37 set TC_PERL_BIN=%TC_ROOT%\perl\bin
38 set TC_INCLUDE=%TC_ROOT%\include
39 set TC_LIBRARY=%TC_ROOT%\lib
```

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

12. Deploying TcSS services

Copy these files:

c:\Siemens\SSOSrv\IdentifyService\deployment**IdentifyService.war**

c:\Siemens\SSOSrv>LoginService\deployment**LoginService.war**

to:

c:\Siemens\Tomcat9SSO**webapps**

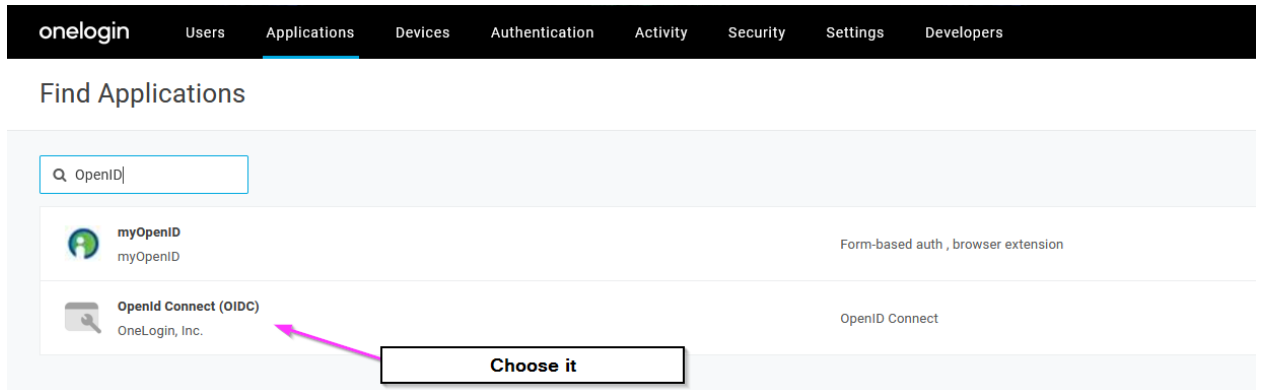
Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

Steps on the OneLogin service side

1. **You should have dev account, it is free for developing purpose** after get it you should create new OpenID integration application



Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

2. Configuring OpenId Connect (OIDC)

[Applications /](#)

OpenId Connect (OIDC)

The screenshot shows the 'Info' tab of the OpenId Connect (OIDC) configuration page. The left sidebar contains a menu with 'Info' selected, followed by 'Configuration', 'Parameters', 'Rules', 'SSO', 'Access', 'Users', 'Privileges', and 'Setup'. The main content area is titled 'Portal' and includes the following fields:

- Display Name:** Teamcenter_tcmfa.sisw.ru (OIDC)
- Tab:** sdistrus
- Visible in portal:** A green toggle switch is turned on.
- Rectangular Icon:** A placeholder image with a checkered background and a wrench icon. Below it, a note says: 'Upload an icon with an aspect-ratio of 2.64:1 as either a transparent .PNG or .SVG'.
- Square Icon:** A placeholder image with a checkered background and a wrench icon. Below it, a note says: 'Upload a square icon at least 512x512px as either a transparent .PNG or .SVG'.

A callout box with the text 'Some name, it does not matter' points to the 'Display Name' field.

[Applications /](#)

OpenId Connect (OIDC)

The screenshot shows the 'Configuration' tab of the OpenId Connect (OIDC) configuration page. The left sidebar is the same as the previous screenshot, with 'Configuration' selected. The main content area is titled 'Application details' and includes the following fields:

- Login Url:** An empty text field.
- Redirect URI's:** A text field containing 'https://TC133AWC6ZQL:8001/LoginService/weblogin/oidc_callback'. A callout box with the text 'See slide 11' points to this field.
- Post Logout Redirect URIs:** A text field containing 'https://TC133AWC6ZQL:3000'. A callout box with the text 'AWC URL' points to this field.

A note below the 'Redirect URI's' field states: 'After the user is authenticated we only allow redirects back to entries on this comma (or new-line) separated list of urls, and HTTPS is required. http://localhost is permitted for development purposes only and should not be used in production.'

[Applications /](#)

OpenId Connect (OIDC)

The screenshot shows the 'SSO' tab of the OpenId Connect (OIDC) configuration page. The left sidebar is the same as the previous screenshots, with 'SSO' selected. The main content area is titled 'Enable OpenID Connect' and includes the following fields:

- Client ID:** A text field containing a long alphanumeric string. A callout box with the text 'See slide 12' points to this field.
- Client Secret:** A text field containing a long alphanumeric string. A callout box with the text 'See slide 12' points to this field. There are links for 'Hide client secret' and 'Regenerate client secret'.
- Issuer URL:** A text field containing 'https://sdistrus-dev.onelogin.com/oidc/2 Well-known Configuration'. A callout box with the text 'See slide 11 This is federation URL' points to this field.
- Application Type:** A dropdown menu with 'Web' selected.
- Token Endpoint:** A dropdown menu with 'Basic' selected.

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

JSON configuration for <https://sdistcrus-dev.onelogin.com/oidc/2/>:

```
{  "acr_values_supported": [    "oneLogin:nist:level:1:re-auth"  ],  "authorization_endpoint": "https://sdistcrus-dev.onelogin.com/oidc/2/auth",  "claims_parameter_supported": true,  "claims_supported": [    "sub",    "email",    "name",    "family_name",    "given_name",    "middle_name",    "nickname",    "preferred_username",    "profile",    "picture",    "birthdate",    "gender",    "zoneinfo",    "locale",    "phone_number",    "address",    "role",    "groups",    "email_verified",    "picture_verified",    "profile_verified",    "phone_number_verified",    "address_verified",    "role_verified",    "groups_verified",    "email_verified",    "picture_verified",    "profile_verified",    "phone_number_verified",    "address_verified",    "role_verified",    "groups_verified"  ],  "grant_types_supported": [    "authorization_code",    "implicit",    "password",    "refresh_token"  ],  "id_token_signing_alg_values_supported": [    "RS256",    "RS384",    "RS512",    "ES256",    "ES384",    "ES512",    "PS256",    "PS384",    "PS512"  ],  "issuer": "https://sdistcrus-dev.onelogin.com/oidc/2",  "jwks_uri": "https://sdistcrus-dev.onelogin.com/oidc/2/certs",  "registration_endpoint": "https://sdistcrus-dev.onelogin.com/oidc/2/register",  "request_parameter_supported": false,  "request_uri_parameter_supported": false,  "response_modes_supported": [    "query",    "fragment"  ],  "response_types_supported": [    "code",    "token",    "id_token",    "id_token token",    "code token",    "code id_token",    "code id_token token",    "token id_token",    "token id_token token",    "code id_token token"  ],  "scopes_supported": [    "openid",    "email",    "profile",    "phone",    "address",    "role",    "groups"  ],  "subject_types_supported": [    "public"  ],  "token_endpoint": "https://sdistcrus-dev.onelogin.com/oidc/2/token",  "token_endpoint_auth_methods_supported": [    "client_secret_basic",    "client_secret_post",    "none"  ],  "userinfo_endpoint": "https://sdistcrus-dev.onelogin.com/oidc/2/me",  "userinfo_signing_alg_values_supported": [    "RS256",    "RS384",    "RS512",    "ES256",    "ES384",    "ES512",    "PS256",    "PS384",    "PS512"  ],  "code_challenge_methods_supported": [    "S256"  ],  "introspection_endpoint": "https://sdistcrus-dev.onelogin.com/oidc/2/token/introspection"}
```

Values and their corresponding fields:

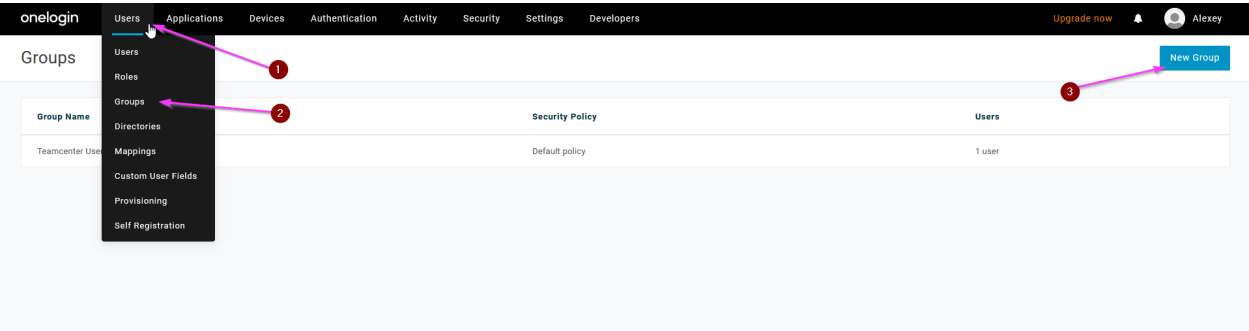
- 100: `authorization_endpoint`
- 102: `token_endpoint`
- 104: `token_endpoint`
- 105: `jwks_uri`
- 107: `introspection_endpoint`

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

3. Creating User Group in OneLogin account

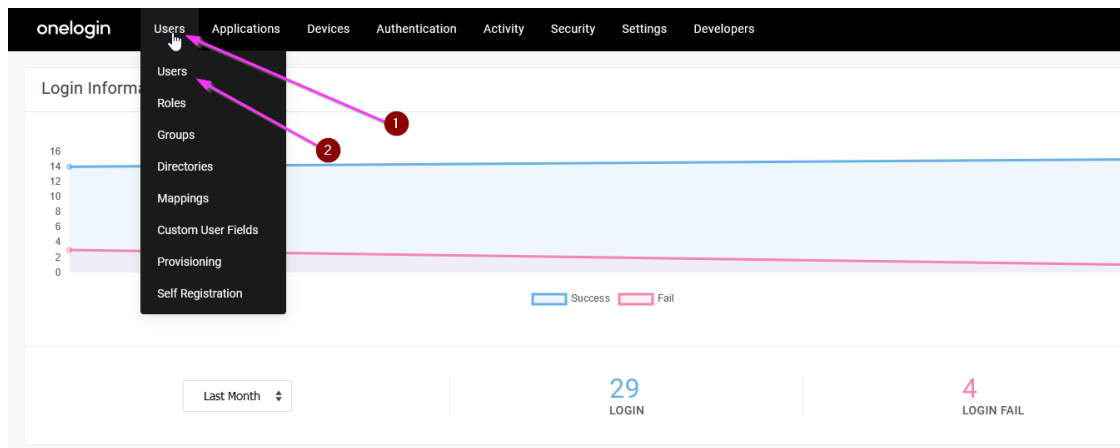


Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

4. Adding and configuring users at OneLogin account



The user profile page for 'Ivan Ivanov' shows a sidebar with 'User Info' selected. The main area contains a form with fields for: First name (Ivan), Last name (Ivanov), Email (ivan@gmail.com), Username (ivan), Phone number (+79175366761), Manager (Alexey Sedoykin), Company, Department, and Title. A callout box points to the 'Username' field with the text: 'It will be presented as preferred_username in OpenID response, and it will be use like a Teamcenter Login'.

The user profile page for 'Ivan Ivanov' shows the 'Authentication' tab selected in the sidebar. The main area contains a form with fields for: Group (Teamcenter Users), Trusted IDP (NONE), Authenticated by (OneLogin), User security policy (group policy is Default policy), and Open ID (ivan). A callout box points to the 'Group' field with the text: 'It will be presented as preferred_username in OpenID response, and it will be use like a Teamcenter Login'.

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

The screenshot displays the Okta user management interface for a user named Ivan Ivanov. The interface is divided into several sections:

- User Info:** Located at the top left, it includes fields for User Info, Authentication, and Applications. A red circle with the number 1 is placed next to the Applications link.
- Roles:** Located in the middle left, it shows a list of roles with a green checkmark next to the 'Default' role. A red circle with the number 2 is placed next to the 'Default' role.
- Applications:** Located on the right, it shows a list of applications. A red circle with the number 3 is placed next to the 'Teamcenter_tcmfa.sisw.ru (OIDC)' application. A red circle with the number 4 is placed next to the 'Admin-configured' application. A red circle with the number 5 is placed next to the 'More Actions' dropdown menu.

Numbered callouts (1-5) are present, indicating specific areas of interest or steps in a process:

- 1: Points to the 'Applications' link in the left sidebar.
- 2: Points to the 'Default' role in the 'Roles' section.
- 3: Points to the 'Teamcenter_tcmfa.sisw.ru (OIDC)' application in the 'Applications' list.
- 4: Points to the 'Admin-configured' application in the 'Applications' list.
- 5: Points to the 'More Actions' dropdown menu in the top right corner.

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

5. Adding MFA feature

onelogin Users Applications Devices Authentication Activity Security Settings Developers Upgrade now Alexey

Authentication Factors

Authentication Factors

Factor

OneLogin Protect

Display Name

Users

OneLogin Protect

1 User

New Auth Factor Save

Policies

Type Policy Name

User Default policy

New User Policy New App Policy

Policies / 301442

Default policy

Sign In

Password

Account Recovery

Session

MFA

IP Addresses

Customization

Require trusted device

☐ Device Trust Required

☒ Allow self-installation

Certificate expires in

1 Year

One-time passwords

☒ OTP Auth Required

Available factors:

☒ OneLogin Protect

Note: Changes to OTP factors won't affect Password Update settings.

Phone number for SMS

☒ Allow user to change phone number for OneLogin SMS

User can change phone number used for SMS verification from their profile. (Note that a directory mapping will overwrite a user-entered phone nu

Policies / 301442

Default policy

Sign In

Password

Account Recovery

Session

MFA

IP Addresses

Customization

Enter one or more IP addresses or ranges separated by spaces, for example 192.128.20.14 192.130.1.1-192.130.1.40

☒ Ignore X-Forwarded-For header IP addresses

If checked, OneLogin will only evaluate the gateway IP address and ignore the IP addresses from the X-Forwarded-For header.

☐ MFA Bypass for Trusted Device

If checked, OneLogin will bypass MFA when an user is authenticating from a trusted device.

Enforcement Settings

OTP required for

All users

OTP required at

At every login

Security cookie expiration

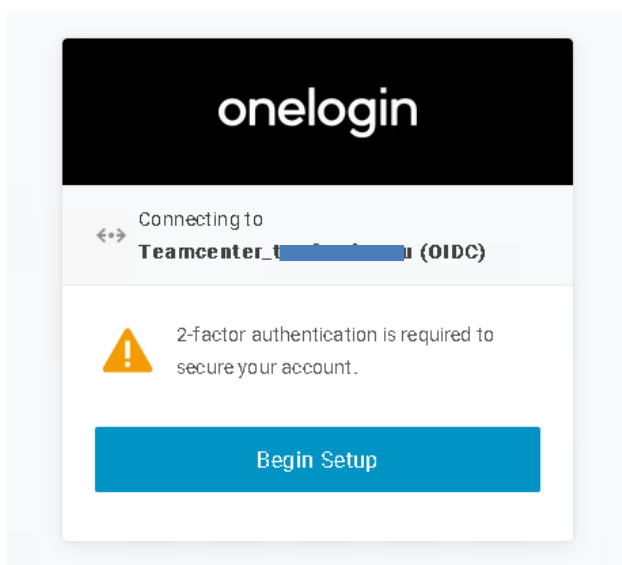
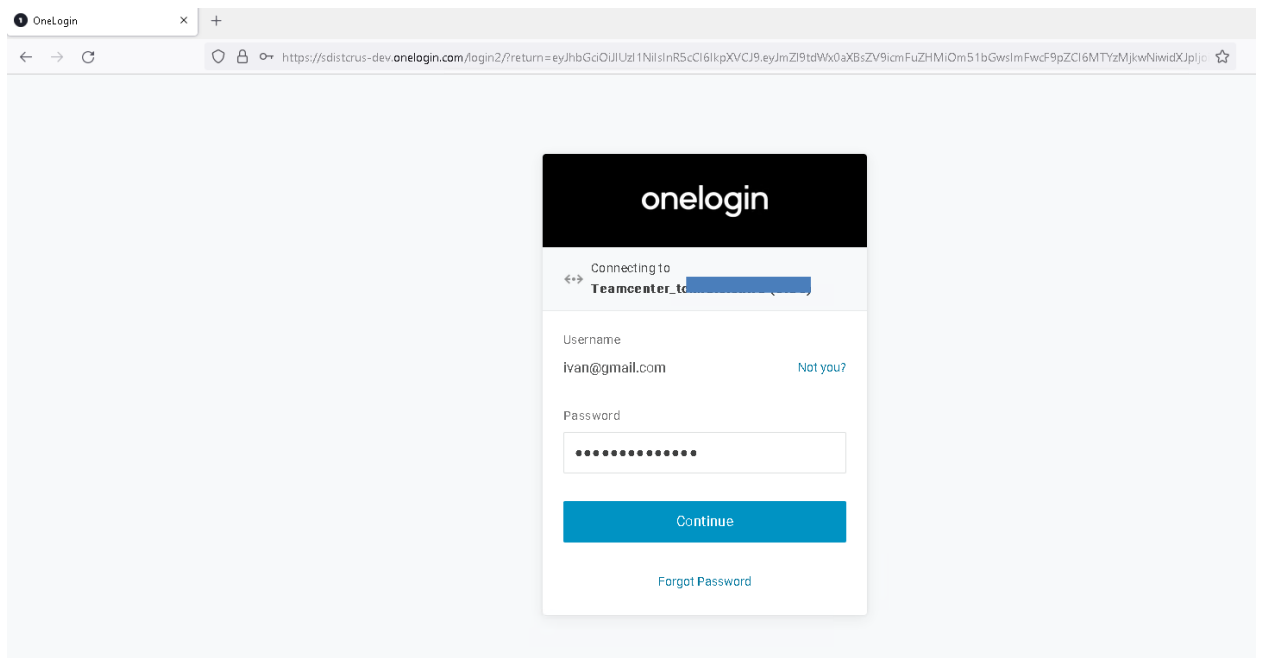
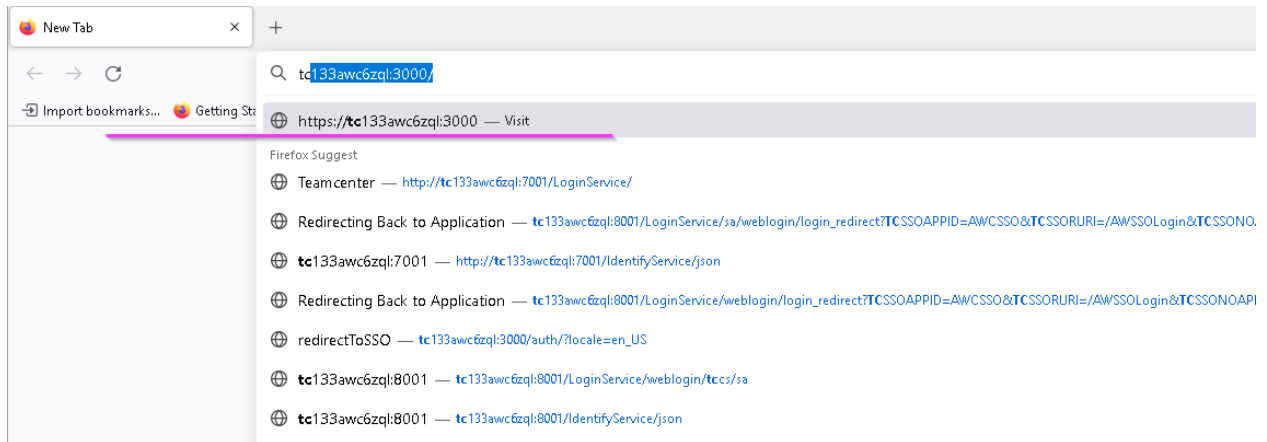
days

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

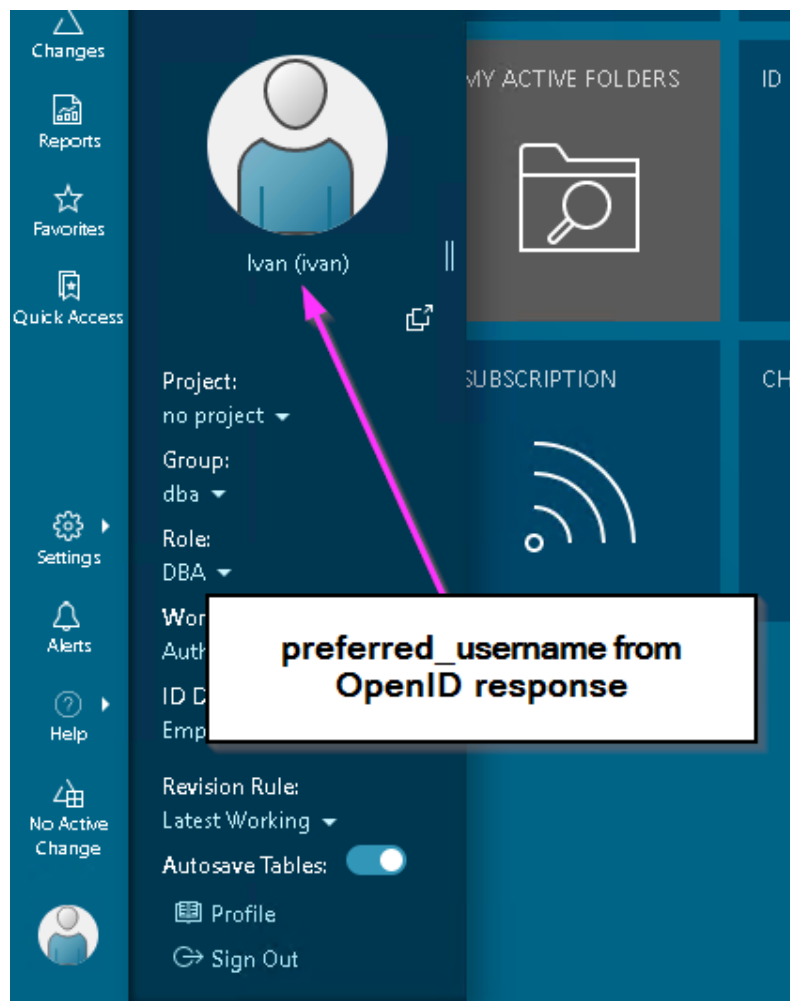
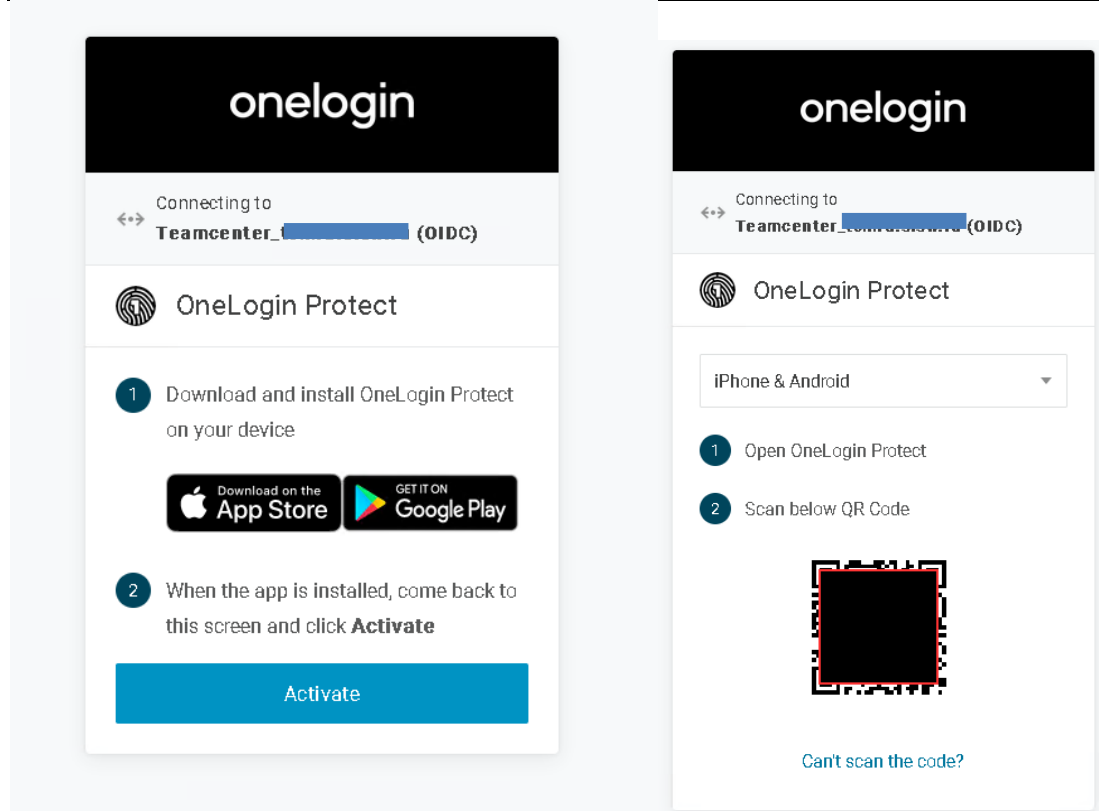
How can I check all this?!



Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!



THAT'S ALL, gOod LUCK!

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!

Authorship: <https://www.linkedin.com/in/sedoykin>

Disclaimer:

No guarantees or responsibilities are provided. You perform all actions at your own risk!
