

File name	OpenID SSO
Author	YouTestMe
Confidentiality	Public
Last save date	Thursday, May-26-2022 at 5:08:00 PM

Table of Contents

1	Introduction	1
2	Setting up the OpenID Connect integration	1
2.1	How to debug attribute mappings	6
2.2	Sign in with OpenID	8

1 Introduction

This article explains how to set up the OpenID Connect integration, an authentication, and authorization scheme that allows your users to log in with an external system's ID.

2 Setting up the OpenID Connect integration

As an example, we will use Microsoft Azure AD as the OpenID Connect Identity Provider. Some basic information regarding the OpenID Connect integration can be found on the following link:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-v2-protocols>.

You need to register your application on the Identity Provider (IP) side. The procedure can be found on the following link: <https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app>.

On the YouTestMe (YTM) side, you need to set up the OpenID configuration.

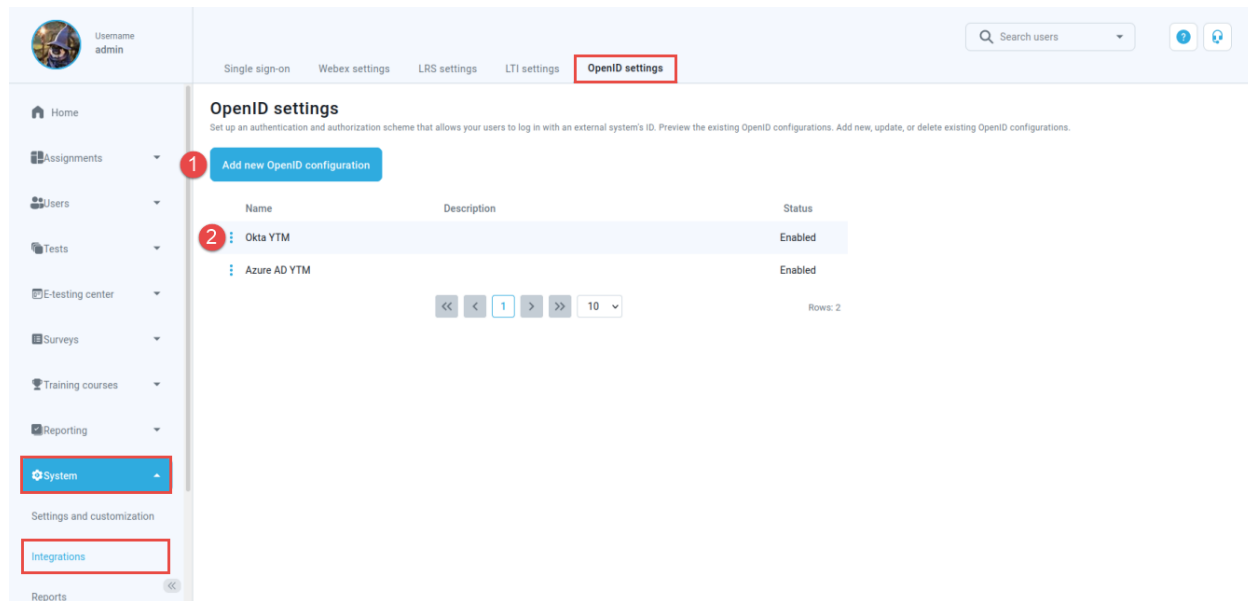
To access the OpenID configuration panel, navigate to the „**System**“ section in the main menu, then select „**Integrations**“. Then, choose the "**OpenID Settings**" tab.

All the existing OpenID configurations (if any) are displayed in a table with their names, descriptions, statuses, and available actions.

1. By clicking the "**Add new OpenID configuration**" button, a new pop-up window will appear where you can enter the parameters of a new OpenID configuration (*For more information on how to add a new configuration, please read instructions below*).

- By clicking the „Edit“ option from the three vertical dots icon, a pop-up window (with the same fields as the one for adding a new configuration) will appear. There you can change the parameters of the selected OpenID configuration.

An OpenID configuration can be deleted by clicking the **trash can** icon from the three vertical dots icon.



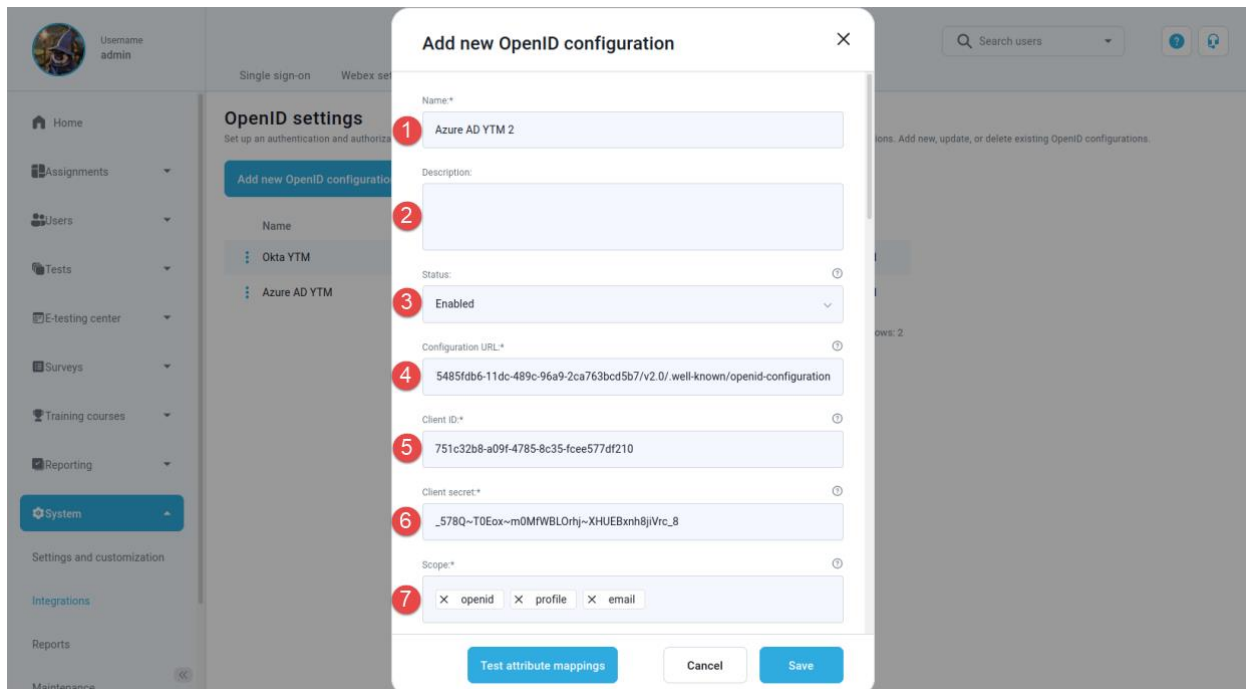
When creating a new or updating an existing OpenID configuration, you have to enter the parameters for configuring the client (YTM web app) and IP.

For configuring the IP side, the Global administrator role is required. For configuring the YTM side, the Administrator role is required.

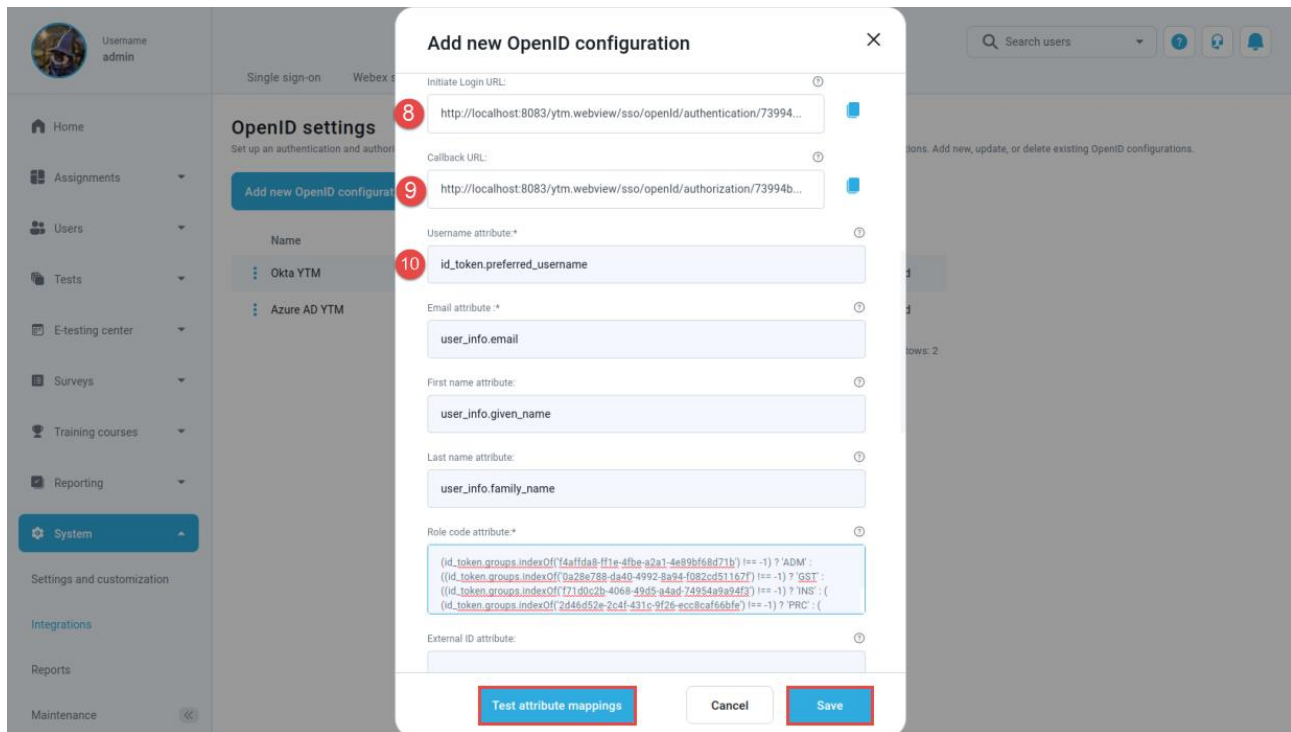
The steps are:

- Enter the desired name of the OpenID configuration.
- Optionally enter a description of the OpenID configuration.
- Choose the status of the OpenID configuration. If the status is "**Enabled**," users will be able to sign in with the IP specified in the configuration. Otherwise, they will not be able to do so.
- Insert the "**Configuration URL**" - a URL where the configuration of the OpenID IP can be found. For example, authorization endpoint URL, token endpoint URL, supported scopes, etc. The OpenID IP provides this parameter under **Azure Active Directory > App registrations > <YOUR-APPLICATION> > Endpoints > OpenID Connect metadata document**.
- Insert the "**Client ID**" - a public identifier for the client (YTM). The OpenID IP provides it under "**Application (client) ID**" when registering the application. [Client ID link](#)
- Insert the "**Client secret**" - a value the client uses to exchange an authorization code for a token. The OpenID IP provides it when registering the application. For the OpenID IP, you can generate this under <YOUR-APPLICATION> > **Certificates & secrets > New client secret**. [Client secret link](#)
- Choose the "**Scope**" to specify what access privileges are requested as part of the authorization. On the IP side, scopes are managed under **Azure Active Directory > App registrations > <YOUR-APPLICATION> > API Permissions**.

Select **Add a permission > Microsoft Graph > Delegated permissions** to add a new scope. The **"openid"** scope is mandatory, and the rest are optional. Usually, **"profile"** and **"email"** are included as well.



8. The **"Initiate Login URL"** parameter is automatically generated in the configuration by YTM. This is the URL of the YTM endpoint that initiates the sign-in flow upon receiving a request. When the OpenID IP redirects to this endpoint, the client is triggered to send an authorization request. This parameter is optional, and if used, it is copied from the configuration window and pasted to the appropriate field in the app settings on the IP side.
9. The **"Callback URL"** parameter is automatically generated in the configuration by YTM. This is the URL of the callback endpoint where the OpenID IP redirects the user browser and sends security tokens after authentication. On the IP side, this parameter might be referred to as "redirect URI" as well, and you have to copy it from the configuration window and paste it there (<YOUR-APPLICATION> > **Authentication > Add a platform > Web**). [Redirect URI link](#)



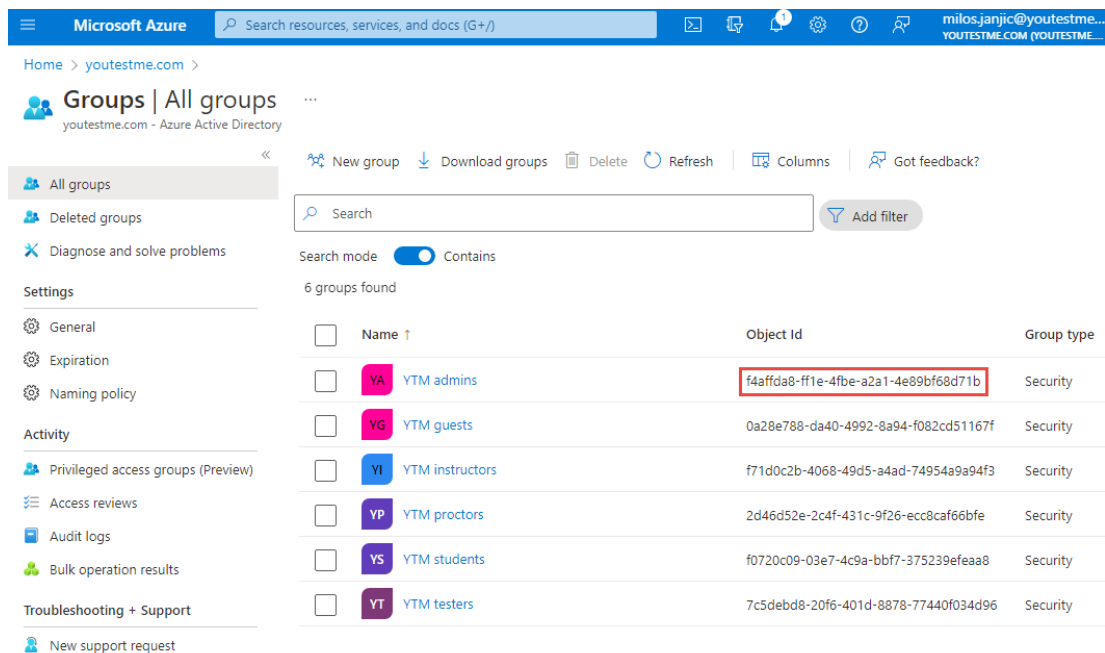
10. Mapping of the Authorization server (IP) claims to the YTM user profile attributes. Claims are name/value pairs that contain information about a user. The claims can be sent using the access token, the ID token, or the user info endpoint data. An [access token](#) is an authorization in the form of a string given to an application to access the protected data of a user. An [ID token](#) is a string that contains information about a user and proves that the user has been authenticated. The [UserInfo endpoint](#) returns claims about an authenticated user.

The mapping from the claims can be specified for each YTM user account attribute. The default values are set for the most important attributes (username, email, first name, last name, role). These values are only suggestions, and they can be changed. Every value represents a JavaScript expression based on which the corresponding attribute is evaluated.

- a. The expression `"id_token.preferred_username"` can be used to obtain the **"Username attribute"** value (assuming that the `"preferred_username"` claim is included in the ID token on the IP side). This means that the result of evaluating `"id_token.preferred_username"` will be mapped to **"Username attribute"**. A username is a unique identifier for the user, based on which it is determined whether the received claims are used to update an existing one or to create a new user.
- b. `"user_info.email"` can be mapped to the **"Email attribute"** (assuming that the data sent by the UserInfo endpoint includes the `"email"` claim).
- c. As a special case of a JavaScript expression, a constant represented by a string value (surrounded by single or double quotes) can be mapped to an attribute. The default value for the **"Role code attribute"** field is 'ATT', representing the Student role. Note that the code value is expected here, not the full role name.

To check the role codes for YTM user roles, navigate to "Users" in the main menu and select "Roles and permissions".

- i. Under the **Azure Active Directory > App registrations > <YOUR-APPLICATION> > Token configuration > Add optional claims**, you can add claims to be sent in access token/id token. Under **Add groups claim**, you can add group claims to the tokens.
- ii. Under **Azure Active Directory > Groups > New Group**, you can add a new group of users. For example, we created several groups corresponding to the user roles in our application and assigned users to these groups. We determine the user role based on the group they belong to. The image below shows the groups we created.



Every group is identified by its "Object Id ", which we use in the JavaScript expression for the "Role code attribute":

```
(id_token.groups.indexOf('f4affda8-ff1e-4fbe-a2a1-4e89bf68d71b') !== -1) ? 'ADM' :
((id_token.groups.indexOf('0a28e788-da40-4992-8a94-f082cd51167f') !== -1) ? 'GST' :
((id_token.groups.indexOf('f71d0c2b-4068-49d5-a4ad-74954a9a94f3') !== -1) ? 'INS' : (
(id_token.groups.indexOf('2d46d52e-2c4f-431c-9f26-ecc8caf66bfe') !== -1) ? 'PRC' : (
(id_token.groups.indexOf('7c5debd8-20f6-401d-8878-77440f034d96') !== -1) ? 'TST' : 'ATT'
))));
```

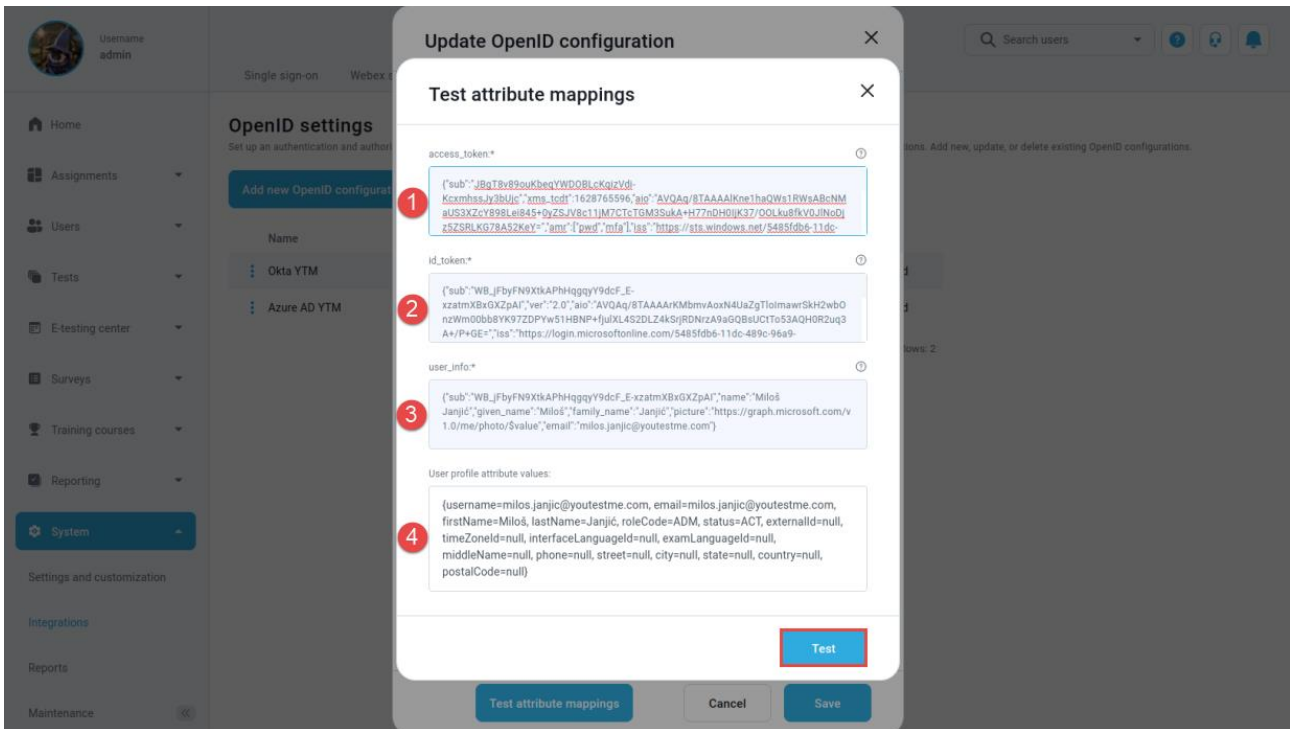
From the example above, a user who belongs to the YTM admins group (in IP side) will have an administrator role in YTM, a user who belongs to the YTM instructors group, an instructor role, etc.

2.1 How to debug attribute mappings

By clicking the "Test attribute mappings" button, a new pop-up window opens, which is used to test the mappings between the IP's claims and YTM user profile attributes. This provides a possibility to manually define strings in JSON format that simulate data received by the IP and examine the values of user profile attributes for that particular case.

When the pop-up window opens:

1. Enter a JSON string that simulates the payload of an access token received by the IP.
2. Enter a JSON string that simulates the payload of an id token received by the IP.
3. Enter a JSON string that simulates the data received as the response to a request to the UserInfo endpoint of the IP.
4. By clicking the "Test" button, the user profile attribute values corresponding to the above IP data are displayed.



Below are examples for access token payload, id token payload, and UserInfo data in JSON format:

- access token payload


```
{
  "sub": "JBgT8v89ouKbeqYWDOBLcKqizVdi-KcxmhssJy3bUjc",
  "xms_tcdt": "1628765596",
  "aio": "AVQAq/8TAAAIKne1haQWw1RWsABcNMUS3XZcy898Lei845+0yZSJV8c11jM7CTcTGM3SukA+H77nDH0iK37/OOLku8fkV0JINoDjz5ZSRLKG78A52KeY=",
  "amr": [
    "pwd",
    "mfa"
  ],
  "iss": "https://sts.windows.net/5485fdb6-11dc-489c-96a9-2ca763bcd5b7/",
  "app_displayname": "GetCertified",
  "signin_state": [
    "kmsi"
  ],
  "oid": "bf62a491-dd2b-4712-acb8-72cc96b74381",
  "platf": "8",
  "tid": "5485fdb6-11dc-489c-96a9-2ca763bcd5b7",
  "acr": "1",
  "puid": "10032001F3B78532",
  "wids": [
    "62e90394-69f5-4237-9190-012177145e10",
    "b79fbf4d-3ef9-4689-8143-76b194e85509"
  ],
  "xms_st": {
    "sub": "WB_jFbyFN9XtkAPhHggqyY9dcF_E-xzatzmXBxGXzpaI",
    "exp": "1651222042",
    "ipaddr": "141.95.241.10",
    "iat": "1651217597",
    "scp": "openid profile email",
    "ver": "1.0",
    "idtyp": "user",
    "uti": "FRt1hRsfNEOsboG8YJYAA",
    "given_name": "Miloš",
    "aud": "00000003-0000-0000-c000-000000000000",
    "tenant_region_scope": "EU",
    "unique_name": "milos.janjic@youtestme.com",
    "upn": "milos.janjic@youtestme.com",
    "nbf": "1651217597"
  }
}
```

```
:1651217597,"appidacr":"1","rh":"0.AS8Atv2FVNwRnEiWqSynY7zVtwMAAAAAAAAAAwAAAAAAAAAAvAKk","appid":"751c32b8-a09f-4785-8c35-fcee577df210","name":"Miloš Janjić","family_name":"Janjić","acct":0}
```

- **id token payload**

```
{ "sub": "WB_jFbyFN9XtkAPhHqgqyY9dcF_E-xzatzmXBxGXZpAl", "ver": "2.0", "aio": "AVQAq/8TAAArKMbmVaoxN4UaZgTlOImawrSkH2wbOnzWm00bb8YK97ZDPYw51HBNP+fjuXL4S2DLZ4kSrijRDNrzA9aGQBsUCtTo53AQH0R2uq3A+/P+GE=", "iss": "https://login.microsoftonline.com/5485fdb6-11dc-489c-96a9-2ca763bcd5b7/v2.0", "groups": ["f4affda8-ff1e-4fbc-a2a1-4e89bf68d71b"], "ae30426b-ef6c-442f-a561-0039784d83ce"}, "oid": "bf62a491-dd2b-4712-acb8-72cc96b74381", "preferred_username": "milos.janjić@youtestme.com", "uti": "FRt1hRsfNEOsOG8YJYYAA", "given_name": "Miloš", "tid": "5485fdb6-11dc-489c-96a9-2ca763bcd5b7", "aud": "751c32b8-a09f-4785-8c35-fcee577df210", "nbf": "1651217597", "rh": "0.AS8Atv2FVNwRnEiWqSynY7zVt7gyHHWfoIVHjDX87ld98hAvAKk", "wids": ["62e90394-69f5-4237-9190-012177145e10", "b79fbf4d-3ef9-4689-8143-76b194e85509"], "name": "Miloš Janjić", "exp": "1651221497", "iat": "1651217597", "family_name": "Janjić", "email": "milos.janjić@youtestme.com" }
```

- **UserInfo**

```
{ "sub": "WB_jFbyFN9XtkAPhHqgqyY9dcF_E-xzatzmXBxGXZpAl", "name": "Miloš Janjić", "given_name": "Miloš", "family_name": "Janjić", "picture": "https://graph.microsoft.com/v1.0/me/photo/$value", "email": "milos.janjić@youtestme.com" }
```

Imagine you make an error when inserting attribute mappings (for example, you misspell a token/endpoint name or enter a non-existent claim for a given token/endpoint). You will get the error shown in the image below, where you can see the values of the access token, ID token, and UserInfo endpoint data. You can copy these three parameters to the corresponding fields of the **"Test attribute mappings"** window and analyze them by modifying the mappings in JSON strings.

HTTP Status 400 – Bad Request

Type Status Report

Message Can't map attribute "email" with mapping "email: user_info.email" for parameters: access_token: ("sub":"WB_jFbyFN9XtkAPhHqgqyY9dcF_E-xzatzmXBxGXZpAl","ver":"2.0","aio":"AVQAq/8TAAArKMbmVaoxN4UaZgTlOImawrSkH2wbOnzWm00bb8YK97ZDPYw51HBNP+fjuXL4S2DLZ4kSrijRDNrzA9aGQBsUCtTo53AQH0R2uq3A+/P+GE=", "iss": "https://login.microsoftonline.com/5485fdb6-11dc-489c-96a9-2ca763bcd5b7/v2.0", "groups": ["f4affda8-ff1e-4fbc-a2a1-4e89bf68d71b"], "ae30426b-ef6c-442f-a561-0039784d83ce"}, "oid": "bf62a491-dd2b-4712-acb8-72cc96b74381", "preferred_username": "milos.janjić@youtestme.com", "uti": "FRt1hRsfNEOsOG8YJYYAA", "given_name": "Miloš", "tid": "5485fdb6-11dc-489c-96a9-2ca763bcd5b7", "aud": "751c32b8-a09f-4785-8c35-fcee577df210", "nbf": "1651217597", "rh": "0.AS8Atv2FVNwRnEiWqSynY7zVt7gyHHWfoIVHjDX87ld98hAvAKk", "wids": ["62e90394-69f5-4237-9190-012177145e10", "b79fbf4d-3ef9-4689-8143-76b194e85509"], "name": "Miloš Janjić", "exp": "1651221497", "iat": "1651217597", "family_name": "Janjić", "email": "milos.janjić@youtestme.com"} id_token: ("sub":"WB_jFbyFN9XtkAPhHqgqyY9dcF_E-xzatzmXBxGXZpAl","ver":"2.0","aio":"AVQAq/8TAAArKMbmVaoxN4UaZgTlOImawrSkH2wbOnzWm00bb8YK97ZDPYw51HBNP+fjuXL4S2DLZ4kSrijRDNrzA9aGQBsUCtTo53AQH0R2uq3A+/P+GE=", "iss": "https://login.microsoftonline.com/5485fdb6-11dc-489c-96a9-2ca763bcd5b7/v2.0", "groups": ["f4affda8-ff1e-4fbc-a2a1-4e89bf68d71b"], "ae30426b-ef6c-442f-a561-0039784d83ce"}, "oid": "bf62a491-dd2b-4712-acb8-72cc96b74381", "preferred_username": "milos.janjić@youtestme.com", "uti": "FRt1hRsfNEOsOG8YJYYAA", "given_name": "Miloš", "tid": "5485fdb6-11dc-489c-96a9-2ca763bcd5b7", "aud": "751c32b8-a09f-4785-8c35-fcee577df210", "nbf": "1651217597", "rh": "0.AS8Atv2FVNwRnEiWqSynY7zVt7gyHHWfoIVHjDX87ld98hAvAKk", "wids": ["62e90394-69f5-4237-9190-012177145e10", "b79fbf4d-3ef9-4689-8143-76b194e85509"], "name": "Miloš Janjić", "exp": "1651221497", "iat": "1651217597", "family_name": "Janjić", "email": "milos.janjić@youtestme.com"} user_info: ("sub":"WB_jFbyFN9XtkAPhHqgqyY9dcF_E-xzatzmXBxGXZpAl","name":"Miloš Janjić","given_name":"Miloš","family_name":"Janjić","picture":"https://graph.microsoft.com/v1.0/me/photo/\$value","email":"milos.janjić@youtestme.com")

Description The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

Apache Tomcat/9.0.41

2.2 Sign in with OpenID

On the login page, click the **"Sign in with OpenID"** button to display the available OpenID configurations.

The user initiates the OpenID authentication and authorization flow by clicking one of them.

The flow can also be initiated externally to the YTM web application by redirecting to the location specified by the **"Initiate Login URL"** parameter.



Username

admin

Password

.....

[Forgot your password?](#)

Sign In

Not registered yet? [Create account](#)

Sign in with OpenID ▾

Okta YTM

Azure AD YTM

The purpose of this training course:

1. Every employee should be certified for a number of skills
2. Every employee should observe the application and testing process end to end and provide feedback and suggestions on what can be improved.
3. **Have very high standards and be very critical – this is the best way to make high-quality software.**
4. Feedback should be sent by email to ["ytm-support@youtestme.com"](mailto:ytm-support@youtestme.com)

For more information on creating an account and sign in, please check this [link](#).

[Contact support](#)

When signing in to the application first time, the user account will be created, and all mapped attributes will be transferred to YTM.

Each new logging into the application (through OpenID) with the same username will update the user profile.