SaaS Offer Technical Tools by Madatech

Documentation

SaaS Offer Technical Tools by Madatech is a set of modules including Landing page and Webhook endpoint intended for the ISVs who partner with Microsoft and integrate their services with Marketplace SaaS Offer. Also, this solution may be beneficial for Teams App developers.

When publishing Marketplace SaaS Offer, you are required to provide a landing page URL and webhook endpoint. Landing page of your application is intended to allow you confirm the activation of the subscription. You might want to process internal procedures when someone purchase a new subscription. The webhook endpoint is intended to notify about any change the client might make on marketplace including renewing, upgrading the plan or quantity. Also, you need to provide a tool for your client to manage the licenses and assigning/revoking license to a user. Our solution is covering all these aspects.

Currently our app includes the following modules:

- **Landing Page module**
  - Subscription activation,
    - Triggering external endpoint when user activates a subscription,
    - Public endpoint to trigger when ISV finish activating the subscription on his side.
  - License Management,
    - Viewing all purchased subscriptions,
    - Assigning a license to a user,
    - Viewing the users that are assigned to a license,
    - Revoking a license from a user.
  - Log management
    - Logging any action performed by user either on portal or in marketplace,
    - Sending each log to an endpoint provided by the ISV.
  - Purchasing subscriptions (coming soon)
    - Purchasing new subscriptions inside a portal,
    - Upgrading currently purchased subscription plan,
    - Upgrading currently purchased subscription quantity.
- **Webhook Endpoint module**
  - Handling activities
    - Public endpoint for marketplace to trigger,
    - Support for both key validation and claim validation of the requests,
    - Identifying activities received from Marketplace and adjusting information on ISV’s side,
    - Logging any activity performed by user in marketplace,
    - Sending each log to an endpoint provided by the ISV.

Planning an offer

Before you will start to deploy any of these modules you need to prepare the following things:

Activation Endpoint – if you need to perform any procedure on your side before activating the subscription then you will need to provide HTTPS Rest API endpoint to receive information about subscriptions requested to be activated. The endpoint will receive requests that includes header parameter MarketplaceToken. Keep this token, you will need this token later when requesting to activate the subscription after you finished on your side. You will receive Subscription model as a JSON in a payload:

```
{
  "id": "<guid>", // purchased SaaS subscription ID
  "subscriptionName": "Contoso Cloud Solution", // SaaS subscription name
  "offerId": "offer1", // purchased offer ID
  "planId": "silver", // purchased offer's plan ID
  "quantity": 20, // number of purchased seats, might be empty if the plan is not per seat
  "subscription": { // full SaaS subscription details, see Get Subscription APIs response body for full description
    "id": "<guid>",
    "publisherId": "contoso",
    "offerId": "offer1",
    "name": "Contoso Cloud Solution",
    "saasSubscriptionStatus": " PendingFulfillmentStart ",
    "beneficiary": {
      "emailId": "test@test.com",
      "objectId": "<guid>",
      "tenantId": "<guid>",
      "puid": "<ID of the user>"
    },
    "purchaser": {
      "emailId": "test@test.com",
      "objectId": "<guid>",
      "tenantId": "<guid>",
      "puid": "<ID of the user>"
    },
    "planId": "silver",
    "term": {
      "termUnit": "P1M",
      "startDate": "2022-03-07T00:00:00Z",
      "endDate": "2022-04-06T00:00:00Z"
    },
      "autoRenew": true/false,
    "isTest": true/false,
    "isFreeTrial": false,
    "allowedCustomerOperations": <CSP purchases>["Read"] <All Others> ["Delete", "Update", "Read"],
      "sandboxType": "None",
      "lastModified": "0001-01-01T00:00:00",
      "quantity": 5,
    "sessionMode": "None"
  }
}
```

Log Share Endpoint – Provide an endpoint to receive any activity performed by the user. You will receive the following payload:

```
{
  "id": "<guid>", // Action ID
  "type": "<string>", // Action type [Resolve, Activate, Assign, Unassign]
  "initiatorUserId": "<guid>", // Guid of the user who performed the action
  "date": "<datetime>", // Date and time of the action
  "subscriptionInfoId": "<guid>", // subscription ID from database (only if the action relates to a subscription)
  "subscriptionUserId": "<guid>", // subscription user ID from database (only if the action relates to a user)
}
```

Database connection string – Provide a connection string to a database. Use this script to create the database:

```
IF OBJECT_ID(N'[__EFMigrationsHistory]') IS NULL

BEGIN
    CREATE TABLE [__EFMigrationsHistory] (
        [MigrationId] nvarchar(150) NOT NULL,
        [ProductVersion] nvarchar(32) NOT NULL,
        CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY ([MigrationId])
    );
END;
GO

BEGIN TRANSACTION;
GO

CREATE TABLE [Subscriptions] (
    [Id] uniqueidentifier NOT NULL,
    [SubscriptionId] uniqueidentifier NOT NULL,
    [PlanId] nvarchar(max) NOT NULL,
    [TenantId] uniqueidentifier NOT NULL,
    [Quantity] int NOT NULL,
    [StartDate] datetime2 NOT NULL,
    [EndDate] datetime2 NULL,
    [Status] nvarchar(max) NOT NULL,
    [BillingCycle] nvarchar(max) NOT NULL,
    [AcquiredLicenses] int NOT NULL,
    CONSTRAINT [PK_Subscriptions] PRIMARY KEY ([Id])
);
GO

CREATE TABLE [SubscriptionUsers] (
    [Id] uniqueidentifier NOT NULL,
    [UserId] uniqueidentifier NOT NULL,
    [UserUpn] nvarchar(max) NOT NULL,
    [SubscriptionInfoId] uniqueidentifier NOT NULL,
    [Status] nvarchar(max) NOT NULL,
    CONSTRAINT [PK_SubscriptionUsers] PRIMARY KEY ([Id]),
    CONSTRAINT [FK_SubscriptionUsers_Subscriptions_SubscriptionInfoId] FOREIGN KEY ([SubscriptionInfoId]) REFERENCES
[Subscriptions] ([Id]) ON DELETE CASCADE
);
GO

CREATE TABLE [SubscriptionActions] (
    [Id] uniqueidentifier NOT NULL,
    [Type] nvarchar(max) NOT NULL,
    [InitiatorUserId] uniqueidentifier NOT NULL,
    [Date] datetime2 NOT NULL,
    [SubscriptionInfoId] uniqueidentifier NULL,
    [SubscriptionUserId] uniqueidentifier NULL,
    CONSTRAINT [PK_SubscriptionActions] PRIMARY KEY ([Id]),
    CONSTRAINT [FK_SubscriptionActions_SubscriptionUsers_SubscriptionUserId] FOREIGN KEY ([SubscriptionUserId])
REFERENCES [SubscriptionUsers] ([Id]) ON DELETE CASCADE,
    CONSTRAINT [FK_SubscriptionActions_Subscriptions_SubscriptionInfoId] FOREIGN KEY ([SubscriptionInfoId]) REFERENCES
[Subscriptions] ([Id]) ON DELETE CASCADE
);
GO

CREATE INDEX [IX_SubscriptionActions_SubscriptionInfoId] ON [SubscriptionActions] ([SubscriptionInfoId]);
GO

CREATE INDEX [IX_SubscriptionActions_SubscriptionUserId] ON [SubscriptionActions] ([SubscriptionUserId]);
GO

CREATE INDEX [IX_SubscriptionUsers_SubscriptionInfoId] ON [SubscriptionUsers] ([SubscriptionInfoId]);
GO

INSERT INTO [__EFMigrationsHistory] ([MigrationId], [ProductVersion])
VALUES (N'20221226173006_InitialCreate', N'7.0.1');
GO

COMMIT;
GO
```

PLEASE NOTE: We do not store any data on our side. All data stored in this database.

Registering Offer

Open our portal and go to the Home page. Press Create new offer. Fill the form as instructed. You will be redirected to the Offer page. Choose the licenses for the offer to activate the offer. After saving it the portal will display instructions on how to create resources on Azure. After you created app services upload app service publishing profiles in the portal. We will contact you for a detail. In 3-5 working days we will deploy the app to your app service. After we finish deploying the app in portal you will see the offer marked as Live.