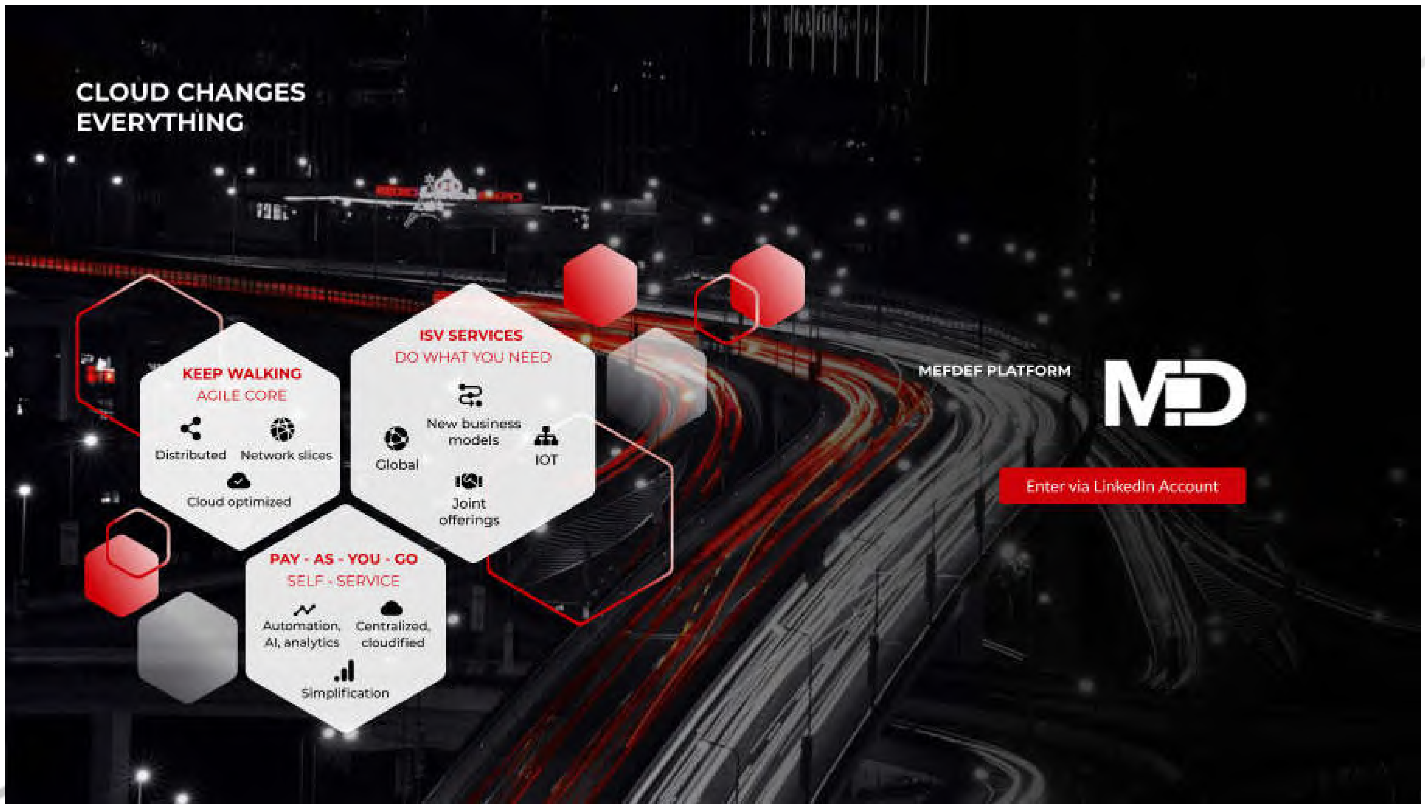


MEF.DEV PLATFORM STARTER GUIDE

Introduction for the Developer when Working with the Serverless Architecture Application Development Platform

Annotation

The document describes the first steps of a .NET developer when developing applications for further publication in the MEF.DEV platform based on the Domain Driven Design reference architecture and loosely coupled code.

Sergey V. Polovnikov
Sergiy.polovnikov@natec.tech

Contents

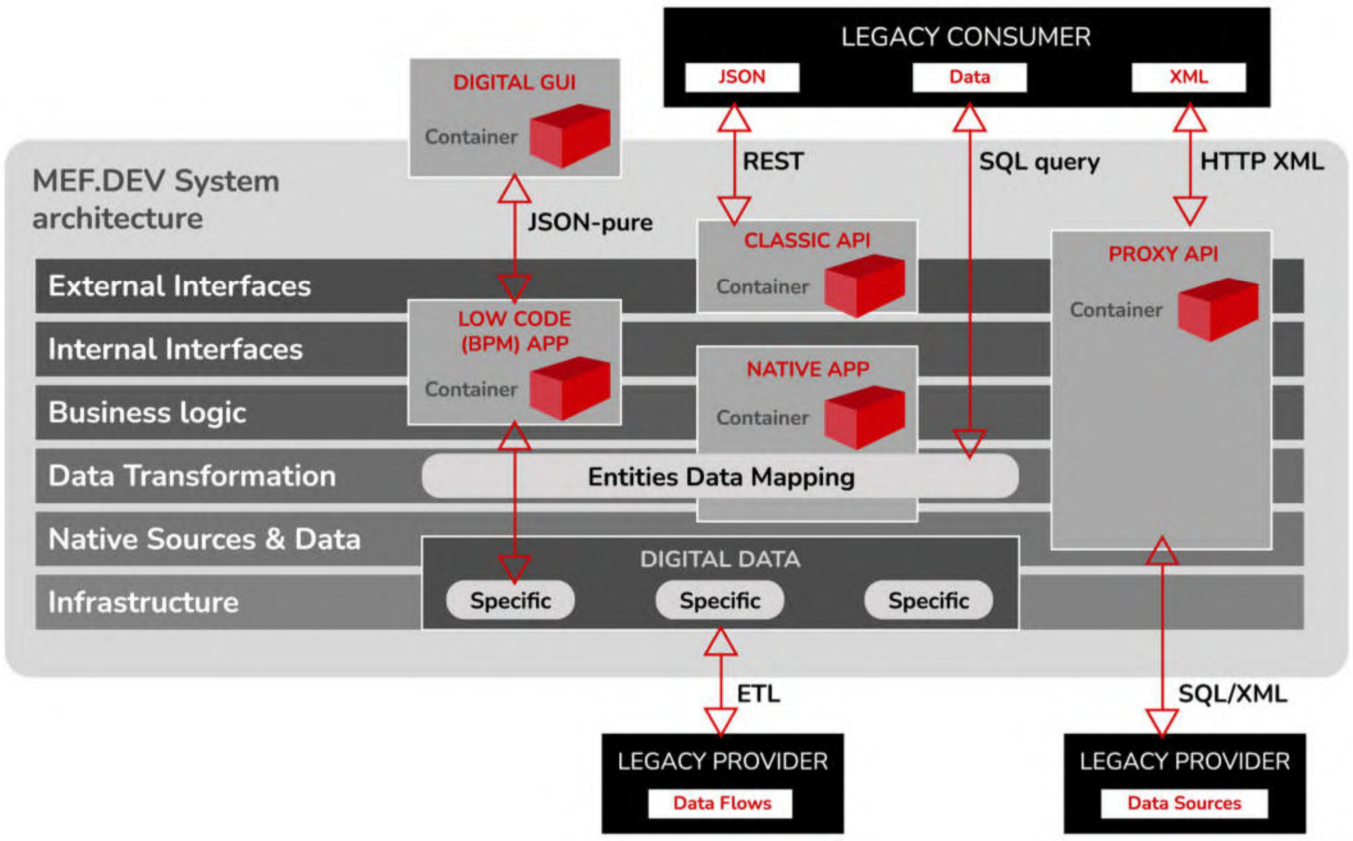
Revision History

Rev	Date	Author	Description
1.0	2021-04-30	Sergey Polovnikov	First version

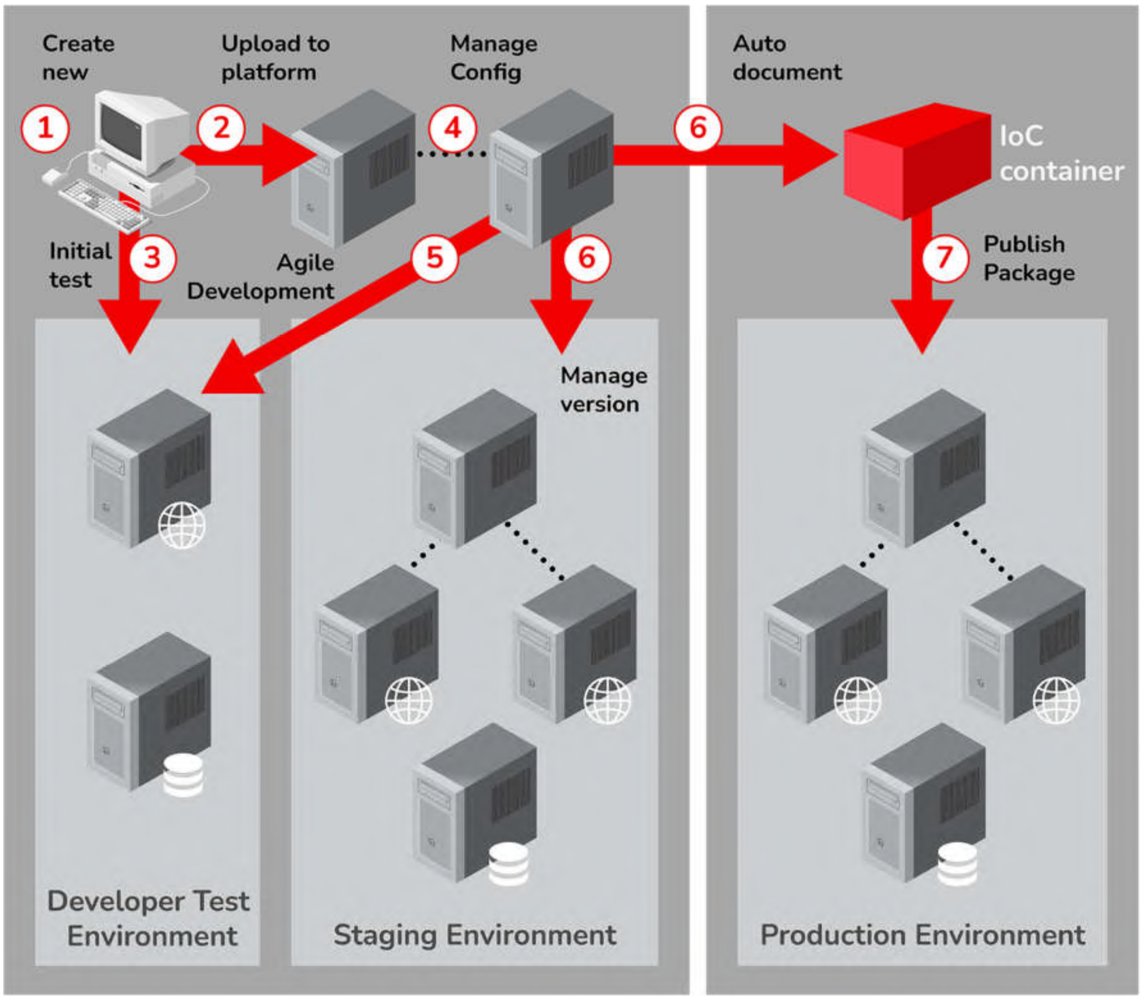
Introduction

Introducing MEF. DEV

MEF. DEV is application development, hosting, and management platform that can help operators accelerate the transformation of legacy BSS features. The platform supports the domain-driven design and business analysis with the ability to generate code based on Domain Driven Design models and a unified development process. It provides a collaborative approach to software development that helps companies get a software solution for business several times faster than if it were done by a team of internal or hired developers.



Acceleration is achieved by forming a joint project team of various teams with a narrow specialization for developing a specific product in several parallel streams at once, each of which will write its piece of code, then the results of their programming will be combined into a single whole. Thus, the end customer, firstly, saves time and gets the product faster, and secondly, receives twice as many helpful resource for development. Developers learn to work in maximum synchronization with other teams with documentation updated in real mode, thereby minimizing alterations and getting the opportunity to work with each specific team at any time convenient for it from anywhere in the world.



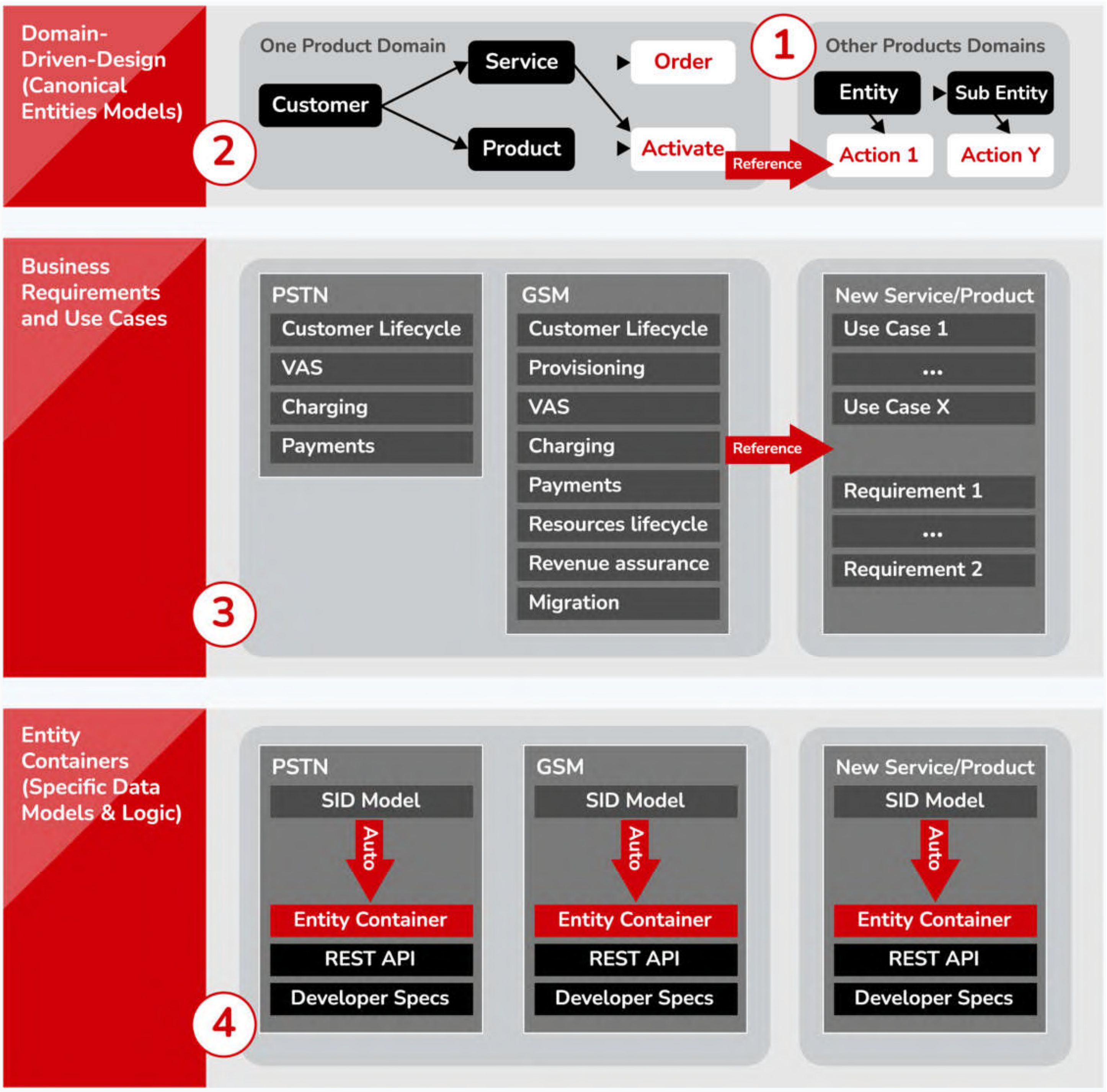
Essential advantages of the MEF.DEV platform:

- in the MEF system. DEV one product or project is written simultaneously by different development teams
- use loosely coupled code that is automatically assembled into a single unit
- source code in MEF. DEV can be generated and hosted not in the cloud, but in-house, i.e. inside the enterprise
- transparency of data models and auto-generation of the technical specification of each version of such loosely coupled code
- automatic code generation of updated data models that support such patterns as model-first and database-first, including using a reverse engineering approach

MEF.DEV platform provides accelerated development, serverless hosting, and Agile application development capabilities for enterprise enterprises. The platform aims to increase business agility, especially in legacy environments, by standardizing legacy-> Digital transformation processes. The platform has a graphical user interface that simplifies integrating and deploying applications.

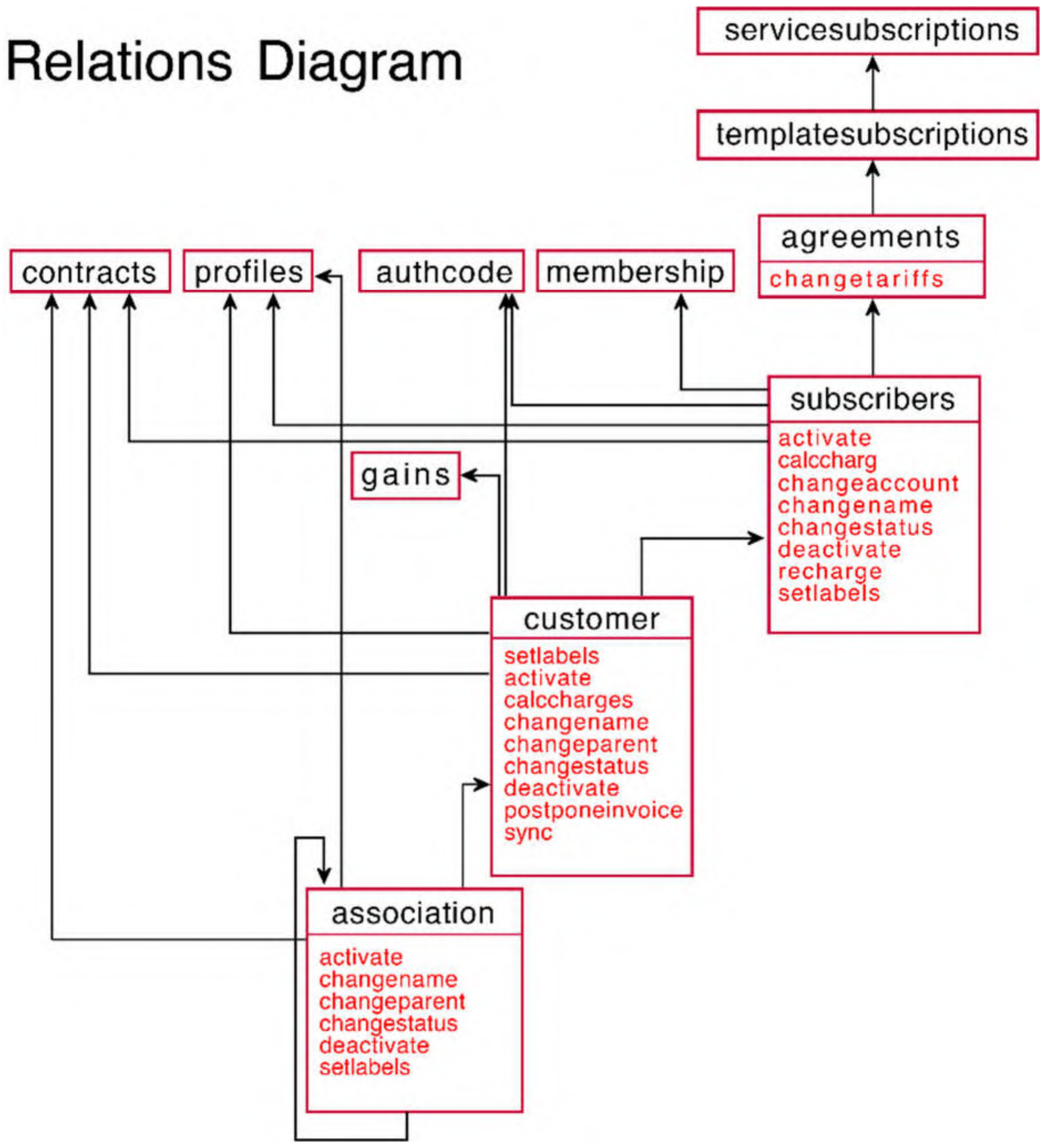
Entities as the basis of a reference architecture

The MEF framework automatically displays unique URI values. DEV after publishing a specific version of the data models (entities). Entities provide access to business logic and specific (native) data storage through **composition**, much more than Dependency injection or Inversion of Control. Composition gives Enterprise developers a more straightforward setup and sequential management in larger projects without worrying about when, where, and how a particular application/block/module is configured. Regardless of how they use the embedded application/block/module, they get precisely the implementation wanted.



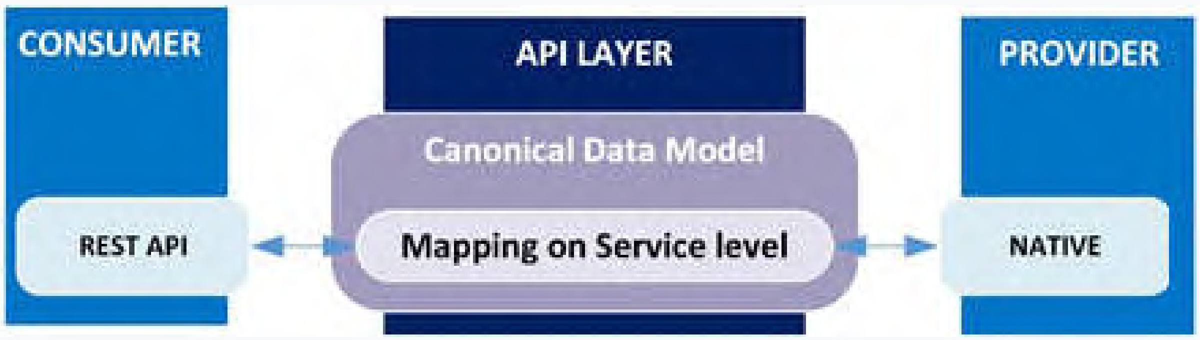
1. Creating a concrete Enterprise Reference Architecture begins with the definition and standardization of domain services or products.
2. Based on these definitions, entity and action models are developed. These include common types and behaviors about all services/products for further description. For each service/product, specific resources, interactions, and models are created. Some of these models refer to multiple entities (for example, there is a significant overlap between the Customer and Group models).
3. Business requirements (BR) and use cases (UseCases) are specific to each service/product and define functional requirements, use cases, business processes, transitions between states, etc. related to particular entities (e.g., activation action for services or physical installation of a service)
4. Next, MEF. Dev maps entity and action models to data schemas, then exported to pluggable IoC containers, automatically generating documentation, use cases, and instructions for the developer or external APIs.

Visualizing the contents of a specific IoC container (actually a developer-specific implementation) gives transparency to all other platform participants.



an example of a BSS.Entities plugin visualization

A significant advantage is the principles of exposure, which, together with bringing custom data models to generally accepted (canonical) ones, provide **opportunities for flexible development by various developers without diving into the details of the final implementation, including the data layer**:

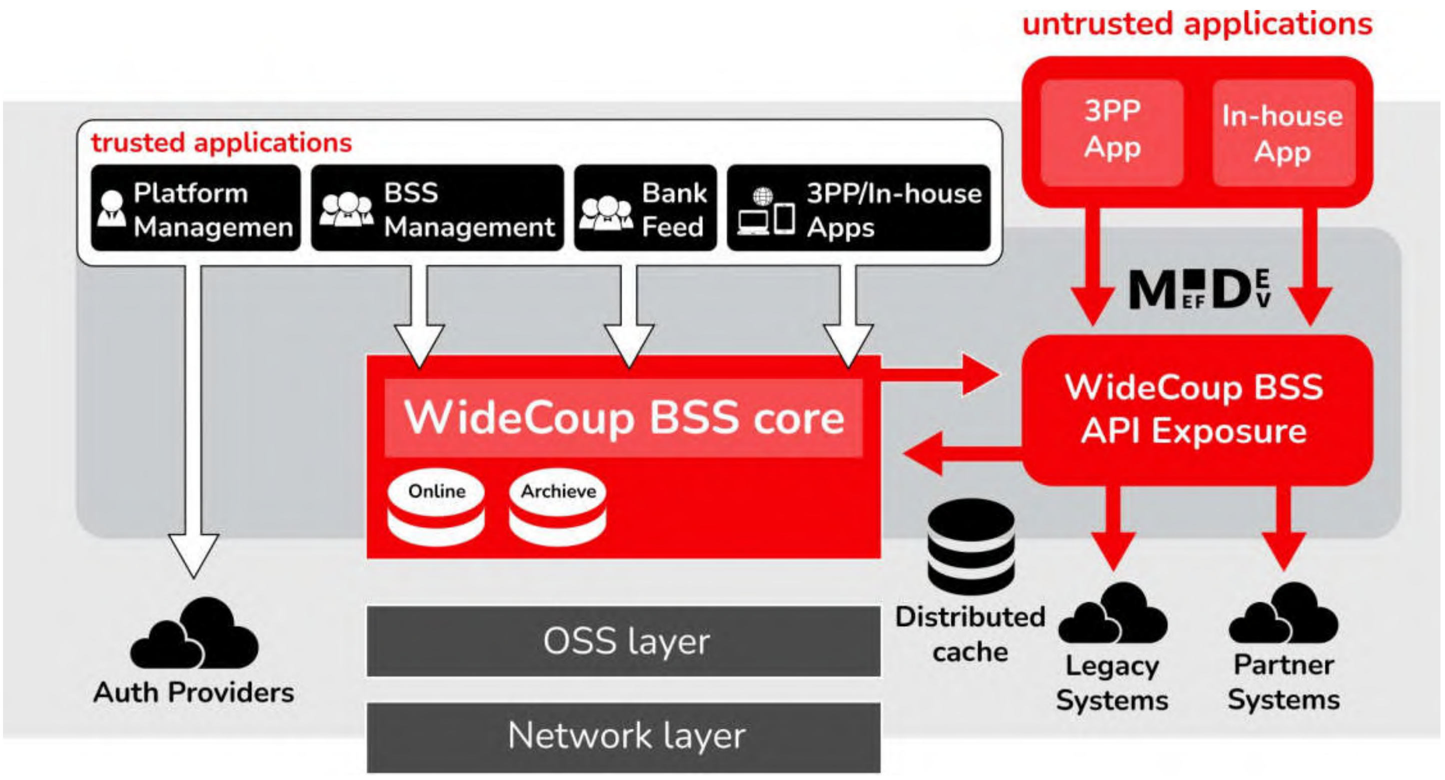


Platform Description

The user interface of the MEF framework. Dev is implemented as a developer self-service web portal based on a plugin-oriented approach, where each plugin is the final result of the work of a particular developer or development team. The current version of the platform supports the following plugin options:

- **Portal** (portal APIUI application) **is** a rich custom Web application for interactive interaction with the end-user. This application assumes the presence of a GUI user interface, the development of which begins with a blank canvas, on which the developer can place various interface elements to implement business logic through calls to the backend of the platform, namely backend plug-ins. Data type the interaction is trusted, and the Canvas diagram is reflected as a Trusted application.
- **Canvas** (canvas UI app) is a simple Web application that implements an integrated interaction of different Web services within a single user interface. Typically used as an interactive tool on a secure website that provides a layout of both linked and unrelated external Web applications, services, and links. Portal applications are focused on providing data in an easy-to-understand format and are a simplified option portal application.
- **A model-driven** application is a software server application whose functions or behavior are based on or controlled by some evolutionary application models of target things (entities). As the basis of the platform's reference architecture, entity application models are available as a public part of the export of a loosely coupled approach to programming. It is essential to understand that shared information and data models are standardized and standardized and are serviced on the platform side as the main elements of the dynamic composition. They can be changed during the execution of the application.
- **A** service (microservice API) is a stand-alone server application in which all application functions run as an independent service. This architectural type allows each service to evolve or be updated without disrupting other applications on the platform, as it does not involve integration at the level of loosely coupled code.

The functionality of the platform includes various operational tasks for the operation and maintenance of plug-ins of different types, providing them with standardized approaches to the authorization and authentication infrastructure, monitoring and logging of work, control and accounting for the use of plug-ins during the life cycle for various tasks, including billing the subscription model. From the point of view of the approach to deployment, the platform supports the Internet and Intranet usage schemes, the main differences of which are in the user authorization scheme - oAuth2.0 and Active Directory, respectively. The WideCoup Billing platform is developed by the NATEC R&D Center on .Net Core and can be deployed on both Linux and Windows architectures.



For the developers themselves, the platform provides many opportunities to speed up the process of developing and deploying programming results, namely:

- authorization and authentication infrastructure that provides the developer with ready-made functionality to control access to the business logic of applications based on the role model
- possibility of code generation based on models and domain-oriented design (domain-driven design)
- the ability to manage the configuration of the application at different levels - the level of developer and end-user (application tenant)
- the possibility of auto-generation of documentation for each version of the application based on standard attributes of decorators. In addition to specifications for data models and classes, the documentation also includes visualization of model dependencies (entity relationships and available actions).
- the ability to manage the current version of the application available on the platform, including for rollback tasks to previous versions in case of inoperability
- the ability to use the UX/UI template library (design UI kit) when developing interactive GUI applications

A design UI kit is a set of resources and tutorials for creating user interfaces for a customer's internal applications. The library's front-end styles were developed by NATEC's design team and are distributed as deployment-ready resource packages that can be included in your application code.

Technical stack for working with the MEF.DEV platform keys .NET Core for server-side backend applications and Angular for mobile and interactive Web application development.

.NET Core, as a developer platform, provides functionality for cross-platform and modularity, an essential difference of which is the separation of the runtime from the library. Also, because .NET Core is modular, every component of the MEF framework. Dev is updated through a separate NuGet package manager. This allows developers to update their modules individually and independently of each other. As a result, the application is in the MEF.DEV platform can work with individual modules and is not dependent on updating the entire platform.

In addition to the modularity of the Framework itself, the MEF framework. DEV sets rules for the programmer based on abstraction methods, not through implementations. The most powerful tool for this is the interface approach, making it possible not to be tied to a specific implementation. Interfaces allow the developer to link classes in a very accessible form, so they can be developed and tested separately from external dependencies or with a minimum number of them if complete isolation is not possible. This source code implementation is called loosely coupled code. The platform provides an infrastructure for using it for other developers, thereby guaranteeing a collaborative approach to software delivery in the Agile agile development paradigm.

In terms of front-end development, the platform supports Google's lightweight, fast, and affordable JavaScript framework, designed to develop single-page applications. This framework makes it possible to build interactive and dynamic web applications that require less effort and code. The peculiarity of Angular is that it allows you to use HTML as a template language, and then extend the HTML syntax to express the application’s components. Using data binding and dependency injection, you can eliminate most of the code you would have to write. Overall, Angular is a framework for building large-scale, high-performance, and easy-to-maintain web applications by:

- Typescript is the primary template language, but applications can also be written using languages such as Dart or JavaScript.
- Angular aims to develop Single Page Applications
- Applications written in Angular are compatible with various browsers
- Clean and accurate user interface design with simple routing
- Angular offers server-side rendering that speeds up the loading of the initial page and therefore improves SEO by making it easier to crawl dynamic pages.

It is essential to note the orientation of the MEF. The DEV platform primarily serves high-load applications by dynamically scaling to provide enough resources to handle current application loads. The main scaling element is the distributed cache, an external platform service for developer applications. By using the Distributed Cache, application data:

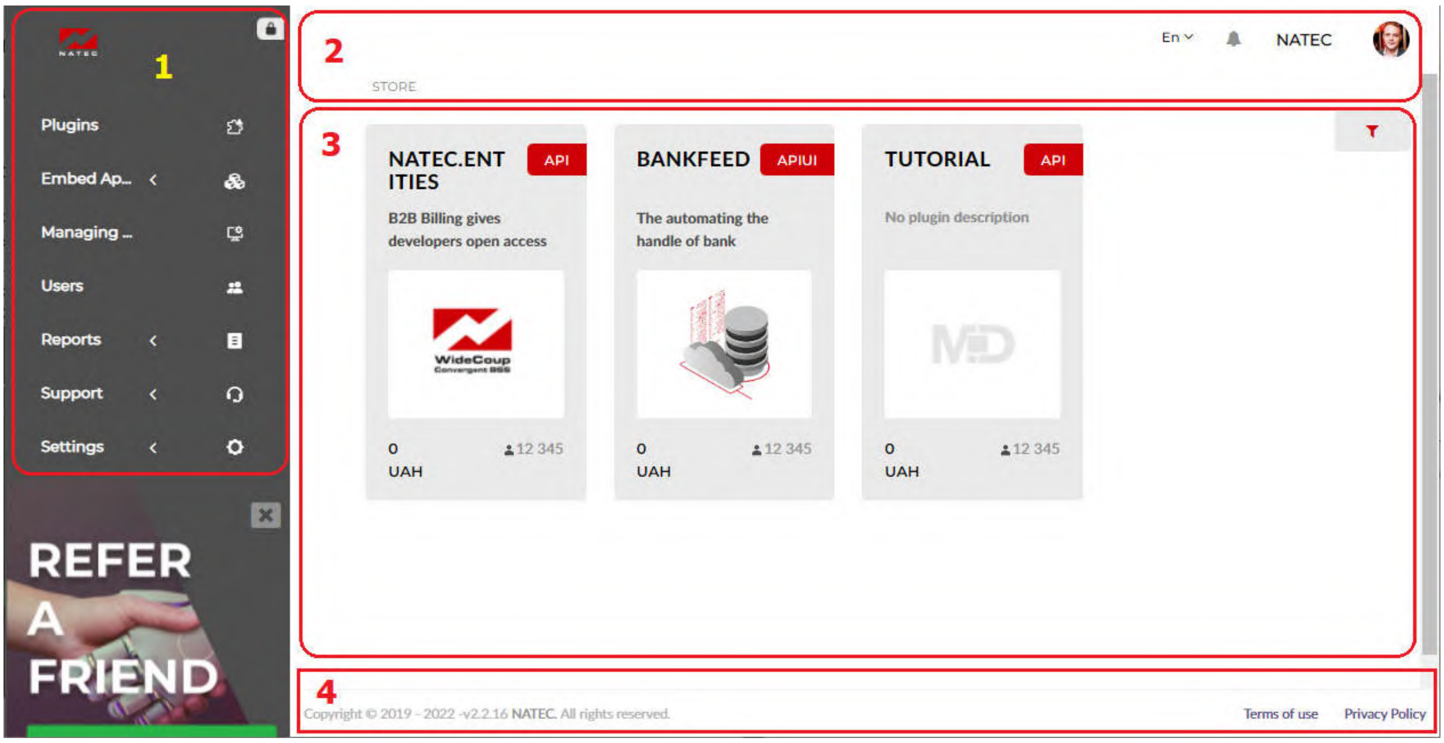
- Are coherent (consistent) for requests to multiple servers.
- Withstand server restarts and application deployment.
- Do not use local memory.

The distributed cache configuration is based on the implementation; by default, the platform uses a configuration based on MS SQL Server. Redis and NCache implementations are also available. The application interacts with the cache using the IDistributedCache interface regardless of the implementation.

Platform Interface

The concept of working with the platform interface consists of four elements:

1. Left sidebar
2. Upper header
3. App workspace
4. Bottom footer



The left sidebar is used to select platform features and contains the following sections:

- **Plugins** – designed for developers to create and configure applications
- **Embed Apps** – Displays available interactive Canvas or Portal apps
- **Managing Plugins** – designed to manage the tenant's configuration and subscriptions to other developers' applications
- **Users** —Designed to manage tenant users, including user roles and user access to tenant applications
- **Reports** – designed to monitor the use of tenant applications and errors
- **Support** – designed for calls to the technical support of the platform and access to the notification information
- **Settings** – designed for the tenant, user profile, and other settings

Top header – the selected interface language, the presence of notifications, the name of the selected tenant, as well as the current user, and the menu for switching to the profile are displayed.

The stage displays the functionality of the selected page.

The bottom footer contains the version of the platform and links to the terms of use of the platform (/terms) and privacy policy (/privacy).

Common usage scenario

Depending on the user's role, there may be different scenarios for using the platform, but the most common scenario for third-party developers is as follows:

- A new developer enters the platform and registers as a new user. To go to the platform, a referral link is used, which the new user received among his friends
- After successful registration, an individual tenant is created for the user, and the user can go to the Plugins section and create the first application (plugin) based on ready-made samples and documentation in the public domain
- After successfully describing, configuring, and self-testing the application, the developer directs the application for verification through the context menu of the table in the Plugins section
- After successful verification of the application, the application becomes available in the Store for use by other users of the platform
- A platform user selects an app from the Store and adds the app to their tenant through the button of the same name in the app's description
- If necessary, the user configures the application for his tenant; for server applications, the administrator creates a set of credentials to access the applications
- in the case of a multi-user tenant, the application is assigned as the tenant administrator to many users and becomes available for use

The application lifecycle includes the following states:

- **New** - The app has not yet been downloaded to the platform. Not available to anyone.
- **draft** - the application is under development and is presented as an intermediate version. Not available to anyone except the developers of the tenant
- **Review** – the application is sent for review and awaits a decision
- **Rejected** – the application has not been successfully tested and needs improvement.
- **Active** – The application has been successfully validated and is now available to all users to add to the tenant and use when the user assigns the application as a tenant administrator
- **Stopped** – The tenant administrator suspends the application for tenant users.
- **Suspended** – the application is removed from the tenant and is no longer available for use
- **Deactivated** – the previously tested application has been removed by the developer and is not available for use by other users
- **Deleted** – the application under development has been removed from the platform

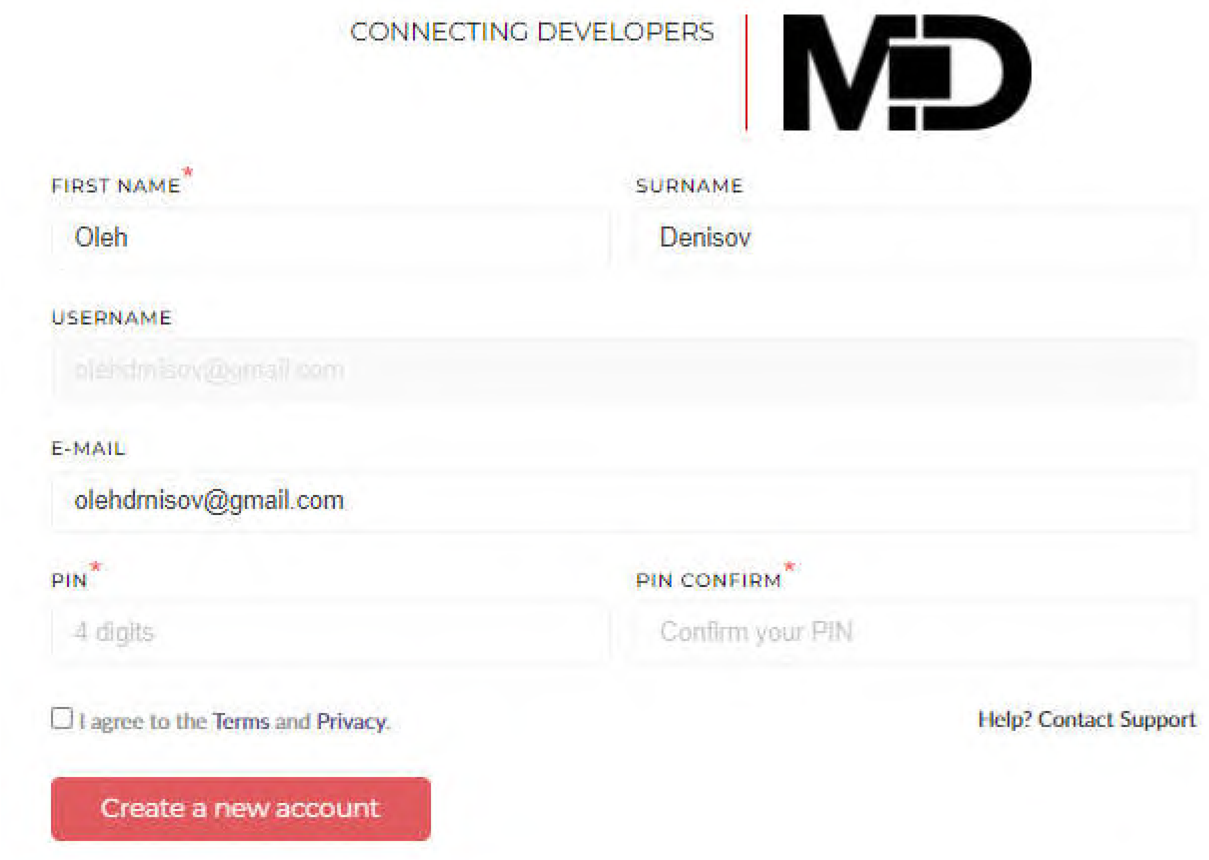
In addition to the primary states described above, additional states are available that can be used in other scenarios:

- **Requested** –The tenant requests the application by a user who is not a tenant administrator or requires confirmation from the application developer

- **FinBlock** – the application is not available for use in the tenant due to the presence of the tenant's debt. Used in the case of a paid subscription to the application.
- **Paused** - The app is paused. Used in the case of a paid subscription to the app.

User Registration (/register)

This page is used to register a new user in the system and contains information about the user in the system, part of the information the system automatically takes from LinkedIn.
IMPORTANTLY! The user needs to specify the PIN twice (this value will be used to unlock the screen of applications after the timeout expires), as well as read the terms of use of the platform (/terms) and privacy policy (/privacy).



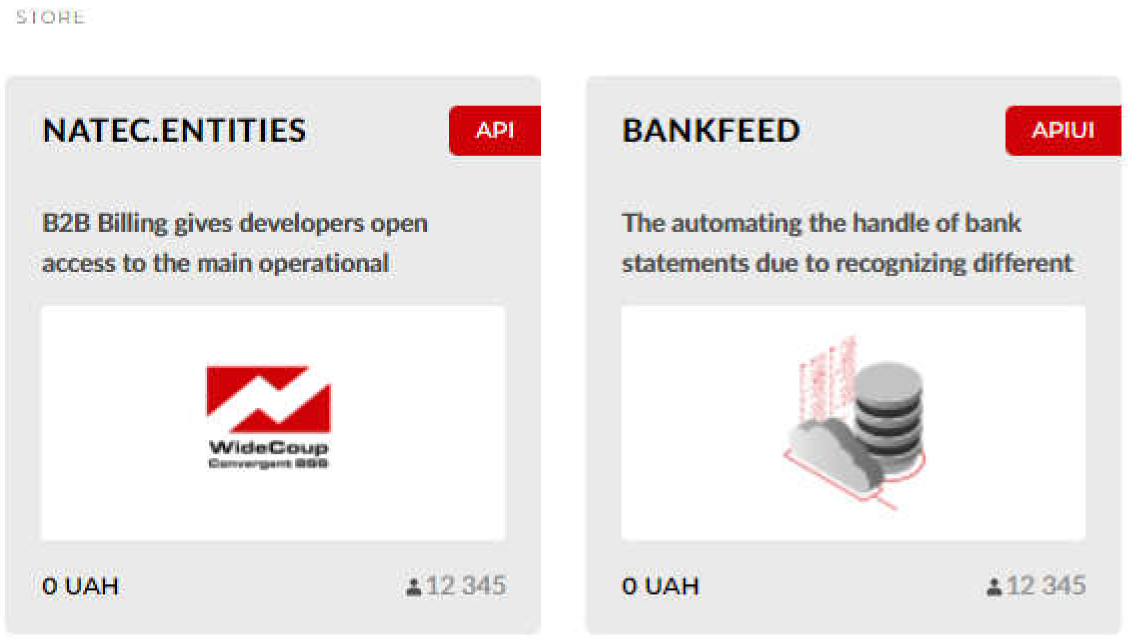
CONNECTING DEVELOPERS

FIRST NAME*: Oleh
SURNAME: Denisov
USERNAME: olehdmisov@gmail.com
E-MAIL: olehdmisov@gmail.com
PIN*: 4 digits
PIN CONFIRM*: Confirm your PIN

☐ I agree to the Terms and Privacy.

Shop(/store)

This page is used to view and subscribe to existing plugins on the platform. This page contains information about the plugin, its creator, and the subscription price.



STORE

NATEC.ENTITIES API

B2B Billing gives developers open access to the main operational

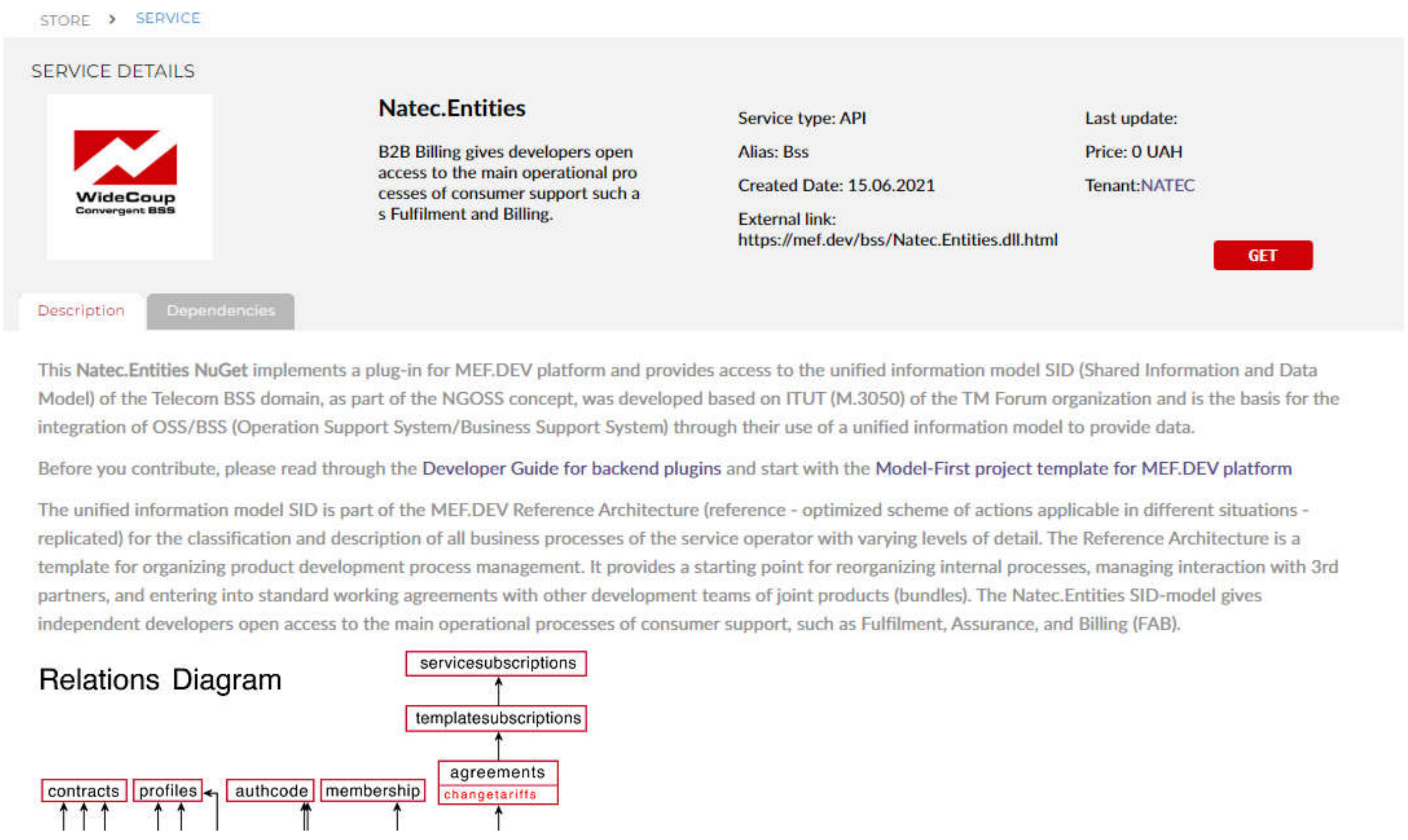
0 UAH 12 345

BANKFEED APIUI

The automating the handle of bank statements due to recognizing different

0 UAH 12 345

The user can study the detailed information about the application by going to the application page and adding the plugin to his tenant for further use - for this purpose; the **GET** button is used.



STORE › SERVICE

SERVICE DETAILS

Natec.Entities

B2B Billing gives developers open access to the main operational pro cesses of consumer support such a s Fulfilment and Billing.

Service type: API

Alias: Bss

Created Date: 15.06.2021

External link: https://mef.dev/bss/Natec.Entities.dll.html

Last update:

Price: 0 UAH

Tenant:NATEC

GET

Description | Dependencies

This Natec.Entities NuGet implements a plug-in for MEF.DEV platform and provides access to the unified information model SID (Shared Information and Data Model) of the Telecom BSS domain, as part of the NGOSS concept, was developed based on ITUT (M.3050) of the TM Forum organization and is the basis for the integration of OSS/BSS (Operation Support System/Business Support System) through their use of a unified information model to provide data.

Before you contribute, please read through the Developer Guide for backend plugins and start with the Model-First project template for MEF.DEV platform

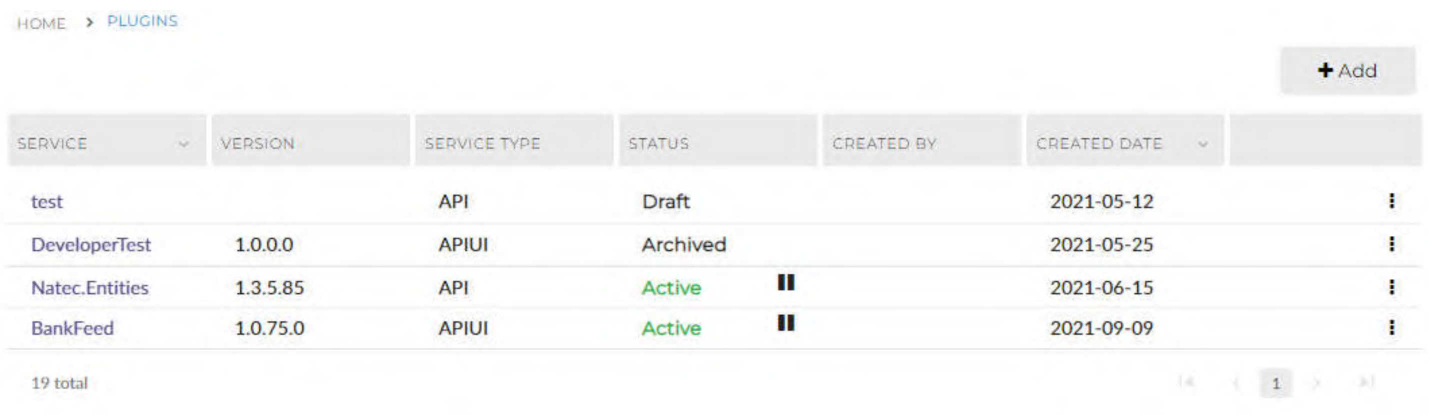
The unified information model SID is part of the MEF.DEV Reference Architecture (reference - optimized scheme of actions applicable in different situations - replicated) for the classification and description of all business processes of the service operator with varying levels of detail. The Reference Architecture is a template for organizing product development process management. It provides a starting point for reorganizing internal processes, managing interaction with 3rd partners, and entering into standard working agreements with other development teams of joint products (bundles). The Natec.Entities SID-model gives independent developers open access to the main operational processes of consumer support, such as Fulfilment, Assurance, and Billing (FAB).

Relations Diagram

Some applications contain dependencies on other plugins – information about this is presented on the same name tab. This information should be considered when adding an application to a tenant because with the application you are adding, its dependencies will also be added to the tenant and will require configuration (if applicable).

Plugins (/console/services/my-plugins)

This page is used to view, create, edit, and delete plugins on the platform. When you go to this section, the tabular part of the page contains a list of plugins and those available for manipulation by a particular user.



HOME > PLUGINS

+ Add

SERVICE	VERSION	SERVICE TYPE	STATUS	CREATED BY	CREATED DATE	
test		API	Draft		2021-05-12	⋮
DeveloperTest	1.0.0.0	APIUI	Archived		2021-05-25	⋮
Natec.Entities	1.3.5.85	API	Active ❚❚		2021-06-15	⋮
BankFeed	1.0.75.0	APIUI	Active ❚❚		2021-09-09	⋮

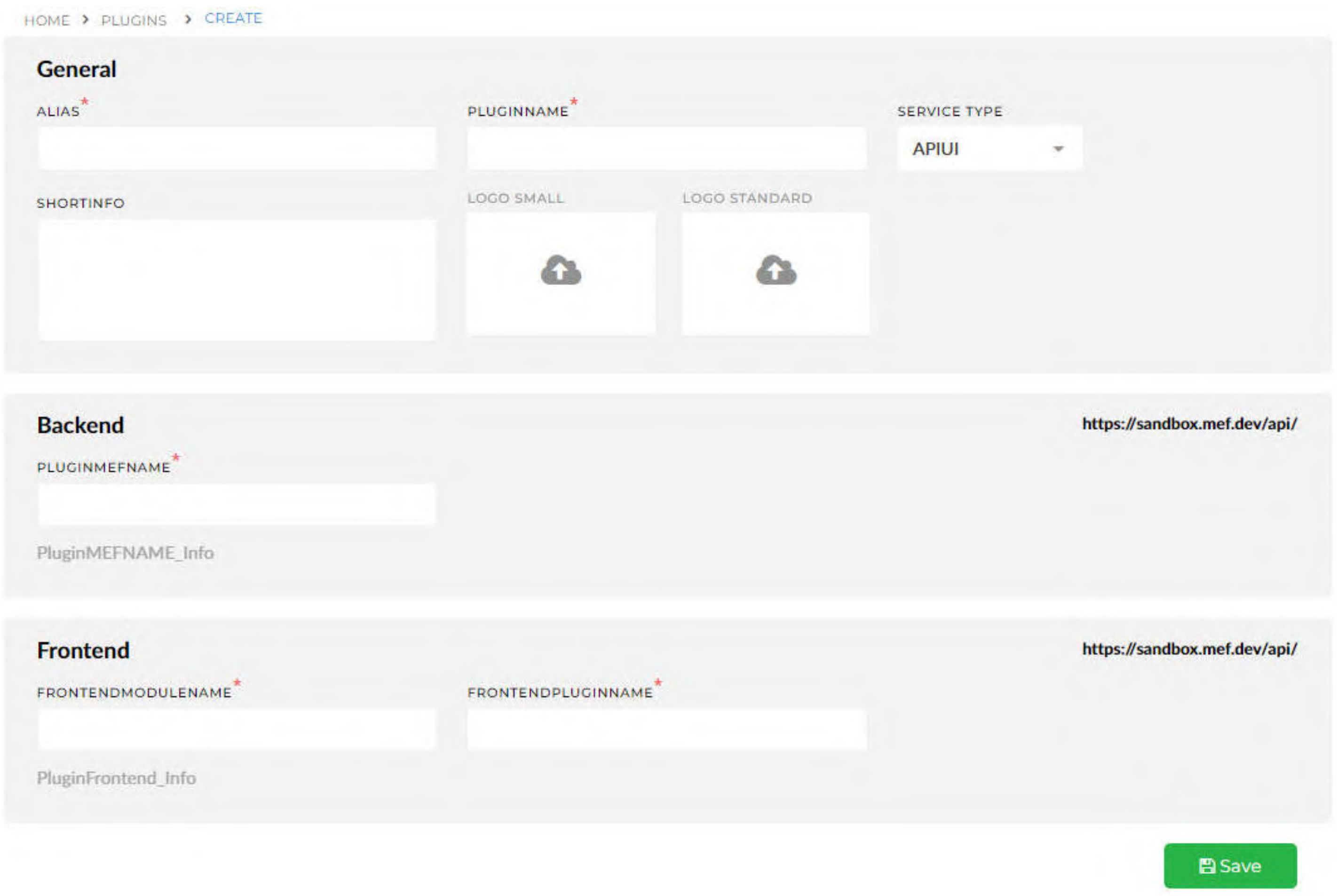
19 total

Table description:

- **VERSION** - the current version of the plugin is indicated in the column.
- **SERVICE TYPE** – the plugin type is specified in the column.
- **STATUS** – the plugin status is indicated in the column.
- **CREATED BY** – the column indicates who created the plugin.
- **CREATED DATE** – the date of creation of the plugin.

Adding a new plugin occurs after clicking on the "**+Add**" button. This opens a new plugin creation page (/console/services/create). Required fields are marked with an asterisk. In the **SERVICE TYPE** field, you must specify the type of plugin to be created. By default, the type of API plug-in to be created. The current version of the platform supports the following plugin options:

- **Portal** (portal APIUI application) **is** a rich custom Web application for interactive interaction with the end-user. This application assumes the presence of a GUI user interface, the development of which begins with a blank canvas, on which the developer can place various interface elements to implement business logic through calls to the backend of the platform, namely backend plug-ins. Data type the interaction is trusted, and the Canvas diagram is reflected as a Trusted application. A distinctive feature of these applications is the presence of a user configuration for accessing on-premises data located in the perimeter of the end user's enterprise. This approach ensures that various requirements for personal data protection are met in cases where data stored in the cloud is impossible or undesirable. An example of a plugin of this type is the tutorial-ui-plugin.
- **Canvas** (UI canvas app) is a simple Web application that implements an integrated interaction of different Web services within a single user interface. Typically used as an interactive tool on a secure website that provides a layout of both linked and unrelated external Web applications, services, and links. Portal applications focus on providing data in an easy-to-understand format and implement business logic changes or manipulations with data without access to the server part of the platform (do not require the presence of backend plugins) and are simplified versions of the portal application. Examples of this type of plugin are simple-UI-plugin-calculator and ui-plugin-example.
- **A model-driven** (backend API app) is a software server application whose functions or behavior are based on or controlled by some evolutionary application models of target things (entities). As the basis of the platform's reference architecture, entity application models are available as a public part of the export of a loosely coupled approach to programming. It is essential to understand that shared information and data models are standardized and standardized and are serviced on the platform side as the main elements of the dynamic composition. They can be changed during the execution of the application. A distinctive feature of these applications is the ability to automatically generate the source code of the application after changing models and the ability to use a custom configuration to access local data located in the perimeter of the end user's enterprise. An example of a plugin of this type is the model-first-backend-plugin.
- **A service** (microservice API) is a stand-alone server application in which all application functionality runs as an independent service. This architectural type allows each service to evolve or be updated without disrupting other applications in the platform because it does not involve integration at the level of loosely coupled code. A distinctive feature of these applications is that there is no user configuration to access the user's local data and implements A request-response approach to interoperability for the use of external, independent functions.


HOME › PLUGINS › CREATE

General

ALIAS*

PLUGINNAME*

SERVICE TYPE: APIUI

SHORTINFO

LOGO SMALL

LOGO STANDARD

Backend

https://sandbox.mef.dev/api/

PLUGINMEFNAME*

PluginMEFNAME_Info

Frontend

https://sandbox.mef.dev/api/

FRONTENDMODULENAME*

FRONTENDPLUGINNAME*

PluginFrontend_Info

The required fields depend on the type of application selected:

- **Portal** – requires filling in the Backend and Frontend sections
- **Canvas** – requires you to fill in the Frontend section
- **Model-driven** – requires filling in the Backend section
- **Service** – requires filling in the Backend section

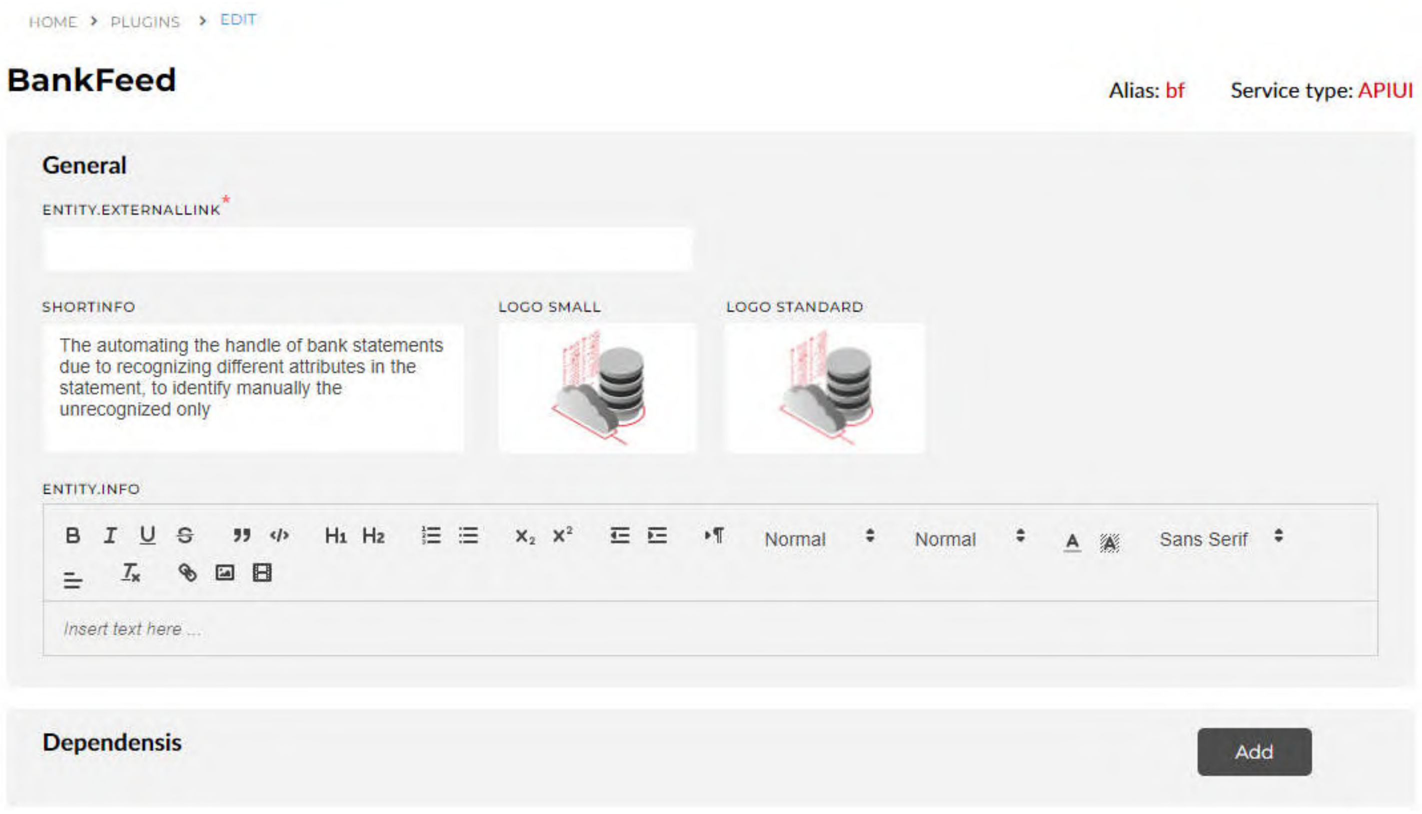
The Backend and Frontend sections include the following required fields:

- **ALIAS** is a unique value for your application domain, such as BSS; if you do not know the value, specify the value in the format ddmmyyhhmm. Understanding that this value will be part of the URI when accessing your application is essential. You can enter Latin and numerals and use special characters "-" and "_."
- **PLUGINNAME** – A unique (within the store) friendly name for your app, must fully match its intended purpose in the form of a noun, such as a Calculator. You can enter Latin and numbers without using special characters other than a space.
- **PLUGINMEFNAME** – the unique (within alias) name of the application backend must fully match the namespace value of your backend project. It is allowed to enter Latin and numbers without using special characters
- **FRONTENDMODULENAME** is a unique name for the Angular application (corresponds to the name value from the package.json configuration). Used by the framework to identify the package.
- **FRONTENDPLUGINNAME** is the unique module name for the Angular application (corresponding to the value from the plugin.module.ts configuration).

After entering all the necessary fields, click the save button, and the system will open a new page for editing the plugin (/console/services/id/edit). To successfully register the plugin, the user must enter the required fields and load the package of the NuGet application depending on the selected type of plugin to create and click save.

IMPORTANTLY! Before you download a package, you can compile it based on a pre-generated project, such as a project template for a Model-driven application.

To autogenerate the source code, use the Generate API project button. When clicked, the system generates a project in the Visual Studio solution format based on a file with entity models uploaded by the user. Differences in different types of projects can be found in the documentation on Github.



BankFeed

Alias: bf

Service type: APIUI

General

ENTITY.EXTERNALLINK*

SHORTINFO

The automating the handle of bank statements due to recognizing different attributes in the statement, to identify manually the unrecognized only

LOGO SMALL

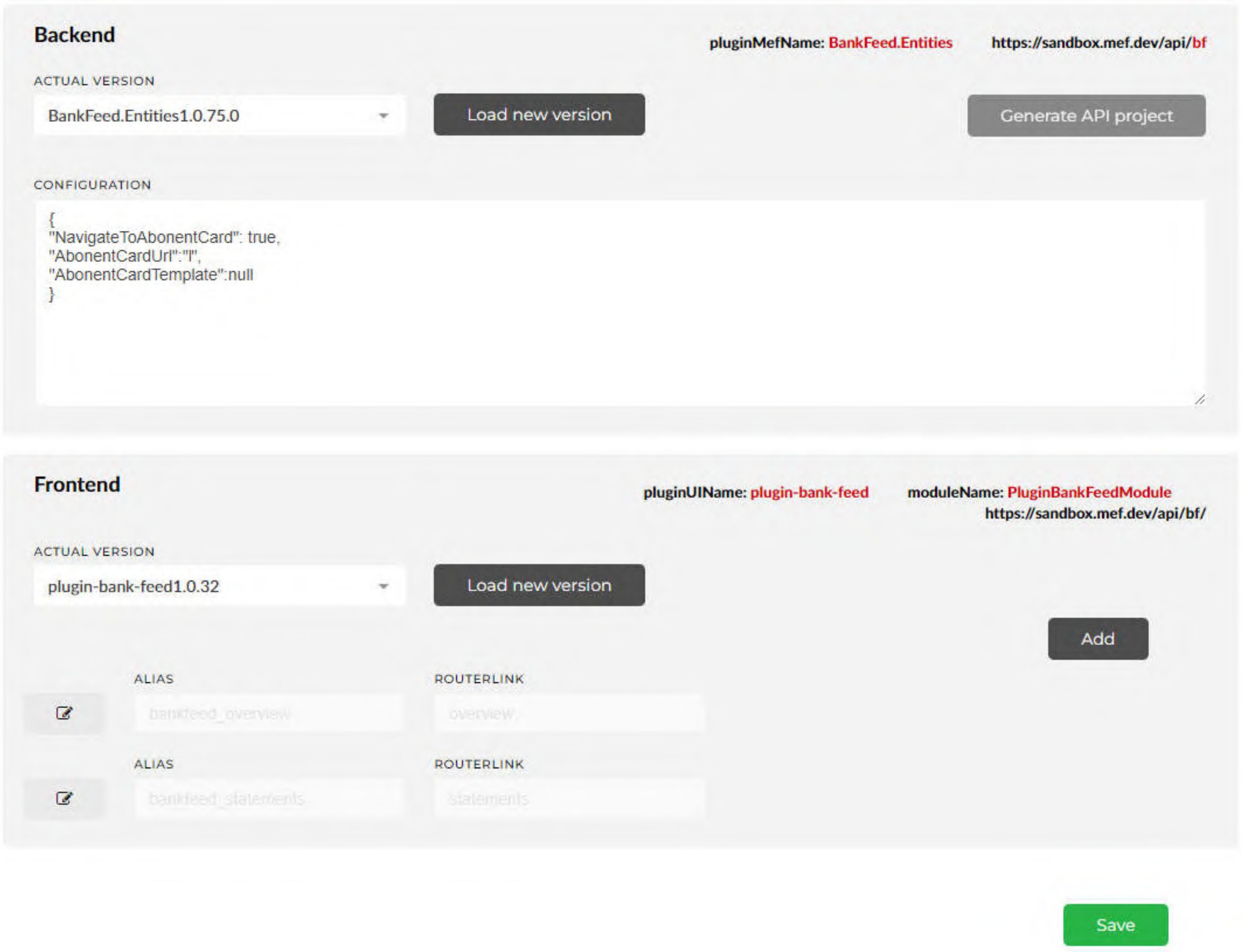
LOGO STANDARD

ENTITY.INFO

Dependensis

The GENERAL information block contains information for information support of users when navigating in the store, in particular:

- **ENTITY. EXTERNALLINK** —A link to a source of additional information about the app. For Canvas or Portal apps, it can contain a direct link to the Web App's start page while waiting for quick navigation to the app from the Store.
- **SHORTINFO** – a short description of the application of no more than 150 characters.
- **LOGO SMALL** – application icon for indication on the dependencies tab of the application description
- **LOGO STANDARD** - app icon to display in the store
- **ENTITY.INFO** – an extended description of the application, may contain hyperlinks and pictures



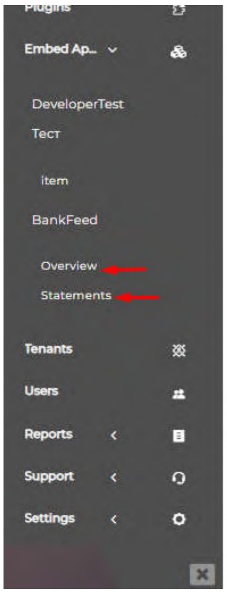
Backend
pluginMefName: BankFeed.Entities
https://sandbox.mef.dev/api/bf
ACTUAL VERSION
BankFeed.Entities1.0.75.0
Load new version
Generate API project
CONFIGURATION

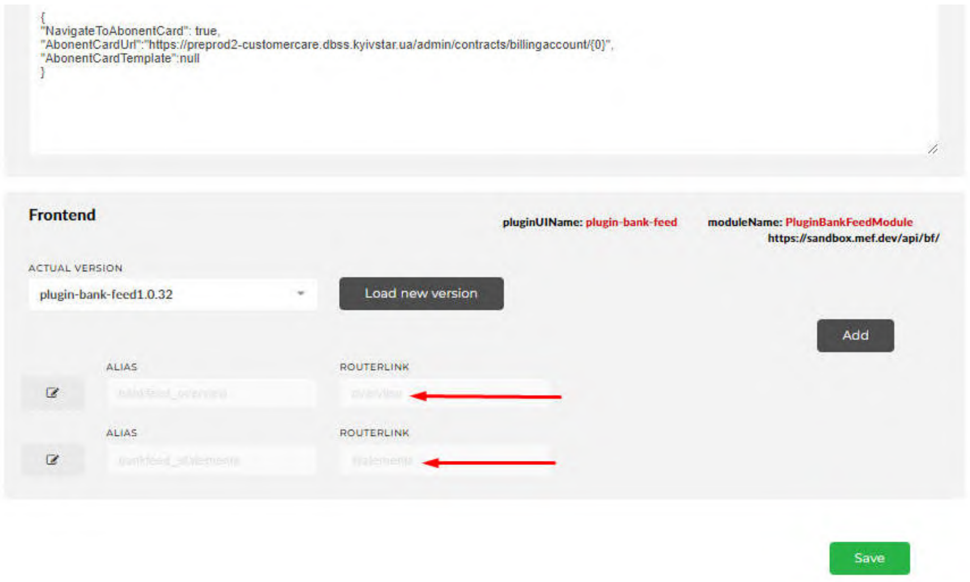
```
{
"NavigateToAbonentCard": true,
"AbonentCardUrl":"I",
"AbonentCardTemplate":null
}
```

Frontend
pluginUIName: plugin-bank-feed
moduleName: PluginBankFeedModule
https://sandbox.mef.dev/api/bf/
ACTUAL VERSION
plugin-bank-feed1.0.32
Load new version
Add

ALIAS	ROUTERLINK
bankfeed_overview	overview
bankfeed_statements	statements

For interactive plug-ins, it is essential to display them in the left side menu, for this purpose **there are attributes ALIAS** and **ROUTERLINK** - they can be specified in different languages.



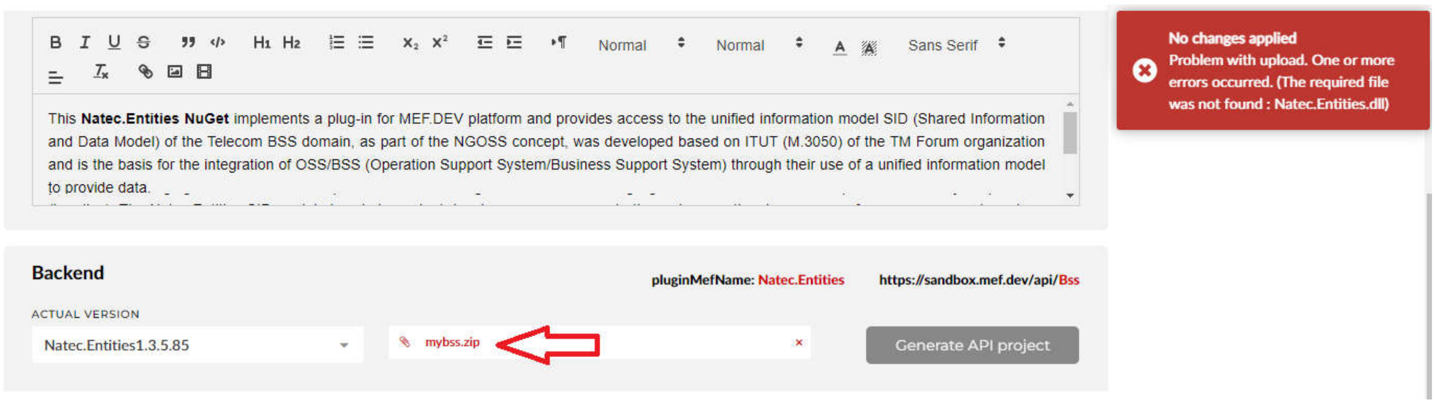


```
{
"NavigateToAbonentCard": true,
"AbonentCardUrl":"https://preprod2-customercare.dbss.kyivstar.ua/admin/contracts/billingaccount/{0}",
"AbonentCardTemplate":null
}
```

Frontend
pluginUIName: plugin-bank-feed
moduleName: PluginBankFeedModule
https://sandbox.mef.dev/api/bf/
ACTUAL VERSION
plugin-bank-feed1.0.32
Load new version
Add

ALIAS	ROUTERLINK
bankfeed_overview	overview
bankfeed_statements	statements

After filling in all the necessary information, you need to click the **SAVE** button. In case of errors in filling or building plug-ins (incorrectness of the loaded project), the system will display an error message.



This **Natec.Entities NuGet** implements a plug-in for MEF.DEV platform and provides access to the unified information model SID (Shared Information and Data Model) of the Telecom BSS domain, as part of the NGOSS concept, was developed based on ITUT (M.3050) of the TM Forum organization and is the basis for the integration of OSS/BSS (Operation Support System/Business Support System) through their use of a unified information model to provide data.

No changes applied
Problem with upload. One or more errors occurred. (The required file was not found : Natec.Entities.dll)

Backend

pluginMefName: Natec.Entities https://sandbox.mef.dev/api/Bss

ACTUAL VERSION

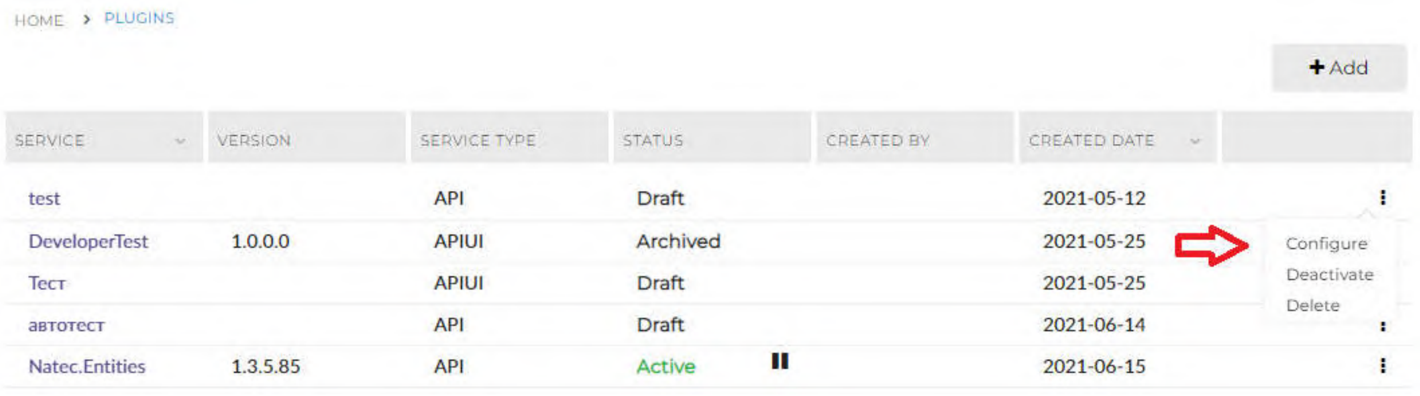
Natec.Entities1.3.5.85

mybss.zip

After successfully saving the plugin, the following plugin management functions from the context menu of the table become available to the user:

- **Send for review** - used to send the plugin for verification for further display in the store and use by other users of the platform
- **Stop** - used to terminate access to the plugin for all platform users temporarily. After stopping, this function is inverted in Run to resume access
- **Deactivate** - used to permanently terminate access to the plugin to all users of the platform
- **Delete** - used to delete the draft plugin

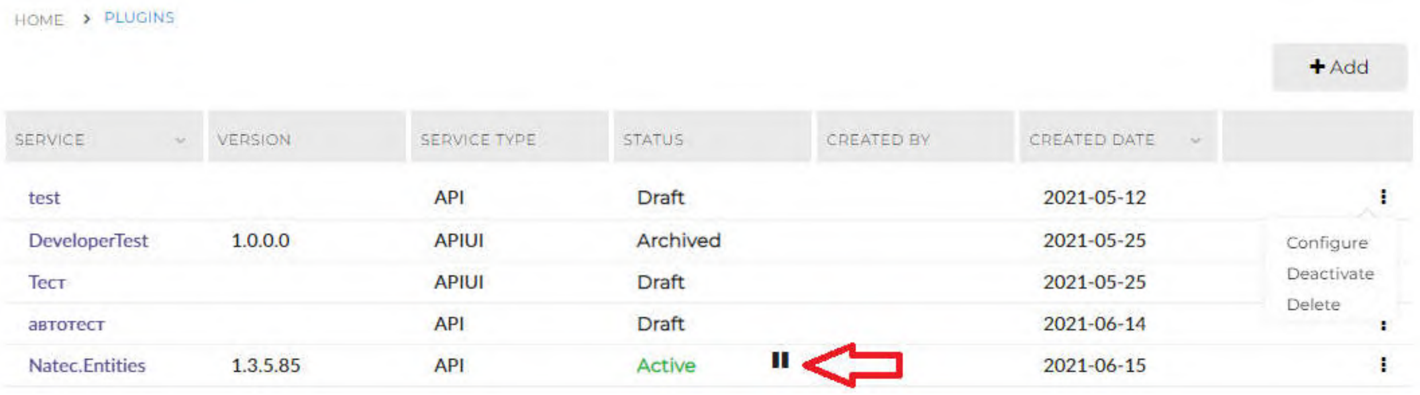
To edit the desired plugin, you need to click on three dots and select the Configure menu item, after which the system will open the plugin editing page.



SERVICE	VERSION	SERVICE TYPE	STATUS	CREATED BY	CREATED DATE
test		API	Draft		2021-05-12
DeveloperTest	1.0.0.0	APIUI	Archived		2021-05-25
Тест		APIUI	Draft		2021-05-25
автотест		API	Draft		2021-06-14
Natec.Entities	1.3.5.85	API	Active		2021-06-15

To remove the desired plugin, you need to click on three dots and select the Uninstall menu item, after which the system will uninstall the plugin.

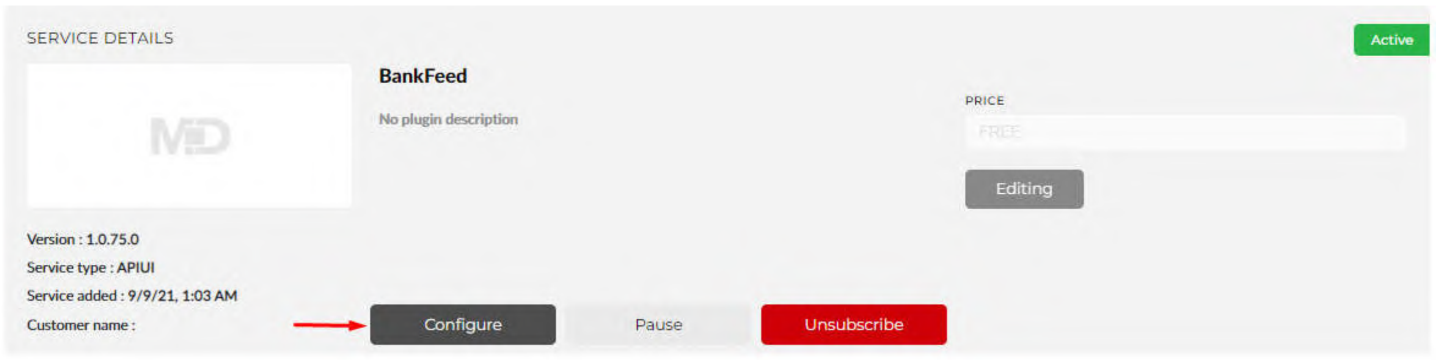
You can change the plugin's status from **Active** to **Suspended** and back again by clicking on the button in the Status column.



SERVICE	VERSION	SERVICE TYPE	STATUS	CREATED BY	CREATED DATE
test		API	Draft		2021-05-12
DeveloperTest	1.0.0.0	APIUI	Archived		2021-05-25
Тест		APIUI	Draft		2021-05-25
автотест		API	Draft		2021-06-14
Natec.Entities	1.3.5.85	API	Active		2021-06-15

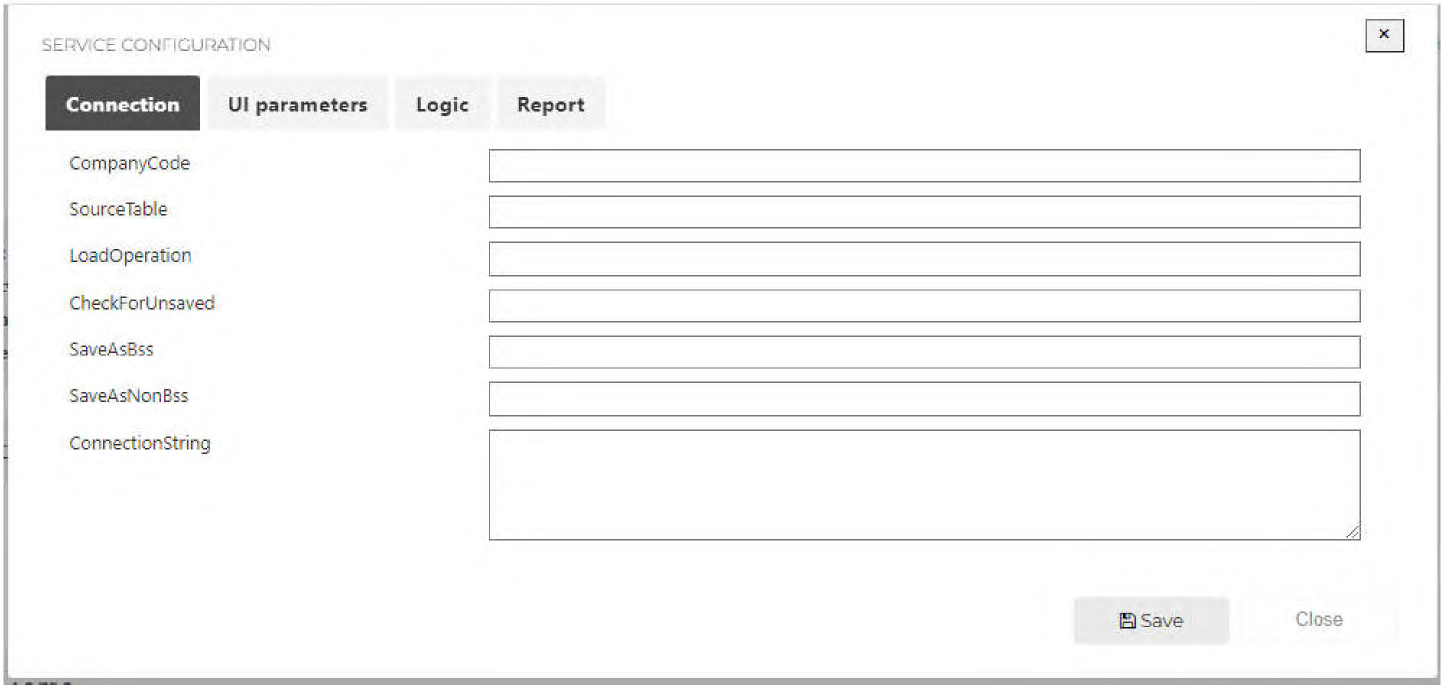
Managing Plugins (/console/services)

This page is used to manage the plugin, namely to unsubscribe from the plugin (remove from the tenant), configure the plugin for use in the tenant, and understand the plugin's current status.



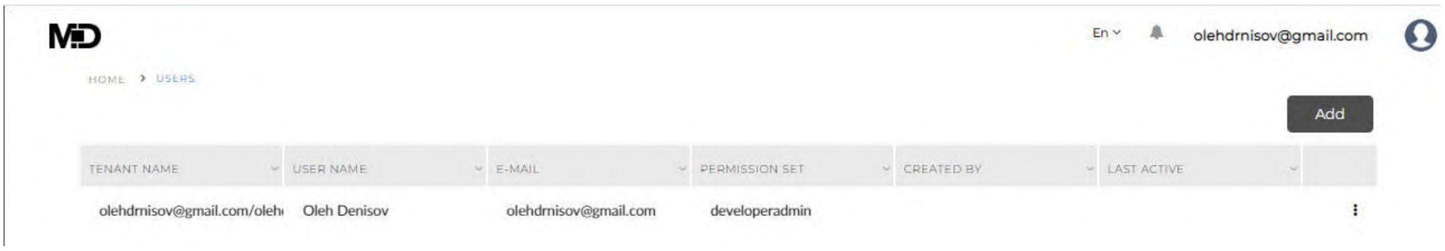
Version : 1.0.75.0
Service type : APIUI
Service added : 9/9/21, 1:03 AM
Customer name :

To display the configuration of the plugin, the **CONFIGURE** button is used. It is essential to understand that the presence and structure of the configuration are determined by the developer of the plugin - thereby following the developer's recommendations when configuring the plug-in.



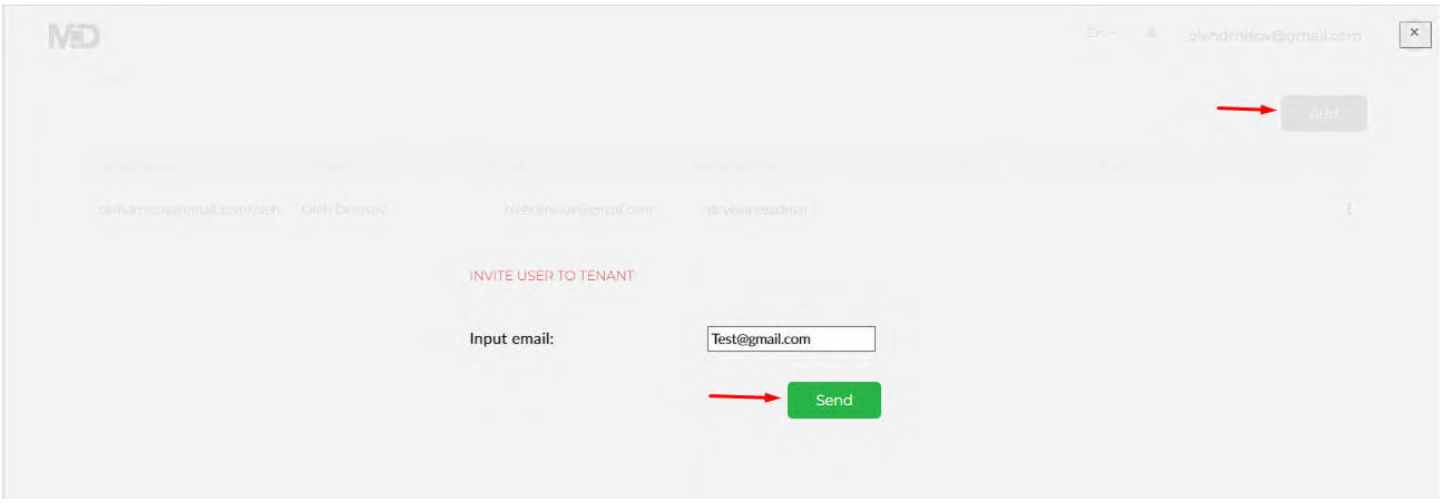
Users (/console/manage/users)

Use this page to manage this tenant's users and add to the tenant.



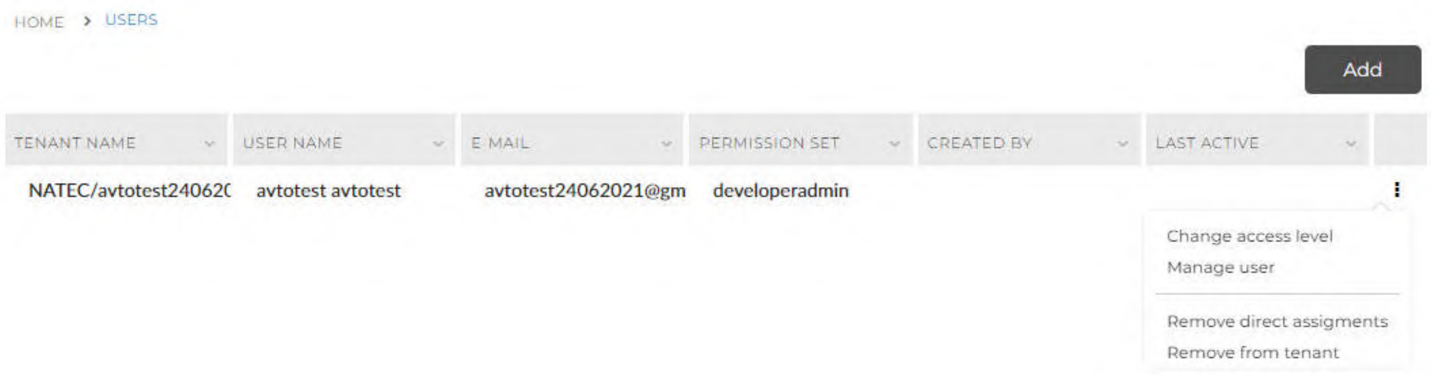
TENANT NAME	USER NAME	E-MAIL	PERMISSION SET	CREATED BY	LAST ACTIVE
olehdrnisov@gmail.com/oleh	Oleh Denisov	olehdrnisov@gmail.com	developeradmin		

To add a new user, he must be pre-registered on the platform. If the user is not registered on the platform, you can send him an invitation with your referral link - in this case, he will receive an e-mail with a link to join the platform.



User control in the section of the tenant is carried out by clicking on three dots in the table line, while the following actions are available:

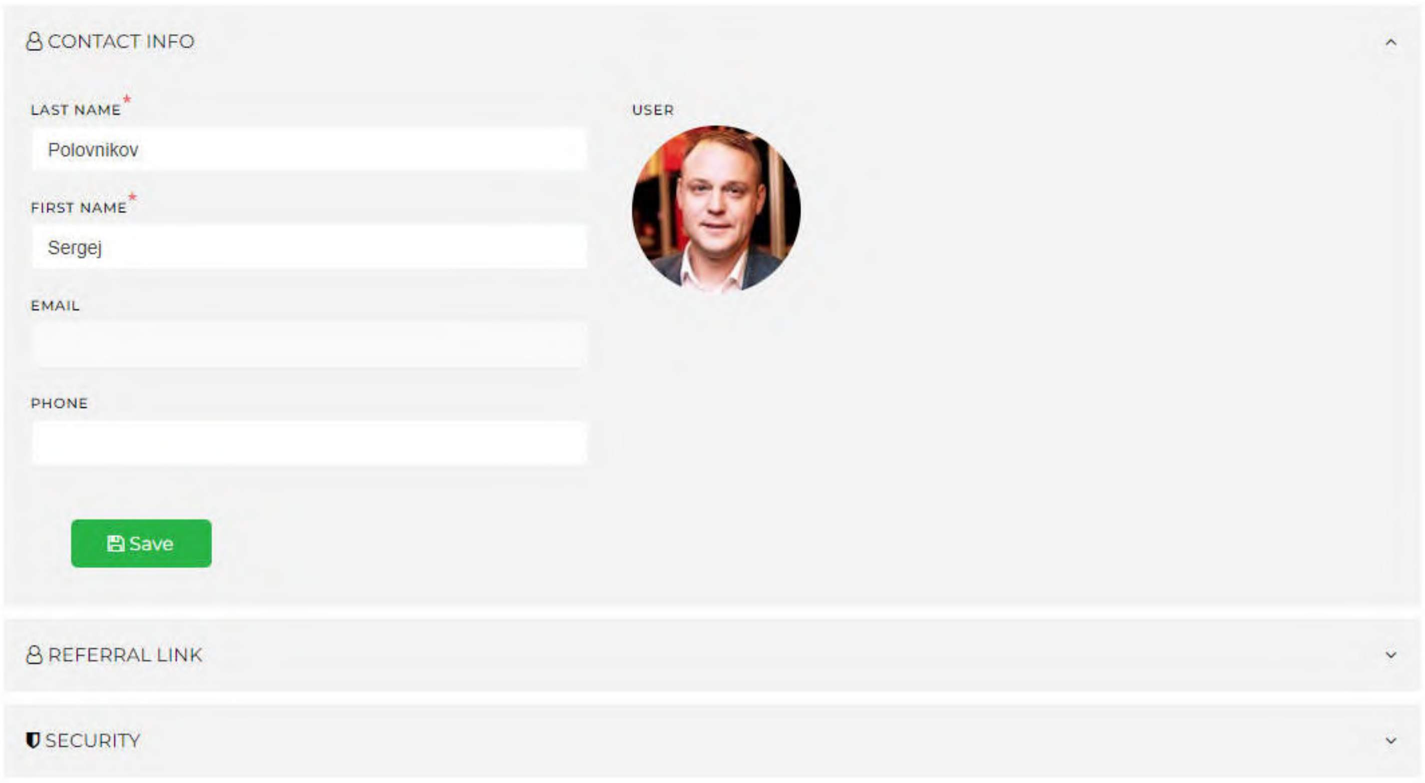
- **Change access level** - Manage user access in this tenant. It is essential to understand that you cannot manage user rights in other tenants where you are not an administrator. Also, you cannot change the role yourself.
- **Manage user** – Manage a user's subscription to apps from a tenant
- **Remove direct assignments** - disconnect all assigned applications from this user
- **Remove from tenant** – The option excludes the user from the tenant. It is essential to understand that the user himself is not removed from the platform. Also, you cannot remove yourself from your tenant.



TENANT NAME	USER NAME	E MAIL	PERMISSION SET	CREATED BY	LAST ACTIVE
NATEC/avtotest24062(	avtotest avtotest	avtotest24062021@gm	developeradmin		

Console/settings/profile

This page is used to view, edit, delete an account and get a referral link on the platform. Also, this page contains contact information for the user and his referrer (the one who was invited to the platform)


CONTACT INFO

LAST NAME*
Polovnikov

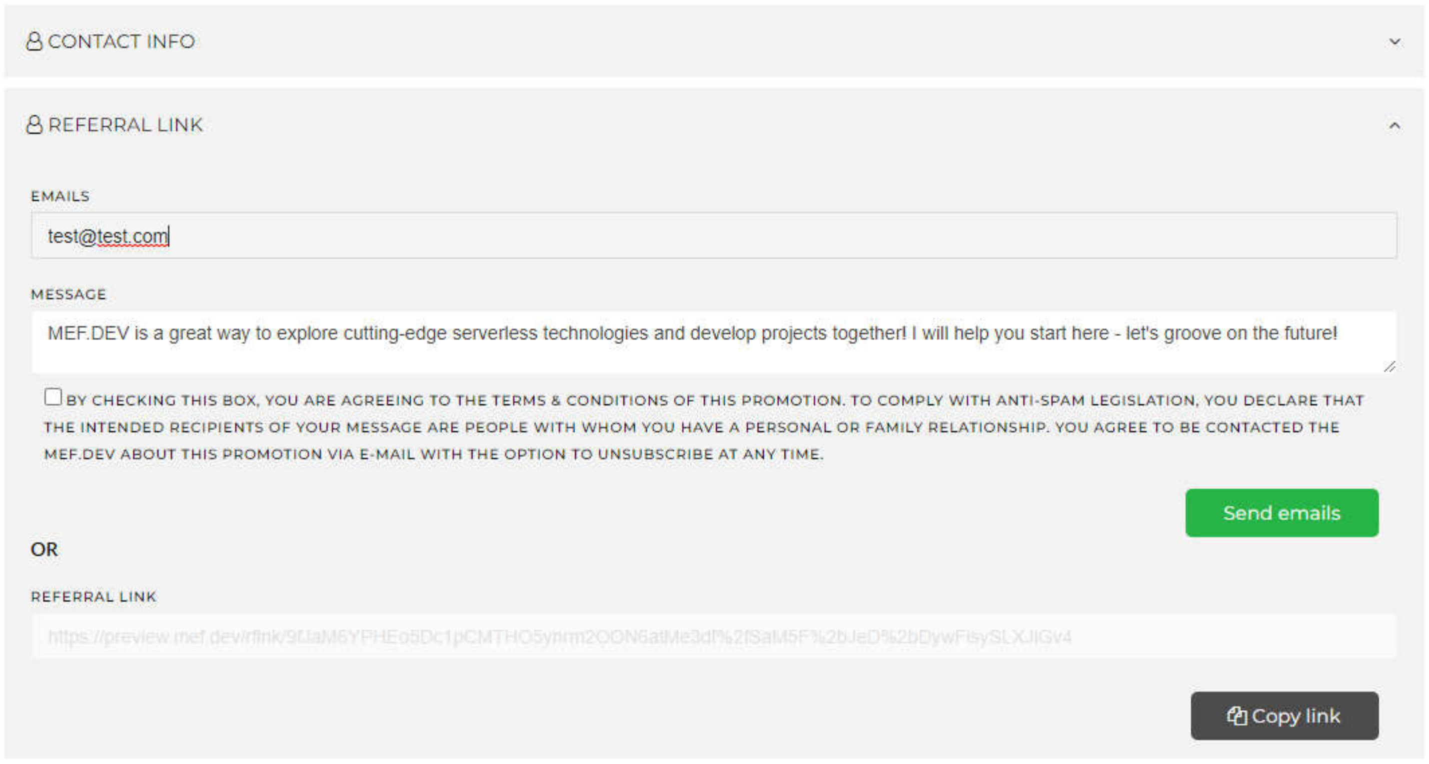
FIRST NAME*
Sergej

EMAIL

PHONE

USER

To receive a referral link or send an invite with a referral link via email, you need to disclose the **REFFERAL LINK** section



EMAILS
test@test.com

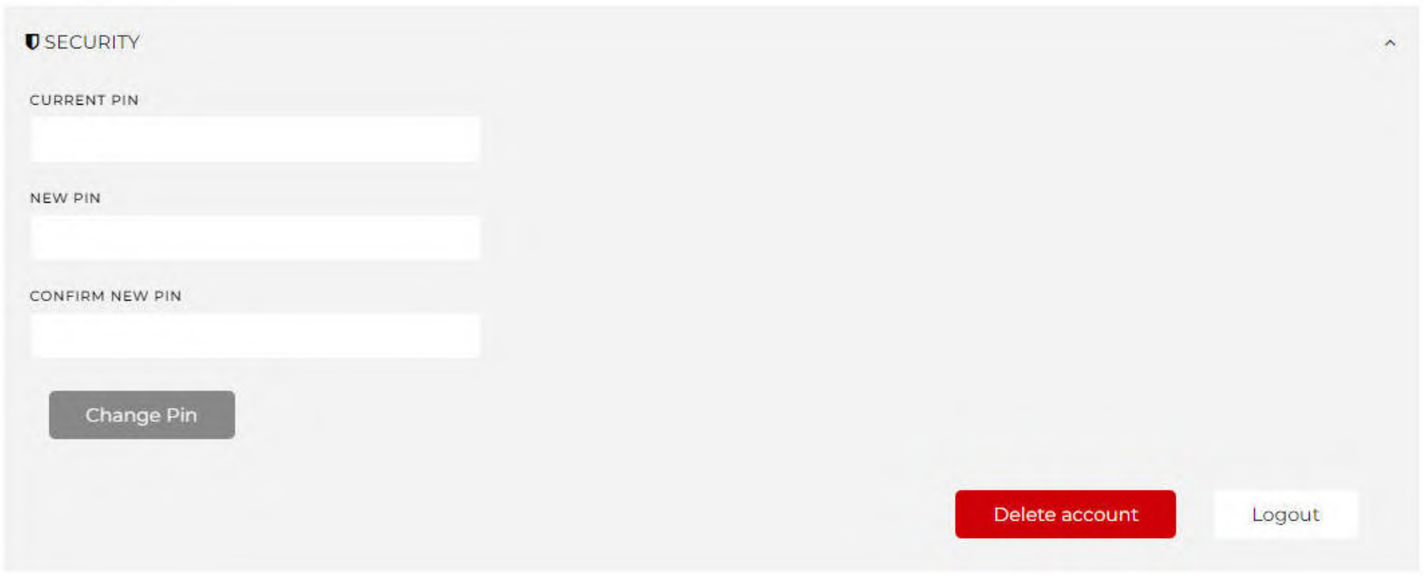
MESSAGE
MEF.DEV is a great way to explore cutting-edge serverless technologies and develop projects together! I will help you start here - let's groove on the future!

☐ BY CHECKING THIS BOX, YOU ARE AGREEING TO THE TERMS & CONDITIONS OF THIS PROMOTION. TO COMPLY WITH ANTI-SPAM LEGISLATION, YOU DECLARE THAT THE INTENDED RECIPIENTS OF YOUR MESSAGE ARE PEOPLE WITH WHOM YOU HAVE A PERSONAL OR FAMILY RELATIONSHIP. YOU AGREE TO BE CONTACTED THE MEF.DEV ABOUT THIS PROMOTION VIA E-MAIL WITH THE OPTION TO UNSUBSCRIBE AT ANY TIME.

OR

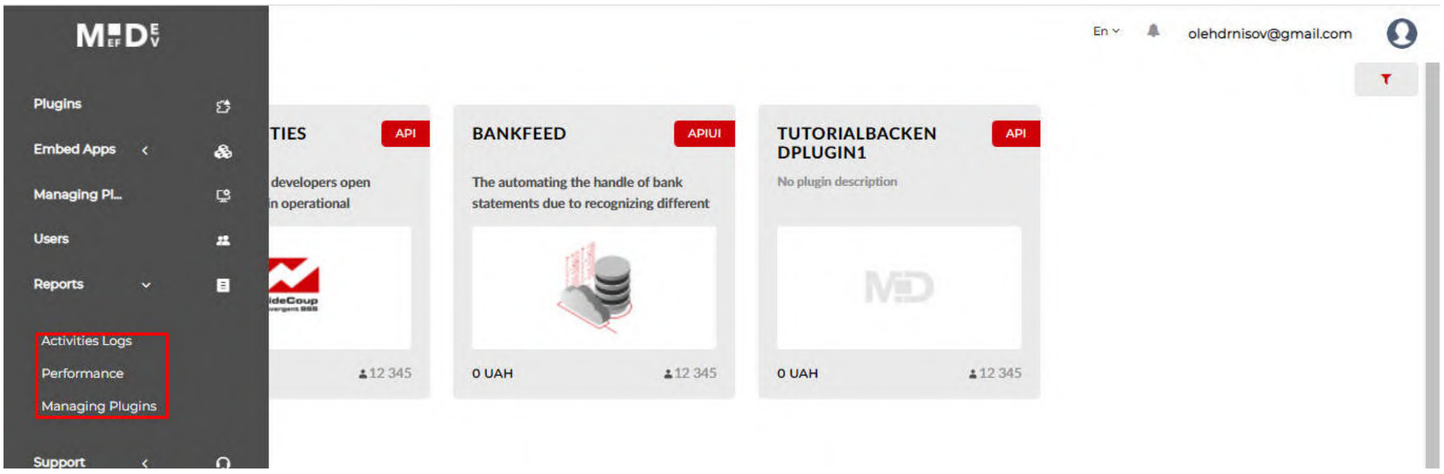
REFERRAL LINK
https://preview.mef.dev/rflink/9fJaM6YPHEo5Dc1pCMTHO5ynm2OON6atMe3dl%2fSaM5F%2bJeD%2bDywFisySLXJIGv4

To delete your account, you need to open the **SECURITY** section

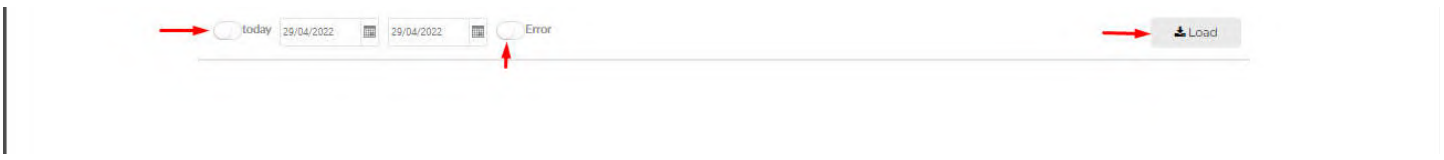


Reports section (/console/reports/...)

This section contains **the Activities**, **Performance**, and **Services** pages, which reflect various reports on application usage in a given tenant. That is used to monitor events in plug-ins and control access to plug-ins.



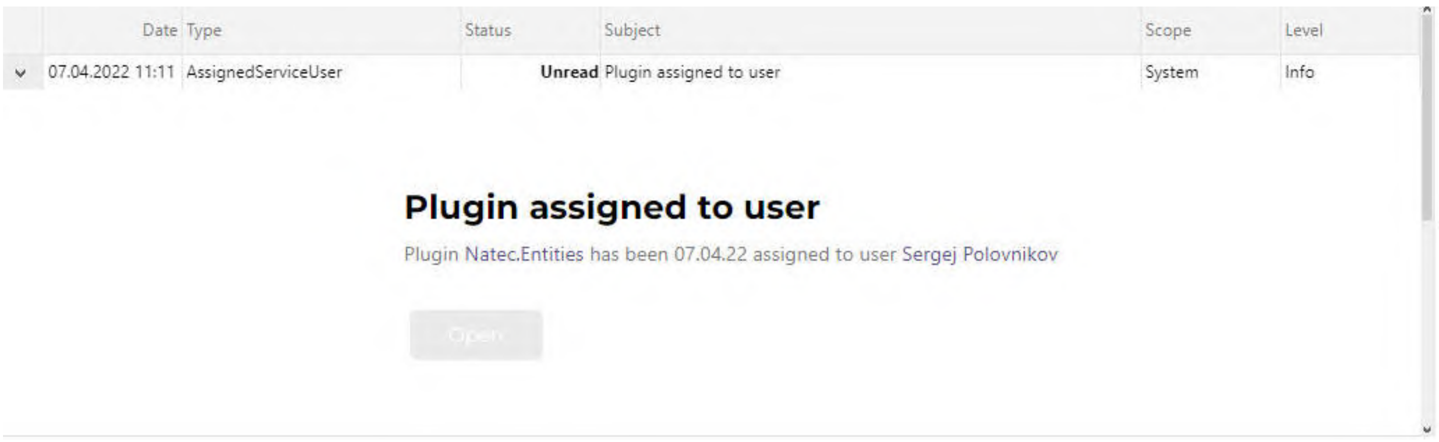
Each of the reports allows you to filter - you can specify the period and the feature to display error data. Clicking the **Load** button shows the data.



Support section (/console/support/...)

This section contains **the Messages/list** and **Notifications** pages used to display messages from the platform and view notification events generated by the platform.

Platform notifications are sent to the specified email of the user, and are also available for viewing on the **Notifications** page

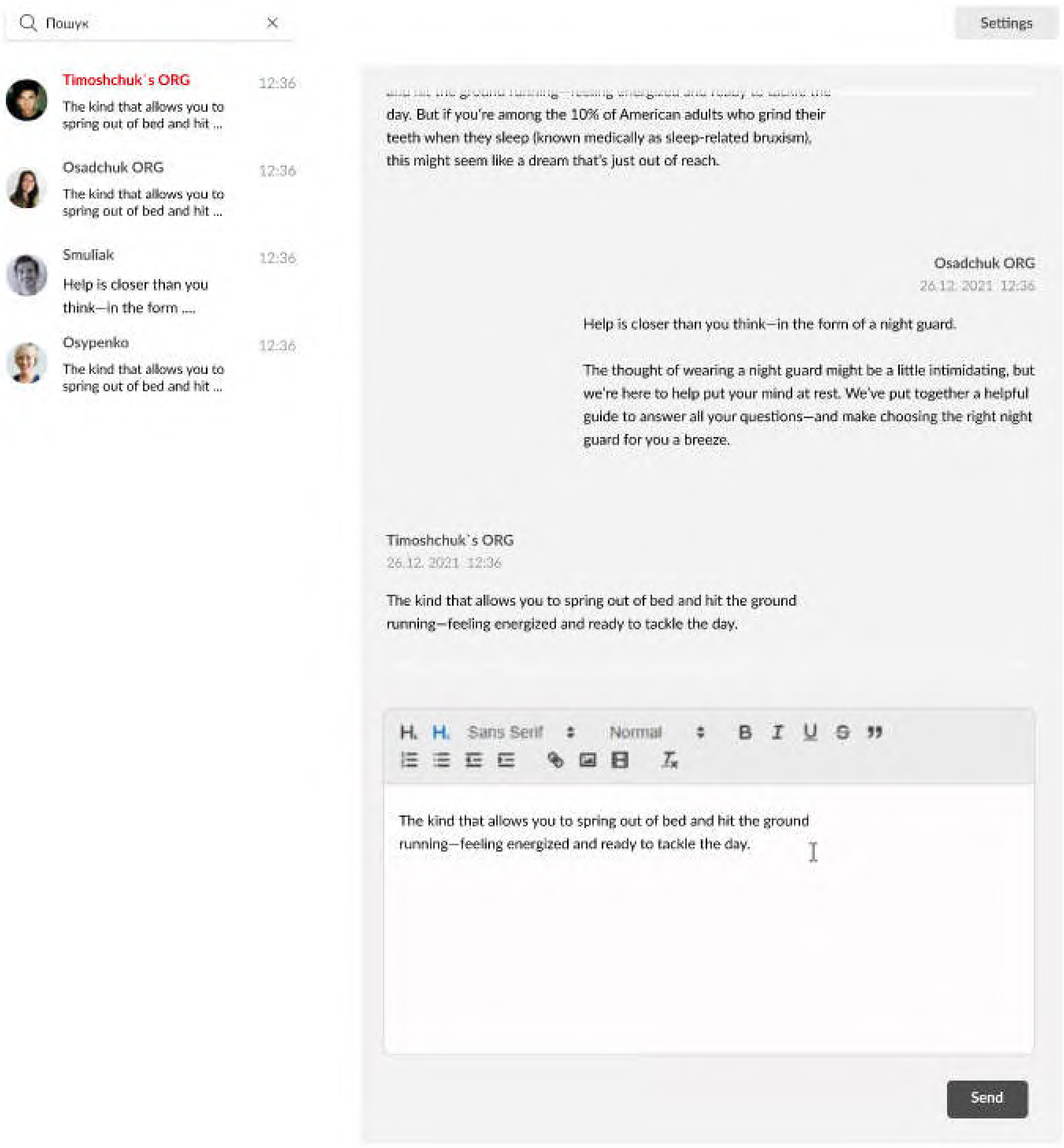


	Date	Type	Status	Subject	Scope	Level
⌄	07.04.2022 11:11	AssignedServiceUser	Unread	Plugin assigned to user	System	Info

Plugin assigned to user

Plugin Natec.Entities has been 07.04.22 assigned to user Sergej Polovnikov

User requests to the platform are available for viewing on the **Messages/list** page and reflect the communication history within each specific request



Timoshchuk`s ORG 12:36
The kind that allows you to spring out of bed and hit ...

Osadchuk ORG 12:36
The kind that allows you to spring out of bed and hit ...

Smuliak 12:36
Help is closer than you think—in the form ....

Osypenko 12:36
The kind that allows you to spring out of bed and hit ...

day. But if you're among the 10% of American adults who grind their teeth when they sleep (known medically as sleep-related bruxism), this might seem like a dream that's just out of reach.

Osadchuk ORG
26.12. 2021 12:36

Help is closer than you think—in the form of a night guard.

The thought of wearing a night guard might be a little intimidating, but we're here to help put your mind at rest. We've put together a helpful guide to answer all your questions—and make choosing the right night guard for you a breeze.

Timoshchuk`s ORG
26.12. 2021 12:36

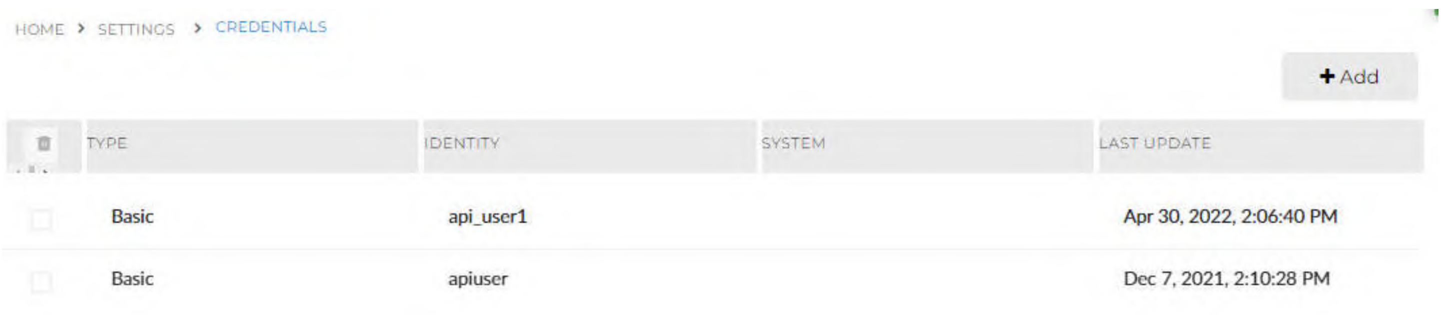
The kind that allows you to spring out of bed and hit the ground running—feeling energized and ready to tackle the day.

The kind that allows you to spring out of bed and hit the ground running—feeling energized and ready to tackle the day.

/console/settings/...) section

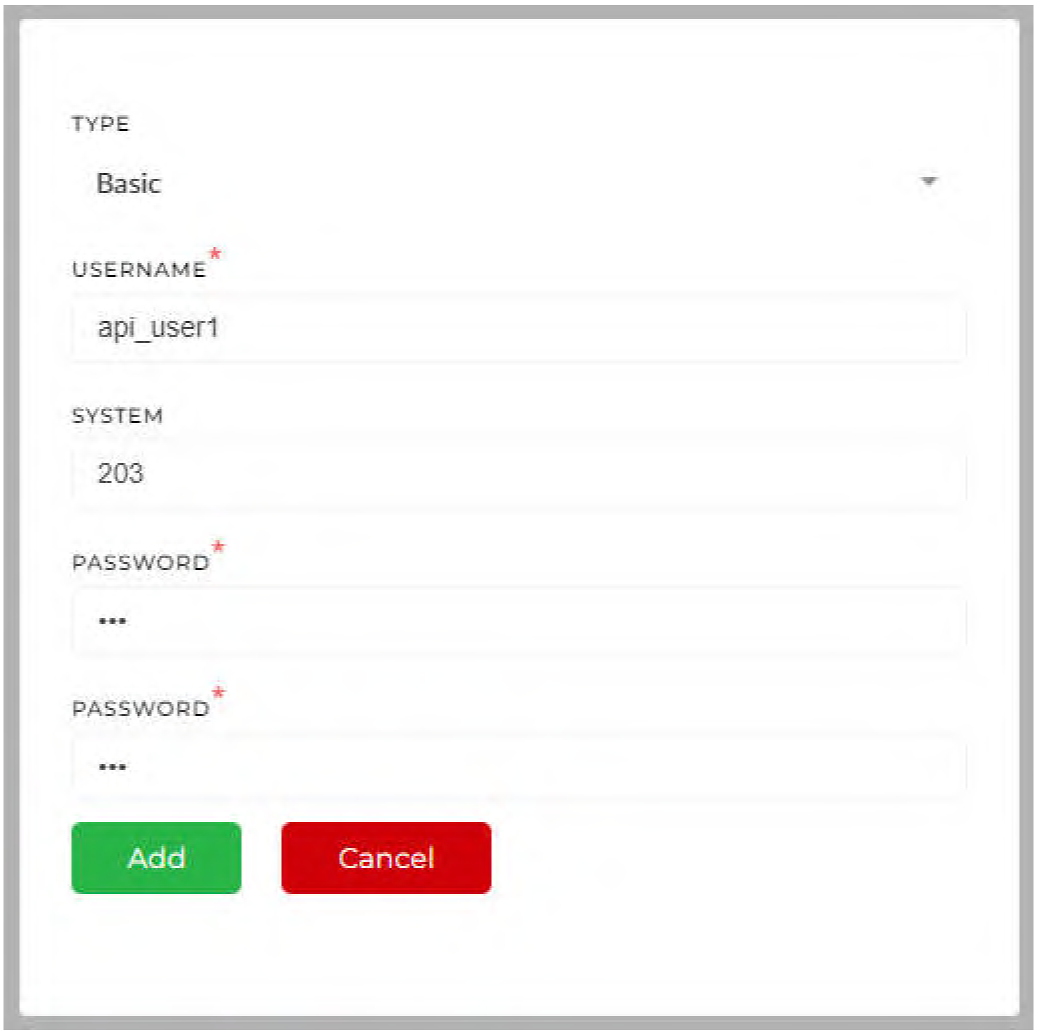
This section contains **a Redirects** page for managing credentials for accessing server applications through API requests, duplicates **the Profile** menu item, and a **Tenant** page for changing the tenant if a user participates in multiple tenants.

By going to the Credentials page (/console/settings/credentials), the user can create new or cancel existing credentials, for example, for Basic authorization of the new API.



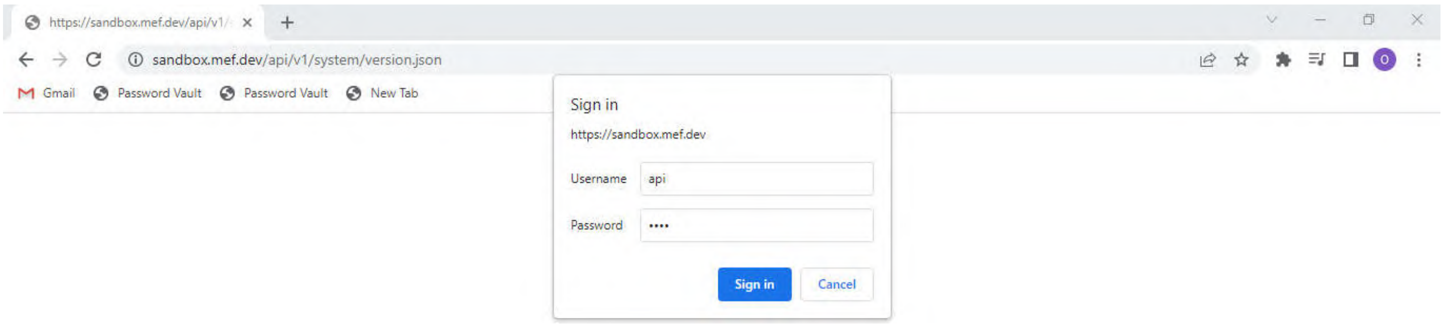
	TYPE	IDENTITY	SYSTEM	LAST UPDATE
	Basic	api_user1		Apr 30, 2022, 2:06:40 PM
	Basic	apiuser		Dec 7, 2021, 2:10:28 PM

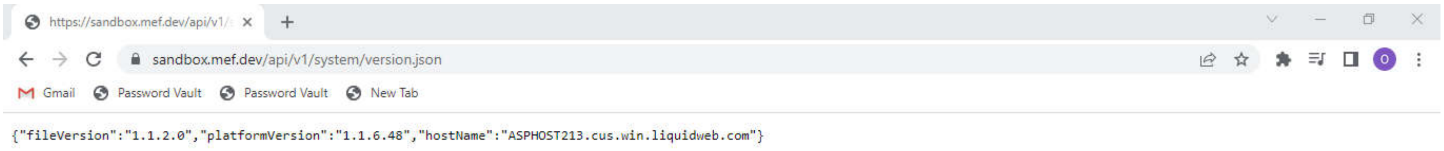
To create a new record, click the Add button and specify the requested fields in the pop-up window.



TYPE	Basic
USERNAME*	api_user1
SYSTEM	203
PASSWORD*	•••
PASSWORD*	•••

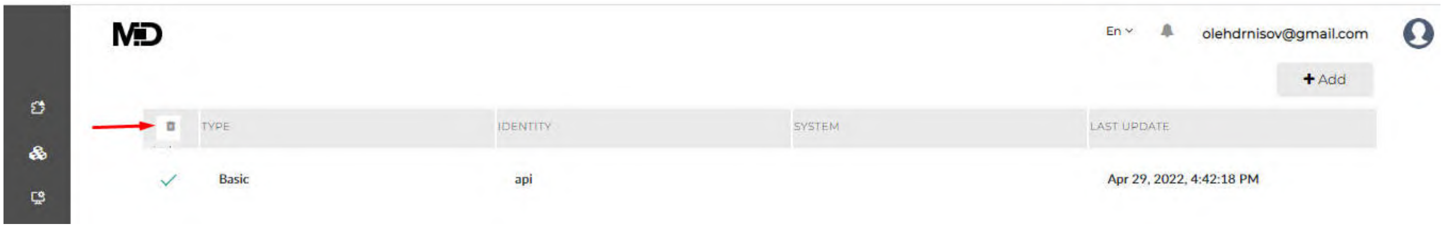
You can check the generated Credentials by inserting the URI of the plug-in access, or the platform version checker public resource https://sandbox.mef.dev/api/v1/system/version.json, into the browser and entering these access parameters.




https://sandbox.mef.dev/api/v1/
sandbox.mef.dev/api/v1/system/version.json
Gmail
Password Vault
Password Vault
New Tab

```
{"fileVersion":"1.1.2.0","platformVersion":"1.1.6.48","hostName":"ASPHOST213.cus.win.liquidweb.com"}
```

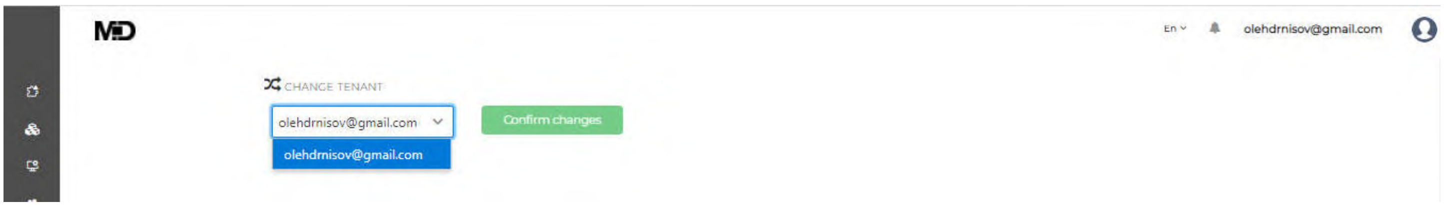
IMPORTANTLY! The password cannot be changed; there is only the option to delete and re-create.



	TYPE	IDENTITY	SYSTEM	LAST UPDATE
✓	Basic	api		Apr 29, 2022, 4:42:18 PM

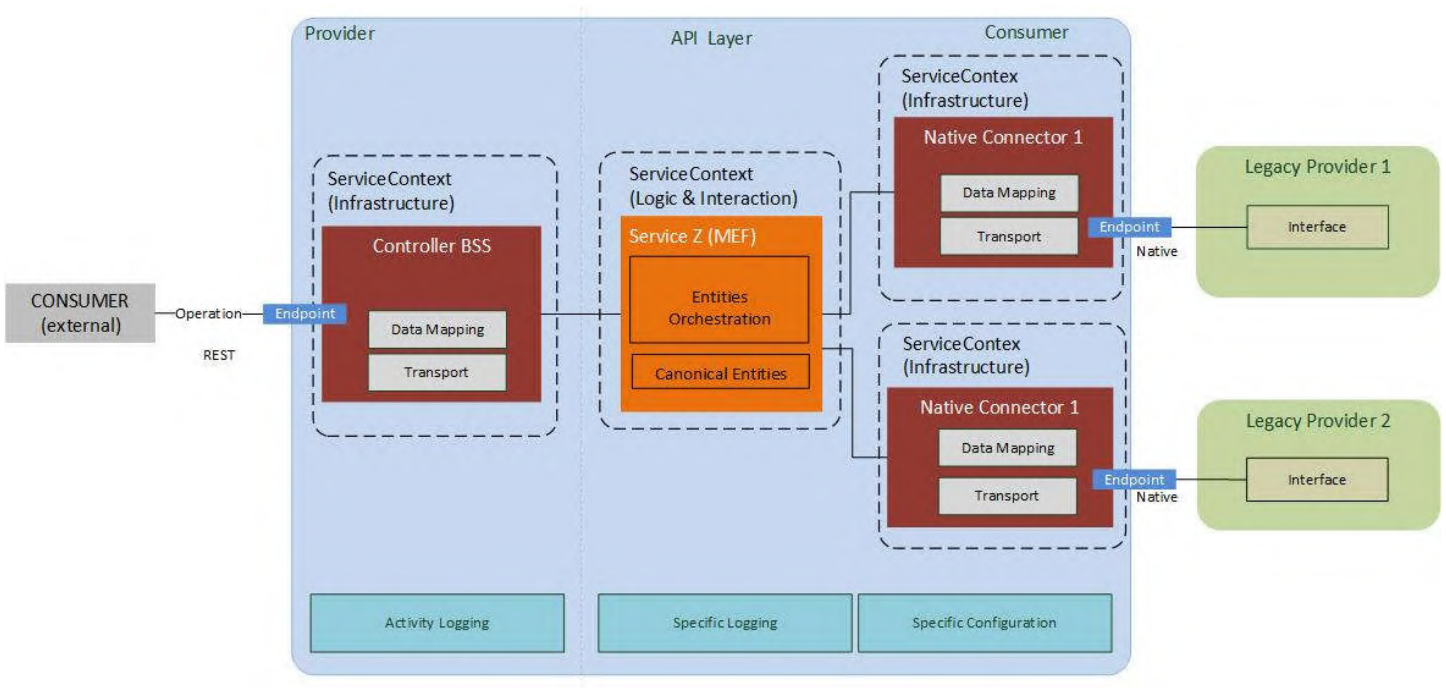
Switch between tenants

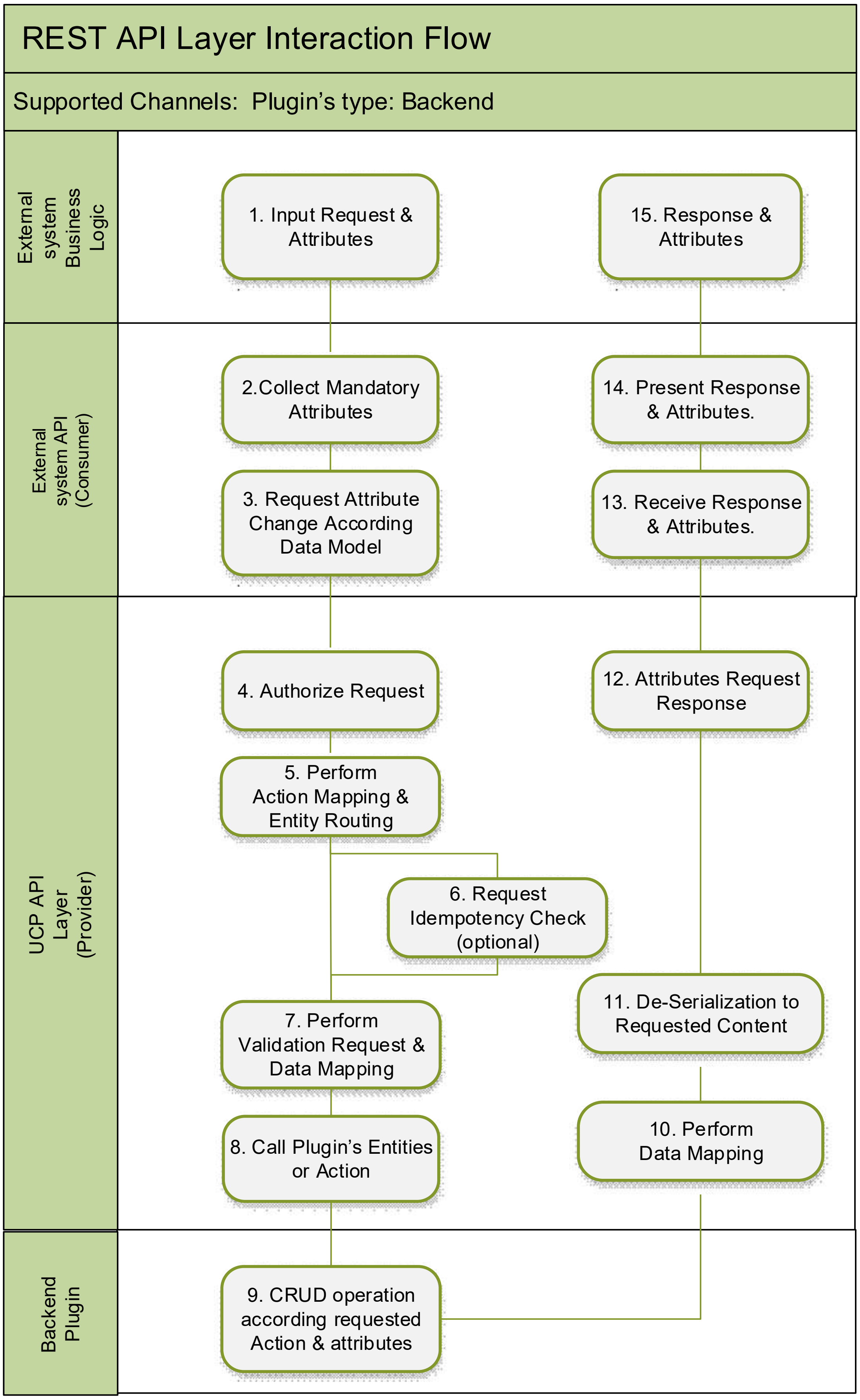
On the Current tenant (/console/settings/tenant) page, a user can switch to a different tenant if the user has access to other tenants.



Background Legacy Interaction

In addition to the advantages of reusing IoC containers, the platform allows external use of plug-ins through an automatically generated REST interface for plug-ins. This approach makes it possible to quickly integrate "old" systems without the need for refinement on their side.



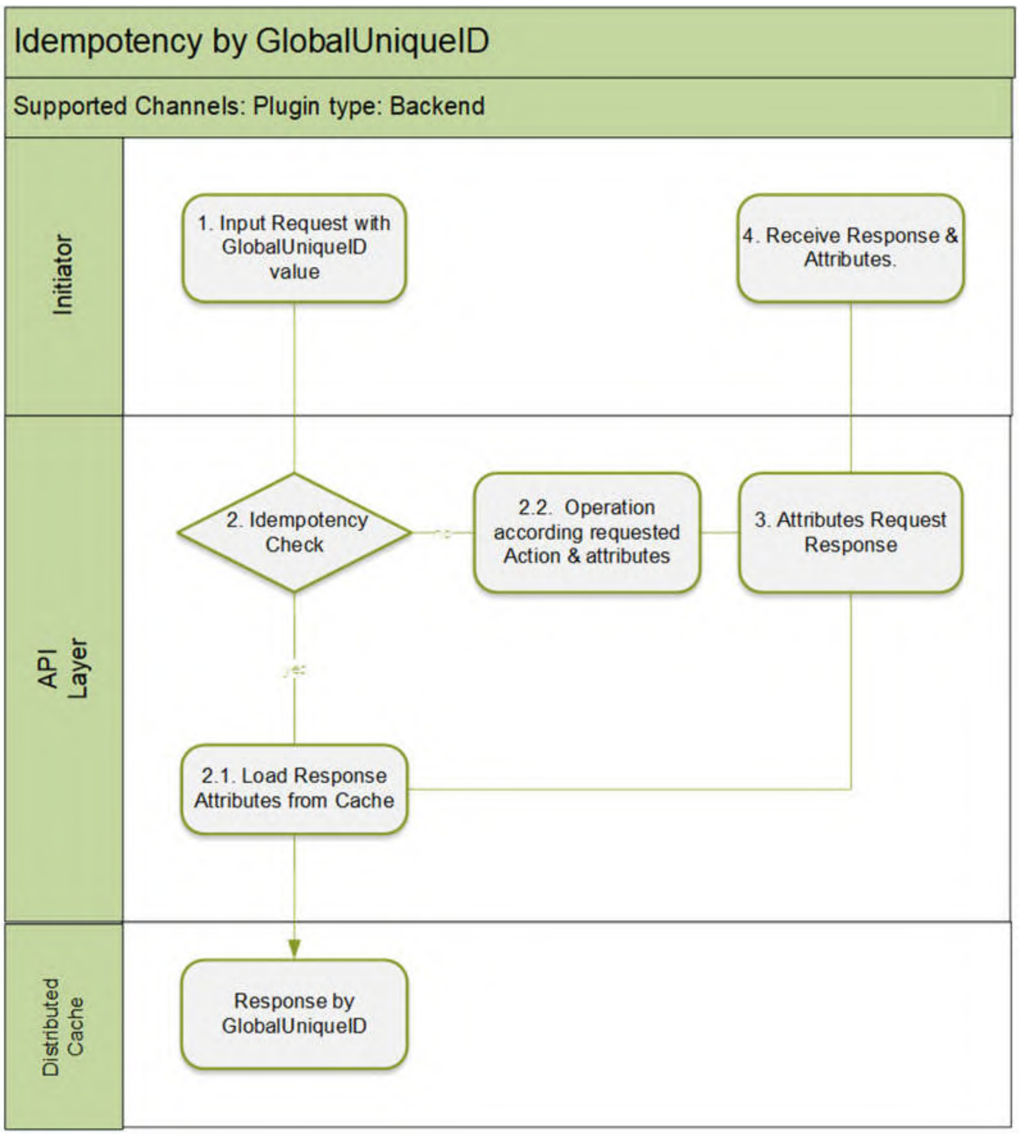

REST API Layer Interaction Flow
Supported Channels: Plugin's type: Backend
External system Business Logic
External system API (Consumer)
UCP API Layer (Provider)
Backend Plugin
1. Input Request & Attributes
2.Collect Mandatory Attributes
3. Request Attribute Change According Data Model
4. Authorize Request
5. Perform Action Mapping & Entity Routing
6. Request Idempotency Check (optional)
7. Perform Validation Request & Data Mapping
8. Call Plugin's Entities or Action
9. CRUD operation according requested Action & attributes
10. Perform Data Mapping
11. De-Serialization to Requested Content
12. Attributes Request Response
13. Receive Response & Attributes.
14. Present Response & Attributes.
15. Response & Attributes

Service Flow

Layer API: Implements rest HTTP interface, authorization, caching, and call idempotency.
Backend Plugin: implements business logic, such as CRUD operations for entities

Idempotency

The global_unique_id option is used to avoid duplication of data insertion. Each successful CRUD operation, including for the POST method, is stored in the platform's distributed cache. In case of receiving a second request, the API returns a previously saved response from the cache. The request is identified as repeated by the value of the global_unique_id.


Idempotency by GlobalUniqueID
Supported Channels: Plugin type: Backend
Initiator
API Layer
Distributed Cache
1. Input Request with GlobalUniqueID value
4. Receive Response & Attributes.
2. Idempotency Check
2.2. Operation according requested Action & attributes
3. Attributes Request Response
2.1. Load Response Attributes from Cache
Response by GlobalUniqueID

API Authorization

For authorization on API methods, support for various types of authorization is implemented in 2 modes of platform deployment - **Internet** and **Intranet**; the supported authorization types for each type of deployment are listed below:

Internet:

- Basic Authorization
- oAuth 2.0

Intranet:

- URL (token)
- Basic Authorization
- Kerberos
- NTLM