
Article

[Sergey Mikhailenko](#) · Nov 17, 2020 5m read

[Open Exchange](#)

Organization of message notification and provision of information to users of the messenger Telegram using two bots.

Hello developers!

I submitted for the contest a solution that would allow organizing message notifications and providing information to users of the messenger Telegram. This is achieved by creating two bots: an administrator and an informant. The administrator manages the list of users connected to the bot informant and their access rights.

The Solution based on project

<https://github.com/intersystems-community/TelegramAlerts>

During the installation and configuration process, we will create an informant bot and an admin bot, which will allow the informant bot to provide users with the requested content.

Installation with ZPM

```
zpm:USER>install appmsw-telestat
```

Installation with Docker

Prerequisites

Make sure you have [git](#) and [Docker desktop](#) installed.

Installation

Clone/git pull the repo into any local directory

```
$ git clone https://github.com/SergeyMi37/appmsw-telestat.git
```

Open the terminal in this directory and run:

```
$ docker-compose build
```

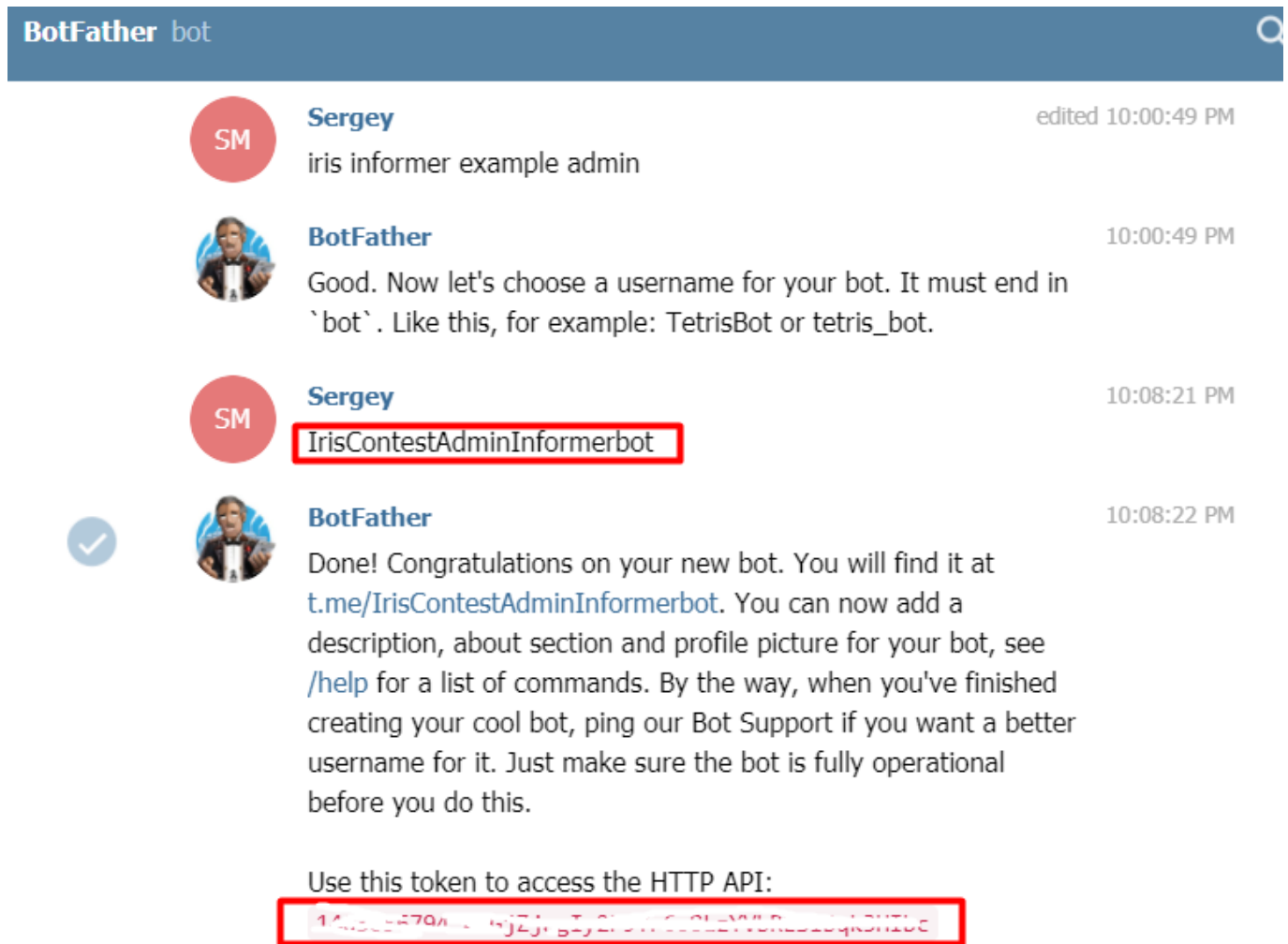
3. Run the IRIS container with your project:

```
$ docker-compose up -d  
$ docker-compose exec iris iris session iris
```

Before starting the installation

If you do not have a Telegram messenger account, then this is just a reason to create it.

So, let's create an administrator bot. To do this, find BotsFather, join it and enter the /newbot command.



Then we will create an informant bot and save their names and tokens to enter when installing the module using the package manager.

Open IRIS terminal:

```
USER>
USER>zpm
zpm:USER>install appmsw-telestat
```

Or if from docker

```
zpm:USER>load /opt/irisapp

[appmsw-telestat]      Module object refreshed.
[appmsw-telestat]      Validate START
[appmsw-telestat]      Validate SUCCESS
[appmsw-telestat]      Compile START
[appmsw-telestat]      Compile SUCCESS
[appmsw-telestat]      Activate START
[appmsw-telestat]      Configure START
Shall we enter names and tokens ? [y,n] <y> y
```

```
Enter the username of the admin bot IrisContestAdminInformerbot
Enter the token to access of the admin bot 794:AAGjZjag9Yr6LzYVbRESIBqk3HIbc
Enter the number phone of the admin bot 77777777 [77777777]
Enter the username of the informer bot IrisContestInformerbot [IrisContestInformerbot
]
Enter the token to access of the informer bot sY5NxkS0QXRXdjxmbrIWJWLOA
```

If you make a mistake, or BotFather changed the token, you can always retry later by performing a utility `##class(appmsw.telestat.API.util).Init()`
Product items changed successfully

```
[appmsw-telestat]      Configure SUCCESS
[appmsw-telestat]      Activate SUCCESS
```

The program memorized names, tokens and phone numbers in the table `appmsw.telestat.Bots`
After that, we will open the product and launch it.

Configuring a bot admin

Find the created admin bot in Telegram, connect to it and execute the command `/start`
And we will show him our phone number. Exactly the one that we entered during the installation.
If all is well, we will receive a message:
Your number has been successfully accepted OK

otherwise:
Your number is not included in the allowed table. Check the correctness of the initial data

Now `ChatId` is attached to the administrator's phone.
You can test notifications to the Admin bot with a command in the terminal

```
user>zwrite ##class(appmsw.telestat.API.util).ToAdmin("Contest")
```

Configuring a bot informant

We will find the informant in the telegram created by the bot and connect to it by pressing the START button.
The product service will prepare a message and also offer to show the phone number.

The bot admin will receive a message about sending the phone, and by selecting the Allow or Deny buttons, you will make a decision that will come in response to the bot informant.

But even if access to information was denied, `ChatId` and the phone number were entered in the `appmsw.telestat.Bots` table and the bot informant can be sent a message using the utility

```
user>zwrite ##class(appmsw.telestat.API.util).ToInformer("7971111111",,, "Hello don't
be sad")
```

The configuration work has been completed and you can see what is available out of the box.

For bot admin, if you enter `/start`

iris informer example admin

Bot administration service for tracking Ensemble and IRIS servers. Can take commands:

/GetLastAlerts - Get last alerts. Server: 'hp-msw'

/ServersStatus - Get a list of monitored instances

/Userlist - Get a list of users receiving information about servers and their status

For the administrator bot, it is possible to view and edit user attributes with the /Userlist command

Another command /GetLastAlerts is implemented more as an example.

And the appmsw.telestat.TelegramServiceAlert service is configured to periodically check system messages and if their level of importance is more than 2, display them to all users connected to the bot informant who have the notification field set to yes

The list of commands and content is expanding. It is enough to create your own class similar to appmsw.telestat.API.commands And a method GetCommands For a list of commands and GetAlerts To get content on them.

Teams and content can be differentiated between users by groups.

This solution has been configured and tested in Long polling mode. But it can be configured in Webhook mode too.

[#Interoperability](#) [#InterSystems](#) [IRIS](#)

[Check the related application on InterSystems Open Exchange](#)

Source

URL: <https://community.intersystems.com/post/organization-message-notification-and-provision-information-users-messenger-telegram-using-two>