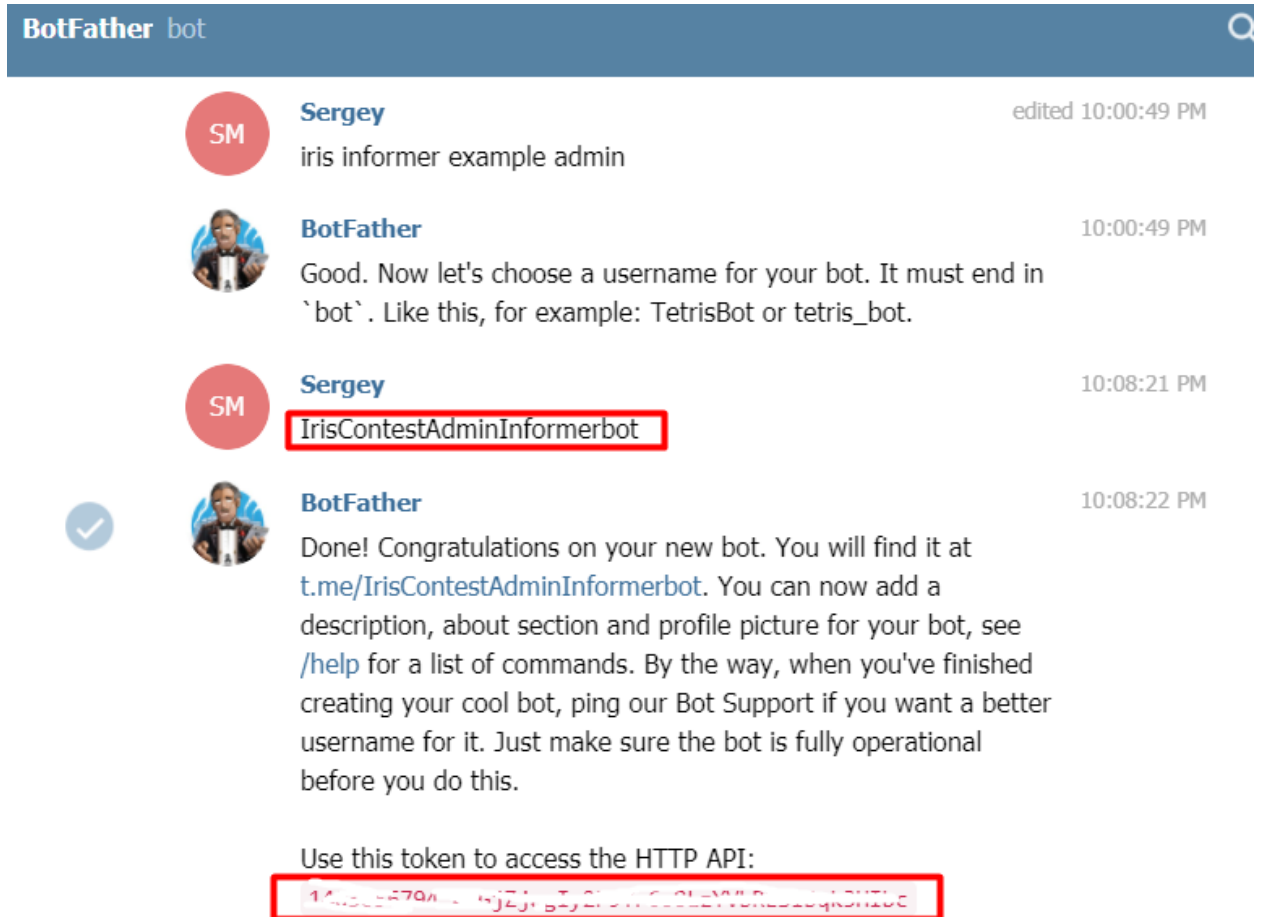


Before starting the installation

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 a bot bot and save their tokens for input when installing the module using the package manager.

You are creating two bots. The first is the admin and the second is the informer. The first one will be used only by you as an administrator, and the name of the second bot can be distributed to those users whom you allow to monitor the use of your service.

You are creating two bots. The first is the admin and the second is the informer. The first one will be used only by you as an administrator, and the name of the second bot can be given to those persons whom you assign the right to use your service.

Open IRIS terminal:

```
USER>
USER>zpm
zpm:USER>install appmsw-telealerts
Or if from docker
```

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

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

```
Enter the token to access of the admin bot 123:456zzzzzzzzz  
Enter the number phone of the admin bot 79771234567  
Enter the token to access of the informer bot 123:789xxxxxxx
```

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

The program memorized names, tokens and phone numbers in the table `appmsw.telestat.Bots`

Name	Alias	Group	Address	Name Inst	Name User	Body	Name Instance	Name	Instance	State	Subsignature	ID	ID Key	Instance

To register an IRIS instance that we will monitor, you need to follow the procedure:
Do `##class(appmsw.telestat.API.inst).EditInst("test2021",1,"_system","SYS","http://iris-test:52773")`

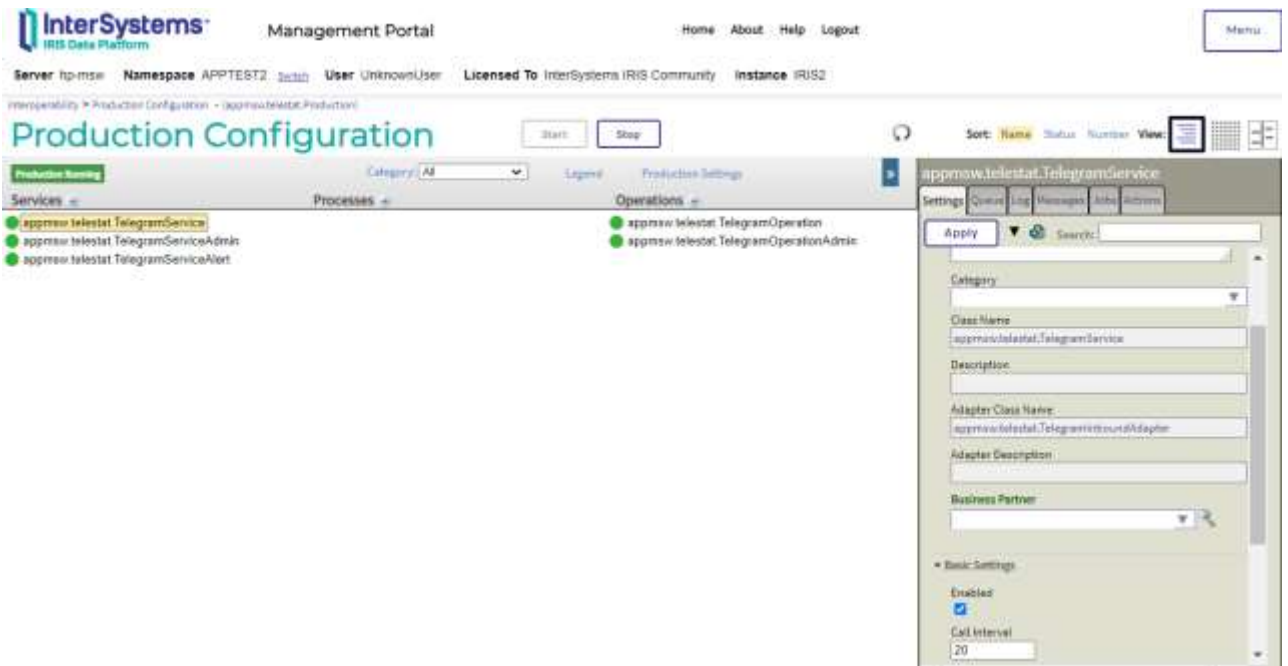
There may be several of them.

If you need to exclude from the monitoring list, then enter 0 in the second parameter:

Do `##class(appmsw.telestat.API.inst).EditInst("test2021",0,"_system","SYS","http://iris-test:52773")`

All instances that are included in the list must have the package installed `apptools-admin`

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 it our phone number. Exactly the one that we entered during the installation. This must be done from a mobile phone, and the number is not transmitted from the browser version of the messenger.

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 `apmsw.telestat.Bots` table and the bot informant can be sent a message using the utility

```
user>zwrite ##class(apmsw.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:

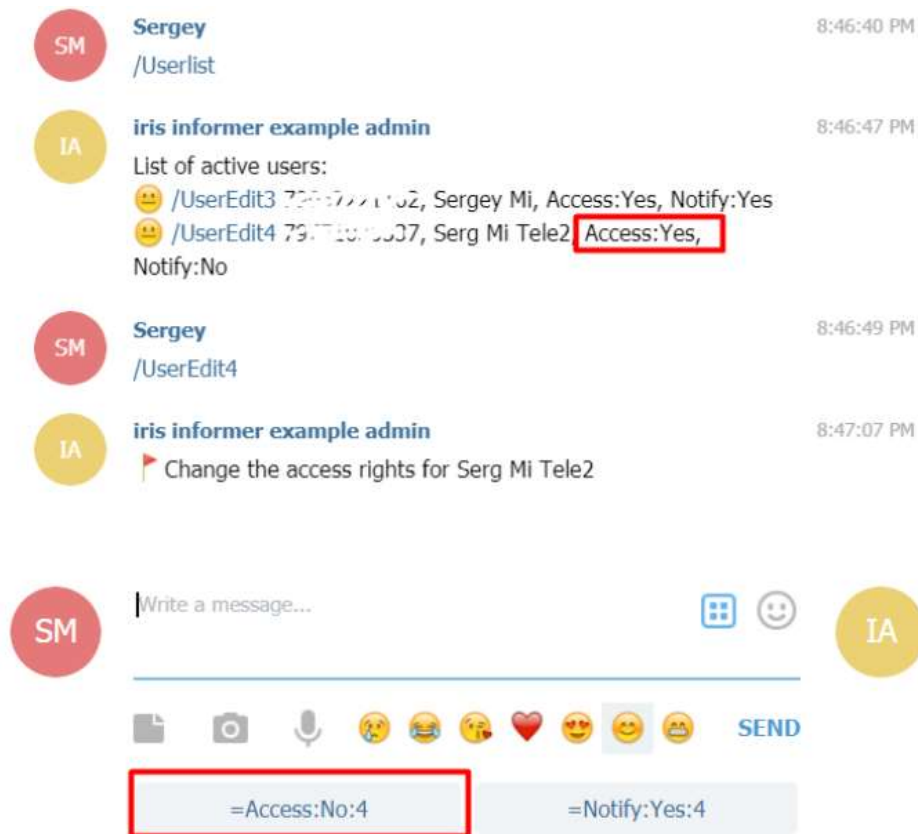
```
/GetLastAlerts - Get last alerts. Server: 'hp-msw'
```

```
/ServersStatus - Get a list of monitored instances
```

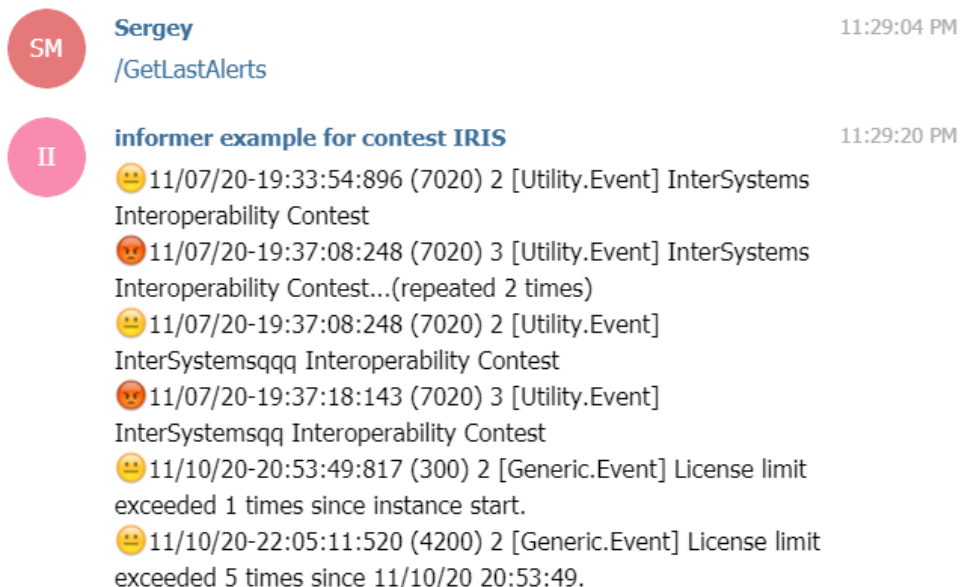
```
/Userlist - Get a list of users receiving information about servers and their status
```

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

```
the /Userlist command
```



Another command `/GetLastAlerts` will print the last 10 messages from the messages.log from each instance.



And the `appmsw.telestat.TelegramServiceAlert` service is configured to periodically check system messages and, if their significance level is more than 2, display them to all users connected to the informant bot who have the right to receive notifications.

Let's check this by executing the terminal method:

```
do ##class(%SYS.System).WriteToConsoleLog("InterSystems Iris Test" ,,3)
```

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. SSL configuration is created automatically.