
Article

[Nikolay Solovyev](#) · Apr 25, 2018 4m read

Sending Alerts from Ensemble via Telegram

The features of this API

allow you to create bots with a wide range of functionality including receiving payments.

With the help of the telegram bot, I solved a simple task - sending Alerts from Ensemble to Telegram.

Advantages: Alerts come to the mobile phone, a notification appears, so there is no need to install any additional applications (in contrast to the solution <https://community.intersystems.com/post/sending-alerts-mobile-phone-using-pushover-httpoutboundadapter>).

The bot can be extended by adding new commands, for example, for managing the Production or for solving other tasks in Iris, Ensemble or Cache.

"Telegram" is a well-known instant messenger, which provides an API for creating bots. If you want to start by reviewing the code - [github project](#)

First about a few Telegram features you need to keep in mind when creating a bot:

- Telegram is popular due to its security. For example, when a user starts chatting with a bot - the bot does not know the phone number of the user. This is good for the user - nobody can collect telephone numbers. It is also impossible to send a message just knowing the phone number.
- Telegram supports 2 ways to send updates (those messages that users write to the bot) to your server:
 - Long polling. To receive incoming updates - call API method after necessary time intervals.
 - Webhook. Register your URL, and whenever there is an update for the bot, telegram will send an HTTPS POST request to the specified URL.

I realized both options.

How this bot works and what is done in the Ensemble

1. Any interaction begins with sending the command /start.
The reply: a text message with a greeting and a list of available commands.
2. The user sends the command /subscribe.
The reply contains a special button that allows you to request user 's phone number. This is necessary for identification the user and determine which messages need to be sent to him.

3. Clicking this button user confirms sending his phone number. Ensemble checks whether this number is at the employee table. And if so, the chat ID is saved.
4. When an error occurs in one of the Ensemble production components, Ensemble automatically creates an AlertRequest message and sends it to the business operation Ens.Alert. Business operation Ens.Alert searches all employees having a chat ID and sends a message about the problem to each of them.
5. The /alert command (which sends a test Alert) is also provided for testing.

Features

1. To store the binding of employee's phone number and chat ID in Ensemble, I chose Lookup table. The keys are the phone numbers, the values are telegram chat ID ' s. When a user subscribes to receive notifications, the chat ID is updated.
2. To receive updates, Long polling mechanism was implemented first. In Ensemble, this is very simple - just add a business service and specify the call interval. After a specified interval, Ensemble will access the Telegram API for updates.
Minus of this approach is that the user can notice the delay between sending the command and the bot response. Plus - you do not need to provide access to the Ensemble server from the Internet.
3. The implementation of webhook is not a problem, creating a REST JSON service in Cache is described many times and in details. But when implementing webhook, you need to take into account that Telegram sends messages only over HTTPS, so you need to configure SSL on your web server (I used a self-signed certificate).

The sequence of steps for adding the specified functionality in your Production

1. Create a bot using a special bot "BotFather"
<https://telegram.me/BotFather>
you will receive a token
2. Create a client SSL configuration in Ensemble, which is necessary, because the message is sent over HTTPS.
3. Import classes from <https://github.com/intersystems-community/TelegramAlerts> (package Telegram):
API.cls - class implementing methods for working with API.
TelegramOutboundAdapter.cls - outbound adapter
TelegramOperation.cls - business-operation for sending messages to Telegram
TelegramInboundAdapter.cls - inbound adapter
TelegramService.cls - business service
Msg.TextRequest.cls, Msg.ButtonRequest.cls - messages classes
AlertOperation.cls - business-operation for sending Alerts
RESTBroker.cls - REST-broker for webhook
4. Create a Lookup table in the Ensemble portal

(<https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=ECONFIGotherlookuptables>) and add one or more rows to it, using a phone number (without +) as a key - the phone numbers of those who need to receive notifications.

5. Add the TelegramOperation.cls business-operation to the Production and configure it. In its settings, you must specify TelegramToken and SSLConfiguration

To receive notifications, you must perform 6 or 7.

6. If you use long polling - add the TelegramService business service to your Production and configure it - you must specify TelegramToken, SSLConfiguration, TableName (Lookup table name). Set the call interval.

7. If you use webhook

- Create a web application and specify the class Telegram.RESTBroker.cls as a broker.
- Configure SSL in the Web server.
- Register webhook - for this you need to call the SetWebhook method of Telegram.API class
- Add the TelegramService business service to your Production, specify the TableName parameter. Specify poolsize = 0.

8. Add AlertOperation.cls, with the name Ens.Alert and configure parameters: TelegramToken, SSLConfiguration, TableName.

9. Done!

[#Ensemble](#) [#InterSystems](#) [IRIS](#)

Source URL: <https://community.intersystems.com/post/sending-alerts-ensemble-telegram>