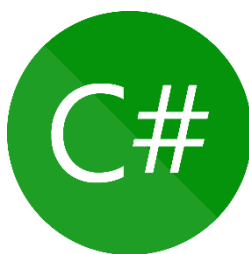


Notificaciones vía Telegram





Índice

Configuraciones de Telegram	3
Crear cuenta personal.....	3
¿Grupo o Conversación?	4
TokenKey	4
Chat ID	4
Envío de Mensajes	4
Integración con C#	5
Librería de Telegram	5
Agregar las siguientes líneas	5
Ejecutar el request.....	5
Parámetros por Main(String args[])	5
Obtener el .exe de la solución:	6
Integración con SSIS.....	7
Tabla de Configuración	7
Paquete de SSIS dtsx.....	7



Configuraciones de Telegram

Crear cuenta personal

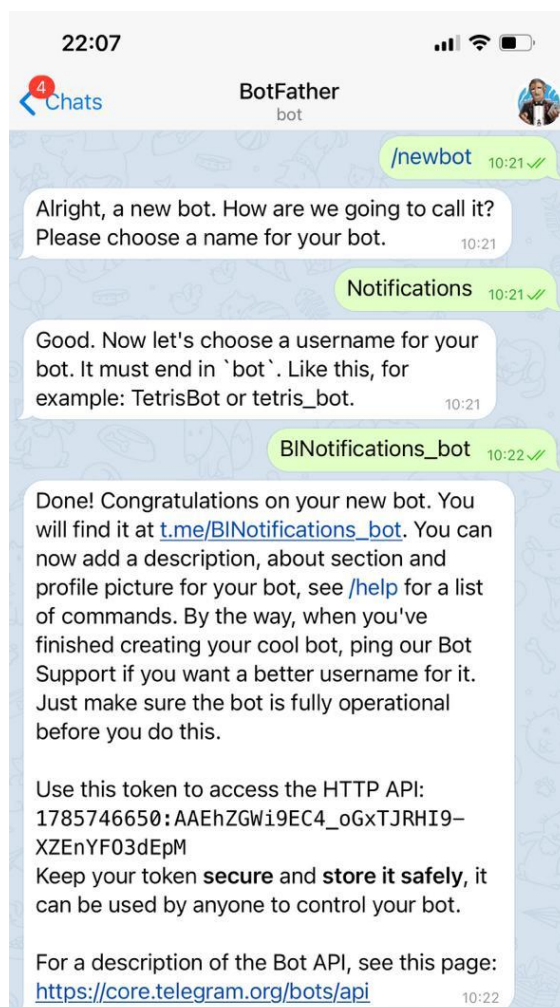
Registrarse en Telegram.

Luego se debe [iniciar una conversación con BotFather](#).

Una vez que estamos en la conversación con BotFather, se debe solicitarle la creación de un nuevo Bot.

Utilizando el comando: **/newbot**.

Luego, se nos solicitara un nombre, y un username.





¿Grupo o Conversación?

Dependerá de si solo nosotros queremos recibir las notificaciones del Bot, o si se desea que estas sean recibidas por varios usuarios de Telegram.

TokenKey

Token que se deberá enviar en todas las request a la API de Telegram, a modo de validación. Este token es único y no puede volver a generarse. Es enviado por **BotFather** al momento de crear el nuevo bot.

Chat ID

Es un código único que identifica a la conversación. Para obtenerlo primero debemos enviar un mensaje de prueba al grupo/conversación en donde se encuentra el bot. Luego se debe ejecutar el siguiente request (con el navegador).

```
https://api.telegram.org/bot<tokenkey>/getUpdates
```

En el caso anterior el token sería el siguiente:

```
https://api.telegram.org/bot1785746650:AAEhZGWi9EC4_oGxTJRHI9-XZEnYFO3dEpM/getUpdates
```

El request que obtenenido nos proporcionara un pequeño log con los mensajes recibidos por el bot con la forma:

```
"message":{"message_id":210,"from":{"id":1503388476,"is_bot":false,"first_name":"Ezedj","language_code":"en"},"chat":{"id":1503388476,"first_name":"Ezedj","type":"private"},"date":1617931601,"text":"/start","entities":[{"offset":0,"length":6,"type":"bot_command"}]},{update_id":266877834,"message":{"message_id":211,"from":{"id":1503388476,"is_bot":false,"first_name":"Ezedj","language_code":"en"},"chat":{"id":1503388476,"first_name":"Ezedj","type":"private"},"date":1617931605,"text":"hola mundo!"}}}
```

Envío de Mensajes

Se realizan mediante la API de Telegram, con el request:

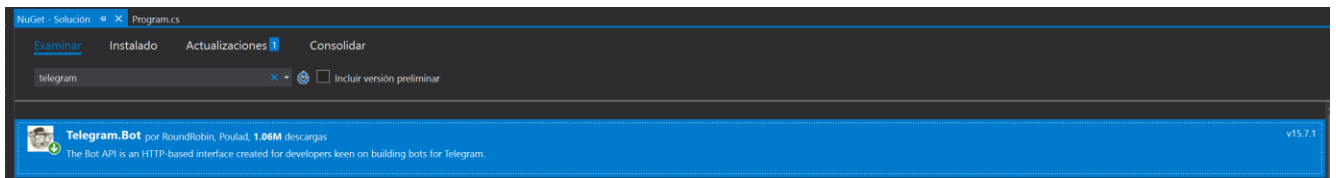
```
https://api.telegram.org/bot<tokenkey>/sendMessage?chat_id=<chatID>&text=<mensaje>
```



Integración con C#

Librería de Telegram

1. Crear un nuevo 'Proyecto de Consola'
2. Agregar la Referencia a la librería de Telegram
 1. En el Explorador de Soluciones -> clic derecho en el proyecto -> *'Administrar paquetes NuGet para la solución'*
 2. Instalar **'Telegram.Bot'** de RoundRobin



3. Haga clic en **Instalar**.

Agregar las siguientes líneas

```
ServicePointManager.Expect100Continue = true;  
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
```

Estas líneas son las encargadas de manejar las excepciones.

Ejecutar el request

Utilizando las siguientes líneas de código

```
string request = $"https://api.telegram.org/bot<tokenKey>/sendMessage?chat_id=<chatID>&text=<mensaje>"  
  
WebClient webCliente = new WebClient();  
  
string response = webCliente.DownloadString(request);
```

Parámetros por Main(String args[])

Con el fin de implementar este código como paquete de SSIS, se puede utilizar el pasaje de argumentos a través de la función main. Esto permite evitar el hardcodeo de argumentos (tokenKey, ChatID y/o mensaje).



De esta forma, se escribe un solo programa en C#, y se varían los parámetros del request dentro de la herramienta en la que se vaya a utilizar. A continuación se explicara la solución utilizando SQL Server Integration Services.

Obtener el .exe de la solución:

El ejecutable de la solución se puede encontrar en la carpeta **/bin/debug** de la solución.

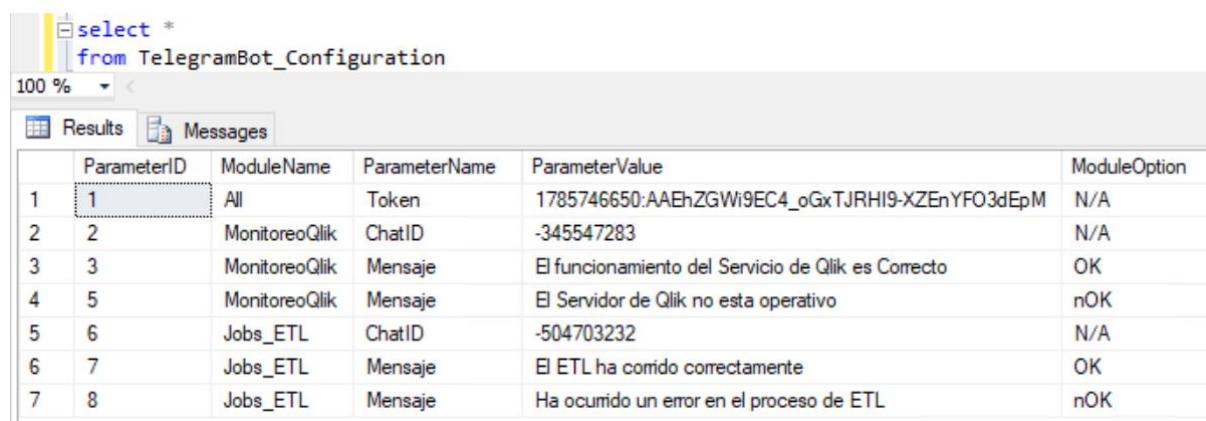


Integración con SSIS

Se detallará la solución utilizada.

Tabla de Configuración

Se creo una tabla que aloje los parámetros variables, para cumplir todos los requisitos, evitando hardcodear parámetros en el código. De esta forma, se utiliza el mismo ejecutable de C# para el envío de todos los mensajes.



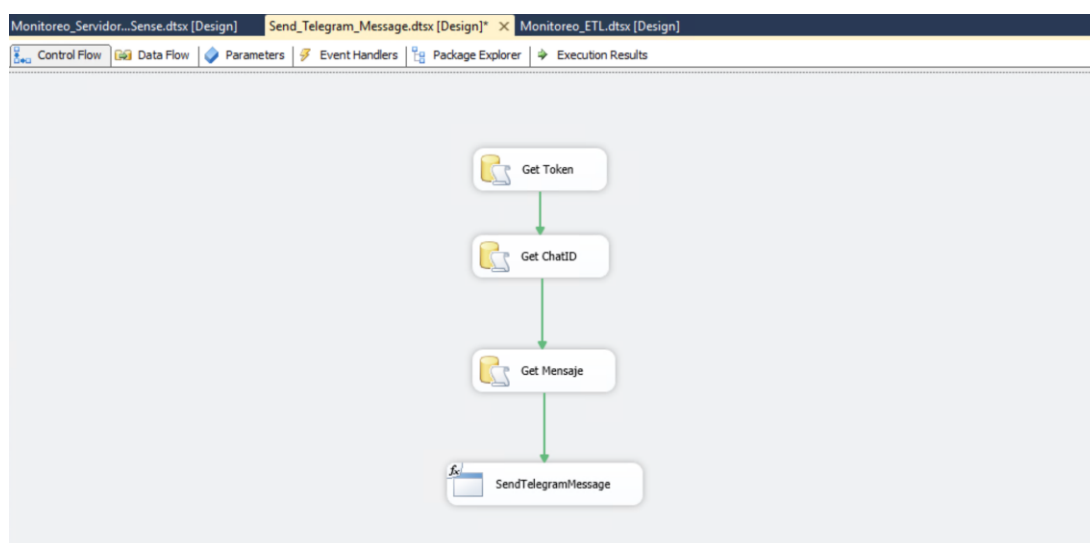
```
select *
from TelegramBot_Configuration
```

	ParameterID	ModuleName	ParameterName	ParameterValue	ModuleOption
1	1	All	Token	1785746650:AAEhZGWi9EC4_oGxTJRHI9-XZEnYFO3dEpM	N/A
2	2	MonitoreoQlik	ChatID	-345547283	N/A
3	3	MonitoreoQlik	Mensaje	El funcionamiento del Servicio de Qlik es Correcto	OK
4	5	MonitoreoQlik	Mensaje	El Servidor de Qlik no esta operativo	nOK
5	6	Jobs_ETL	ChatID	-504703232	N/A
6	7	Jobs_ETL	Mensaje	El ETL ha corido correctamente	OK
7	8	Jobs_ETL	Mensaje	Ha ocurrido un error en el proceso de ETL	nOK

Luego, estos valores son recuperados desde el paquete de SSIS, según corresponda.

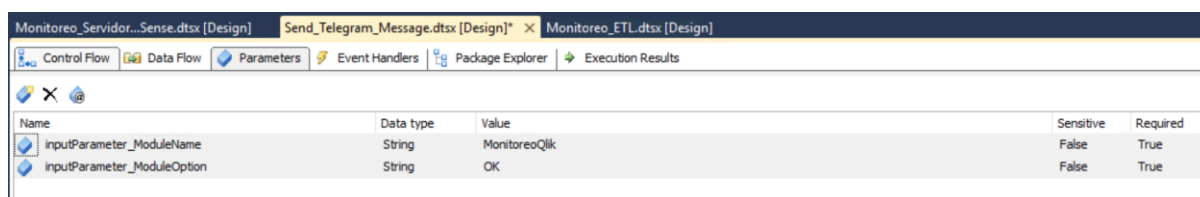
Paquete de SSIS dtsex

Proceso:





Parámetros:



Name	Data type	Value	Sensitive	Required
inputParameter_ModuleName	String	MonitoreoQlik	False	True
inputParameter_ModuleOption	String	OK	False	True

Variables:



Name	Scope	Data type	Value	Expression
input_ModuleOption	Send_Telegra...	String	OK	@[Package:inputParameter_ModuleOption]
input_Modulo	Send_Telegra...	String	MonitoreoQlik	@[Package:inputParameter_ModuleName]
output_ParameterValue_chatID	Send_Telegra...	String	0	
output_ParameterValue_Mensaje	Send_Telegra...	String		
output_Token	Send_Telegra...	String		

Con esta configuración, el paquete recibe dos parámetros: el modulo y el resultado que tiene que comunicar. Ejemplo, si se desea comunicar que ocurrió un error en el ETL:

```
inputParameter_ModuleName="Jobs_ETL"; inputParameter_ModuleOption= "nOK"
```