

Visual Editor Guides



How to verify user's phone number



Table of Contents

1	Introduction	3
2	Use Of Constructor - Verification By Outgoing Call	4
3	Use Of Constructor - Verification By Incoming SMS	8
4	Use Of Redirect - Verification By Outgoing Call	12
5	Use Of Redirect - Verification By Incoming SMS	13

1 Introduction

In some cases a bot has to check if a user has provided a true phone number. It is often needed for confirmation of financial transactions via mobile phone, or when logging into a banking account. MiniApps provides a feature that allows to verify a user's phone number (MSISDN) right within a bot dialog. MiniApps supports two ways of MSISDN verification:

1. **Outgoing call.** First MiniApps asks a user to enter a valid MSISDN in the bot dialog. Once the MSISDN is received, MiniApps requests the user to dial some service phone number from a mobile device associated with that MSISDN. In case of success the user is forwarded to the next bot page automatically.
2. **Incoming SMS.** When MSISDN verification is required, MiniApps asks a user to enter a valid MSISDN in the bot dialog. An SMS with a verification code is then sent to that MSISDN. At the same time a dedicated bot page appears in a messenger where the user should enter the verification code. If the code is correct, the user is forwarded to the next bot page.

If the verification procedure succeeds, MiniApps saves the user's MSISDN, which makes recurring verifications unnecessary.

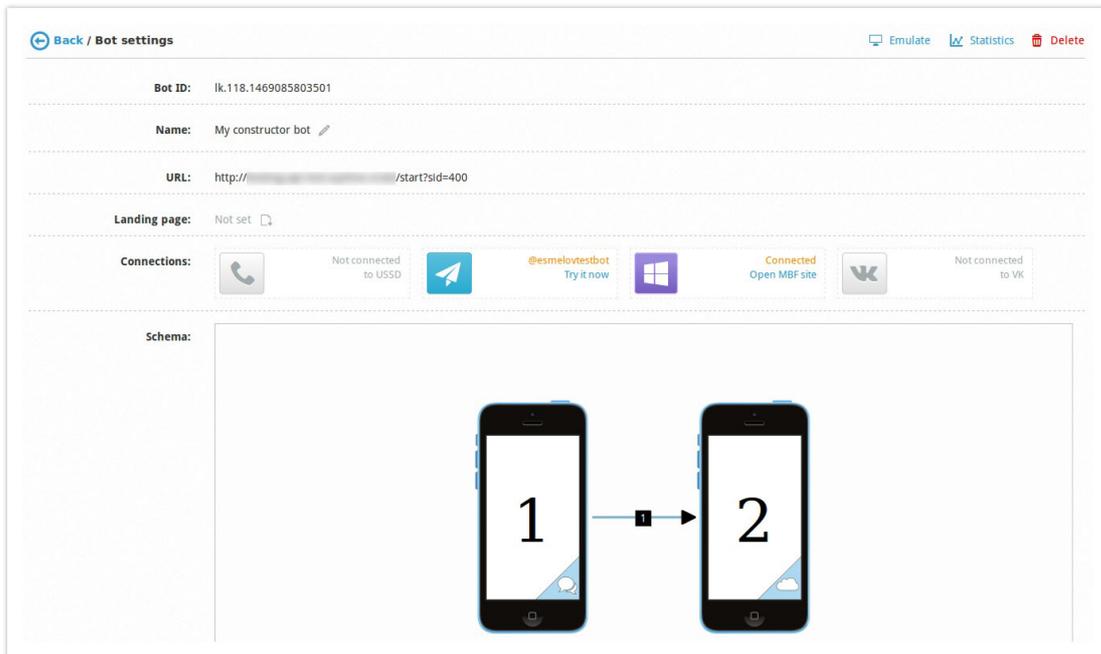
There are two ways to implement phone number verification:

1. [Use of constructor](#) - verification is added through MiniApps bot constructor.
2. [Use of redirect](#) - verification is implemented by HTTP redirect message using API.

Both options are further described below.

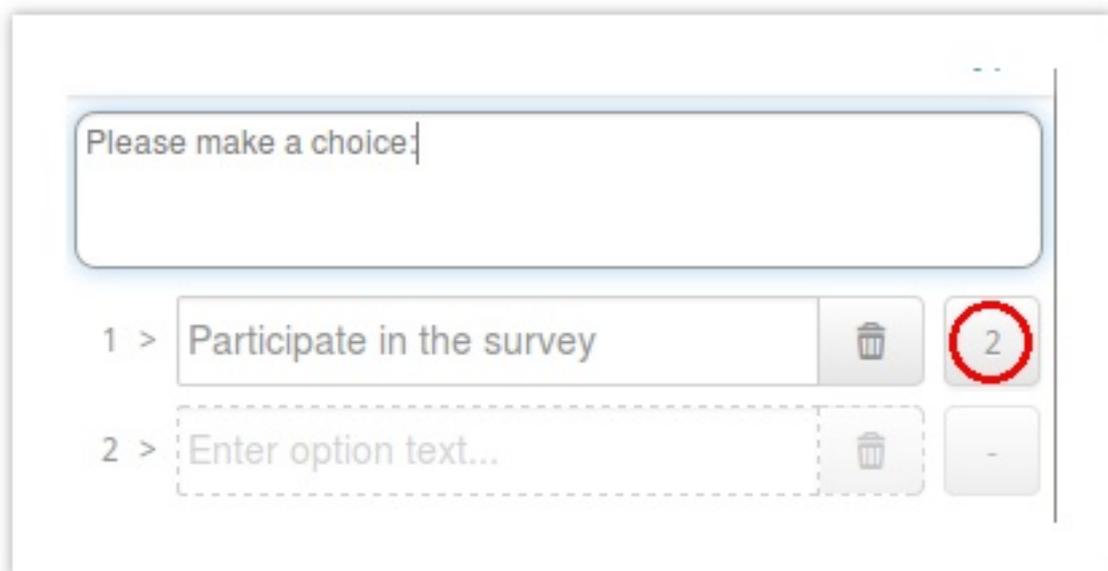
2 Use Of Constructor - Verification By Outgoing Call

11Log in to MiniApps. You will see all your bots. Select a bot created with *Bot Builder Wizard*, or create a new one. Your bot's details similar to the example below will appear:

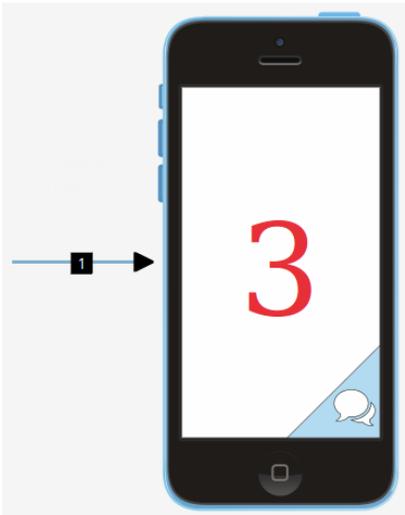


In our example you see two phones, which means two bot pages the user can surf through. Let us add phone verification as a prerequisite for transition to the second page.

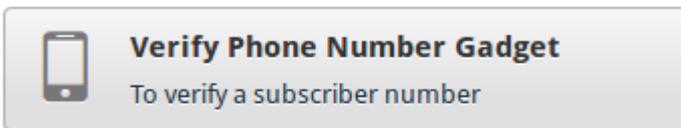
Click on the picture with the phones. The constructor opens. Select the first phone and click on the button signifying transition to the second page:



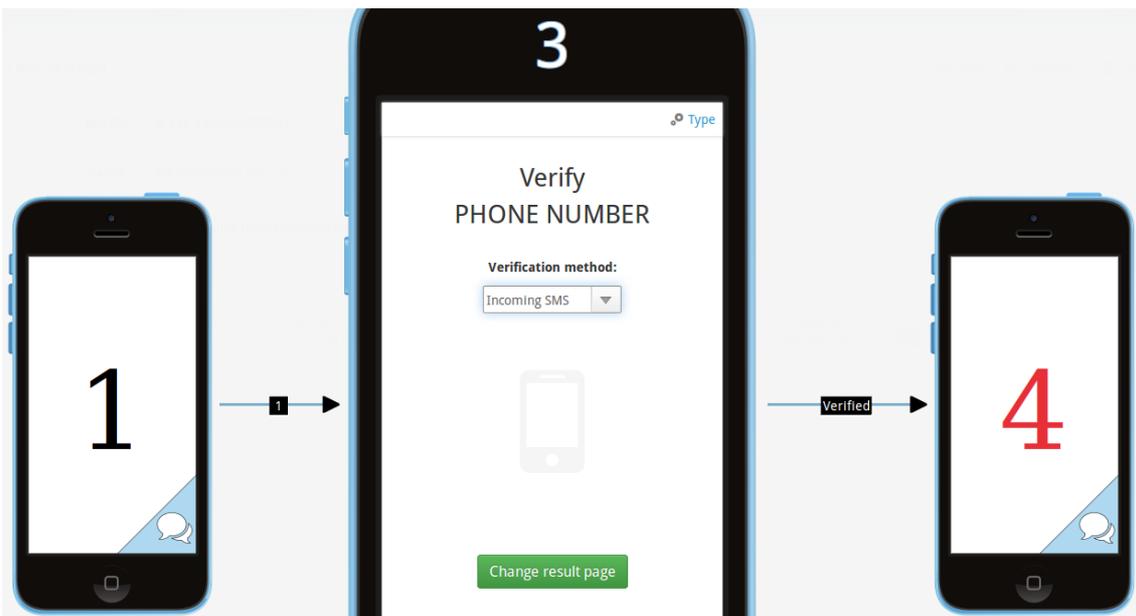
In the upper right corner click on  button. Page 3 appears:



Select it and click [Type](#) link above the edit field. Now select



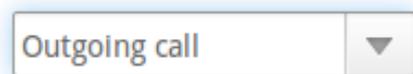
The new bot structure now looks like this:



As described above, the two verification methods available are:

1. **Outgoing call** - the user should dial a service phone number to verify their MSISDN.
2. **Incoming SMS** - the user will receive an SMS with a verification code to be then entered in the corresponding field to confirm.

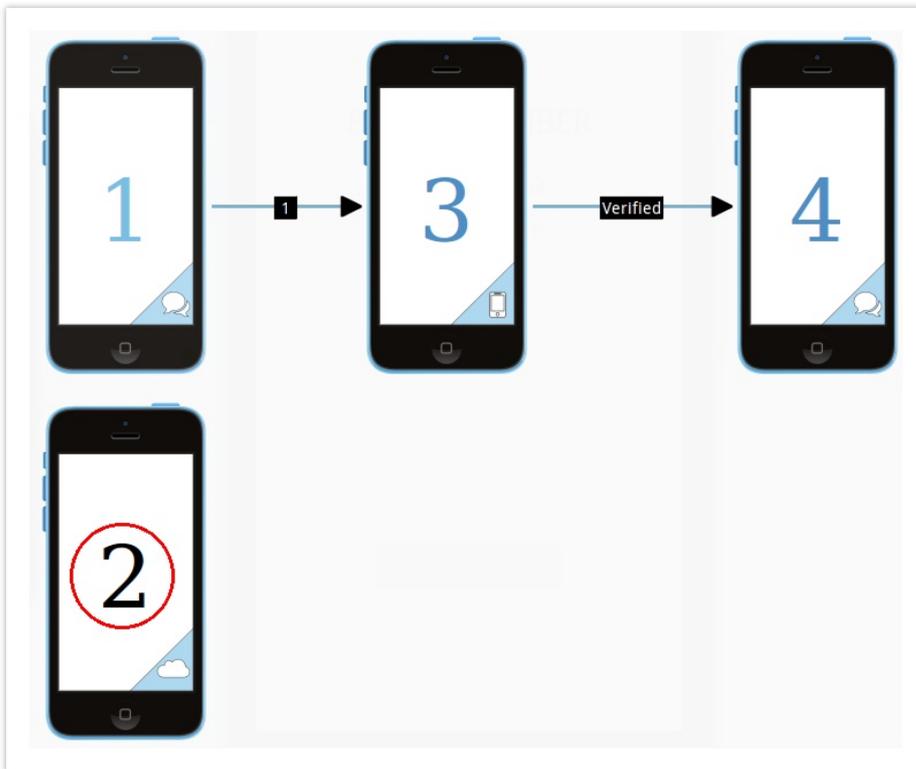
Let us select the **Outgoing call** method:



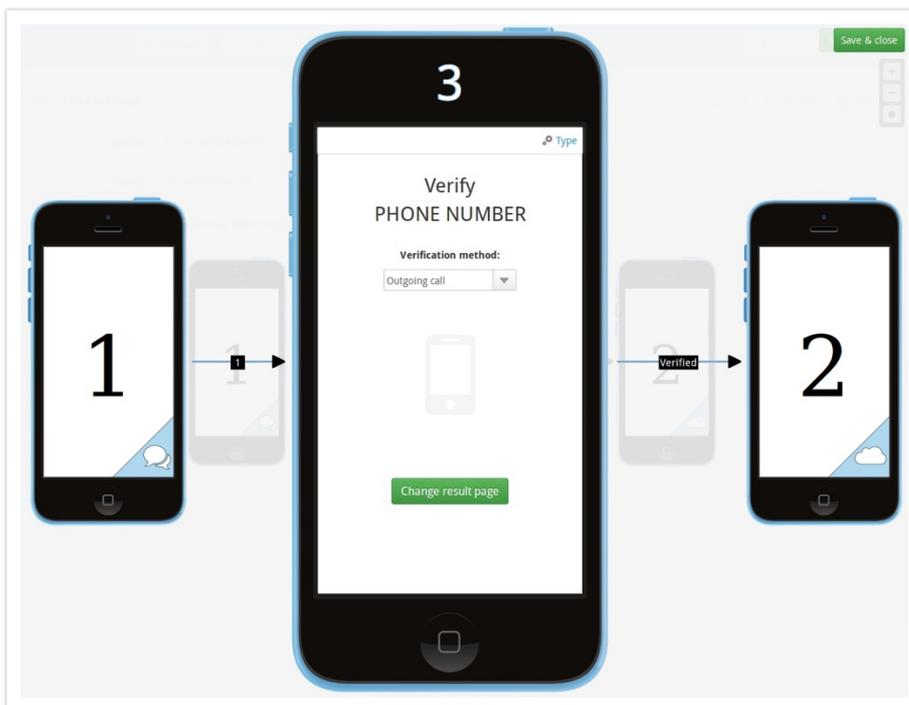
It is normal system behaviour that our Page 3 by default leads to the newly created Page 4 as the next page. To make it leading to Page 2 as the subsequent page, click on

Change result page

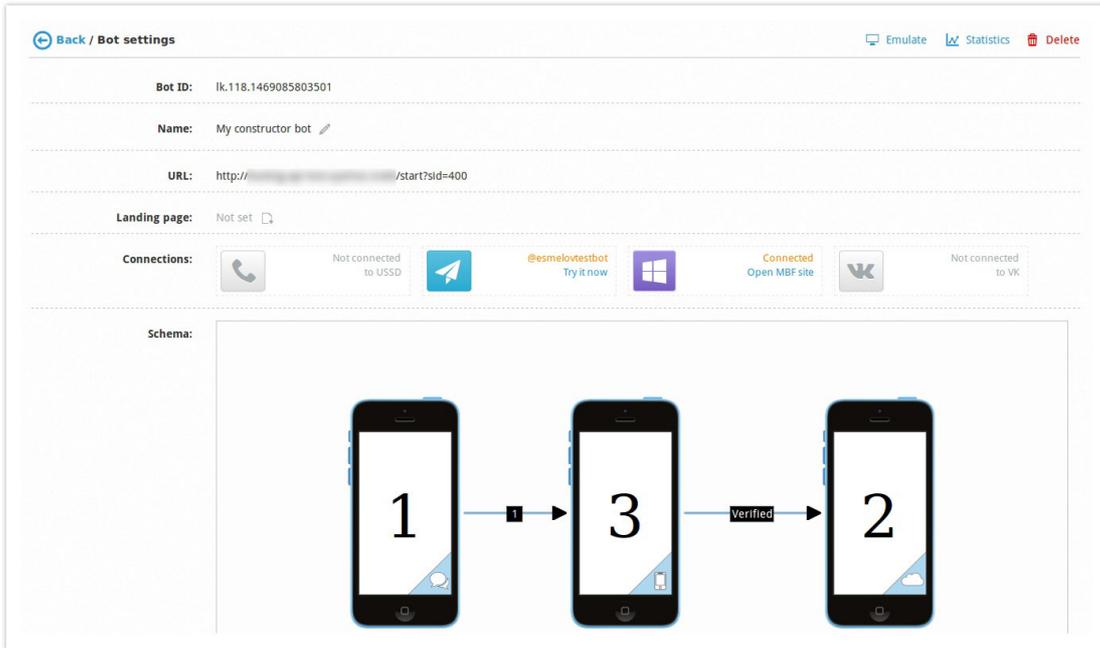
the button and select Page 2:



Now we have the right structure:



To save the structure click on **Save & Close** button and then click on **Save** button in the next window. You now see the updated bot details page:



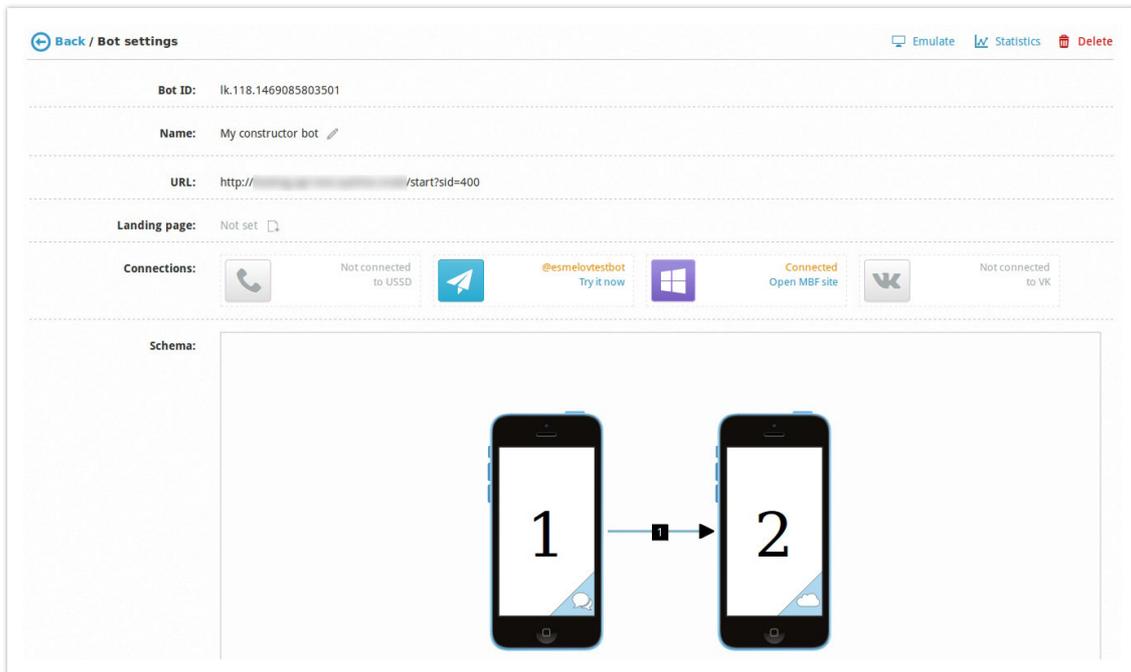
The screenshot displays the 'Bot settings' interface for a bot named 'My constructor bot'. The settings include:

- Bot ID:** Ik.118.1469085803501
- Name:** My constructor bot
- URL:** http://.../start?sid=400
- Landing page:** Not set
- Connections:** A row of connection status cards for various services: Phone (Not connected to USSD), Telegram (@esmelovestbot, Try it now), Windows (Connected, Open MBF site), and VK (Not connected to VK).
- Schema:** A diagram showing a flow of three mobile phones. The first phone displays the number '1', the second displays '3', and the third displays '2'. An arrow points from phone 1 to phone 3, and another arrow points from phone 3 to phone 2, with the word 'Verified' written above the second arrow.

After this change the user have to pass the outgoing call based MSISDN verification procedure in order to move to Page 2.

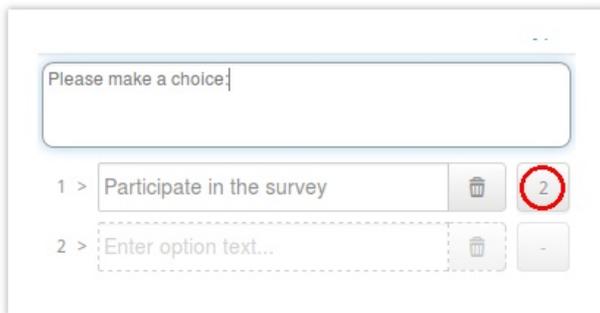
3 Use Of Constructor - Verification By Incoming SMS

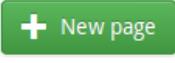
3Log in to MiniApps. You will see all your bots. Select a bot created with *Bot Builder Wizard*, or create a new one. Like in the previous example your bot's details appear:

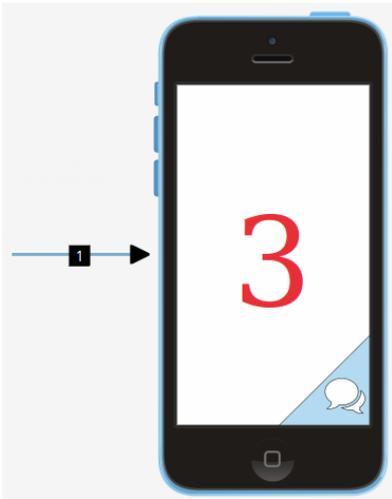


In our example you see two phones, which means two bot pages the user can surf through. Let us add phone verification as a prerequisite for transition to the second page.

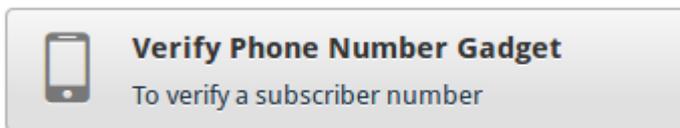
Click on the picture with the phones. The constructor opens. Select the first phone and click on the button signifying transition to the second page:



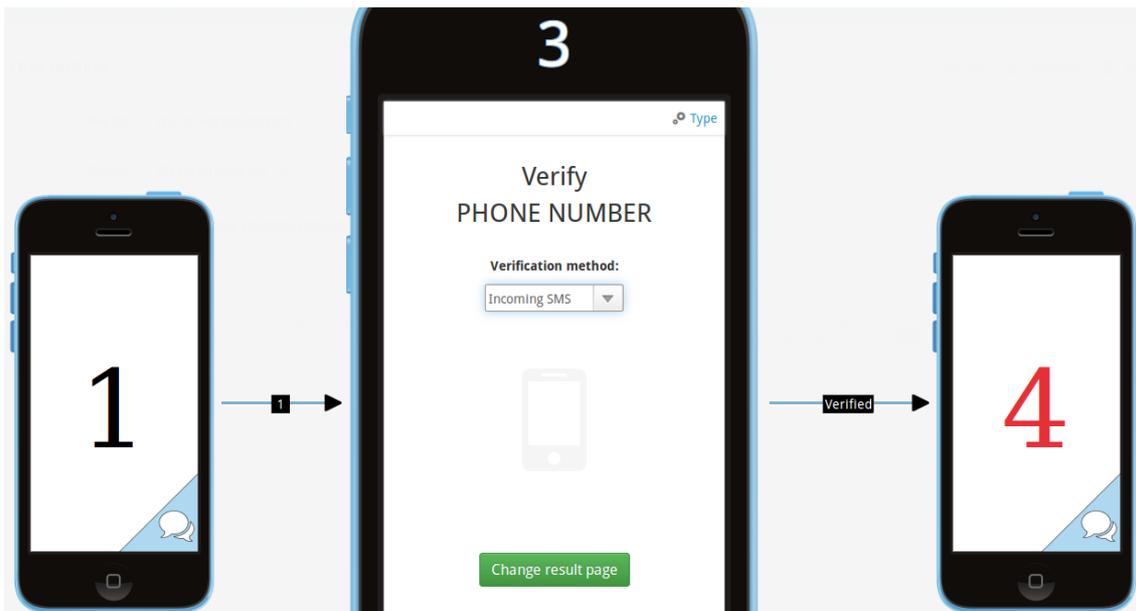
In the upper right corner click on  button. Page 3 appears:



Select it and click  link above the edit field. Now select



The bot structure is updated:

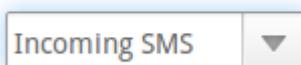


As described above, the two verification methods available are:

1. **Outgoing call** - the user should dial a service phone number to verify their MSISDN.
2. **Incoming SMS** - the user will receive an SMS with a verification code to be then entered in the corresponding field to confirm.

This time let us select the **Incoming SMS** method:

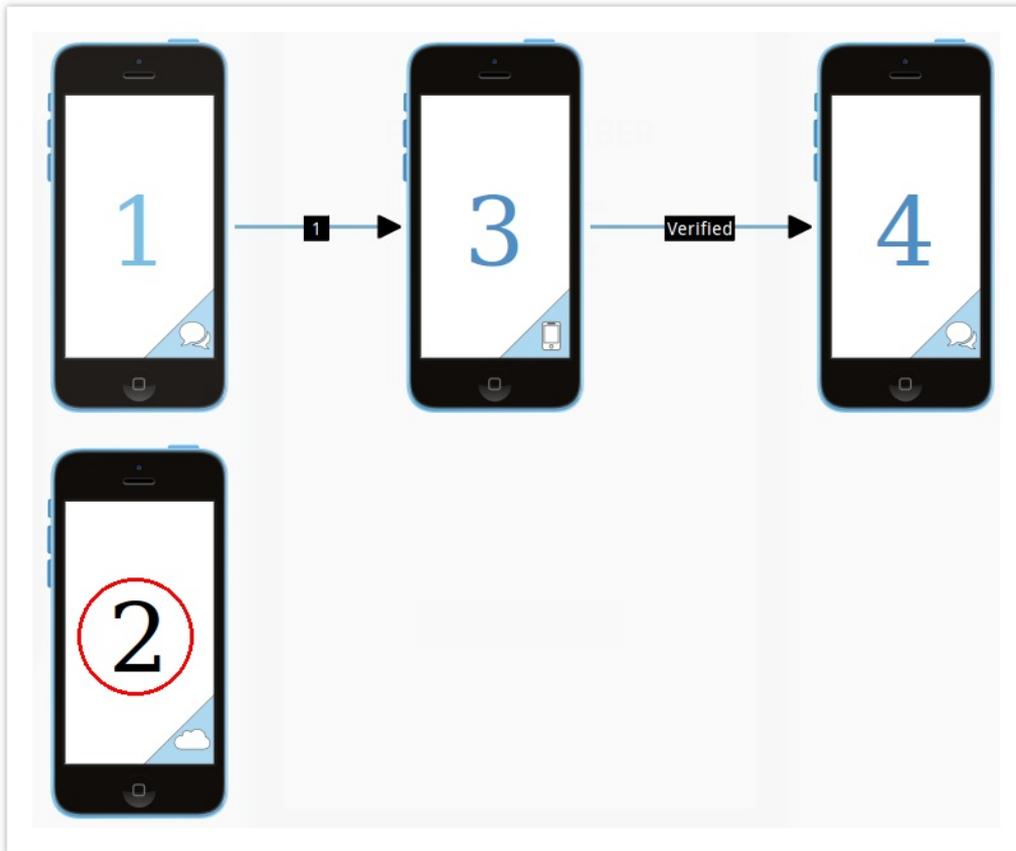
Verification method:



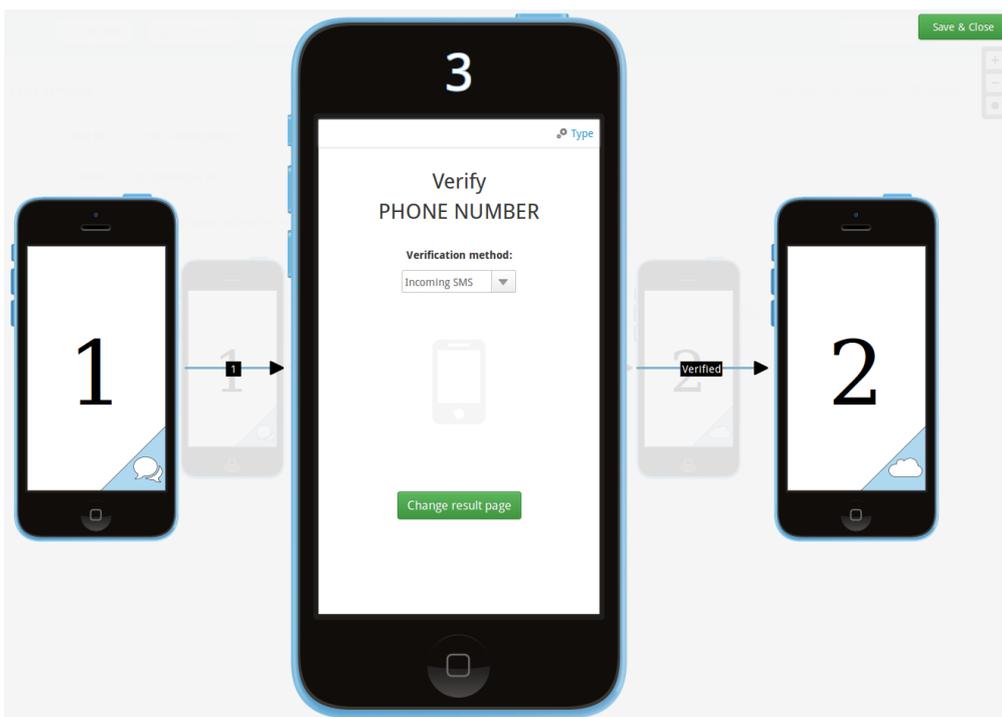
It is normal system behaviour that our Page 3 by default leads to the newly created Page 4 as the next page. To make it leading to Page 2 as the subsequent page, click on

Change result page

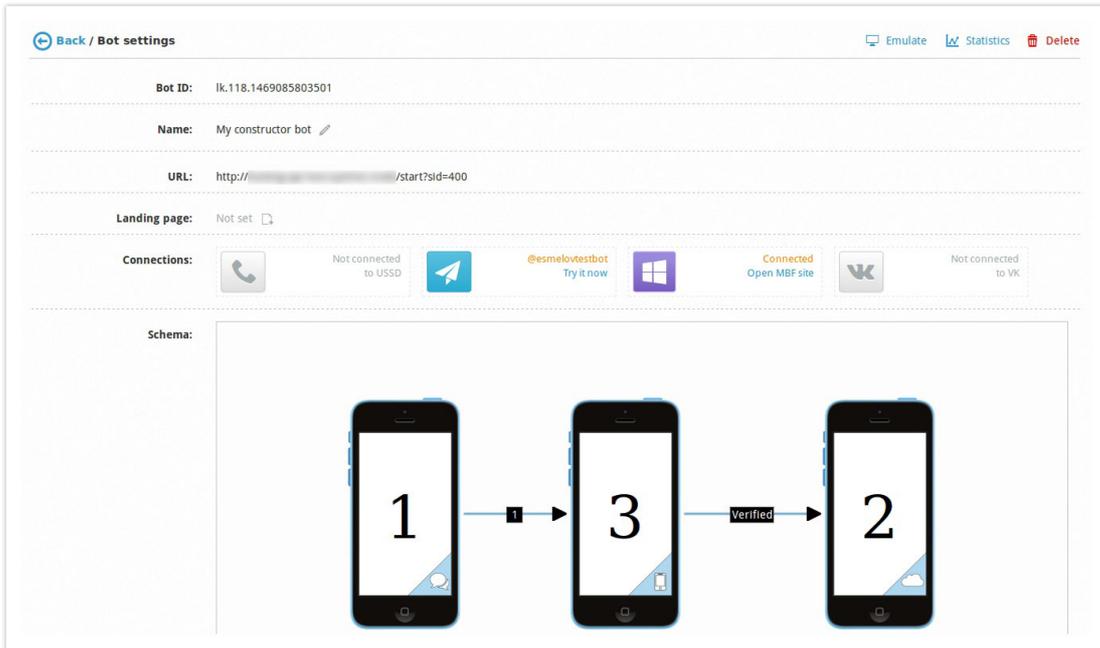
the button and select Page 2:



Now we have the right structure:



To save the structure click on **Save & Close** button and then click on **Save** button in the next window. You now see the updated bot details page:

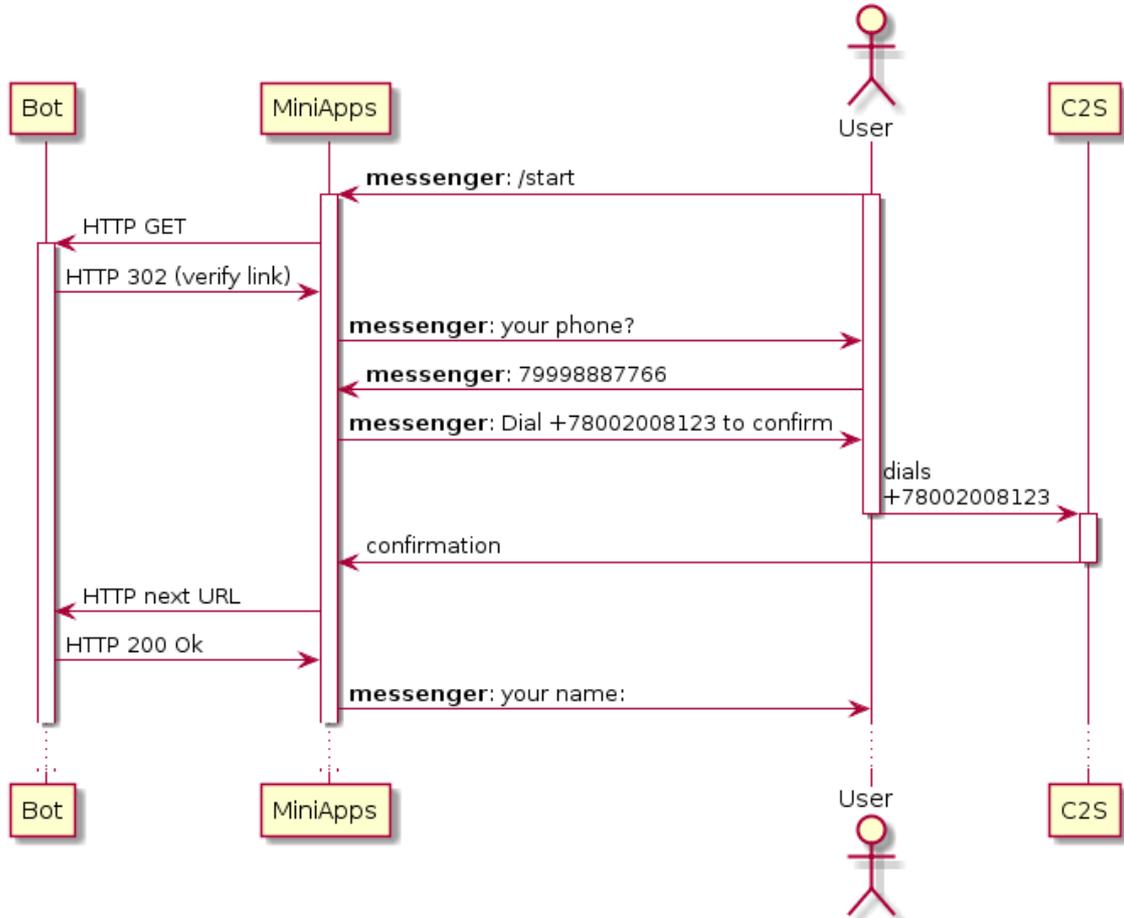


The screenshot shows the 'Bot settings' page for a bot named 'My constructor bot'. The Bot ID is 'lk.118.1469085803501'. The URL is 'http://.../start?sid=400'. The Landing page is 'Not set'. The Connections section shows: Phone (Not connected to USSD), Telegram (@esmelovestbot, Try it now), Windows (Connected, Open MBF site), and VK (Not connected to VK). The Schema section shows a flow: Page 1 → Page 3 → Verified → Page 2.

After this change the user have to pass the incoming SMS based MSISDN verification procedure in order to move to Page 2.

4 Use Of Redirect - Verification By Outgoing Call

If you use MiniApps API to communicate with your users, there is still a way to add MSISDN verification. To do it use the **HTTP Redirect message**. Let us assume that you already have a bot connected to MiniApps personal profile, and you want to add MSISDN verification. Take a look at the next sequence diagram describing MSISDN verification using **Outgoing call** in more detail (the user there uses some kind of messenger to communicate with the bot):



The mentioned **verify link** has the following format:

```

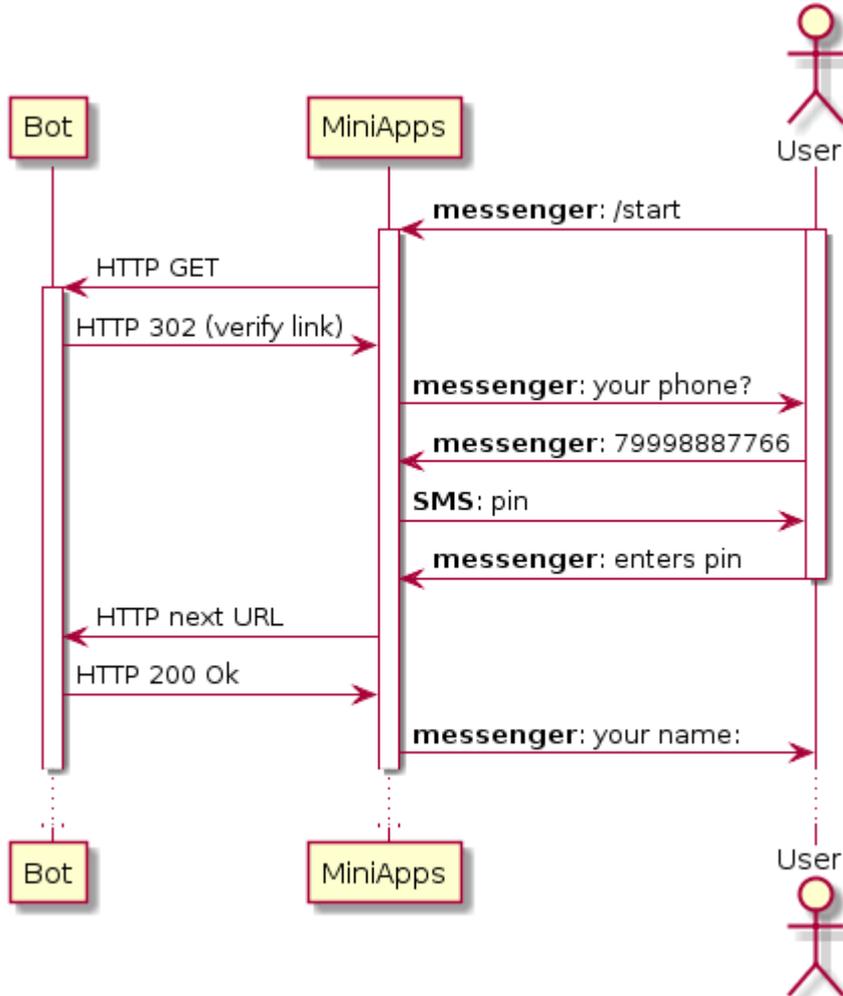
http://plugins.miniapps.run/msisdn-verification?on_success=<next url>&type=c2s
    
```

where the **next url** is a URL that MiniApps should use after verification is done, and **type=c2s** means the **Outgoing call** based verification procedure.

After verification is finished MiniApps sends HTTP GET request to *on_success* URL. The verified phone number is sent in *msisdn* query parameter.

5 Use Of Redirect - Verification By Incoming SMS

A very similar procedure is used to check MSISDN by means of incoming SMS. Let us assume that you already have a bot connected to MiniApps personal profile, and you want to add MSISDN verification. Refer to the next sequence diagram describing MSISDN verification using **Incoming SMS** in more detail (the user there uses some kind of messenger to communicate with the bot):



The mentioned **verify link** has the following format:

```

http://plugins.miniapps.run/msisdn-verification?on_success=<next url>&type=sms
    
```

where the **next url** is a URL that MiniApps should use after verification is done, and **type=sms** means the **Incoming SMS** based verification procedure.

After verification is finished MiniApps sends HTTP GET request to *on_success* URL. The verified phone number is sent in *msisdn* query parameter.