****

**Inbenta Developer Challenge**

**Chatbot Technical Challenge**

*Proof of Completion by © Vincent Seres*

Contents

[1 Content Creation 3](#_Toc142340905)

[2 Generate interface with the default Configuration 4](#_Toc142340906)

[3 AIML (pattern-based responses) Edition 4](#_Toc142340907)

[4 Configure the Chatbot SDK 5](#_Toc142340908)

[5 Hosting your Chatbot demo 6](#_Toc142340909)

[6 Dong it in Style 7](#_Toc142340910)

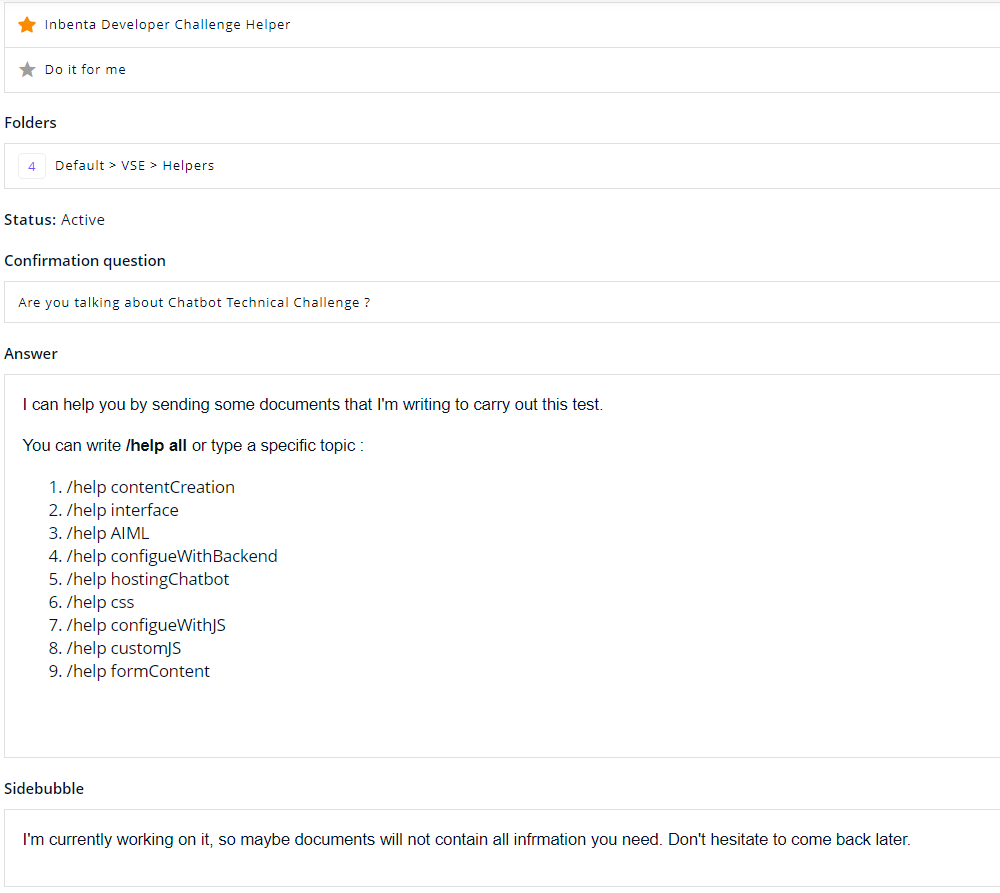
[7 Modify the SDK JS configuration 8](#_Toc142340911)

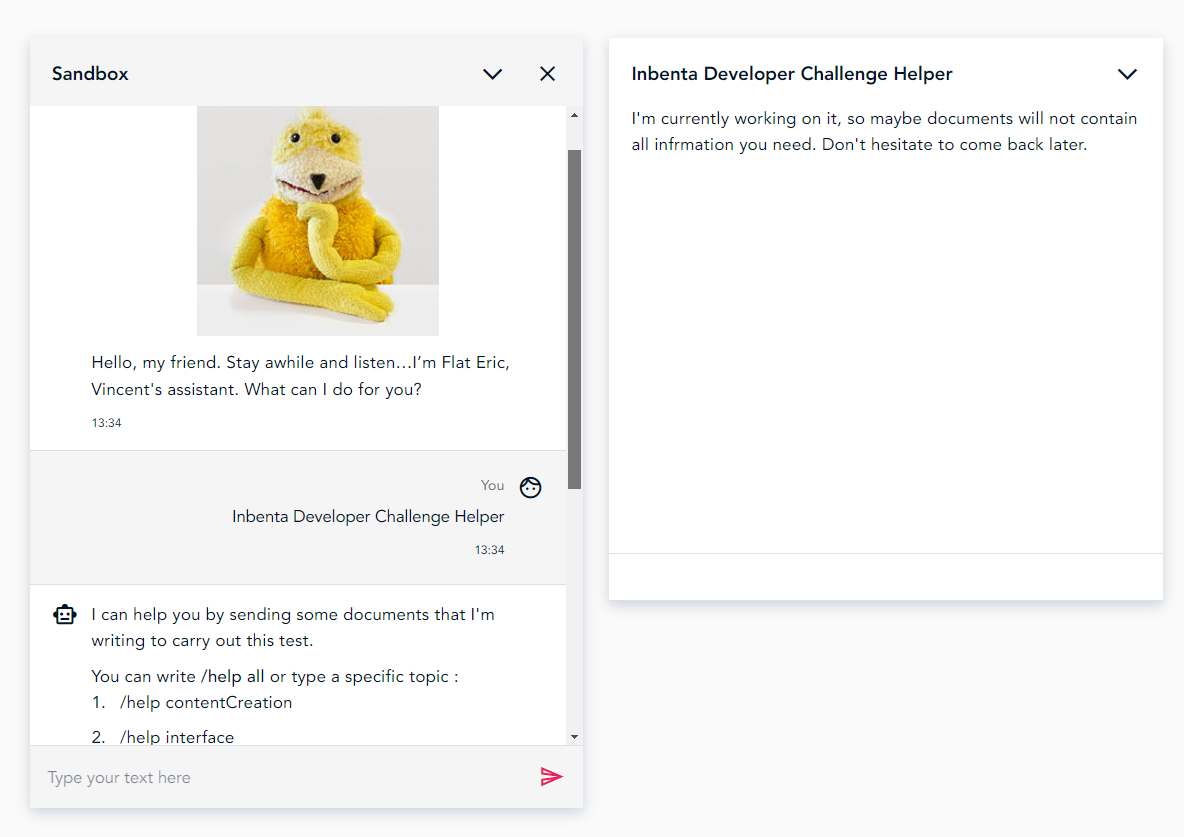
[8 Implement a JavaScript customization 9](#_Toc142340912)

[9 Create a “form” content 10](#_Toc142340913)

# Content Creation

PoC - Screenshot of the content as seen from the Chatbot user interface “se\_1.png”



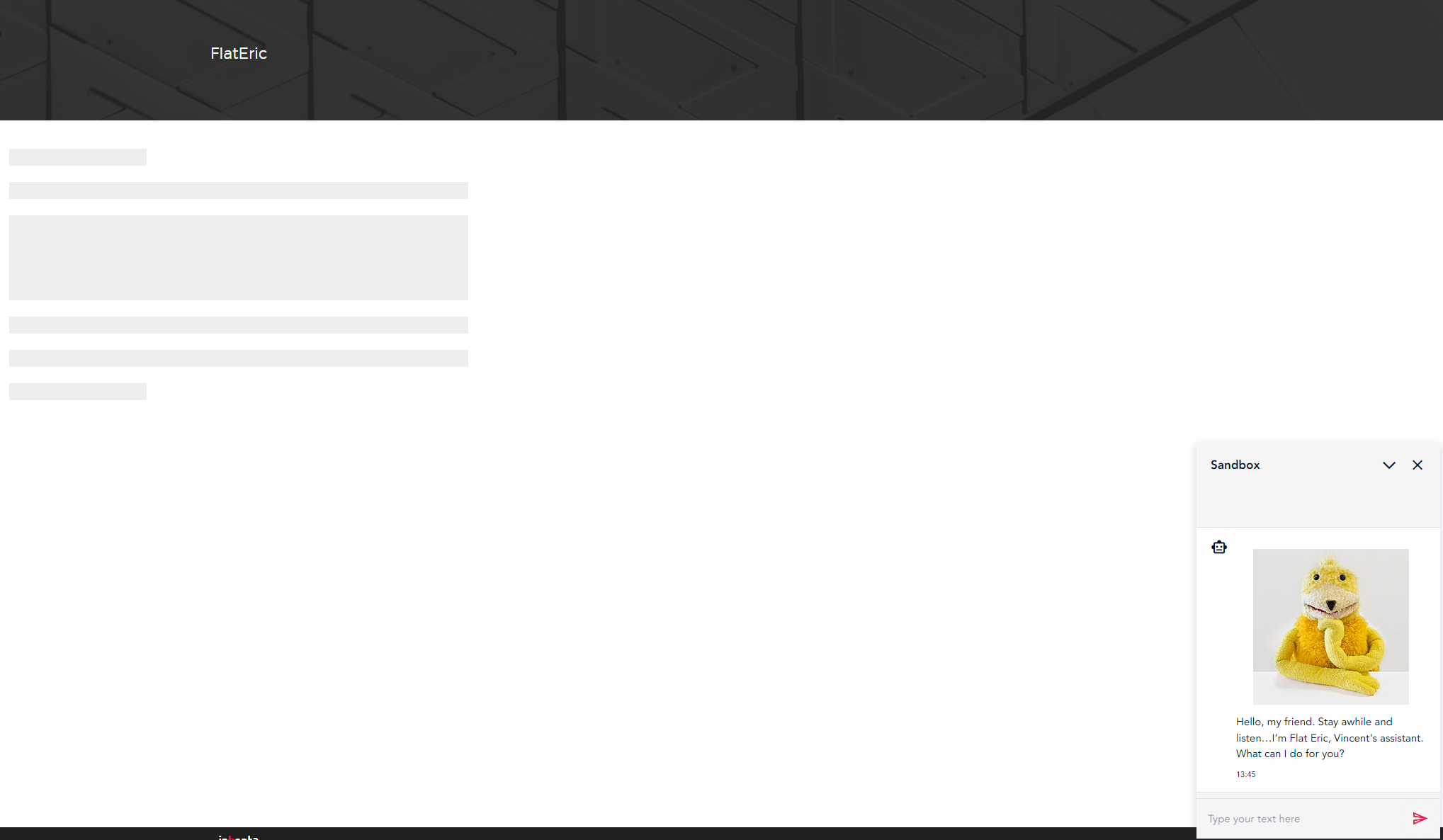


# Generate interface with the default Configuration

PoC - Same screenshot as (1.-)

Public URL (shared from Inbenta platform):

<https://shared-apps.inbenta.com/shared-applications/fa16224fcc7011680ef8909be5101cb453a58003b19ad2b1d5fbede4229f1ebc/d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35>



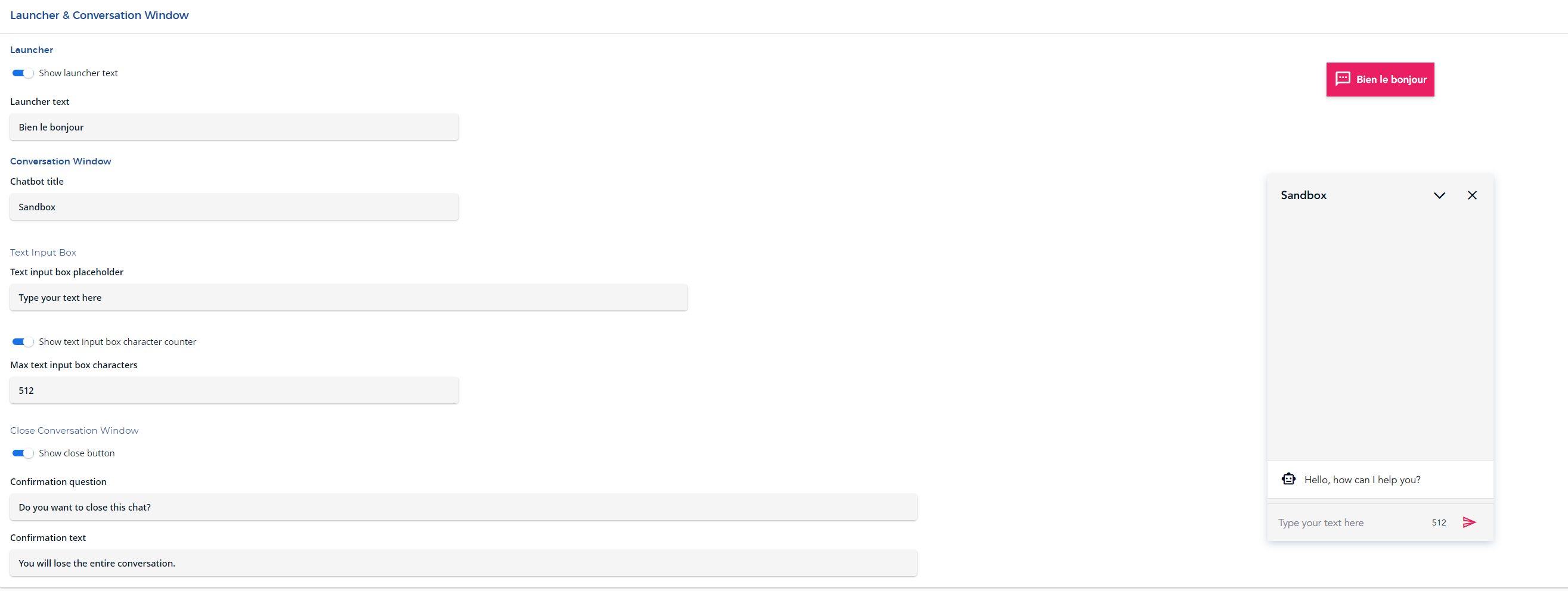
# AIML (pattern-based responses) Edition

PoC - Screenshot of the AIML response as seen from the Chatbot user interface “se\_3.png”

|  |  |
| --- | --- |
| Here some AIML generated for this test:  <https://github.com/oizo1987/Inbenta/tree/main/Chatbot%20Technical%20Challenge/AIML>   * heybot\_en.aiml * diabloQuest\_en.aiml * technicalChallengeHelper.aiml |  |

# Configure the Chatbot SDK

PoC - Screenshot that shows both the title and the ratings as seen from the Chatbot user interface “se\_4.png”



|  |  |
| --- | --- |
|  |  |

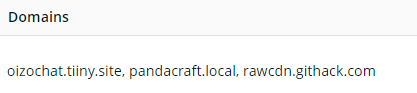
# Hosting your Chatbot demo

PoC - HTML file “se\_5.html”

HTML file available here (with solution for chapter 5, 6, 7, 8 and 9): https://github.com/oizo1987/Inbenta/blob/main/Chatbot%20Technical%20Challenge/www/chatbot\_APIFullClient/index.html

Public URL of hosted html with chatbot integration: <https://oizochat.tiiny.site/>

This domain is set on dev env API config:



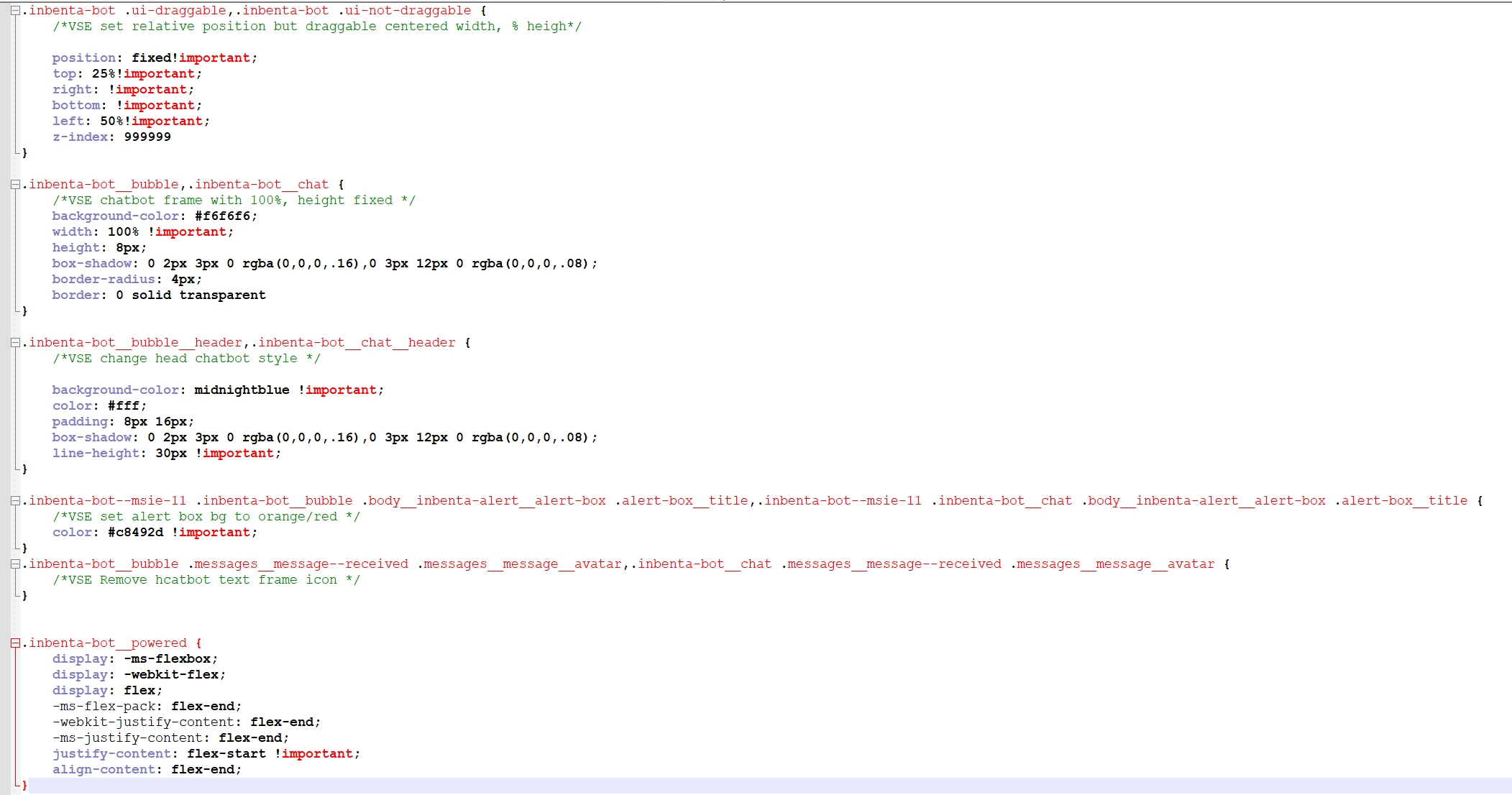
# Dong it in Style

PoC - HTML file with the custom CSS. “se\_6.html”

Done. See link chapter 5.

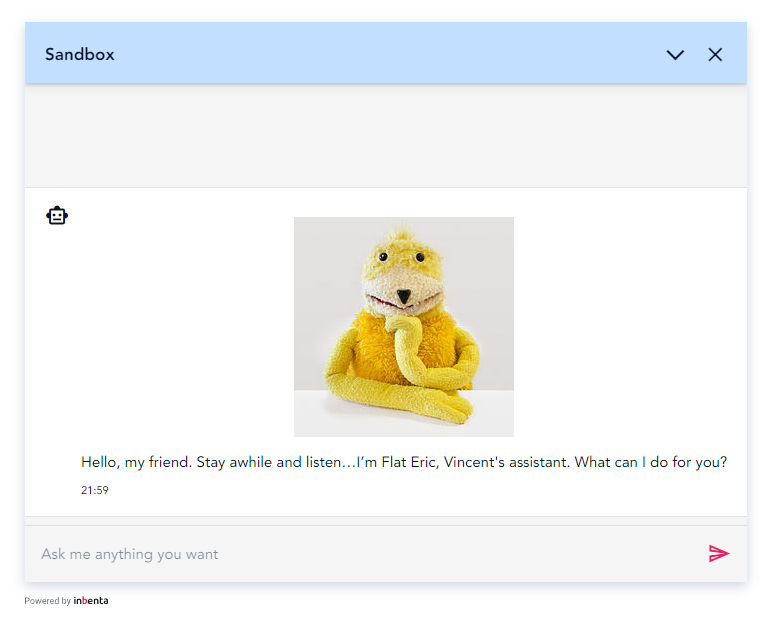
Add the CSS import in html:

I maintained skin enabled and overwrited some attributes.



You can also disable skin and create your own skin:





# Modify the SDK JS configuration

PoC - HTML file with the custom JS. “se\_7.html”

Add to **config** parameter to function InbentaChatbotSDK.buildWithDomainCredentials(authorization,**config**) with this variable -example-:

**closeButton**: {

visible: true

}

**labels**: {

en: {

'enter-question': 'Ask me anything you want'

}

}

**conversationWindow**: {

draggable: false,

position: {

top: 100,

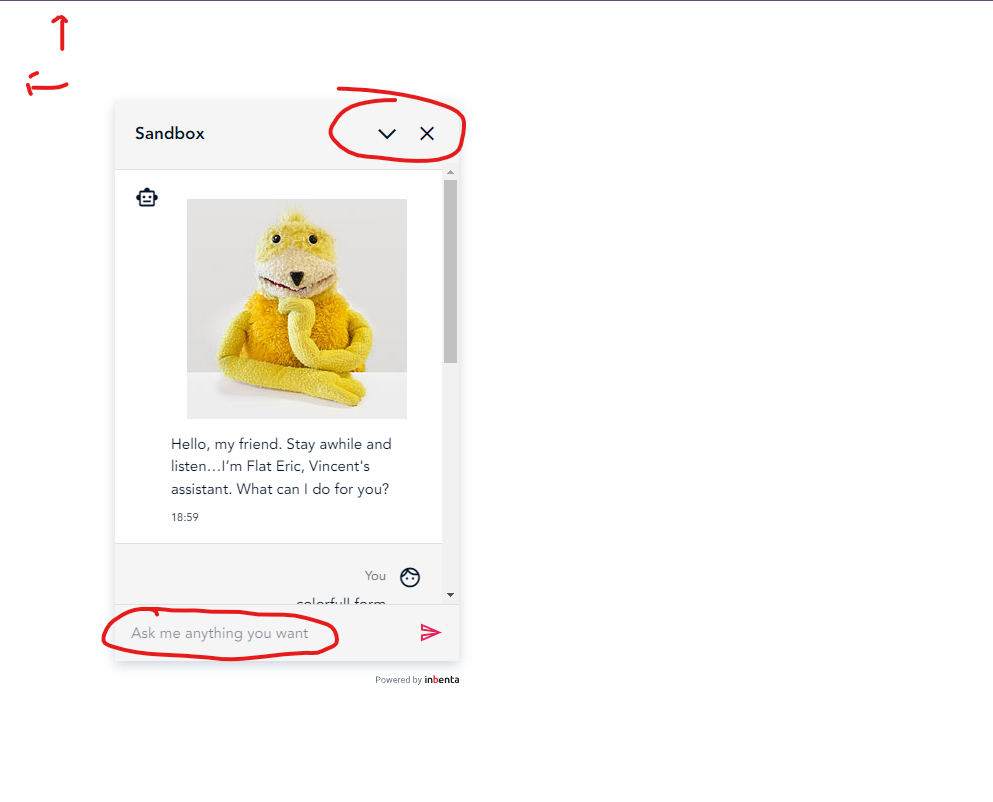
left: 100,

bottom: null,

right: null,

},

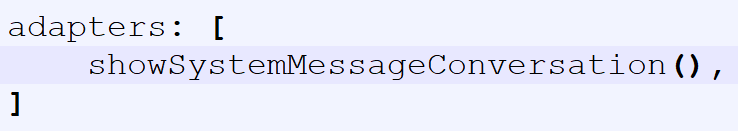
}

*\*Position is overwritted by CSS.*

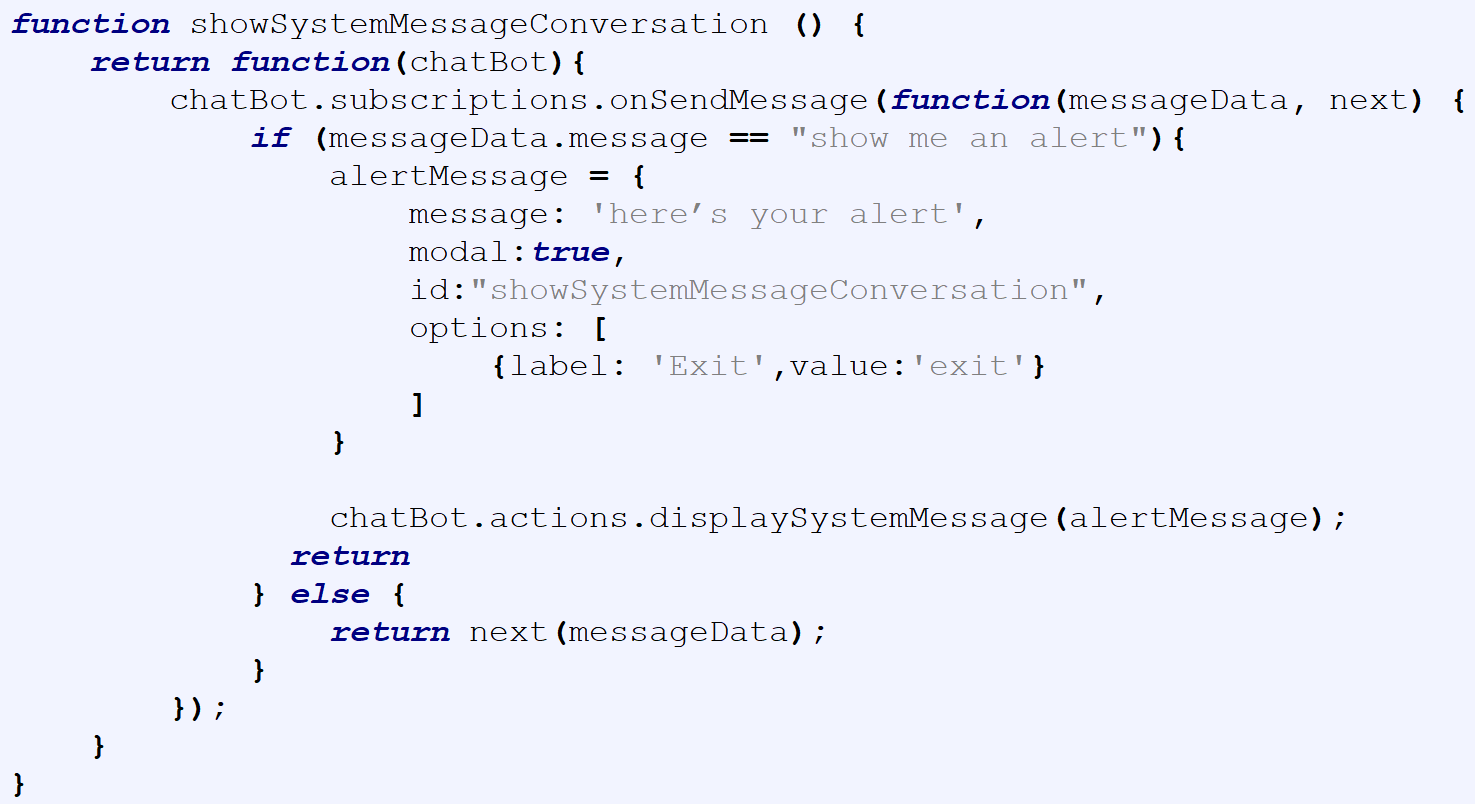
# Implement a JavaScript customization

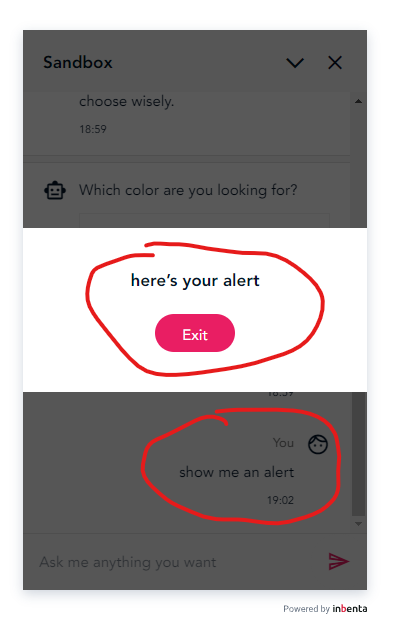
PoC - HTML file with the custom JS. “se\_8.html”

Add adapters to **config** parameter



Then import JS (best solution) or add raw code into yhe html:

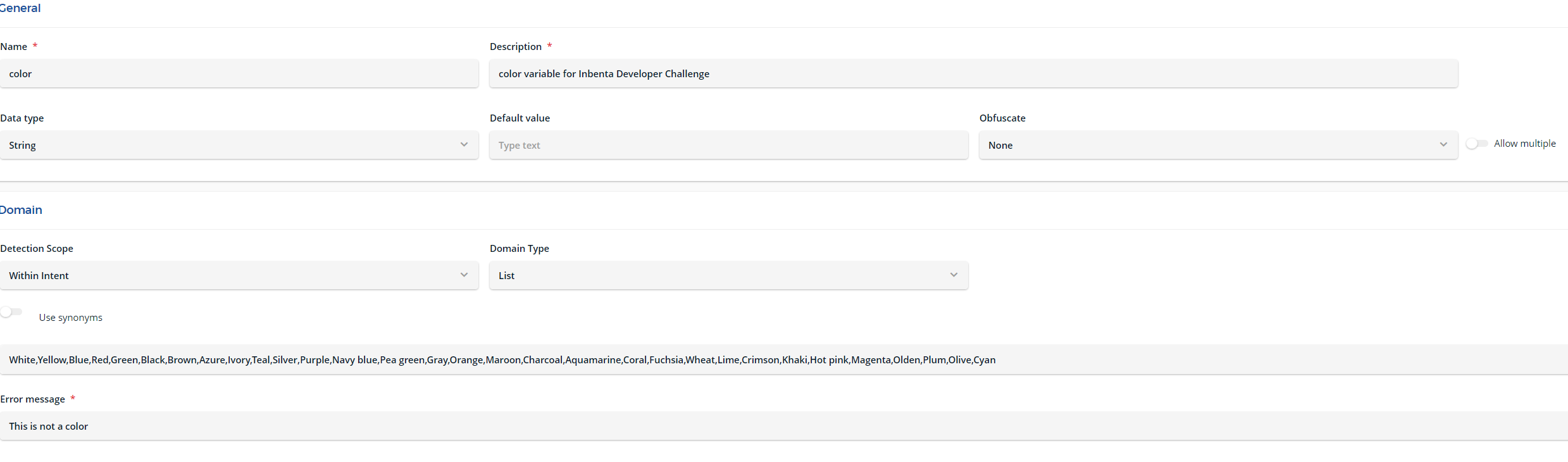




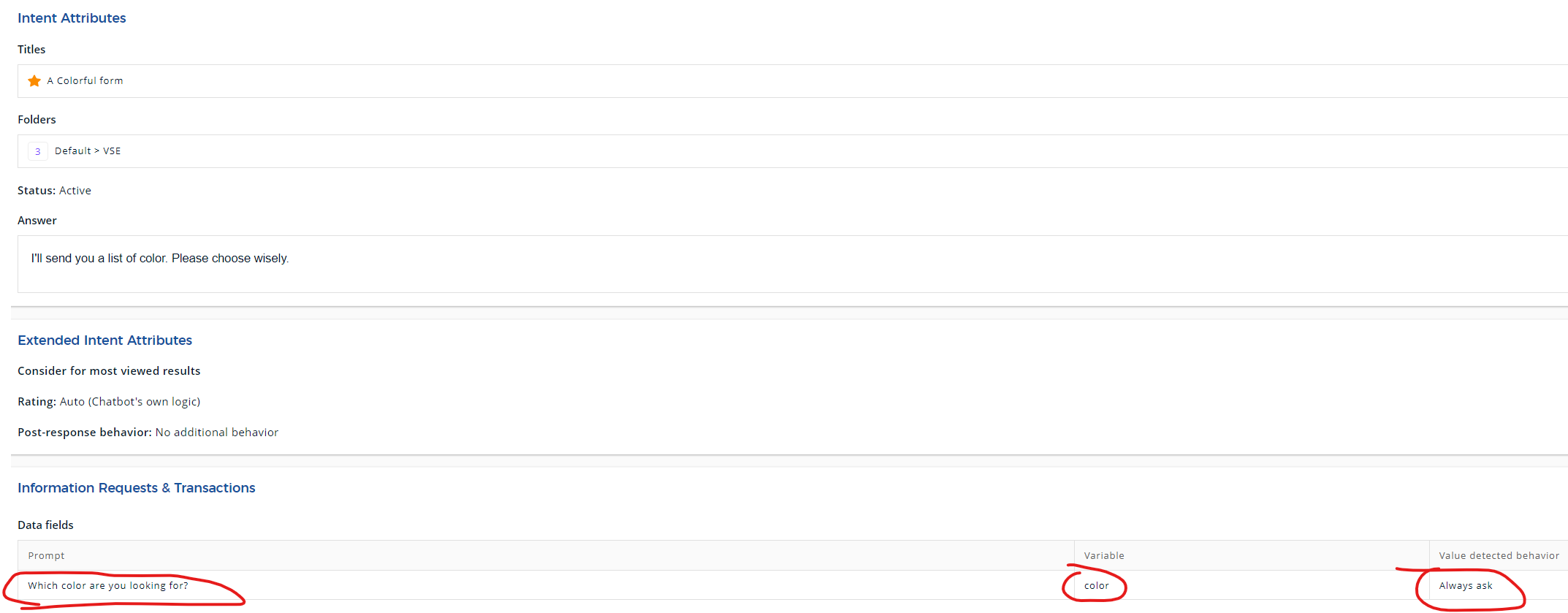
# Create a “form” content

PoC - HTML file with the JS callback. “se\_9.html”

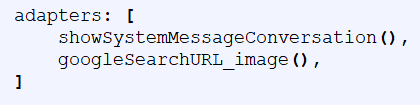
Configure variable

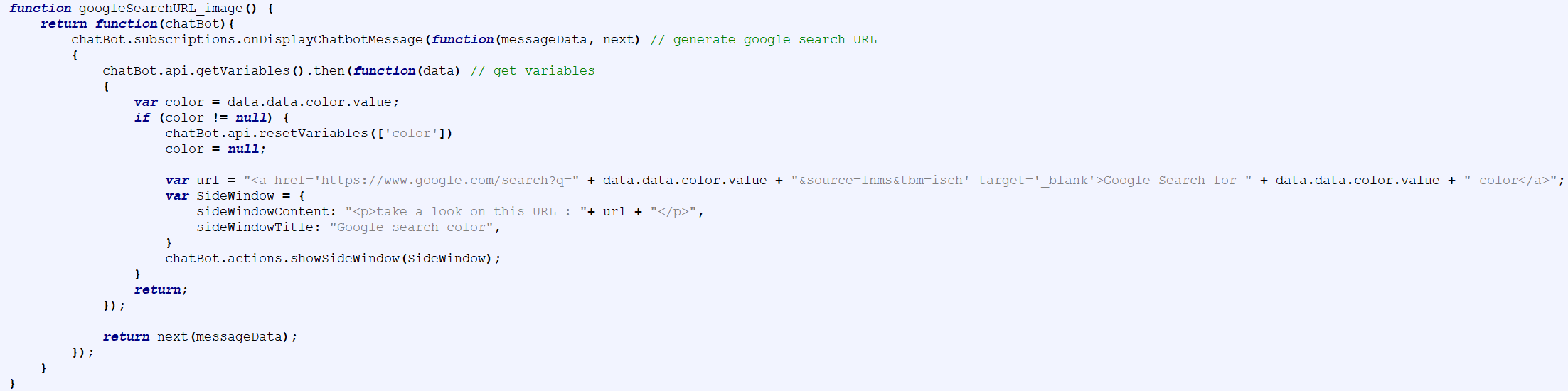


Configure intent



Code subscription and action:





Result :

