

Customização do Webchat gerado por meio do Copilot Studio

1. Objetivo

Documentar o passo a passo para customizar o Webchat gerado pelo Copilot Studio

2. Metodologia

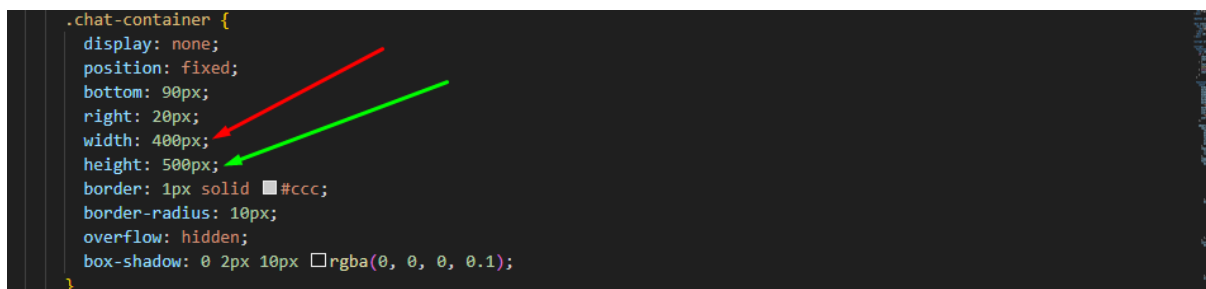
Criamos um exemplo simples em html e javascript para embedar o chat criado pelo Copilot Studio e utilizamos a documentação da Azure para aplicar as personalizações (vide referências em anexo).

3. Customizações

Para testar as customizações, criamos um arquivo index.html (vide código em anexo) para verificar as alterações de fonte, cor, tamanho e outros detalhes. A seguir, detalhamos cada customização:

3.1. Container do chat

Para modificar o tamanho e outras customizações do container do chat, procure o estilo “.chat-container {...}” e edite conforme a preferência:

A screenshot of a code editor showing CSS code for a chat container. The code is as follows:

```
.chat-container {  
  display: none;  
  position: fixed;  
  bottom: 90px;  
  right: 20px;  
  width: 400px;  
  height: 500px;  
  border: 1px solid #ccc;  
  border-radius: 10px;  
  overflow: hidden;  
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}
```

Two arrows are drawn on the code: a red arrow pointing to the 'width: 400px;' line and a green arrow pointing to the 'height: 500px;' line.

3.2. Cabeçalho do chat

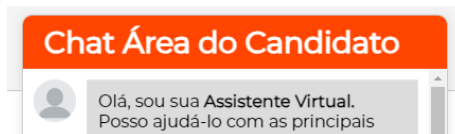
Para modificar a cor e a fonte do cabeçalho do chat, edite o seguinte script em css:

```

.chat-container .chat-header {
  align-items: center;
  background-color: #f4a460;
  color: white;
  display: flex;
  height: 50px;
  padding: 0 20px;
  font-family: "Montserrat", sans-serif;
  font-size: 14px;
}

```

No estilo “.chat-container .chat-header {...}” utilize o “background-color” para alterar a cor de fundo do decorator, “color” para alterar a cor da fonte e “font-family” para alterar a fonte utilizada. Resultado:



3.3. Botão do chat

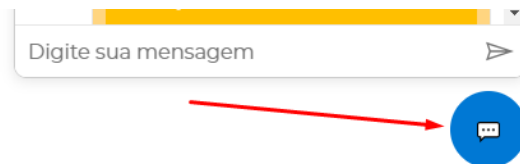
Para editar o formato, tamanho, cores ou fontes do botão do chat, faça as alterações no estilo “.chat-button {...}”:

```

.chat-button {
  position: fixed;
  bottom: 20px;
  right: 20px;
  background-color: #0078d4;
  color: white;
  border: none;
  border-radius: 50%;
  width: 60px;
  height: 60px;
  font-size: 16px;
}

```

Exemplo do botão para abrir e colapsar o chat:



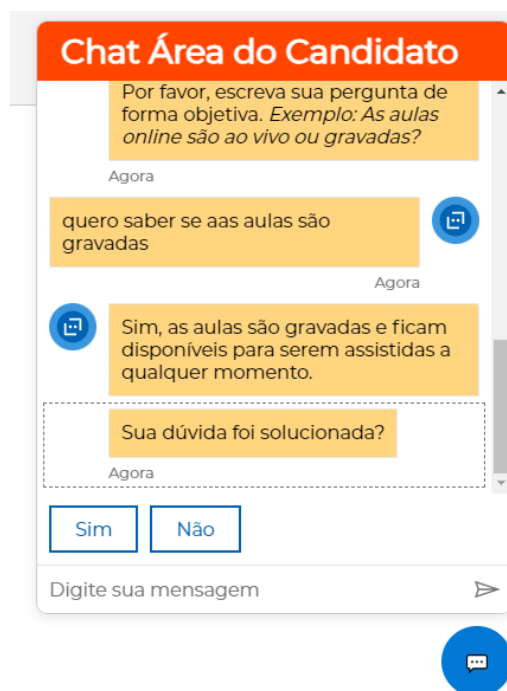
3.4. Customização dos elementos internos do chat

Para customizar os elementos internos do chat, exceto Adaptive Cards, como mensagens, caixas de texto, botão de enviar e imagens dos avatares, modifique o seguinte dicionário em javascript:

```
const styleOptions = {
  hideUploadButton: true,
  backgroundColor: "white",
  bubbleBackground: "#DCDCDC",
  bubbleFromUserBackground: "#DCDCDC",
  primaryFont: "Montserrat",
  botAvatarBackgroundColor: "#FFFFFF",
  botAvatarImage:
    "https://learn.microsoft.com/azure/bot-service/v4sdk/media/logo_bot.svg",
  botAvatarInitials: "AN",
  userAvatarBackgroundColor: "#FFFFFF",
  userAvatarImage:
    "https://learn.microsoft.com/azure/bot-service/v4sdk/media/logo_bot.svg",
  userAvatarInitials: "US",
};
```

A variável "backgroundColor" altera a cor do fundo do chat como um todo. As variáveis "bubbleBackground" e "bubbleFromUserBackground" alteram a cor das caixas de mensagens (cards de texto) do bot e do usuário, respectivamente. Utilize "primaryFont" para selecionar a fonte desejada. Por fim, utilize as configurações de avatar para alterar a imagem e iniciais do bot e usuário.

Exemplo de customização interna do chat (para configurações avançada, consulte a referência [5]):

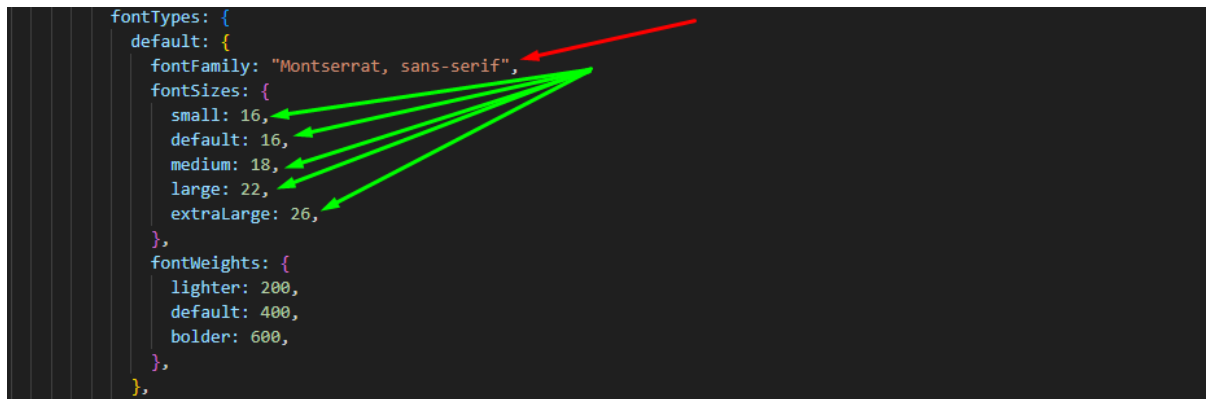


3.4. Customização dos Adaptive Cards

Para estilizar o menu inicial do chat é necessário alterar outro dicionário com as configurações do Adaptive Card. Edite o "adaptiveCardsHostConfig" para modificar a aparência dos elementos que utilizam adaptive cards, no nosso exemplo, o menu inicial.

Para editar tamanhos e estilo de fonte do Adaptive Card, procure por “fontTypes”:

```
fontTypes: {  
  default: {  
    fontFamily: "Montserrat, sans-serif",  
    fontSizes: {  
      small: 16,  
      default: 16,  
      medium: 18,  
      large: 22,  
      extraLarge: 26,  
    },  
    fontWeights: {  
      lighter: 200,  
      default: 400,  
      bolder: 600,  
    },  
  },  
},
```

A screenshot of a code editor showing the 'fontTypes' configuration. A red arrow points to the 'fontFamily' property, and five green arrows point to the 'small', 'default', 'medium', 'large', and 'extraLarge' properties within the 'fontSizes' object.

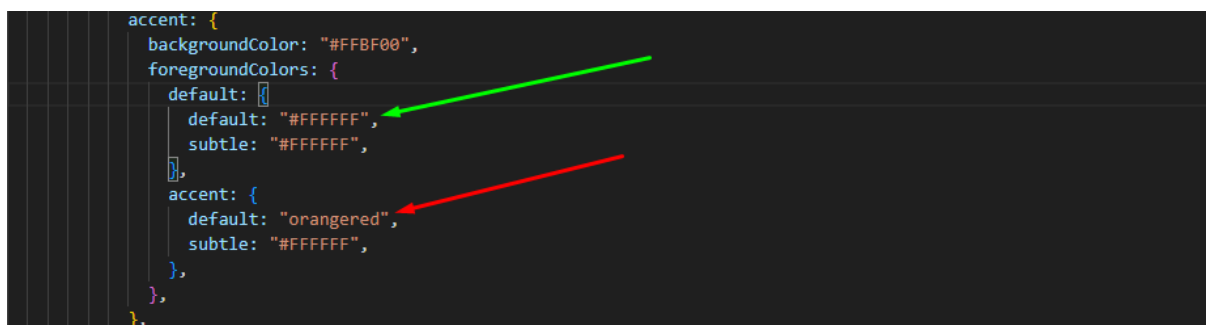
Já a cor e fonte da caixa de texto do Adaptive Card deve ser alterada em no objeto “default” dentro de “containerStyles”:

```
containerStyles: {  
  default: {  
    backgroundColor: "#FFD580",  
    foregroundColors: {  
      default: {  
        default: "#000000",  
        subtle: "#767676",  
      },  
      accent: {  
        default: "#FF0000",  
        subtle: "#FF0000",  
      },  
    },  
  },  
},
```

A screenshot of a code editor showing the 'containerStyles' configuration. A red arrow points to the 'backgroundColor' property, and a green arrow points to the 'default' property within the 'foregroundColors' object.

Podemos alterar a cor e fonte dos botões do Adaptive Card do menu alterando o “accent” dentro de “containerStyles”:

```
accent: {  
  backgroundColor: "#FFBF00",  
  foregroundColors: {  
    default: {  
      default: "#FFFFFF",  
      subtle: "#FFFFFF",  
    },  
    accent: {  
      default: "orangered",  
      subtle: "#FFFFFF",  
    },  
  },  
},
```

A screenshot of a code editor showing the 'accent' configuration. A green arrow points to the 'default' property within the 'foregroundColors' object, and a red arrow points to the 'default' property within the 'accent' object.

Para customizações adicionais dos Adaptive Cards e mais informações sobre o “adaptiveCardsHostConfig” consulte as referências [6] e [7].

4. Conclusão

Embora a customização do Webchat gerado pelo Copilot Studio seja limitada, ela ainda oferece a capacidade de fazer ajustes essenciais para atender a algumas necessidades específicas. O processo descrito, que envolve HTML, CSS e JavaScript, permite personalizar aspectos básicos como o container do chat, cabeçalho, botões e elementos internos. É possível realizar modificações importantes, como alterações de cor, tamanho e fonte, além de ajustes no menu inicial dos Adaptive Cards. Para personalizações mais avançadas ou específicas, é aconselhável consultar a documentação adicional e explorar os recursos fornecidos nas referências.

5. Referências

[1] WEB CHAT CUSTOMIZATION IN THE BOT FRAMEWORK SDK - BOT SERVICE.

Disponível em: [<https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-webchat-customization?view=azure-bot-service-4.0&preserve-view=true>]. Acesso em: [04/09/2024].

[2] CUSTOMIZE THE LOOK AND FEEL OF A COPILOT - MICROSOFT COPILOT

STUDIO. Disponível em: [<https://learn.microsoft.com/en-us/microsoft-copilot-studio/customize-default-canvas?tabs=web>]. Acesso em: [04/09/2024].

[3] GITHUB - MICROSOFT/BOTFRAMEWORK-WEBCHAT: A HIGHLY-

CUSTOMIZABLE WEB-BASED CLIENT FOR AZURE BOT SERVICES. Disponível em: [<https://github.com/microsoft/BotFramework-WebChat>]. Acesso em: [04/09/2024].

[4] BOTFRAMEWORK-WEBCHAT/SAMPLES/02.BRANDING-STYLING-AND-CUSTOMIZATION/A.BRANDING-WEB-CHAT AT MASTER ·

MICROSOFT/BOTFRAMEWORK-WEBCHAT. Disponível em:

[<https://github.com/microsoft/BotFramework-WebChat/tree/master/samples/02.branding-styling-and-customization/a.branding-web-chat#why-stylesetoptions>]. Acesso em: [04/09/2024].

[5] BOTFRAMEWORK-

WEBCHAT/PACKAGES/API/SRC/DEFAULTSTYLEOPTIONS.TS AT MASTER ·

MICROSOFT/BOTFRAMEWORK-WEBCHAT. Disponível em:

[<https://github.com/microsoft/BotFramework-WebChat/blob/master/packages/api/src/defaultStyleOptions.ts>]. Acesso em: [04/09/2024].

[6] PROVIDE ABILITY TO MODIFY ADAPTIVECARDHOSTCONFIG · ISSUE #2034 · MICROSOFT/BOTFRAMEWORK-WEBCHAT. Disponível em: [\[https://github.com/microsoft/BotFramework-WebChat/issues/2034\]](https://github.com/microsoft/BotFramework-WebChat/issues/2034). Acesso em: [04/09/2024].

[7] ADAPTIVECARDS/SAMPLES/HOSTCONFIG/WEBCHAT.JSON AT MASTER · MICROSOFT/ADAPTIVECARDS. Disponível em: [\[https://github.com/microsoft/AdaptiveCards/blob/master/samples/HostConfig/webchat.json\]](https://github.com/microsoft/AdaptiveCards/blob/master/samples/HostConfig/webchat.json). Acesso em: [04/09/2024].

6. Anexos

Index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Página Simples com Chat</title>
    <style>
      @font-face {
        font-family: "Montserrat";
        src: url("./Montserrat.ttf");
      }

      body {
        margin: 0;
        padding: 0;
        height: 100vh;
        display: flex;
        flex-direction: column;
        justify-content: space-between;
      }

      .header {
        text-align: center;
        padding: 20px;
        background-color: #f4f4f4;
        border-bottom: 1px solid #ccc;
      }

      .chat-button {
        position: fixed;
        bottom: 20px;
```

```

    right: 20px;
    background-color: #0078d4;
    color: white;
    border: none;
    border-radius: 50%;
    width: 60px;
    height: 60px;
    font-size: 16px;
}

.chat-container {
    display: none;
    position: fixed;
    bottom: 90px;
    right: 20px;
    width: 400px;
    height: 500px;
    border: 1px solid #ccc;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

.chat-container .chat-header {
    align-items: center;
    background-color: orangered;
    color: white;
    display: flex;
    height: 50px;
    padding: 0 20px;
    font-family: "Montserrat", sans-serif;
    font-size: 14px;
}

.chat-container #webchat {
    height: calc(100% - 50px);
    overflow: hidden;
    width: 100%;
}
</style>
</head>
<body>
    <div class="header">
        <h1>Página com Chat</h1>
    </div>

    <button class="chat-button" onclick="toggleChat()">💬 </button>

```

```

<div class="chat-container" id="chatContainer">
  <div class="chat-header">
    <h1>Chat Área do Candidato</h1>
  </div>
  <div id="webchat" role="main"></div>
</div>

<script src="https://cdn.botframework.com/botframework-
webchat/latest/webchat.js"></script>

<script>
function toggleChat() {
  var chatContainer = document.getElementById("chatContainer");
  if (
    chatContainer.style.display === "none" ||
    chatContainer.style.display === ""
  ) {
    chatContainer.style.display = "block";
    initializeWebChat();
  } else {
    chatContainer.style.display = "none";
  }
}

function initializeWebChat() {
  const styleOptions = {
    hideUploadButton: true,
    backgroundColor: "white",
    bubbleBackground: "#DCDCDC",
    bubbleFromUserBackground: "#DCDCDC",
    primaryFont: "Montserrat",
    botAvatarBackgroundColor: "#FFFFFF",
    botAvatarImage:
      "https://learn.microsoft.com/azure/bot-
service/v4sdk/media/logo_bot.svg",
    botAvatarInitials: "AN",
    userAvatarBackgroundColor: "#FFFFFF",
    userAvatarImage:
      "https://learn.microsoft.com/azure/bot-
service/v4sdk/media/logo_bot.svg",
    userAvatarInitials: "US",
  };

  adaptiveCardsHostConfig = {
    spacing: {
      small: 3,

```



```

    default: 8,
    medium: 20,
    large: 30,
    extraLarge: 40,
    padding: 10,
  },
  separator: {
    lineThickness: 1,
    lineColor: "#EEEEEE",
  },
  supportsInteractivity: true,
  fontTypes: {
    default: {
      fontFamily: "Montserrat, sans-serif",
      fontSizes: {
        small: 16,
        default: 16,
        medium: 18,
        large: 22,
        extraLarge: 26,
      },
      fontWeights: {
        lighter: 200,
        default: 400,
        bolder: 600,
      },
    },
  },
  monospace: {
    fontFamily: "'Courier New', Courier, monospace",
    fontSizes: {
      small: 16,
      default: 16,
      medium: 18,
      large: 22,
      extraLarge: 26,
    },
    fontWeights: {
      lighter: 200,
      default: 400,
      bolder: 600,
    },
  },
},
containerStyles: {
  default: {
    backgroundColor: "#DCDCDC",
    foregroundColors: {

```

```

        default: {
            default: "#000000",
            subtle: "#767676",
        },
        accent: {
            default: "#FF0000",
            subtle: "#FF0000",
        },
    },
    },
    accent: {
        backgroundColor: "#c9c9c9",
        foregroundColors: {
            default: {
                default: "#000000",
                subtle: "#000000",
            },
            accent: {
                default: "#0F3669",
                subtle: "#000000",
            },
        },
    },
    },
    },
    imageSizes: {
        small: 40,
        medium: 80,
        large: 160,
    },
    actions: {
        maxActions: 100,
        spacing: "default",
        buttonSpacing: 8,
        showCard: {
            actionMode: "inline",
            inlineTopMargin: 8,
        },
        actionsOrientation: "vertical",
        actionAlignment: "stretch",
    },
    adaptiveCard: {
        allowCustomStyle: true,
    },
    imageSet: {
        imageSize: "medium",
        maxImageHeight: 100,
    },
    },

```

```

factSet: {
  title: {
    color: "default",
    size: "default",
    isSubtle: false,
    weight: "bolder",
    wrap: true,
    maxWidth: 150,
  },
  value: {
    color: "default",
    size: "default",
    isSubtle: false,
    weight: "default",
    wrap: true,
  },
  spacing: 8,
},
};

const tokenEndpointURL = new URL(
  "https://c6541a989433e2588d8b5f038e90f4.eb.environment.api.powerplatform.com/powervirtualagents/botsbyschema/crd7f_faqAreaDoCandidato/directline/token?api-version=2022-03-01-preview"
);

const locale = document.documentElement.lang || "en";
const apiVersion = tokenEndpointURL.searchParams.get("api-version");

(async function () {
  const [directLineURL, token] = await Promise.all([
    fetch(
      new URL(
        `/powervirtualagents/regionalchannelsettings?api-version=${apiVersion}`,
        tokenEndpointURL
      )
    )
  ]).then((response) => {
    if (!response.ok) {
      throw new Error(
        "Failed to retrieve regional channel settings."
      );
    }
    return response.json();
  });
})();

```

```

        .then(({ channelUrlsById: { directline } }) => directline),
        fetch(tokenEndpointURL)
        .then((response) => {
            if (!response.ok) {
                throw new Error("Failed to retrieve Direct Line token.");
            }
            return response.json();
        })
        .then(({ token }) => token),
    ]);

    const directLine = WebChat.createDirectLine({
        domain: new URL("v3/directline", directLineURL),
        token,
    });

    const subscription = directLine.connectionStatus$.subscribe({
        next(value) {
            if (value === 2) {
                directLine
                    .postActivity({
                        localtimezone:
                            Intl.DateTimeFormat().resolvedOptions().timeZone,
                        locale,
                        name: "startConversation",
                        type: "event",
                    })
                    .subscribe();

                subscription.unsubscribe();
            }
        },
    });

    WebChat.renderWebChat(
        { directLine, locale, styleOptions, adaptiveCardsHostConfig },
        document.getElementById("webchat")
    );
})();
}
</script>
</body>
</html>

```