
Doc-PRO Widget Integration Guide

Version 1.0 – May 2025

1. Description

The Doc-PRO widget is an embeddable module that enables users to upload documents through an intuitive interface. It can be easily integrated into any webpage using a simple `<iframe>` element.

2. Basic Integration

Embed the widget into your HTML using the following code:

```
<iframe
  src="https://widget.botmaster.ai/default/uploader"
  frameborder="0"
  allow="camera *"
  id="docpro-widget"
  style="width: 100%; height: 600px;">
</iframe>
```

3. CSS Customization

Customize the look of the widget using CSS variables and font definitions:

```
.font-family {
  font-family: "Open Sans", sans-serif;
}

@layer base {
  :root {
    --primaryColor: 22 54 86;
    --secondaryColor: 248 133 44;
    --backgroundColor: 255 255 255;
    --textColor: 18 42 70;
    --detailColor: 229 231 235;
  }
}
```

4. postMessage Communication

After embedding the iframe, the browser console will show:

[WIDGET] widget-loaded - postMessage

This indicates the widget is ready but lacks configuration. Send the following postMessage object:

```
{
  "type": "widget-config",
```

```
"data": {
  "token":
  "ZGV2b3BzK25vbWFpbC1kZW1vQGRhemVyb2xhYi5jb206UGFzc3dvcmQkMQ==",
  "configuration": {
    "botId": "66d072ea5ec720af22364e10",
    "customer": "DOMESTICO | BUSINESS",
    "market": "LIBERO | TUTELATO",
    "commodity": "GAS | LUCE | DUAL",
    "cssUrl": "INSERT_HERE_PATH_CSS"
  }
}
```

5. Example: Angular Integration

Send configuration when widget is loaded:

```
@HostListener('window:message', ['$event'])
onMessage(event: { data: any }) {
  if (event.data.type === 'widget-loaded') {
    const message = {
      type: 'widget-config',
      data: {
        token: 'YOUR_ENCODED_TOKEN',
        configuration: {
          botId: '66d072ea5ec720af22364e10',
          customer: 'DOMESTICO',
          market: 'LIBERO',
          commodity: 'GAS',
          cssUrl: 'https://yourdomain.com/style.css'
        }
      }
    };
    const iframe = document.getElementById('docpro-widget') as HTMLIFrameElement;
    iframe.contentWindow?.postMessage(message, '*');
  }
}
```

6. Example: React Integration

React hook to send configuration to the widget once loaded:

```
useEffect(() => {
  const handleMessage = (event) => {
    if (event.data.type === "widget-loaded") {
      const message = {
        type: "widget-config",
        data: {
```

```
token: "YOUR_ENCODED_TOKEN",
configuration: {
  botId: "66d072ea5ec720af22364e10",
  customer: "DOMESTICO",
  market: "LIBERO",
  commodity: "GAS",
  cssUrl: "https://yourdomain.com/style.css"
}
};
const iframe = document.getElementById("docpro-widget");
iframe.contentWindow.postMessage(message, "*");
}
};
window.addEventListener("message", handleMessage);
return () => window.removeEventListener("message", handleMessage);
}, []);
```

7. Receiving Widget Responses

Listen for postMessages and parse the response:

```
@HostListener('window:message', ['$event'])
onMessage(event: { data: any }) {
  console.log(event.data); // Response from widget
}
```

8. JSON Response Structure

The widget sends a JSON response like the following:

```
{
  "codiceCliente": {
    "value": "318974832",
    "confidence": 93.76
  },
  "fornitore": {
    "value": "Enel Energia S.p.A.",
    "confidence": 51.07
  },
  "commodity": {
    "value": "gas naturale",
    "confidence": 87.46
  },
  "codiceFiscale": {
    "value": "RSSMRA80A01H501U",
    "confidence": 93.78
  },
}
```



```
"nomeTitolareFornitura": {  
  "value": "ANGIOLINA",  
  "confidence": 45.33  
}  
}
```