

medsolv.MRM

Dokumentation OBS-WIDGET

VERSION: 8/2026



Intention

Im Folgenden finden Sie eine Dokumentation zur Einbindung des OBS-Widget auf ihrer Webseite.

Inhalt

Intention.....	2
Inhalt	2
Einbettung in Website	3
Inline-Rendering.....	3
Custom Properties	3
Remote Access on Widget	4

Einbettung in Website

Empfohlene Methode das OBS Widget in einer Website einzubinden:

```
<div id="obs-widget"

    data-tenant-id="EXTERNAL_TENANT_ID"

    data-deployment="INSTALLATION_DEPLOYMENT"

    data-mode="fab"

    data-reschedule-token="url:rescheduleToken"

></div>

<script src="BASE_URL/widget.js" async></script>
```

Für mrm.med wäre es:

INSTALLATION_DEPLOYMENT: mrm.med

BASE_URL: <https://widget.arzttermin.med>

- Das Modul Online-Terminassistent muss  Administration -> Produktinformationen aktiviert sein.
- Damit kann unter  Administration -> Online-Service der Dienst mit den gewünschten Settings konfiguriert werden.
- Es muss unter  Administration -> Benachrichtigungen mindestens 1 Benachrichtigungsdienst konfiguriert sein (SMS Versand oder Mailversand).

Inline-Rendering

Alternativ kann das Widget inline gerendert werden.

- Dazu kann data-mode="inline" eingestellt werden. Das Widget rendert dann als iframe im gegebenem div (id="obs-widget")
- Für Inline ist es empfehlenswert eine feste height anzugeben (mindestens 30rem), da sich das iframe nicht mittels height auto oder height 100% automatisch anpassen kann.

Custom Properties

Mit data-fab-style kann ein typischer Style string wie data-fab-style= „bottom: 20px; right: 20px; scale: 0.8“ übergeben werden, womit der Style des Floating Action Buttons beeinflusst werden kann. Dabei sollte hauptsächlich die Position und der Scale angepasst werden.

Remote Access on Widget

Die Farben (primär, sekundär) sowie das Theme können später direkt in MRM eingestellt werden. Die dort gemachten Einstellungen werden sofort übernommen und beim nächsten Render berücksichtigt.

Mit `data-dummy-mode="true"` kann für Entwicklungszwecke der Entwicklungsmodus eingeschaltet werden, wobei keine echten Daten, sondern reine Dummy Daten geladen werden. Damit kann die Funktionalität und Optik des Widgets auf einer Website getestet werden.

Remote Access on Widget

Sobald das Widget initialisiert ist, sendet es ein event mit dem type „obs-app-ready“. Ab dann kann auf das iframe zugegriffen werden.

Entweder direkt über das source des app-ready Events (zum Beispiel zum rekonfigurieren der Widgetdaten:

```
window.addEventListener("message", function (event) {  
    if (event.data.type !== "obs-app-ready") return  
    console.log("[OBS Demo] App ready, sending reconfigure  
message")  
    event.source.postMessage(  
        {  
            type: "obs-reconfigure",  
            data: {  
                tenantId: cfg.tenantId || "master-tenant",  
                deployment: cfg.deployment || "local-dev",  
            },  
        },  
        "*",  
    )  
})
```

Remote Access on Widget

Das Widget kann außerdem bei Bedarf über ein custom browser event geöffnet werden:

```
function remoteOpenFab() {  
    const iframe = document.querySelector('#obs-widget-iframe')  
    if (!iframe) {  
        console.warn('[OBS Demo] Iframe nicht gefunden')  
        return  
    }  
    iframe.contentWindow.postMessage({ type: 'obs-remote-open' },  
    '*')  
}
```