



Nationale Feedback-Komponente (NFK) - Integrationsleitfaden

Änderungshistorie

Dokumentversion	Datum	Geänderte Kapitel	Beschreibung der Änderung
1.0	28.01.2022		Initiale Fassung des Integrationsleitfadens
1.1	14.11.2022	3 4	Aktualisierung der Angaben bei den Parametern Aktualisierung der Einbindung in das bestehende Angebot
1.2	14.02.2023	3 4 4.5	Ergänzung des Parameters <code>test</code> Aktualisierung der Einbindung in das bestehende Angebot Ergänzung zur Änderung der Darstellung bei der Integration
1.3	20.07.2023	3 4.6	Aktualisierung der Kontextdaten <code>region</code> und <code>issue</code> als erforderliche Parameter Ergänzung Kapitel zur Content-Security-Policy
1.4	16.11.2023	3	Ergänzung des Parameters <code>PrivacyURL</code>
2.0	24.05.2024	5.5, 6	Erweiterung um eingebetteten Feedback-Dialog (Inline-Einbindung); Erweiterung um Kapitel <i>Sonstige Hinweis</i> ; Ergänzung Änderungshistorie und übergreifende Aktualisierungen, inkl. Umbenennung von LeiKa-Schlüssel zu FIM-Leistungsschlüssel
2.1	28.11.2024	5.6 1, 5.1, 5.4, 5.5, 6.2 4	Erweiterung um empfangbare Events Umstellung auf feedback.gov.de Umbenennung von LeiKa-Schlüssel/FIM-Leistungsschlüssel zu Leistungsschlüssel
2.2	25.04.2025	5, 5.1, 5.2, 5.4, 5.5, 5.5, 5.6, 6, 6.1	Änderung der Einbindung von <code>bmi-widget.js</code> zu <code>widget.js</code> und Umbenennung des Formulars und der dazugehörigen Elemente
2.3	27.05.2025	6.2	Unterstützung für Style-Nonce.
2.4	17.03.2026	6 → 5.7, 5.8 5.7	Integration von Kapitel 6 (Sonstiges) in Kapitel 5 (Einbindung) Dokumentation der seitenspezifischen CSS-Anpassung
2.5	17.06.2026	4	Ergänzung zum Parameter <code>region</code> (Bundes ARS)
2.6	04.09.2026	5.5	Integration des Inline-Dialogs ohne gesondertes JavaScript

Inhalt

1.	Überblick	4
2.	Registrierung des Portals	4
3.	Konfiguration der gewünschten Instrumente	5
4.	Kontextdaten	5
5.	Einbindung in das bestehende Angebot als Dialog	6
5.1	Integrationsskript	7
5.2	Aktionselement zum Öffnen des Feedback-Dialogs	7
5.3	Laden des Feedback-Dialogs.....	8
5.4	Asynchrone Alternative für das Integrationsskript.....	8
5.5	Eingebetteter Feedback-Dialog	9
5.6	Events	9
5.7	Änderung der Darstellung	10
5.8	Content-Security-Policy	14

1. Überblick

Dieses Dokument dient als Anleitung für die Integration der Nationalen Feedback-Komponente in bestehende Webseiten mit Informationen, Online-Verfahren sowie Unterstützungsdienste¹.

Die Nationale Feedback-Komponente stellt eine einfache Möglichkeit zur Verfügung, um die Anforderungen der EU-Durchführungsverordnung 2020/1121 hinsichtlich **Feedback** in bestehende Angebote zu integrieren. Hierfür müssen nur zwei Zeilen zu bestehenden Webseiten hinzugefügt werden.

Das Beispiel für die Integration unter <https://stage.feedback.gov.de/demofeedback/dialog/info-service> zeigt die Einbindung in ein Demoportal.

Folgende Schritte sind nötig, um die Nationale Feedback-Komponente in eine bestehende Webseite einzubinden:

- Registrierung des Portals in der Administrationsoberfläche (]init[)
- Konfiguration der gewünschten Feedback-Instrumente in der Administrationsoberfläche (]init[)
- Identifizierung der Kontextdaten
- Einbindung in das bestehende Angebot

Für einen ersten Test der Einbindung können die Schritte *Registrierung des Portals* und *Konfiguration der gewünschten Instrumente* übersprungen und das Demoportal sowie die Standard Feedback-Instrumente verwendet werden. Auch wenn bereits ein Portal mit Instrumenten existiert, über welches Sie sich anbinden möchten, können die ersten beide Schritte übersprungen werden. In diesen Fällen sind nur *Identifizierung der Kontextdaten* und *Einbindung in das bestehende Angebot als Dialog* durchzuführen.

2. Registrierung des Portals

Bevor die Feedback-Komponente in einem neuen Portal eingebunden werden kann, muss das Portal in der Administrationsoberfläche registriert werden. Dort wird die eindeutige Portal-ID, die Konfiguration der Standardfarben, des Standardlogos, der Standardsprache und die Portal-Administratoren des Portals hinterlegt.

Die initiale Registration des Portals muss durch einen Administrator von]init[erfolgen.

Zum kurzfristigen Testen kann das bereits fertig konfigurierte Testportal mit der Portal-ID "demofeedback" verwendet werden, welches eigene Beispielfarben vorgibt (siehe Kapitel *Kontextdaten*). Das so übermittelte Feedback kann nicht durch Sie eingesehen werden. Wenn Sie sich über ein bestehendes Portal anbinden möchten (z.B. ein Länder- oder Ressortportal), benötigen Sie die registrierte Portal-ID mit den dazugehörigen Instrument-IDs. Verwenden Sie für weitergehende Nachfragen das Kontaktformular unter <https://feedback.gov.de>.

¹ ehemals Hilfs- und Problemlösungsdienste

3. Konfiguration der gewünschten Instrumente

Die im Rahmen des Feedbacks abgefragten Informationen können in der Administrationsoberfläche konfiguriert werden.

Für die in der EU-Durchführungsverordnung 2020/1121, Kapitel III, Artikel 6 geforderten Informationen sind bereits Instrumente angelegt, die direkt verwendet werden können.

Für die Abfrage von zusätzlichen Informationen müssen neue Instrumente angelegt werden. Die initiale Anlage neuer Instrumente kann nur durch einen Administrator von]init[in der Administrationsoberfläche erfolgen. Für jedes Instrument können anschließend weitere Administratoren definiert werden, die die einzelnen Schritte und Fragen des neuen Instruments konfigurieren dürfen.

4. Kontextdaten

Die Nationale Feedback-Komponente benötigt einige Kontextdaten, um die Feedback-Eingabe erfolgreich darzustellen zu können. Diese Kontextdaten werden beim Öffnen der Nationalen Feedback-Komponente mitgegeben.

Es werden folgende Parameter unterstützt:

Parameter	erforderlich / optional	Beschreibung
portalId	erforderlich	Gibt die ID an, die dem Portal in der Administrationsoberfläche zugewiesen wurde (siehe Kapitel <i>Registrierung des Portals</i>).
instrumentId	erforderlich	Die ID des Instruments, welches die Konfiguration des gewünschten Feedback-Dialoges enthält. Diese wird über die Administrationsoberfläche definiert. Zum Testen können folgende Werte der Standardinstrumente verwendet werden: • online-proc für Online-Verfahren • info-service für Webseiten mit Informationen • assist-service für Unterstützungsdienste (wird nur für diese angeboten) Möchte man kein Freitextfeld nutzen, so können die folgenden IDs verwendet werden: • online-proc-noft für Online-Verfahren • info-service-noft für Webseiten mit Informationen
language	optional	Die Sprache der oder des aktuellen Nutzenden als ISO 639-1 Code. Aktuell werden nur "de" und "en" unterstützt, perspektivisch können aber weitere Sprachen hinzukommen. Falls der Parameter nicht angegeben wird oder ungültig ist, wird die in der Portalkonfiguration hinterlegte Sprache angezeigt.

Parameter	erforderlich / optional	Beschreibung
submitUrl	optional	Die URL für die der Dialog geöffnet wurde. Wird mit der aktuellen URL des Browsers befüllt, falls nicht angegeben (maximal 2047 Zeichen).
privacyUrl	optional	Überschreibt die Datenschutz-URL aus der Portalkonfiguration. Nur wenn hier oder in der Portalkonfiguration eine Datenschutz-URL angegeben ist, wird im Dialog ein Hinweis auf diesen Link ausgegeben.
test	optional	Wird dieser Parameter auf "true" gesetzt, so wird der Feedbackdialog zwar angezeigt und das Feedback wird auch entgegengenommen, es wird aber nicht abgespeichert. Dieser Parameter eignet sich daher zum Testen der Einbindung des Produktivsystems, ohne dass dort eingegebenes Testfeedback übermittelt wird.
issueType	optional	Der Typ des Anliegens (maximal 255 Zeichen), zum Beispiel "FIM-Leistung".
issue ²	erforderlich	ID des Anliegens: Leistungsschlüssel (ehemals: LeiKa-Schlüssel, FIM-Leistungsschlüssel) der angebotenen Leistung (maximal 255 Zeichen).
region	erforderlich	Ort / Region des Anliegens: <i>Amtlicher Regionalschlüssel (ARS)</i> der angebotenen Leistung (maximal 255 Zeichen). Kann kein spezifischer ARS angegeben werden, wird empfohlen den Bundes ARS (lt. FITKO 22.1.2024) zu verwenden --> 000000000000

Die letzten drei Parameter *issueType*, *issue* und *region* sind Metadaten, über die in der Administrationsoberfläche Berechtigungen für die Sichtbarkeit der Feedbacks für die einzelnen Auswerter definiert werden können. Für die Metadaten *issue* und *region* muss das vorgegebene Format (siehe Beschreibung) verwendet werden. Der Parameter *issueType* kann vom Portal frei verwendet werden.

Für ein Beispiel zur Parameterübergabe siehe Kapitel *Aktionselement zum Öffnen des Feedback-Dialogs*.

5. Einbindung in das bestehende Angebot als Dialog

Die Nationale Feedback-Komponente bietet einen Dialog zur Integration in eine bestehende Webseite. Beim Feedback-Dialog handelt es sich um ein dynamisches "Overlay-Popup", welches sich bei Aktivierung über die Webseite legt und den Dialog anzeigt.

Die Einbindung besteht aus drei Komponenten:

1. Einem kleinen Integrationsskript: widget.js (< 1KB)
2. Einem Aktionselement zum Öffnen des Feedbacks-Dialogs, zum Beispiel einem Button
3. Dem eigentlichen JavaScript-Code des Feedbacks-Dialogs, welches beim ersten Öffnen nachgeladen wird

² für Online-Verfahren ist die Übermittlung des Leistungsschlüssels (ehemals LeiKa-Schlüssel oder FIM-Leistungsschlüssel) verpflichtend

Als Alternative zum Feedback-Dialog als "Overlay-Popup" bieten wir auch eine eingebettete "Inline" Variante an. Details dazu finden Sie unter *Eingebetteter Feedback-Dialog*.

5.1 Integrationsskript

Zum Bereitstellen des JavaScript Einsprungpunktes, muss das folgende Skript eingebunden werden:

```
<script src="https://stage.feedback.gov.de/<Ihre Portal-ID>/widget.js"
  id="nfkskWidgetScript" async></script>
```

Codeblock 1 – Dieses Skript an das Ende des HTML-Body Tags hängen

Der Platzhalter **<Ihre Portal-ID>** muss durch die Portal-ID ersetzt werden, die dem Portal zugewiesen wurde.

Die ID des Skripts muss zwingend **nfkskWidgetScript** sein.

Der genannte Server ist nur zum Testen der Integration, das dorthin gelieferte Feedback ist nur auf der Administrationsoberfläche der Stage-Umgebung sichtbar und wird auch nur bei Bedarf an die Testumgebung der EU übermittelt.

Für das Produktivsystem ist **stage.feedback.gov.de** durch **feedback.gov.de** zu ersetzen. Dies kann erst geschehen, nachdem auch auf diesem System die entsprechende Portalkonfiguration erstellt wurde.

5.2 Aktionselement zum Öffnen des Feedback-Dialogs

Um den Feedback-Dialog zu öffnen, muss ein Aktionselement definiert werden. Dies kann zum Beispiel ein Button sein, der den Feedback-Dialog öffnet, sobald man ihn drückt:

```
<button class="nfkskWidgetButton"
  data-context='{ "portalId": "<Portal-ID>", "instrumentId": "online-proc" }'>
  Feedback
</button>
```

Codeblock 2 – Button zum Öffnen des Feedback-Dialogs

Das Aktionselement kann ein beliebiges HTML-Element sein, welches einen `onClick`-Handler erlaubt, also zum Beispiel `<button>`, `<input>`, `<a>`, `<div>` oder `<span>`. Das Element muss die CSS-Klasse `"nfkskWidgetButton"` haben, damit automatisch auf das `onclick`-Event reagiert wird. Alternativ kann ein eigener `onClick`-Handler implementiert werden, der dann `nfkskWidgetButton.init(button)` aufruft.

Das `data-context` Attribut enthält ein JSON-Objekt mit den unter *Kontextdaten* beschriebenen Parametern. Der Platzhalter **<Portal-ID>** muss durch die Portal-ID ersetzt werden, die dem Portal zugewiesen wurde. Das Attribut `instrumentId` kann auch `info-service` en

Die Beschriftung ist frei wählbar und muss nicht notwendigerweise "Feedback" lauten.

Es können auch mehrere Aktionselemente pro Seite integriert werden.

Eine Möglichkeit das Aktionselement erst anzuzeigen, wenn des Integrationsskript geladen wurde, finden Sie im Kapitel *Asynchrone Alternative für das Integrationsskript*.

In JSON sind alle Attributnamen und Textwerte zwingend in `"..."` zu schreiben, eine Verwendung von `'...'` wie in JavaScript oder auch Attributnamen ohne Anführungszeichen sind nicht erlaubt!

5.3 Laden des Feedback-Dialogs

Beim Klick auf das Aktionselement wird das eigentliche JavaScript nachgeladen und das Overlay geöffnet, welches dann die hinterlegten Konfigurationen lädt und das Feedback abfragt. Dies geschieht automatisch durch die Feedback-Komponente, weitere Änderungen an der bestehenden Webseite sind daher nicht nötig.

5.4 Asynchrone Alternative für das Integrationsskript

Möchte man verhindern, dass durch Netzwerkprobleme das Aktionselement ohne Funktion angezeigt wird, kann das Integrationsskript nebst Knopf auch asynchron geladen werden:

```
<script>
(function () {
  if (!document.getElementById('nfkskWidgetScript')) {
    var currentScript = document.currentScript;
    var script = document.createElement('script');
    script.setAttribute('id', 'nfkskWidgetScript');
    script.setAttribute('type', 'text/javascript');

    // URL anpassen!
    script.setAttribute('src',
      'https://stage.feedback.gov.de/<Ihre Portal-ID>/widget.js');

    script.onload = function () {
      var feedbackButton = document.createElement('button');
      feedbackButton.onclick = function () { nfkskWidgetButton.init(this); };

      // Textinhalt, CSS-Klasse und Kontextdaten sollten angepasst werden
      feedbackButton.textContent = 'Feedback';
      feedbackButton.className.add('btn');
      feedbackButton.setAttribute('data-context',
        '{ "portalId": "<Portal-ID>", "instrumentId": "online-proc" }');

      currentScript.parentNode.insertBefore(feedbackButton, currentScript);
    };

    script.onerror = function (event) {
      // Anzeige der Fehlermeldung ist optional
      var errorElement = document.createElement('span');
      errorElement.textContent = 'Fehler! Script konnte nicht geladen werden';
      errorElement.className = 'error';
      currentScript.parentNode.insertBefore(errorElement, currentScript);
    };
    document.body.appendChild(script);
  }
})();
</script>
```

Codeblock 3 – Inline Javascript an Stelle des statischen Buttons einfügen

Dieses Beispiel kann beliebig an die eigenen Bedürfnisse angepasst und auch in eine eigene JavaScript-Datei ausgelagert werden.

5.5 Eingebetteter Feedback-Dialog

Der Feedback Dialog kann optional auch direkt in einer existierenden Webseite eingebettet werden ("Inline"), das heißt ohne ein gesondertes Aktionselement, das den Dialog erst öffnet.

Hierfür ist folgender Code in das HTML der Seite einzubinden:

```
<div id="nfkskWidgetRoot" data-context='
  "portalId": "<Portal-ID>",
  "instrumentId": "online-proc",
  "inline": true,
  "initialFocus": false,
  "issueType": "...",
  "issue": "...",
  "region": "..."
'>
  <!-- An dieser Stelle wird der Dialog angezeigt-->
</div>
<script src="https://stage.feedback.gov.de/<Ihre Portal-ID>/widget.js"
  id="nfkskWidgetScript" async></script>
```

Codeblock 4 – HTML zum Einbetten des Feedback-Dialogs

An der Stelle, an der das HTML-Element mit der ID `nfkskWidgetRoot` in der Seite eingebettet ist, wird der Feedback-Dialog ausgegeben. Wird kein solches HTML-Element gefunden, wird der Dialog an der Stelle der Skript-Einbindung ausgegeben (in dem Fall darf das Skript also nicht im Header der Seite geladen werden).

Der Platzhalter `<Portal-ID>` muss durch die Portal-ID ersetzt werden, die dem Portal zugewiesen wurde.

Für das **Livesystem** ist `stage.feedback.gov.de` durch `feedback.gov.de` zu ersetzen. Dies kann erst geschehen, nachdem auch auf diesem System die entsprechende Portalkonfiguration erstellt wurde.

Die Beschreibung der einzelnen Parameter finden Sie im Kapitel *Kontextdaten*.

Der Dialog hat nach links und rechts keinen Rahmen, sondern wird auf 100% der Breite des HTML-Elements skaliert, damit er in einer Linie mit dem Rest der Seite steht. Falls die konfigurierte Hintergrundfarbe des Dialogs nicht der Hintergrundfarbe der Seite entspricht oder die restlichen Elemente einen Rahmen besitzen, muss daher für das HTML-Element `nfkskWidgetRoot` im CSS der Seite ebenfalls ein zusätzlicher Rahmen definiert werden. Informationen zur weiteren Konfiguration der Darstellung erhalten Sie im Kapitel *Änderung der Darstellung*.

5.6 Events

Während Nutzende die Feedbackkomponente verwenden, werden Events im Browser publiziert, damit die aufrufende Seite auf diese Ereignisse reagieren kann. Diese Events werden mit dem Namen `"nfkskWidget"` versehen und können mittels des Typs im Attribut `"detail.type"` unterschieden werden.

Beispiel für das Registrieren für ein solches Event:

```
window.addEventListener("nfkskWidget", function(event) {
  if (event.detail.type === "submit") {
    console.log("Formular in Schritt " + event.detail.step
      + " erfolgreich abgesendet");
  }
});
```

Codeblock 5 - Eventlistener für das erfolgreiche Absenden des Dialogs

Folgende Eventtypen werden aktuell gesendet:

Eventtyp	Beschreibung	Weitere Attribute von <i>event.detail</i>	Beispiel für <i>event.detail</i>
<i>submit</i>	Wird nach jeder erfolgreichen Übermittlung der Daten gesendet.	<i>description</i> : Eine kurze englische Beschreibung, was abgesendet wurde. <i>step</i> : Der Schritt (0 basiert), in dem das Feedback abgesendet wurde. In den meisten Fällen "0" für das First-Level-Feedback und "2" für das Second-Level-Feedback.	<pre>{ type: 'submit', description: 'Submitted feedback', step: 2 }</pre>
<i>submitError</i>	Wird bei einem Fehlschlag der Übermittlung gesendet.	<i>description</i> : Eine kurze englische Beschreibung, wo der Fehler passiert ist. <i>message</i> : Die Fehlermeldung des Servers. <i>step</i> : Der Schritt (0 basiert), in dem das Feedback abgesendet wurde.	<pre>{ type: 'submitError', description: 'Error during feedback submission', message: "Couldn't find portal 'abc'", step: 0 }</pre>
<i>render</i>	Wird immer aufgerufen, wenn ein neuer Schritt dargestellt wird.	<i>step</i> : Der Schritt (0 basiert), in dem das Feedback abgesendet wurde.	<pre>{ type: 'render', step: 1 }</pre>
<i>close</i>	Wird nach dem Schließen des Dialogs aufgerufen	<i>cause</i> : Der Grund für das Schließen, mögliche Werte: <i>Abort button</i>: das X oben rechts im Dialog wurde gedrückt <i>Close button</i>: die oder der Nutzende hat den Dialog bis zum Ende ausgeführt und dort auf den "Schließen"-Knopf gedrückt	<pre>{ type: 'close', cause: 'Abort button' }</pre>

Der Dialog ist als [Shadow DOM Komponente](#) implementiert, wodurch die Adressierung der enthaltenen Komponenten grundsätzlich nicht möglich ist. Hierauf ist auch bei sonstigen Eventlistenern zu achten (zum Beispiel Focus-Events), da immer der Dialog als Ursprung der Events angegeben wird, nicht die enthaltenen Komponenten.

5.7 Änderung der Darstellung

Pro Portal wird in der Administrations-Oberfläche ein eigenes Set von Farben und Texteingenschaften hinterlegt, mit dem das Aussehen des Feedback-Dialogs gesteuert wird.

Soll der Dialog in unterschiedlichen Diensten, die das gleiche Portal verwenden, mit verschiedenem Aussehen eingebunden werden, so können die Farben sowie die Schriftart und -größe bei der Einbindung überschrieben werden. Hierfür ist zusätzlich zum `data-context` Attribut ein `data-theme`

Attribut am Aktionselement zu definieren (siehe auch Kapitel *Aktionselement zum Öffnen des Feedback-Dialogs*):

```
<button class="nfkskWidgetButton"
  data-context='{ ... }'
  data-theme='{ "textColor": "#3C3C3C", "textBackgroundColor": "#FFFFFF", ... }'>
  Feedback
</button>
```

Codeblock 6 – Konfiguration des Aussehens des Dialogs

Die unterstützten Eigenschaften des `data-theme` Attributs sind:

Name im <code>theme</code> Attribut	Name in der Oberfläche	Beschreibung	Format
<code>textColor</code>	Allgemeiner Text	Textfarbe für alle normalen Elemente	Farbe als Hexadezimalwert, zum Beispiel: #000000
<code>textBackgroundColor</code>	Allgemeiner Hintergrund	Hintergrundfarbe des Dialogs	Farbe als Hexadezimalwert, zum Beispiel: #FFFFFF
<code>headlineColor</code>	Hauptüberschrift	Textfarbe der Überschrift	Farbe als Hexadezimalwert, zum Beispiel: #000000
<code>error</code>	Fehlertext	Textfarbe der Fehlermeldungen	Farbe als Hexadezimalwert, zum Beispiel: #ff0000
<code>textfieldColor</code>	Textfeld	Textfarbe der Texteingabefelder	Farbe als Hexadezimalwert, zum Beispiel: #000000
<code>textfieldBackgroundColor</code>	Textfeld-Hintergrund	Hintergrundfarbe der Texteingabefelder	Farbe als Hexadezimalwert, zum Beispiel: #FFFFFF
<code>footerTextColor</code>	Footer	Farbe des Texts im Footer (nur relevant, falls auch das Kontaktformular verwendet wird)	Farbe als Hexadezimalwert, zum Beispiel: #FFFFFF
<code>footerBackgroundColor</code>	Footer-Hintergrund	Hintergrundfarbe der Fußzeile des Dialogs; wird nur angezeigt, wenn das Overlay verwendet wird. Beim eingebetteten Dialog verwendet die Fußzeile immer die allgemeine Hintergrundfarbe.	Farbe als Hexadezimalwert, zum Beispiel: #5A6272
<code>buttonPrimaryColor</code>	Primärbutton	Textfarbe des "Absenden" und "Weiter"-Knopfes	Farbe als Hexadezimalwert, zum Beispiel: #FFFFFF

Name im <code>theme</code> Attribut	Name in der Oberfläche	Beschreibung	Format
<code>buttonPrimaryBackgroundColor</code>	Primärbutton-Hintergrund	Hintergrundfarbe des "Absenden" und "Weiter"-Knopfes; wird auch für die Farbe der Sterne verwendet	Farbe als Hexadezimalwert, zum Beispiel: #1D7BBB
<code>textFamily</code>	Schriftart	Bezeichnung der verwendeten Schriftart für alle Elemente. Ein Font, der auf der Seite deklariert wird, kann hier referenziert werden.	Analog zur font-family in CSS
<code>textSize</code>	Schriftgröße Text	Größe der normalen Texte in Pixeln; Zeilenhöhen, Abstände und Größe der Eingabeelemente werden davon abhängig berechnet	10-30px (es wird empfohlen, für die Schriftgröße maximal 20px zu verwenden)
<code>headlineSize</code>	Schriftgröße Überschrift	Größe der Überschrift des Dialogs	10-30px

Alle Eigenschaften, die nicht über das `theme` Attribut gesetzt werden oder nicht dem erwarteten Format entsprechen, werden aus der Konfiguration des Portals gelesen.

Änderungen an der Darstellung des Dialogs sollten nur über die Administrations-Oberfläche oder über das `theme` Attribute vorgenommen werden. Es wird nicht empfohlen, mittels eigenem CSS weitere Anpassungen vorzunehmen, da sich die Struktur der HTML-Elemente durch Änderungen der Konfiguration oder durch neue Softwareversionen verändern kann. Überdies ist der Dialog als [Shadow DOM Komponente](#) implementiert, wodurch die Adressierung der enthaltenen Komponenten per CSS nicht möglich ist. Hierauf ist auch bei eigenen Eventlistenern zu achten (zum Beispiel Focus-Events), da immer der Dialog als Ursprung der Events angegeben wird, nicht die enthaltenen Komponenten.

Seitenspezifisches CSS

In begrenztem Umfang können ausgewählte Elemente des Dialogs per CSS angepasst werden. Hierzu werden diese Elemente über `part`-Attribute freigegeben und können mittels `::part(...)` im CSS der Seite selektiert werden. Es kann nur das jeweils adressierte Element angepasst werden, verschachtelte oder umliegende Elemente können nicht verändert werden. Änderungen, die über das `theme`-Attribut vorgenommen werden können, haben eine höhere Spezifität und übersteuern die CSS-Regeln. Die `::part(...)`-Selektoren sind auf das Element `nfksk-widget` anzuwenden.

Liste der verfügbaren part-Namen

Hinweis: Grau hinterlegte Zeilen sind nur in der Supportkomponente verfügbar.

part-Name	Beschreibung
button	Alle Buttons
button-back	Button, der zum vorherigen Inhalt führt
button-cancel	Button, der eine Aktion abbricht
button-close	Button, der das Dialogfenster schließt.
button-next	Button, der zum nächsten Inhalt führt
button-save	Button zum Übernehmen der Änderungen - falls ein Nutzer auf „Bearbeiten“ klickt, dann wird dieser Button ausgegeben, um die Änderungen zu speichern.
button-select	Auswahlbutton bei mehreren Standardanliegen
button-submit	Button, der das Formular abschickt
checkbox	Alle Checkbox-Elemente
dialog	Dies ist der mittig zentrierte Dialog. Hierüber kann zum Beispiel die Hintergrundfarbe des Dialogs verändert werden.
dialog-close-button	Der Button zum Schließen des Dialogs oben rechts.
dialog-content	Dialog Inhalt
dialog-footer	Dialog Fußzeile
dialog-title	Dialog Titel
fieldset	Feldgruppe global
fieldset- <i>{TYPE}</i>	Feldgruppe für individuelle Eingabemöglichkeit Beispiel: fieldset-rating, fieldset-textarea, fieldset-radio
input-error-display	Wenn ein erforderliches Feld nicht ausgefüllt wurde, erscheint ein Fehlertext, der über diesen Selektor geändert werden kann.
label	Feldbeschriftungen
label-radio	Feldbeschriftung für Radio Steuerelemente
radio	Radio-Steuerelemente
radio-group	Gruppe von Radio-Steuerelementen.
rating	Das äußerste Element der Bewertungs-Komponente.
rating-item	Einzelner Stern im Normalzustand.
rating-item-active	Einzelner Stern im ausgewählten Zustand
rating-item-hover	Einzelner Stern, wenn die Maus darüber schwebt.
rating-items	Gruppe der Sterne

part-Name	Beschreibung
richtext	Beschreibungs- und Informationstexte, zum Beispiel der Datenschutzhinweis. Wenn sich in dem Element weitere Elemente befinden, so können diese nur über Vererbung modifiziert werden.
select	Gruppe von Auswahlkacheln, wie beim Standardanliegen oder der Instrumentauswahl
select-widget	Eine einzelne Auswahlkachel
textarea	Mehrzeiliges Textfeld
textarea-counter	Textfelder sind auf eine bestimmte Länge von Zeichen begrenzt. Hierüber kann der Zähler unterhalb eines Textfeldes verändert werden.

Eine Anpassung im CSS kann zum Beispiel so aussehen:

```
nfsk-widget {
  /* Alle Buttons werden in Kapitälchen geschrieben und erhalten einen runden Rand: */
  &::part(button) {
    font-variant: small-caps;
    border-radius: 10px;
  }

  /* Die einzelnen Fragen erhalten einen Rand. */
  &::part(fieldset) {
    border: 1px solid lightgray;
    padding: 5px;
  }
}
```

Codeblock 7 - Beispielhafte Anpassung am Feedback-Dialog

5.8 Content-Security-Policy

Falls die Webseite einen restriktiven *Content-Security-Policy* Header setzt, muss für die Einbindung des Formulars der Formularserver als `connect-src` und `script-src` deklariert werden. Außerdem müssen Inline-Styles erlaubt sein, entweder über die Regel `style-src 'unsafe-inline'` oder über ein *Nonce*:

```
Content-Security-Policy: connect-src ... https://feedback.gov.de/; script-src ...
https://feedback.gov.de/; style-src 'nonce-...eindeutiger einmaliger Nonce...'
```

Codeblock 8 – CSP Header

Für das Stagesystem muss stattdessen `"https://stage.feedback.gov.de/"` verwendet werden. Alternativ kann auf beiden Umgebungen auch `"https://*.feedback.gov.de/"` gesetzt werden.

Wird ein Nonce für `style-src` verwendet, muss dies dem Formular bei der Einbindung mitgeteilt werden:

```
<button class="nfskWidgetButton"
  data-context='{ ..., styleNonce="...Nonce..." }' >Feedback</button>
```

Codeblock 9 - Übergabe des Nonce

Webseiten, die mit *Webpack* arbeiten und dort bereits den Nonce definiert haben, brauchen den Wert nicht mitgeben, da auch die globale Variable `__webpack_nonce__` beachtet wird.