

Änderungsverlauf azh: 30.09.2024 - 07.10.2024

04.10.2024 13:50:54 michaela.jahnke

Replacing file "NOVENTI_Whitepaper_Abrechnung_HiMi_DIN-A4.pdf" to
"/user_upload/azh/Whitepaper/NOVENTI_Whitepaper_Abrechnung_HiMi_DIN-A4.pdf"

Replacing file "NOVENTI_Whitepaper_Abrechnung_HiMi_DIN-A4.pdf" to
"NOVENTI_Whitepaper_Abrechnung_HiMi_DIN-A4.pdf"

04.10.2024 13:37:46 michaela.jahnke

Replacing file "NOVENTI_Whitepaper_Abrechnung_HeiMi_DIN-A4.pdf" to
"/user_upload/azh/Whitepaper/NOVENTI_Whitepaper_Abrechnung_HeiMi_DIN-A4.pdf"

Replacing file "NOVENTI_Whitepaper_Abrechnung_HeiMi_DIN-A4.pdf" to
"NOVENTI_Whitepaper_Abrechnung_HeiMi_DIN-A4.pdf"

04.10.2024 11:57:51 michaela.jahnke

Uploading file "NOVENTI_Update Abrechnung_Blankverordnung Physiotherapie.pdf" to
"/user_upload/azh/Webinare_Downloads/Blankverordnung/"

Uploading file "NOVENTI_Update Abrechnung_Blankverordnung Physiotherapie.pdf" to
"Blankverordnung"

01.10.2024 07:45:18 bgmadmin

be_users:57 (steinegger.a)

/ Root /

Record 'steinegger.a' (be_users:57) was updated. (Online).

Username

kinne0

steinegger.a

Password

Name

Lukas Kinne

Anna Steinegger

Email

lukas.kinne@noventi.de

01.10.2024 07:44:24 bgmadmin

be_users:57 (steinegger.a)

/ Root /

Record 'kinne0' (be_users:57) was inserted on page '[root-level]' (0)

01.10.2024 07:36:40 schneider

tt_content:20979 (iframe aktuell)

/ Root / Webseiten / Rezeptabrechnung und Branchensoftware / Landingpages / NOVENTI Ora /
NOVENTI Ora - Jetzt bestellen /

Record 'iframe aktuell' (tt_content:20979) was updated. (Online).

Text

```

<iframe id="tfa" src="https://noventi.tfaforms.
net/4891678"
net/forms/view/4891679"
height="420" style="width:100%;overflow:hidden;" frameborder="0"></iframe> <script> /*
 * Helper script for iframe-published FormAssembly.com forms. * Auto-resize iframe to
fit content. */ (function () { // Übergabe Parameter ans iFrame var params =
document.location.search; if (params) { params = params .replace("utm_clientid=",
"tfa_136=") .replace("utm_source=", "tfa_137=") .replace("utm_medium=", "tfa_138=")
.replace("utm_campaign=", "tfa_139=") .replace("utm_content=", "tfa_141="); var iframe
= document.getElementById('tfa'); console.log(iframe.src); iframe.src = iframe.src +
params; console.log(iframe.src); } /* * A backwards compatible implementation of
postMessage * by Josh Fraser (joshfraser.com) * released under the Apache 2.0 license.
* * This code was adapted from Ben Alman's jQuery postMessage code found at: *
http://benalman.com/projects/jquery-postmessage-plugin/ * * Other inspiration was taken
from Luke Shepard's code for Facebook Connect: *
http://github.com/facebook/connect-js/blob/master/src/core/xd.js */ var XD = function
() { var interval_id, last_hash, cache_bust = 1, attached_callback, window = this;
return { postMessage: function (message, target_url, target) { if (!target_url) {
return; } target = target || parent; // default to parent if (window['postMessage']) {
// the browser supports window.postMessage, so call it with a targetOrigin // set
appropriately, based on the target_url parameter. target['postMessage'](message,
target_url.replace(/([^\:]+\:\/\/[^\:\/]+).*/, '$1')); } else if (target_url) { // the
browser does not support window.postMessage, so set the location // of the target to
target_url#message. A bit ugly, but it works! A cache // bust parameter is added to
ensure that repeat messages trigger the callback. target.location =
target_url.replace(/#.*$/, '') + '#' + (+new Date) + (cache_bust++) + '&' + message; }
}, receiveMessage: function (callback, source_origin) { // browser supports
window.postMessage if (window['postMessage']) { // bind the callback to the actual
event associated with window.postMessage if (callback) { attached_callback = function
(e) { if ((typeof source_origin === 'string' && e.origin !== source_origin) ||
(Object.prototype.toString.call(source_origin) === "[object Function]" &&
source_origin(e.origin) === !1)) { return !1; } callback(e); }; } if
(window['addEventListener']) { window[callback ? 'addEventListener' :
'removeEventListener']('message', attached_callback, !1); } else { window[callback ?
'attachEvent' : 'detachEvent']('onmessage', attached_callback); } } else { // a polling
loop is started & callback is called whenever the location.hash changes interval_id &&
clearInterval(interval_id); interval_id = null; if (callback) { interval_id =
setInterval(function () { var hash = document.location.hash, re = /^#\d+&/; if (hash
!== last_hash && re.test(hash)) { last_hash = hash; callback({data: hash.replace(re,
'')}); } }, 100); } } } }); /** * Generates a random id (with very high probability
of uniqueness). * @return {string} */ function getRandomId() { return (Math.random() +
0.00001).toString(36).substr(2, 5) + (Math.random() + 0.0000001).toString(36).substr(2,
5); } // The code below runs on the parent page, i.e. the page that embeds // a
FormAssembly form through an <IFRAME> HTML tag. // Find FormAssembly iframe(s) and add
two URL parameters to their "src" attributes: // - hostURL: the URL of this page (the
parent page that embeds the form in it) // - faIframeUniqueId: a unique ID generated
for every IFRAME; it is used // for communication between this page and its IFRAME(s).
try { var iframes = document.querySelectorAll('iframe'); for (var i = 0; i <
iframes.length; i++) { if (iframes[i].src.indexOf(hostURL) === -1) { // Generate a
unique ID and save it in a custom attribute var faIframeUniqueId = getRandomId();
iframes[i].setAttribute('data-fa-iframe-unique-id', faIframeUniqueId); var indexFormId
= iframes[i].src.indexOf('tfaforms.net'); var formId = indexFormId > -1 ?
iframes[i].src.substr(indexFormId + 13, indexFormId + 20) : '';
iframes[i].setAttribute('data-fa-formid', formId); // Get SID from URL var sid = null;
try { sid = decodeURIComponent(window.location.search.split('jsid=')[1].split("&")[0]);
} catch (e) { sid = null; } // Preserve any other parameter. var url = iframes[i].src;
var hash = ""; if (url.indexOf('#') !== -1) { hash = "#" + (url.split('#')[1]); url =
url.split('#')[0]; } var sep = (url.indexOf("?") === -1) ? "?" : "&"; iframes[i].src =
url + sep + "faIframeUniqueId=" + faIframeUniqueId + "&jsid=" + sid + "&hostURL=" +
encodeURIComponent(window.location.href) + hash; } } } catch (e) { // iframe not found
or unexpected error. } // Runs on parent page: receive height parameter and update
IFRAME's height. XD.receiveMessage(function (event) { try { var messageString =

```

```
event.data.split(','); var heightOrCommand = messageString[0]; var messageIframeId =
messageString[1]; var source = messageString[2]; var numIframesApplied = 0; // Find the
origin of the message. In other words, identify // which IFRAME the message has come
from, and apply the new height // to it. var iframes =
document.querySelectorAll('iframe'); for (var i = 0; i < iframes.length; i++) { var
iframeId = iframes[i].getAttribute('data-fa-iframe-unique-id'); var formId =
iframes[i].getAttribute('data-fa-formid'); if (typeof iframeId === "undefined" ||
iframeId === null || iframeId === "" || typeof formId === "undefined" || formId ===
null || formId === "") { iframeId = null; } if (source ==
iframes[i].src.replace(/^https?:/, '') || iframeId == messageIframeId || formId ==
messageIframeId ) { numIframesApplied++; if (!isNaN(heightOrCommand)) { // save scroll
position, to restore after the iframe resize (prevents jumping around) var scrollTop =
window.pageYOffset || document.documentElement.scrollTop; // resize iframe
iframes[i].height = parseInt(heightOrCommand); // restore scroll position, after a
timeout to allow for a repaint. setTimeout( function() {
document.documentElement.scrollTop = document.body.scrollTop = scrollTop; },0); } else
{ if (heightOrCommand == "submitted") { // Will scroll to top of iframe
window.scrollTo(0, iframes[i].getBoundingClientRect().top) } if (heightOrCommand ==
"pagechange") { // Will scroll to top of iframe window.scrollTo(0,
iframes[i].getBoundingClientRect().top) } } } if (!numIframesApplied) { //
console.log('None of the existing IFRAMES matched the URL, so the message has not been
applied. '); } } catch (e) { // unexpected message. } });})(); </script>
```