Die JavaScript-API von expui

Eine Referenzkarte von Exponential UI, der JavaScript-Bibliothek der Administration - Packer-Schlüssel, exp_config(), der Exp-Kern, Serveraufrufe mit Exp.io, Dialoge, Tabellen, Datumsauswahl, Uploads und Werkzeugleisten.

Source file: `doc/ger-de/hilfe/erweiterungsentwicklung/teil-8-referenz/expui-javascript-api.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/extension-development/part-8-reference/expui-javascript-api.md

Teil 8, Referenz 30. Exponential UI (die Erweiterung `expui`, im Browser das Objekt `Exp`) ist seit der Entfernung von YUI die JavaScript-Bibliothek der Administration. Erweiterungen verwenden sie für ihre Administrationsseiten. Diese Seite fasst ihre API-Dokumentation zusammen (`doc/API.md` im expui-Repository, zum Zeitpunkt des Schreibens Version 1.0.0.1); die vollständige Beschreibung jedes Moduls steht in dessen Ordner `doc/modules/`. Wie Sie alten YUI-Code darauf umstellen, steht in Von YUI zu expui.

Laden

Template-Aufruf	Lädt
`{exp_config( hash( 'prefs', array( ... ), 'strings', array( ... ) ) )}`	den Konfigurationsblock der Seite: Voreinstellungen und übersetzte Texte für Ihr Skript
`{ezscript_require( 'exp::core' )}`	jQuery 4 und den Kern
`{ezscript_require( array( 'exp::core', 'exp::io' ) )}`	zusätzlich die Serveraufrufe
`{ezcss_require( 'exp/core.css' )}`	die Design-Tokens (`--exp-*`) und Hilfsklassen

Modulschlüssel: `exp::collapse`, `exp::sticky`, `exp::dialog`, `exp::upload`, `exp::datatable`, `exp::datepicker`, `exp::autosave`, jeweils mit seinem Stylesheet `exp/<module>.css`. **Die Administrationsdesigns laden sie alle auf jeder Seite**, sodass eine Administrationsseite Ihrer Erweiterung meist nur ihr eigenes Skript braucht.

Der Kern (Exp)

API	Tut
`Exp.ready( fn )`	ausführen, sobald die Seite bereit ist
`Exp.i18n( text, params )`	ein Text aus den `strings` von `exp_config()`
`Exp.on( name, fn )`, `Exp.off( name, fn )`, `Exp.emit( name, data )`	seitenweite Ereignisse; versehen Sie Ihre Namen mit einem Präfix (`myext:saved`)
`Exp.register( name, init )`, `Exp.start( root, only )`	Verhalten aus HTML starten (`data-*`-Markup)
`Exp.prefs.get( name, fallback )`, `Exp.prefs.set( name, value )`	Benutzervoreinstellungen
`Exp.keys.bind( combo, fn, options )`	Tastenkürzel

API	Tut
`Exp.token()`	das Formular-Token der Seite
`Exp.reducedMotion`	`true`, wenn der Benutzer weniger Bewegung gewünscht hat

Serveraufrufe (Exp.io)

API	Tut
`Exp.io.call( fn, args, options )`	eine ezjscore-Serverfunktion; standardmäßig per POST; gibt ein Promise mit dem `content` der Funktion zurück und lehnt mit einem `Exp.io.Error` ab
`Exp.io.form( form, options )`	ein Formular mit seinen Dateien absenden und dabei das Formular-Token hinzufügen
`Exp.io.poll( fn, args, options )`	alle `every` ms erneut aufrufen, bis `until( content )` wahr ist
`Exp.io.url( path )`	die Adresse eines Pfads im aktuellen Siteaccess

JavaScript

```
Exp.io.call( 'myext::items', [ 2 ], { method: 'GET' } ).then( function ( items ) {
    // use items
}, function ( error ) {
    // error.kind: signedout, refused, server, network, timeout, invalid
} );
```

Module

Modul	API
`exp::dialog`	`Exp.dialog.open( options )`, `confirm( text )`, `alert( text )`, `form( url, options )`, `create( options )`; Markup `data-exp-dialog`
`exp::datatable`	`$( el ).expDataTable( options )`; Markup `data-exp-datatable`
`exp::datepicker`	`$( el ).expDatePicker( options )`; `window.showDatePicker()`
`exp::upload`	`$( el ).expUpload( options )`; Markup `data-exp-upload`
`exp::collapse`	`Exp.collapse( conf )`, `$( el ).expCollapse( conf )`
`exp::sticky`	`$( toolbar ).expSticky( options )`, `Exp.sticky.start( toolbar, options )`
`exp::autosave`	`new Exp.autosave.AutoSubmit( conf )`, `$( form ).expAutosave( conf )`, `$( el ).expPreview( conf )`

Alle Dialoge geben Promises zurück und verwenden das browsereigene Element `<dialog>`.

Unter einer Content Security Policy

Exponential UI wird aus Dateien geladen, und seine Module starten über `data-*`-Markup, sodass eine damit gebaute Seite kein Inline-Skript braucht (CSP: Inline-Skripte in Dateien verlagern). Übergeben Sie die Texte Ihrer Seite über `exp_config()` statt über ein Inline-Skript.

Verwandte Seiten

- Zurück: Fehlercodes und Modulergebnisse
- Weiter: admin4-Komponenten und -Tokens
- Referenz der ezjscore-Serverfunktionen
- YUI aus Exponential entfernt
- Freie Dokumentation: Diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

> **Werden Sie Exponential-Partner**
>
> Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.
>
> Partner: hier beginnen | Wie Sie sich bewerben

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.