MIGRATING FROM JQUERY
Core Journey to Vanilla JS

Andreas Nedbal / T3DD24

ABOUT ME

ANDREAS NEDBAL

Senior Frontend Developer @ in2code

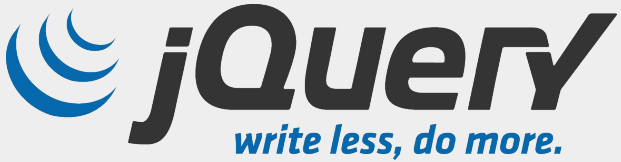
TYPO3 Core Merger: Backend UX

Andreas Kienast 3 hours ago

Hail to @pixeldesu, the mighty jQuery slayer.

👑 1 🪥 2 🧡 1 🪓 1

ABOUT JQUERY



DOM-manipulation library released in 2006

Unified the APIs of different browser vendors with simpler methods

In 2021, the share of the top 10m websites using jQuery was 75%

...and, it's not dead!

jQuery 4.0 Preview released in 2024!

Refactoring

Topics and their Solutions

$(...)

SOLUTIONS

- Understanding the code, then following up with
  - `document.querySelector`
  - `document.querySelectorAll`
- Replacing methods with vanilla counterparts
  - `.attr()` -> `get/setAttribute`
  - `.empty().append(...)` -> `replaceChildren`
- Many, many find/replace regular expressions
  - `.data('test')` -> `/\.data\('([\w\d_]+)'\)/` -> `.dataset.$1`
  - `.addClass('red')` -> `/\.addClass\('([\w\d_]+)'\)/` -> `.classList.add('$1')`
- Lots of manual adjustment afterwards because DOM API has no chaining of methods

BEFORE

```
const $tableElement = $(`table[data-table="${payload.table}"]`);
const $rowElement = $tableElement
  .find(`tr[data-uid="${payload.uid}"]`);
const $panel = $tableElement.closest('.panel');
const $panelHeading = $panel.find('.panel-heading');
const $translatedRowElements = $tableElement
  .find(`[data-l10nparent="${payload.uid}"]`);
```

AFTER

```
const tableElement = document
  .querySelector(
    `table[data-table="${payload.table}"]`
  );
const rowElement = tableElement
  .querySelector(
    `tr[data-uid="${payload.uid}"]`
  );
const panel = tableElement.closest('.panel');
const panelHeading = panel.querySelector('.panel-heading');
const translatedRowElements =
  tableElement.querySelectorAll<HTMLElement>(
    `[data-l10nparent="${payload.uid}"]`
  );
```

BEFORE

```
if ($rowElement.data('l10nparent') === '0' ||
    $rowElement.data('l10nparent') === '') {
    const count = Number(
        $panelHeading
            .find('.t3js-table-total-items')
            .html()
    );
    $panelHeading.find('.t3js-table-total-items').text(count - 1);
}
```

AFTER

```
if (rowElement.dataset.l10nparent === '0' ||
    rowElement.dataset.l10nparent === '') {
    const count = Number(
        panelHeading
            .querySelector('.t3js-table-total-items').textContent
    );
    const tableTotalItems = panelHeading
        .querySelector('.t3js-table-total-items');

    if (tableTotalItems !== null) {
        tableTotalItems.textContent = String(count - 1);
    }
}
```

$(...).on(...)

SOLUTION

jQuery event handlers work differently to vanilla ones, especially considering global events that apply to any element, even new ones.

TYPO3 offers an alternative to jQuery `.on()` with the `RegularEvent` class.

REGULAREVENT

```
constructor(eventName: string, callback: Listener, options:
AddEventListenerOptions)
```

```
public bindTo(element: EventTarget)
```

binds the event handler to the element for the given event (`addEventListener`)

```
public delegateTo(element: EventTarget, selector: string)
```

delegates the event to every element that matches selector inside element (multi-argument `$.on(...)`)

BEFORE

```
$(document).on(
  'change',
  'input[name=unlockDependencyIgnoreButton]',
  (e: JQueryEventObject): void => {
    const $actionButton = $('.t3js-dependencies');
    $actionButton.toggleClass('disabled', !$(e.currentTarget).prop('checked'));
});
```

AFTER

```
new RegularEvent('change', (e: Event, target: HTMLInputElement): void => {
    const actionButton = document.querySelector('.t3js-dependencies');

    if (target.checked) {
        actionButton.classList.remove('disabled');
    } else {
        actionButton.classList.add('disabled');
    }
}).delegateTo(document, 'input[name=unlockDependencyIgnoreButton]');
```

$.ajax

SOLUTION

TYPO3 offers a custom request implementation as a replacement for $.ajax called `AjaxRequest`.

To easily work with the responses from requests, a custom `AjaxResponse` class exists as well.

AJAXREQUEST

```
constructor(url: URL|string)
```

```
async get/post/put/delete(init: RequestInit)
```

common request methods that are Promise resolve-able

```
abort()
```

method to abort the request

AJAXRESPONSE

Automatically gets created by the request methods in `AjaxRequest` and returned in the Promise.

```
public async resolve(expectedType?: string)
```

this method resolves the request and automatically handles the response data and content type either based on the response headers or the expectedType argument

BEFORE

```
$.ajax({
    url: $me.attr('href'),
    dataType: 'html',
    success: (data: any): void => {
        $uploadForm.find('.t3js-upload-form-target').html(data);
    },
});
```

AFTER

```
new AjaxRequest(target.href)
  .get()
  .then(async (response: AjaxResponse): Promise<void> => {
    uploadForm.querySelector('.t3js-upload-form-target')
        .innerHTML = await response.resolve();
});
```

ANIMATIONS
.fadeIn()
.fadeOut()

SOLUTIONS

CSS!

…and…

…removing them entirely! Animations are not really accessible anyway and most of them just were there because they maybe looked “cool” at the time.

BEFORE

```
const $rowElements = $().add($rowElement)
                        .add($translatedRowElements);
$rowElements.fadeTo('slow', 0.4, (): void => {
  $rowElements.slideUp('slow', (): void => {
    $rowElements.remove();
    if ($tableElement.find('tbody tr').length === 0) {
      $panel.slideUp('slow');
    }
  });
});
```

AFTER

```
[rowElement, ...translatedRowElements].forEach((rowElement) => {
  rowElement?.remove();
});

if (tableElement.querySelector('tbody tr') === null) {
  panel.remove();
}
```

```
$('<ul/>').append(
    $('<li/>').append(
        $('<span/>').text('jQuery rocks!')
    )
);
```

JSX...JQUERYX

SOLUTION

Lit

LIT

Lit (also known as LitElements) is a simple library to build web components with.

It provides a base class and a lot of utilities to simplify development of web components.

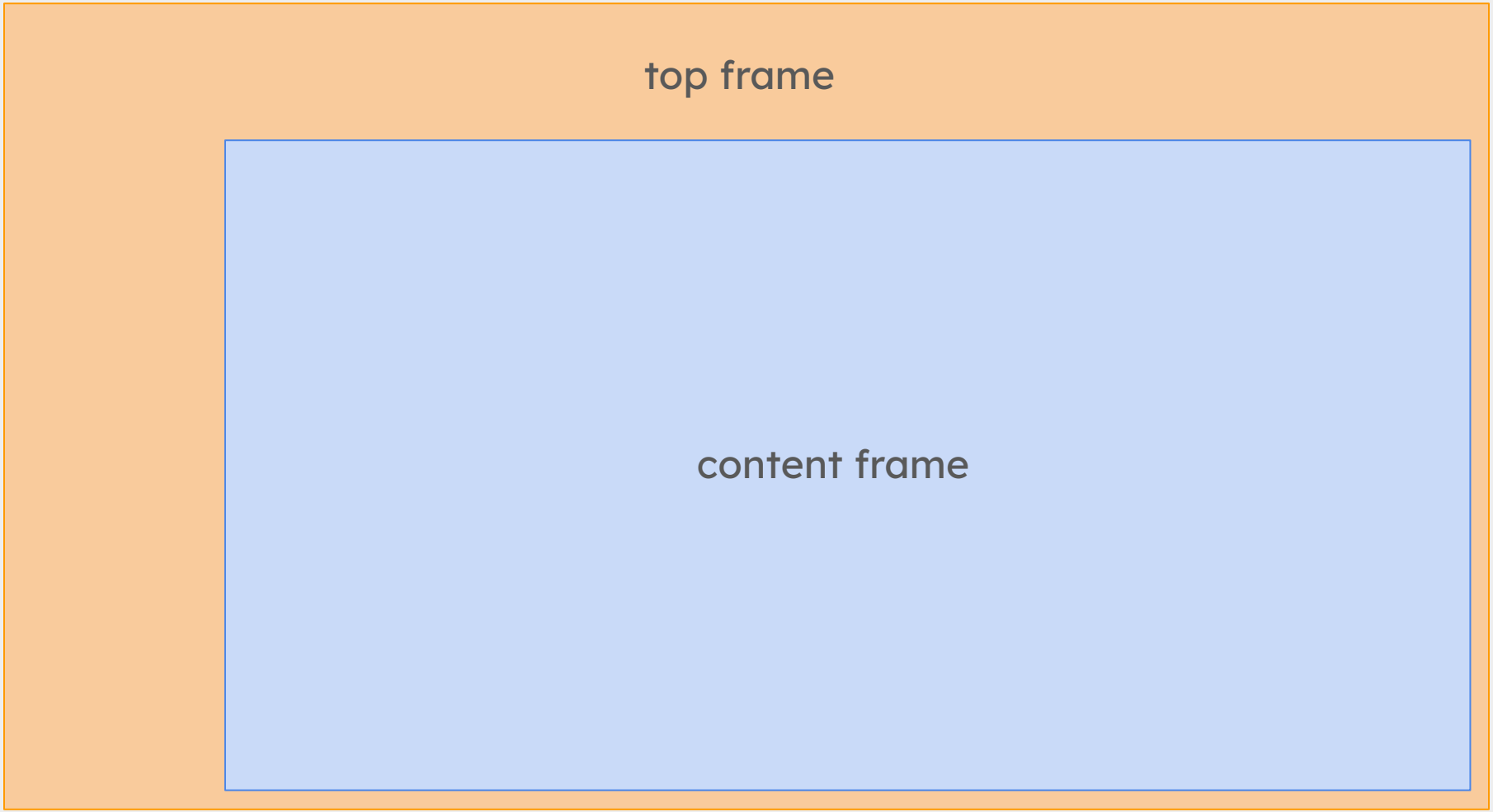
Lit was initially part of the Polymer library by Google, which is in maintenance by now.

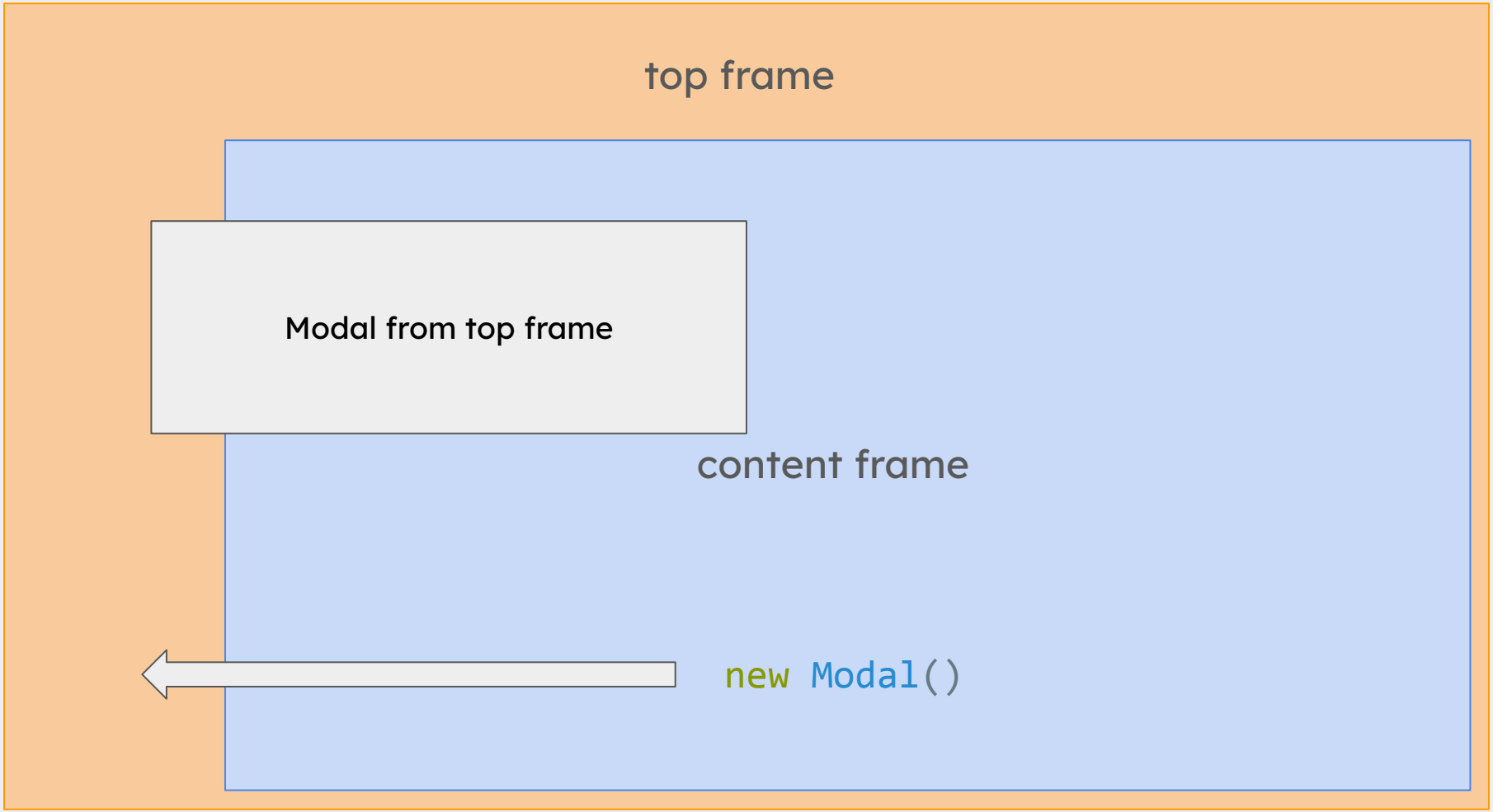
```
    for (const comment of comments) {
        const $panel = $('<div />', { class: 'panel panel-default' });

        if (comment.user_comment.length > 0) {
            $panel.append(
                $('<div />', { class: 'panel-body' }).html(comment.user_comment),
            );
        }

        $panel.append(
            $('<div />', { class: 'panel-footer' }).append(
                $('<span />', { class: 'badge badge-success me-2' }).text(comment.previous_stage_title + ' >
' + comment.stage_title),
                $('<span />', { class: 'badge badge-info' }).text(comment.tstamp),
            ),
        );

        $comments.append(
            $('<div />', { class: 'media' }).append(
                $('<div />', { class: 'media-left text-center' }).text(comment.user_username).prepend(
                $('<div />').html(comment.user_avatar),
                ),
                $('<div />', { class: 'media-body' }).append($panel),
            ),
```

```
protected renderComment(comment: Comment): TemplateResult {
    return html`
      <div class="media">
        <div class="media-left text-center">
          <div>
            ${unsafeHTML(comment.user_avatar)}
          </div>
          ${comment.user_username}
        </div>
        <div class="panel panel-default">
          ${comment.user_comment ? html`
          <div class="panel-body">
            ${comment.user_comment}
          </div>
        ` : nothing}
          <div class="panel-footer">
            <span class="badge badge-success me-2">
              ${comment.previous_stage_title} > ${comment.stage_title}
            </span>
            <span class="badge badge-info">
              ${comment.tstamp}
          </div>
        </div>
      </div>
    `;
}
```

AFTER

The problem with frames, Modals and Lit


top frame
content frame


top frame
Modal from top frame
content frame
new Modal()

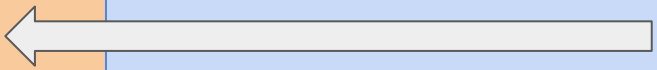
top frame

Modal from top frame

<lit-component/> ???

content frame

```
import LitComponent from '...'

new Modal(
  <lit-component/>
)
```



PROBLEM

In Firefox, loaded modules are constrained to the frames they are loaded from, so the lit-element that the top frame modal tries to render can’t be found. **Technically, this is the correct behaviour.**

Blink/Webkit browsers just don’t seem to care.

SOLUTION

`topLevelModuleImport()`

a method for import calls that can be delegated to the top window from inside frames, so that code used inside modals is loaded in the top window.

BEFORE

```
import '@typo3/backend/some-element'


// ...
```

AFTER

```
await topLevelModuleImport('@typo3/backend/some-element');


// ...
```

Where are we now?

STATE OF JQUERY IN TYPO3 NOW

- 15 explicit imports of jQuery are left in the Core
- 2 are wizards (base class and localization), the rest is EXT:form

TYPO3 10.4
135 explicit imports

TYPO3 13.2
15 explicit imports

ACKNOWLEDGEMENTS

ANDREAS KIENAST

BENJAMIN FRANZKE

So...should you do it?

No!

VERDICT

TYPO3 is a product that tries to keep up with modern technologies (where it makes sense)

In a website project, it’s mainly viable for relaunches.

If you have jQuery in an ongoing project, it’s not worth the effort rewriting everything. If the budget is low, even more so.

jQuery is still maintained and can be kept using.

DISCLAIMER

I showcased some TYPO3-specific JavaScript today.

You are free to use them in your projects, but beware that everything frontend (or "*backend frontend*") related in the Core is internal.

If it's not documented, everything is subject to change at random and might break in your extensions, so be careful when depending on Core JS functionality!

Questions?

Thanks for listening!

Hope you enjoyed this talk :)

Feedback? Want to chat?

Fediverse (Mastodon/…):
@pixel@desu.social

Twitter:
@pixeldesu

TYPO3 Slack:
@pixeldesu

SOURCES

$() patch:
https://review.typo3.org/c/Packages/TYPO3.CMS/+/81810

RegularEvent patch:
https://review.typo3.org/c/Packages/TYPO3.CMS/+/81858

AjaxRequest patch:
https://review.typo3.org/c/Packages/TYPO3.CMS/+/63484

Lit patch:
https://review.typo3.org/c/Packages/TYPO3.CMS/+/81832