

Herzlich Willkommen zum inovex Meetup!



inovex ist ein innovations- und qualitätsgetriebenes IT-Projekthaus mit dem Leistungsschwerpunkt „Digitale Transformation“.

Aktuelle Schwerpunkte:

- Agile Transformation
- Product Development Workshops
- E-Health
- Recommender Systems
- Generative AI

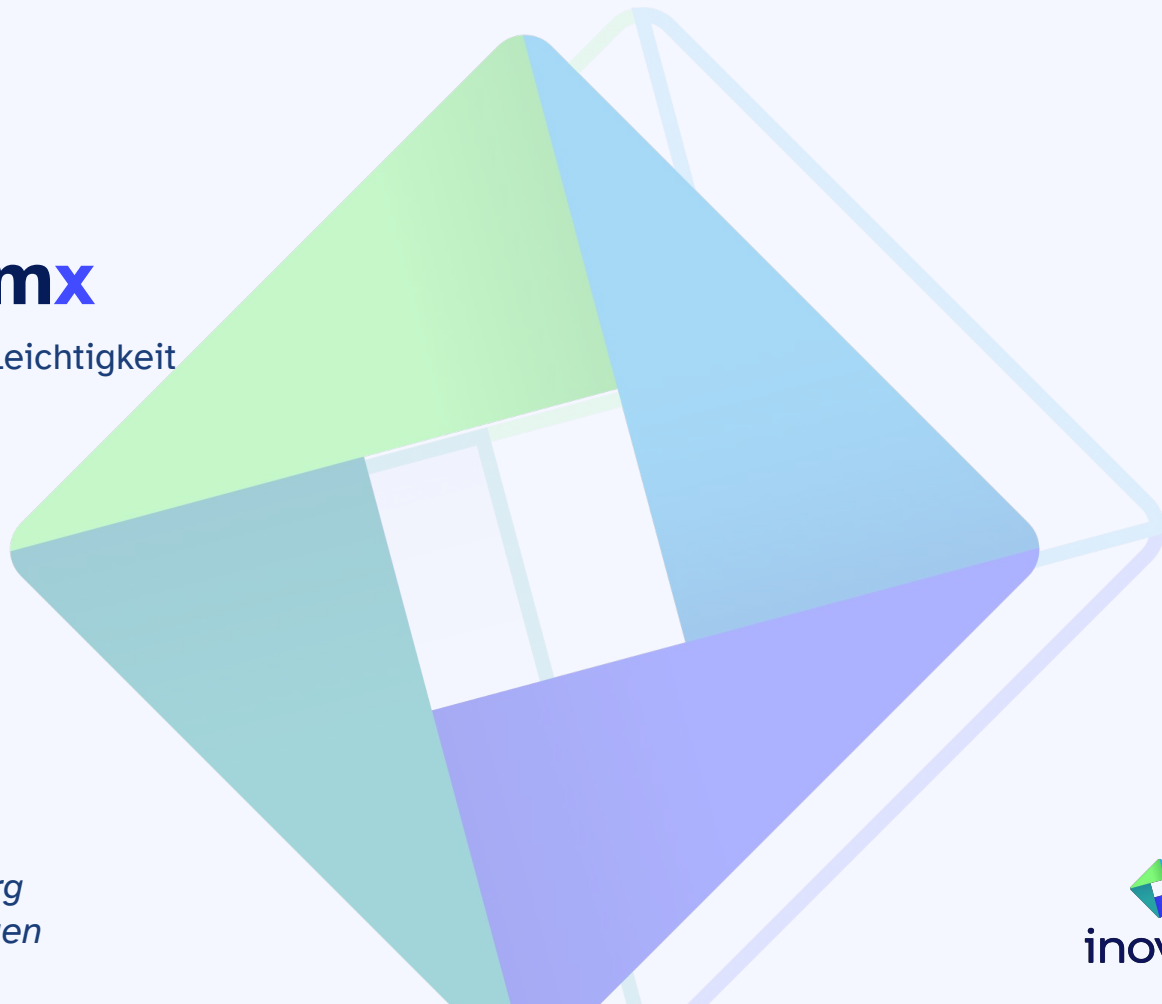


Einführung in htmx

Dynamische Webanwendungen mit Leichtigkeit

Johanna Dolinga

*Karlsruhe · Köln · München · Hamburg
Berlin · Stuttgart · Pforzheim · Erlangen*



Johanna Dolinga



Fullstack Entwicklerin

- Webentwicklung (React & Vue)
- UX
- Agile Development



Johanna Dolinga



@jdolinga

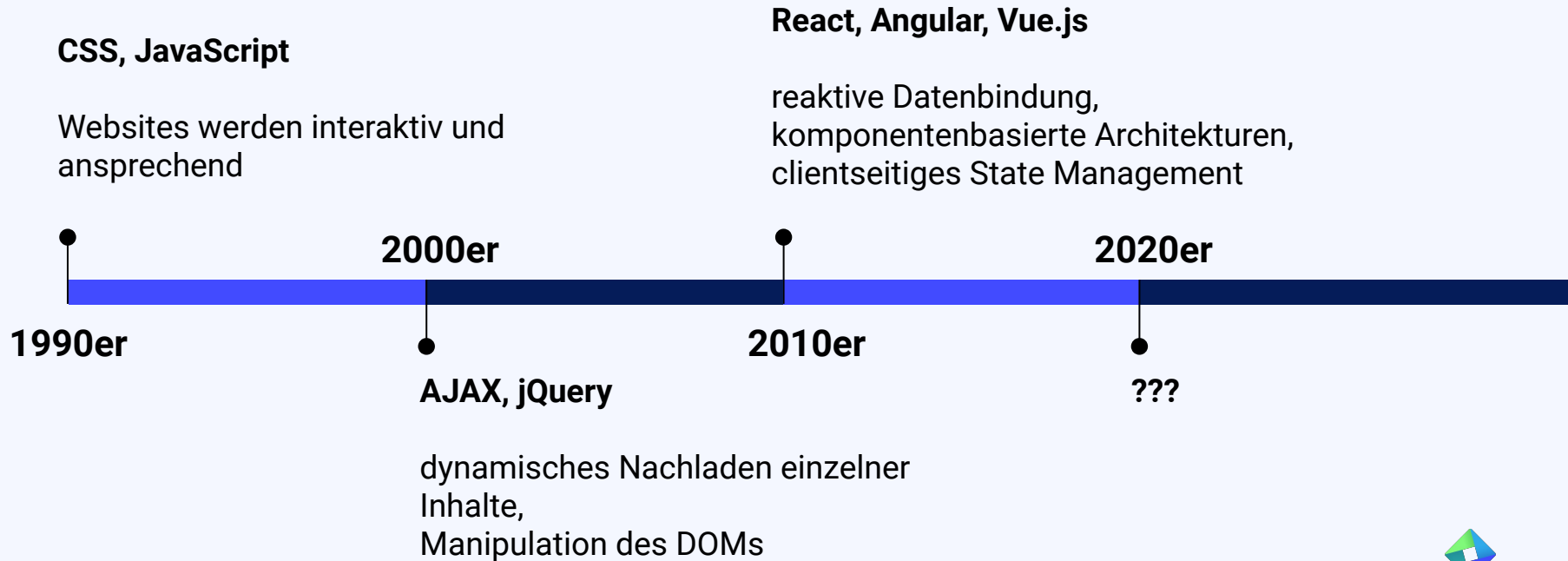
1. htmx - was ist das überhaupt?
2. Deep Dive: Coding mit htmx
3. Evaluation





Rückblick

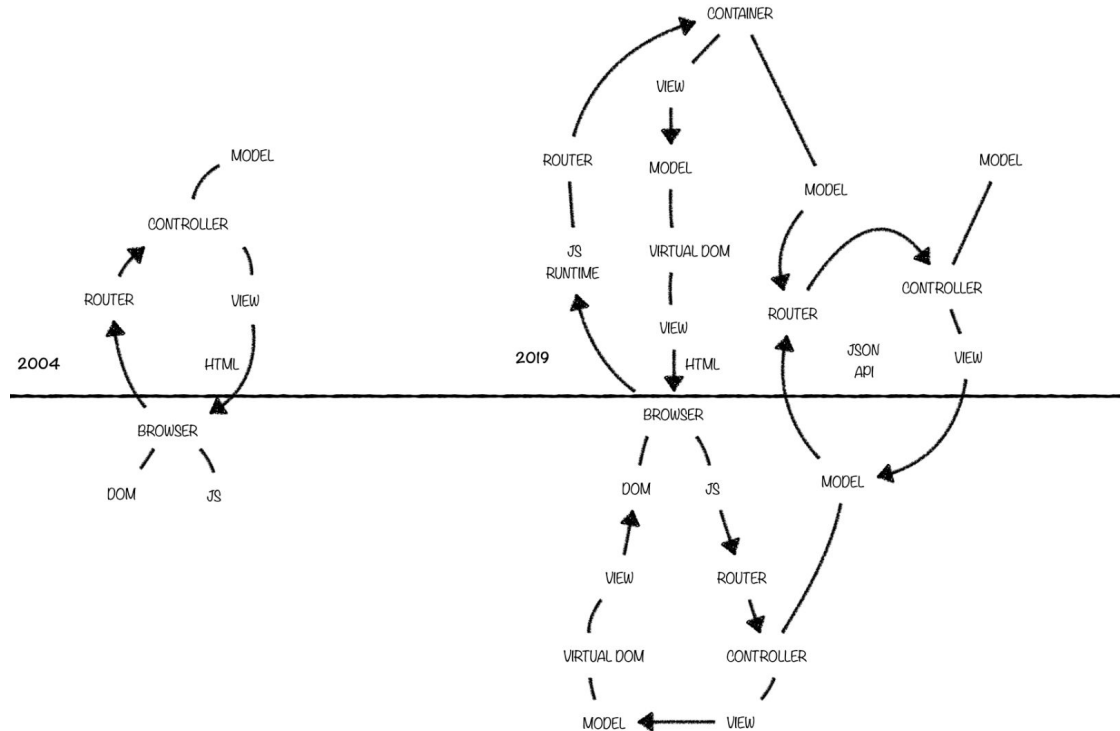
Webentwicklung hat sich in den letzten 30 Jahren grundlegend verändert



Es gibt zahlreiche, nichtfunktionale Anforderungen an eine moderne Webanwendung

- 
- Reibungslose Interaktion
 - Komponentenbasierte Architektur
 - Partielles Nachladen
 - Eine Codebase
 - Effizientes Statemanagement
 - Dokumentation & Community
 - Developer Experience
 - Skalierbarkeit
 - A11y, Security, ...

Die Komplexität von Frontend-Anwendungen hat zugenommen



Ist diese Komplexität notwendig?



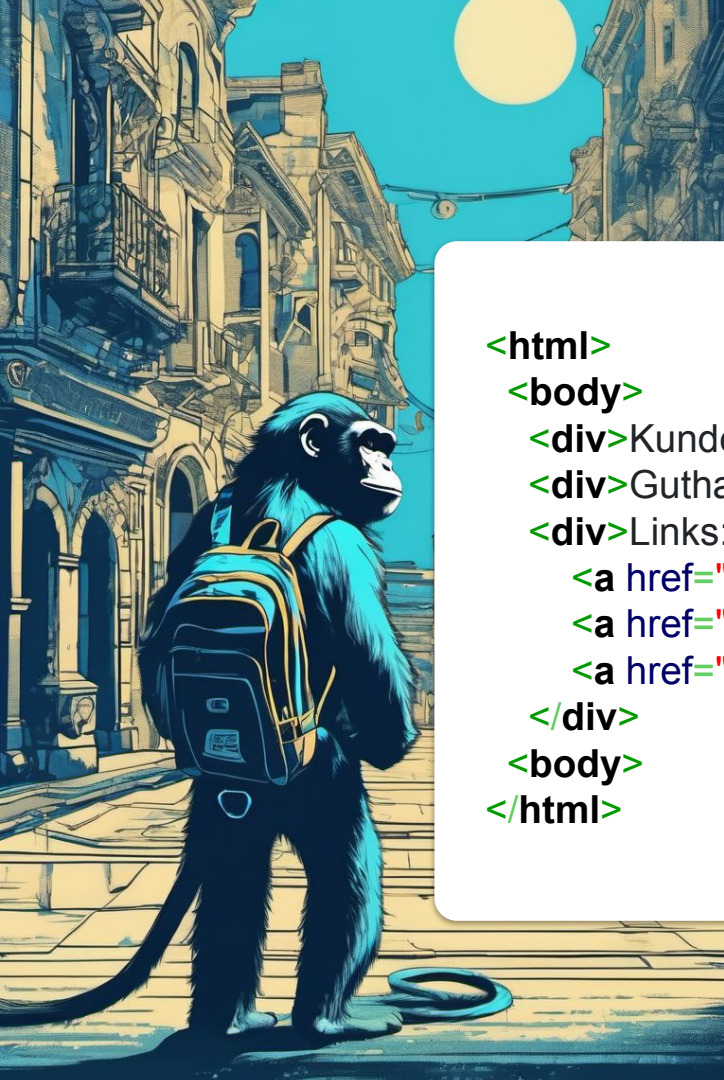


Exkurs: Hypermedia

- Verknüpfung von Medien durch Links
- Hypermedia ist Teil der REST-Leitlinien

HATEOAS:

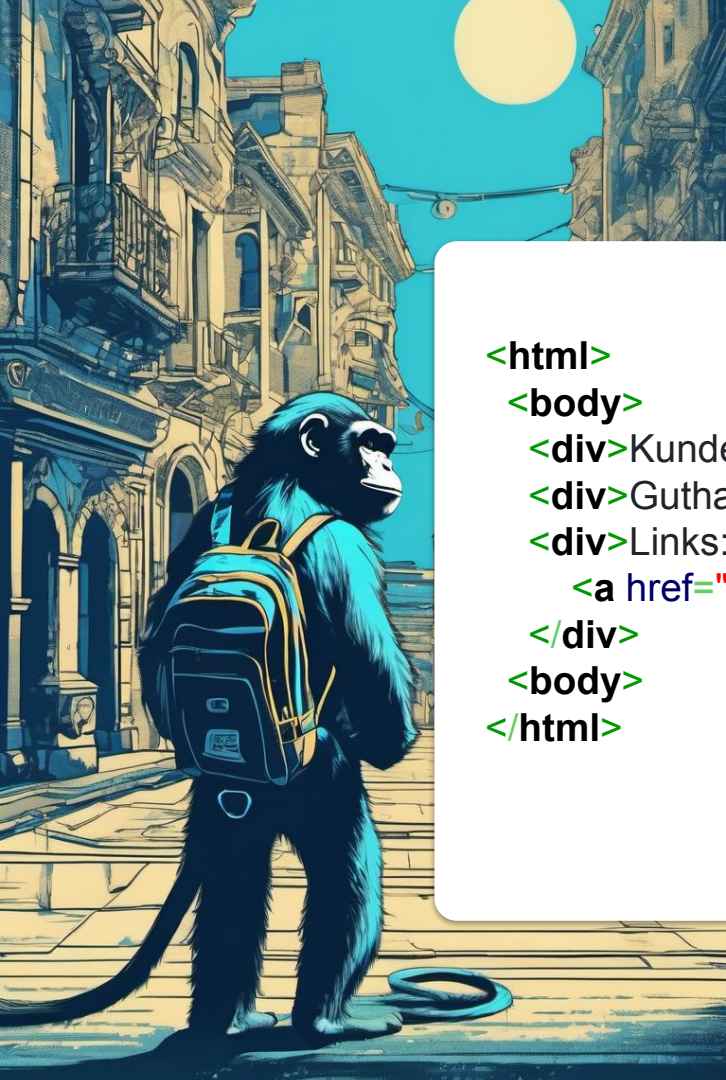
Hypermedia as the Engine of Application State



Exkurs: Hypermedia

```
<html>
<body>
  <div>Kundennummer: 12345</div>
  <div>Guthaben: 100.00 EUR</div>
  <div>Links:
    <a href="/accounts/12345/deposits">Einzahlen</a>
    <a href="/accounts/12345/withdrawals">Auszahlen</a>
    <a href="/accounts/12345/transfers">Überweisen</a>
  </div>
</body>
</html>
```


Exkurs: Hypermedia



```
<html>
<body>
  <div>Kundennummer: 12345</div>
  <div>Guthaben: -50.00 EUR</div>
  <div>Links:
    <a href="/accounts/12345/deposits">Einzahlen</a>
  </div>
</body>
</html>
```

Exkurs: Hypermedia

```
{  
  "account": {  
    "account_number": 12345,  
    "balance": {  
      "currency": "eur",  
      "value": -50.00  
    },  
    "status": "overdrawn"  
  }  
}
```

HTML enthält bereits einige Hypermedia-Funktionen

click und submit
events

<form> , <a>

POST bzw. GET
request

reload page

```
<html>
  <body>
    <div>Kundennummer: 12345</div>
    <div>Guthaben: -50.00 EUR</div>
    <div>Links:
      <a href="/accounts/12345/deposits">Einzahlen</a>
    </div>
  </body>
</html>
```

Durch die Beseitigung dieser willkürlichen Beschränkungen vervollständigt htmx HTML als Hypertext

click und submit
events

jedes Event

<form> , <a>

alle Arten von Tags

POST bzw. GET
request

alle Request-Typen

reload page

Teilweiser Reload

The logo for htmx, featuring a large blue closing tag symbol '</>' followed by the text 'htmx' in a bold, sans-serif font. The 'h' and 't' are black, while the 'm' and 'x' are blue.

htmx ist eine Bibliothek zur dynamischen Aktualisierung von Webseiten ohne JavaScript

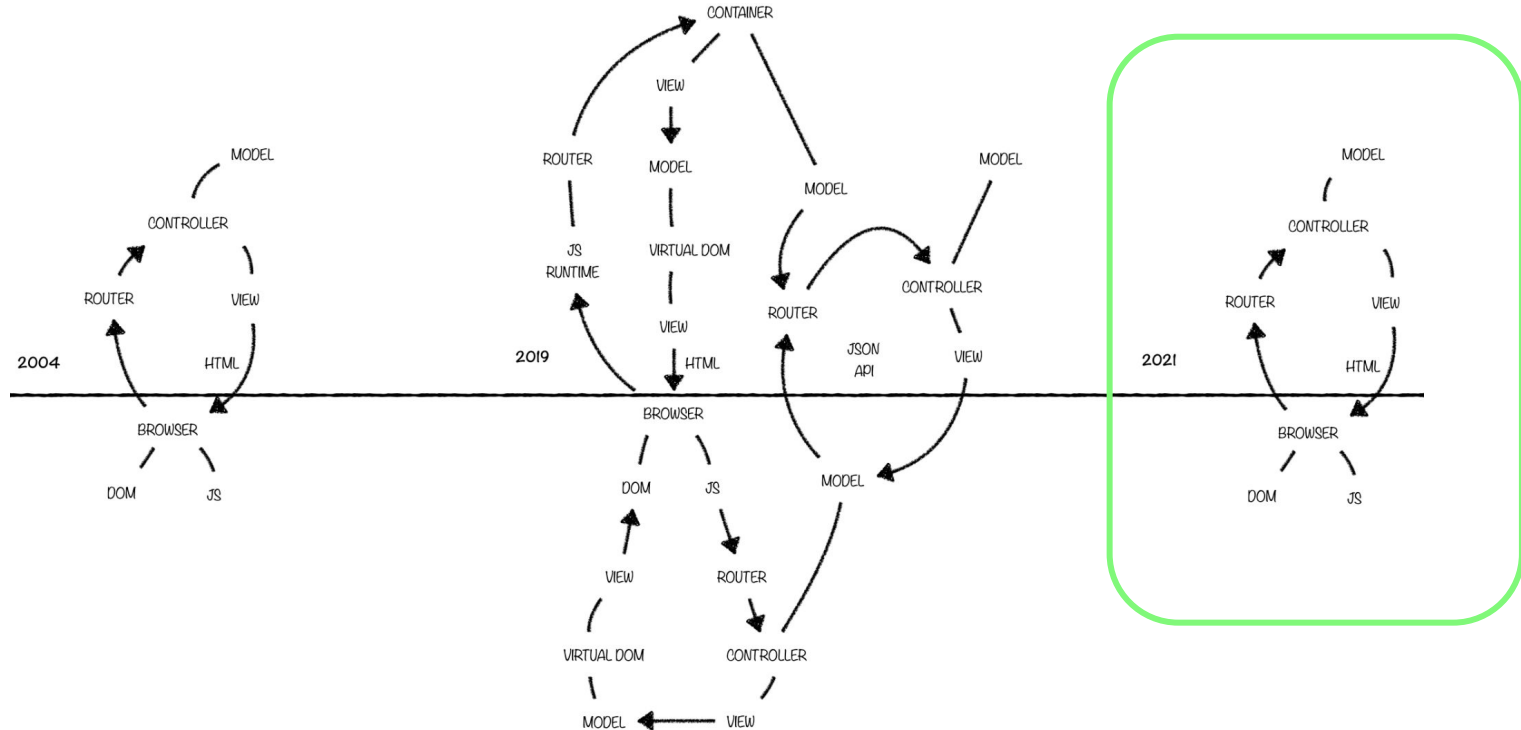
introduction

htmx allows you to access [AJAX](#), [CSS Transitions](#), [WebSockets](#) and [Server Sent Events](#) directly in HTML, using [attributes](#), so you can build [modern user interfaces](#) with the [simplicity](#) and [power](#) of hypertext

htmx is small ([~14k min.gz'd](#)), [dependency-free](#), [extendable](#) & IE11 compatible

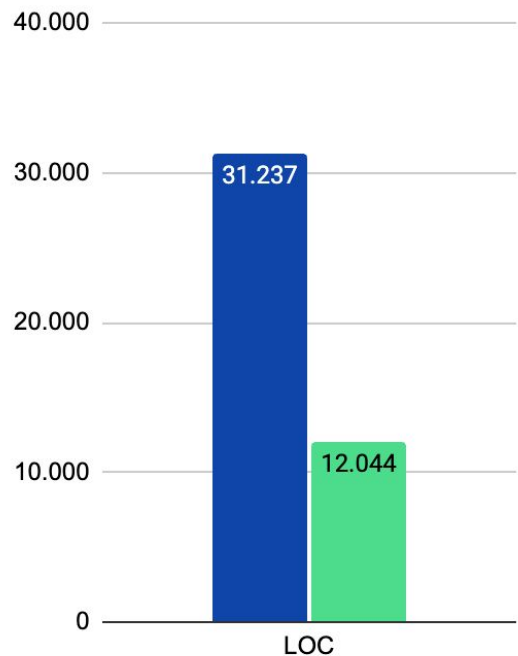


Frontend-Applikationen könnten weniger komplex sein

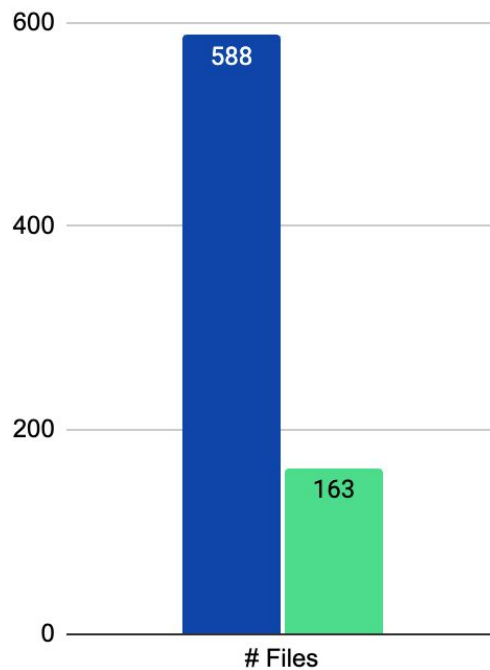


Reduktion von Komplexität ist spürbar

Lines of Code



Anzahl an Files



React

htmx





Wie funktioniert das?



Tech Stack



`</> htmx`

Items

Item 1

Delete

Item 2

Delete

Item 3

Delete

New Item

Add Item

Wie ist die User Experience?



inovex Turniere

SPIELER:IN 1	PUNKTE	SPIELER:IN 2	PUNKTE
Julia	10	Duc	8
SPEICHERN			
Spieler:in hinzufügen			

HIGHSCORES



[Gesamte Bestenliste anschauen](#)

Spielergebnisse			
SPIELER:IN 1	SPIELER:IN 2	ERGEBNIS	
Manuel	Manuel	0 : 10	Löschen
Julia	Johanna	1 : 8	Löschen
Duc	Julia	2 : 1	Löschen
Düc	Duc	6 : 2	Löschen
Dominik	Jonas	2 : 4	Löschen

Seite 1 von 22

[Weiter](#)

htmx kann die meisten nichtfunktionalen Anforderungen an eine moderne Webanwendung bedienen

- 
- Reibungslose Interaktion ✓
 - Komponentenbasierte Architektur ✓
 - Partielles Nachladen ✓
 - Eine Codebase ✓
 - Effizientes Statemanagement ⚠
 - Dokumentation & Community ✓
 - Developer Experience ⚠

- 
- Überblick URLs vs. Endpunkte
 - Codeduplikation für verschiedene States
 - Auf Komponentenstate reagieren
 - Out of the box Funktionalität von Typescript

Fazit

- User Experience wie bei SPA (sofern Internet)
- Leidenschaftsprojekt von Carson Gross, mit viel Humor und Intellekt
- insb. zu empfehlen für Backendler, die keinen Node Server integrieren wollen
- für kleine CRUD Applikationen





“No One Ever Got Fired For
Using React”

Jake Lazaroff

https://x.com/htmx_org?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor