



---

THOMAS SCHUEHLY

# RAPID SERVER-SIDE FULL STACK WEB DEVELOPMENT WITH VIEWCOMPONENTS AND HTMX

18. May 2023

# Thomas Schühly

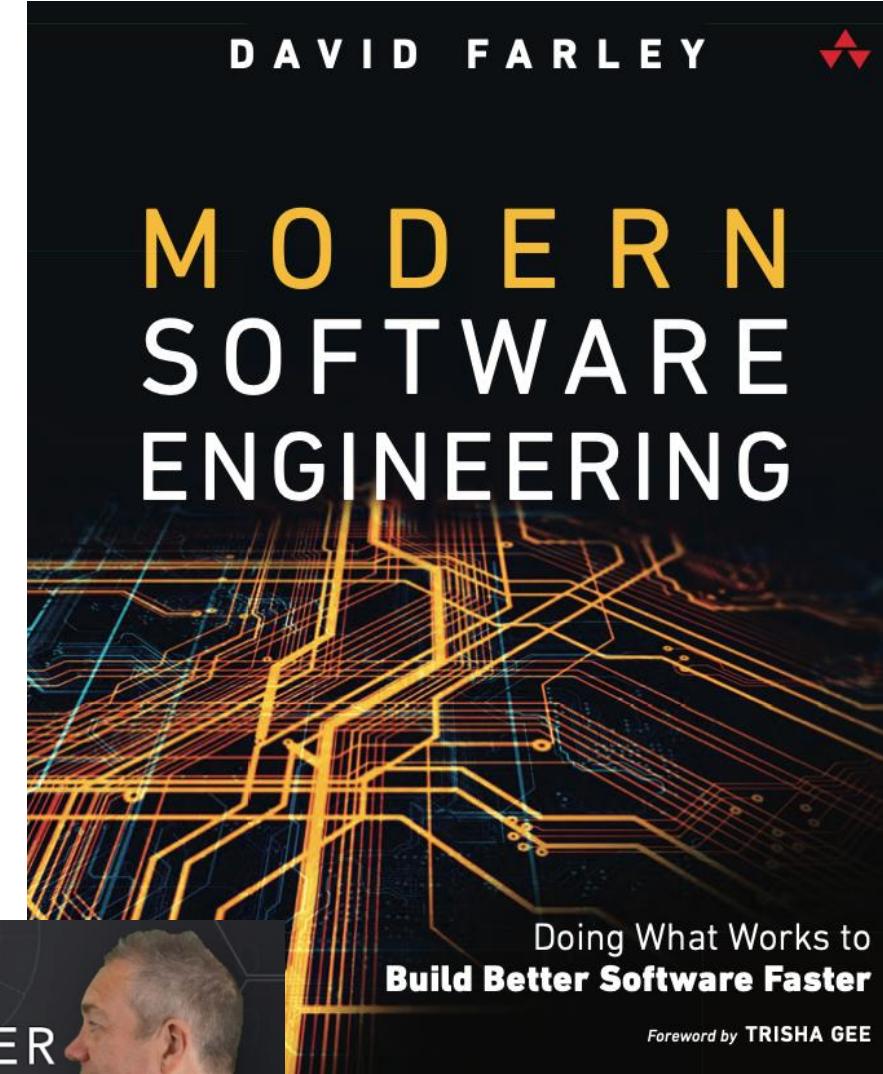
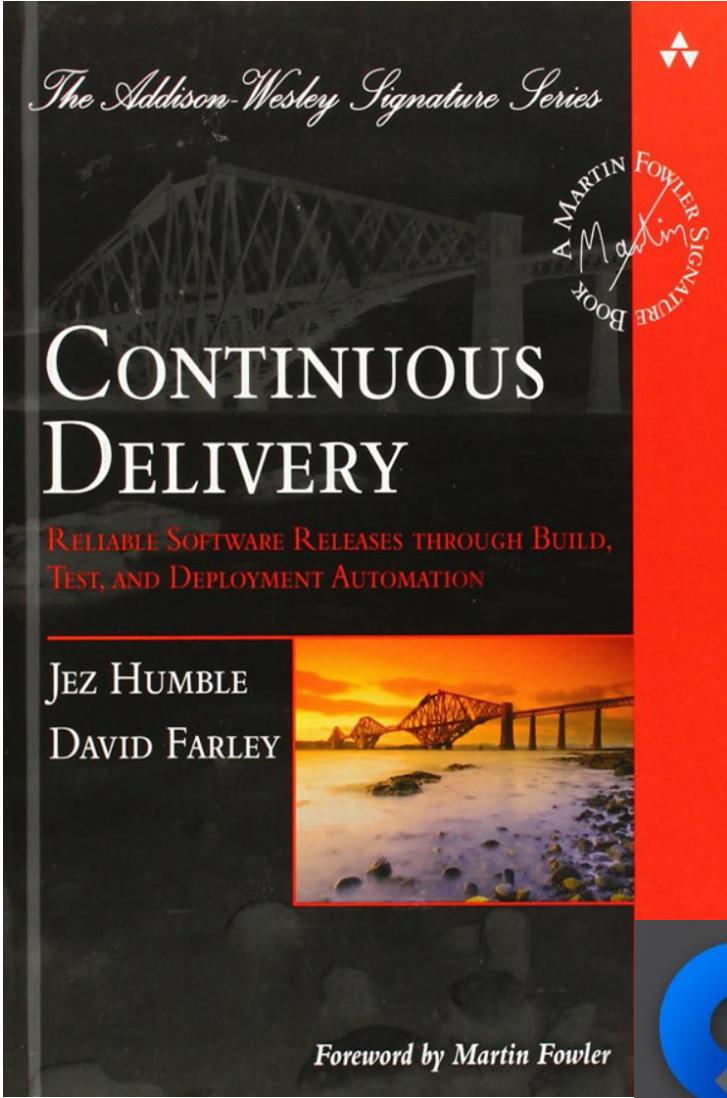


- Spring Boot + Kotlin = ❤️
- Blogging on [tschuehly.de](https://tschuehly.de)
- Process Automation Software Engineer @mimacom Germany in Stuttgart
- Freelance Web Developer
- Open-Source Developer:
  - [github.com/tschuehly/spring-view-component](https://github.com/tschuehly/spring-view-component)
  - [github.com/tschuehly/supabase-security-spring-boot-starter](https://github.com/tschuehly/supabase-security-spring-boot-starter)

Would you describe yourself as a Software Engineer?

# **Software** Engineering

# Software Engineering



<https://www.davefarley.net/>

[https://www.youtube.com/  
@ContinuousDelivery](https://www.youtube.com/@ContinuousDelivery)



# Engineering

„Stifling Bureaucracy and  
heavy weight process  
control“

„Simply writing code  
and nothing else“

“In other disciplines, Engineering is simply the stuff that works. It is practical, pragmatic and more, not less, efficient.”

Dave Farley

*What is Modern Software Engineering?*

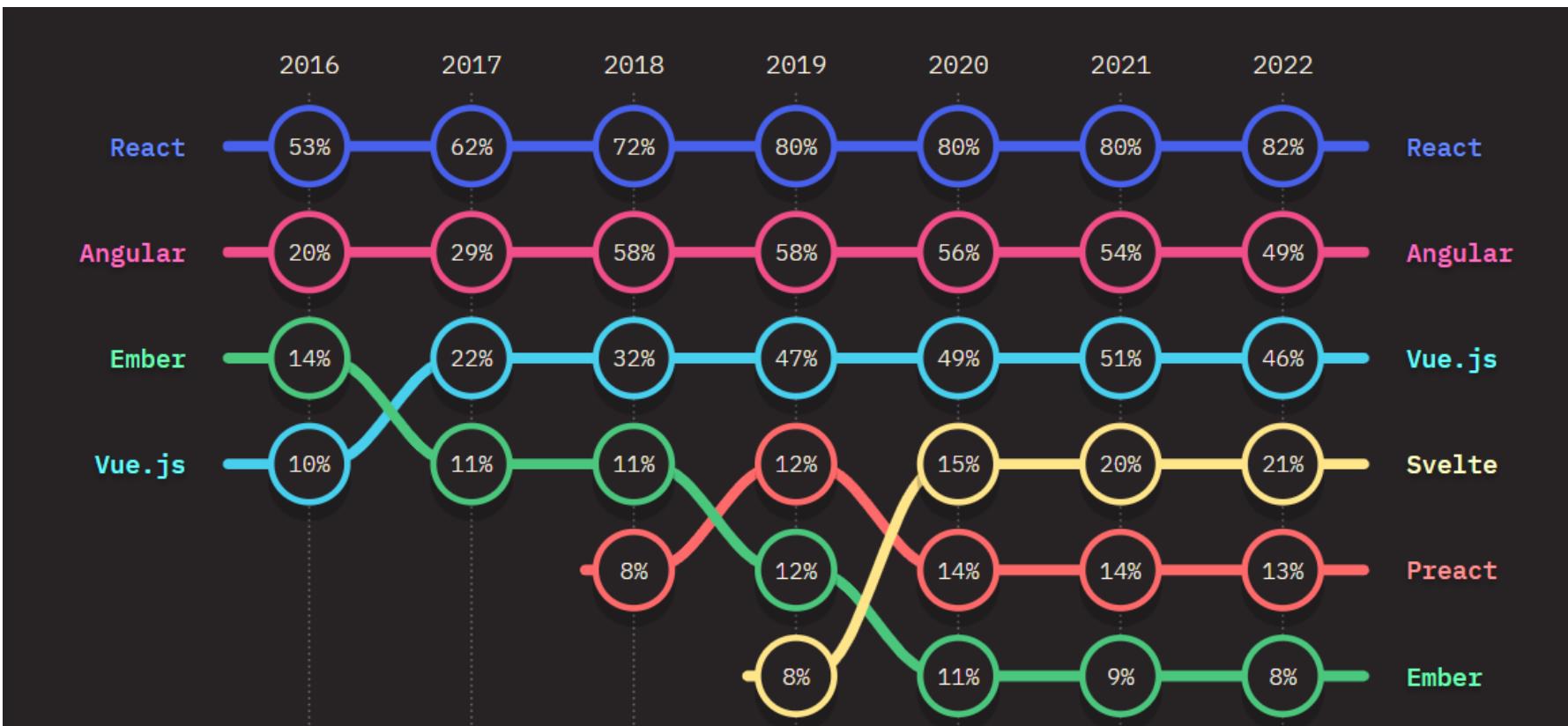
“The systems that modern engineers create are increasingly complex and sophisticated, [ .. ] modern software engineering, needs to focus us on managing that complexity”

Dave Farley

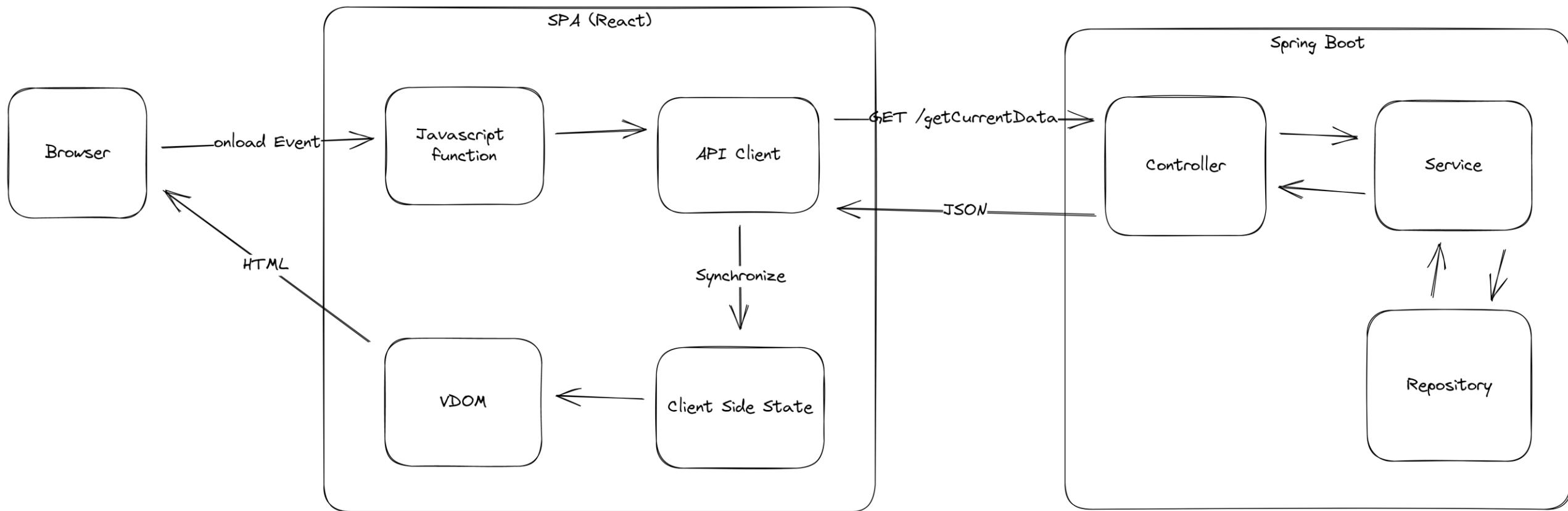
*What is Modern Software Engineering?*



# FRONT-END FRAMEWORKS USAGE OVER TIME



## SPA



After years of relative stability, many are now beginning to question the status quo. [...] on the server Astro, Remix and Next.js (among others) are making us reconsider how much code we really need to ship to the client."

*State of JS 2022*

<https://2022.stateofjs.com/>

**Do we need the complexity of  
“modern” JavaScript frameworks  
for building on the web?**



# Tom MacWright

---

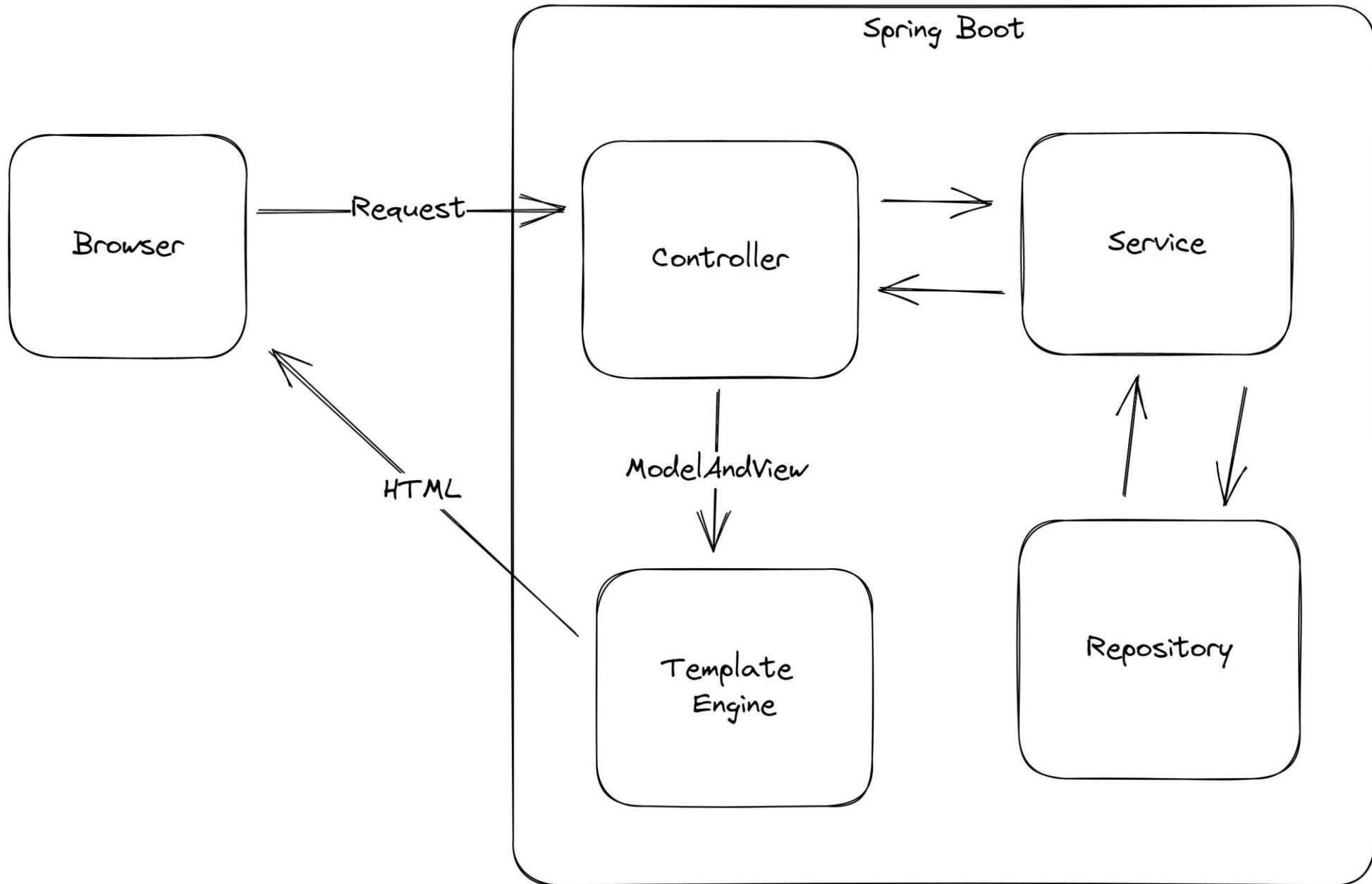
Second-guessing the  
modern web

<https://macwright.com/2020/05/10/spa-fatigue.html>

“There is a sweet spot of React:  
in moderately interactive interfaces.  
Complex forms that require immediate  
feedback,  
UIs that need to move around and react  
instantly.”

- Single Page Applications are an architectural style
- Tradeoffs analysis are skipped
- Many Websites will benefit from the simplicity of server-side logic

# Multi Page Application



**DB TO  
JSON TO  
JS TO HTML**

**DB  
TO HTML**

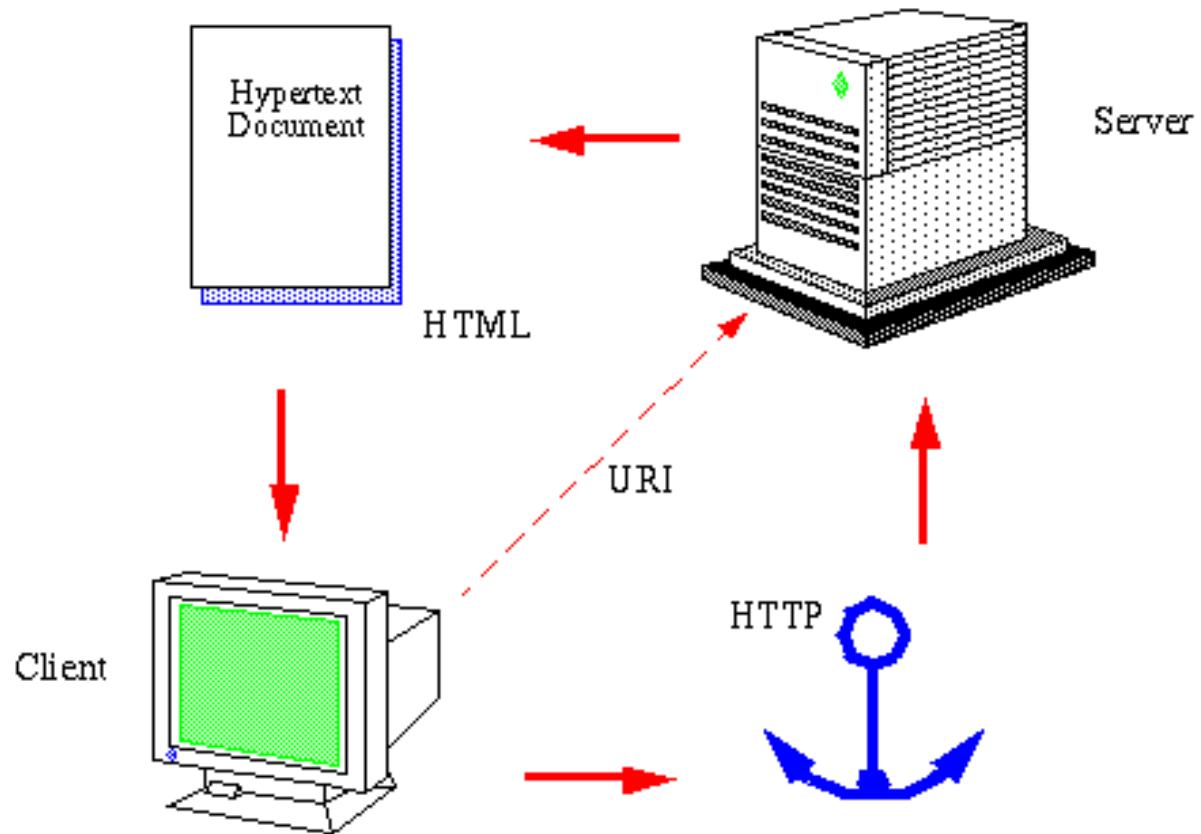


# The World Wide Web

Henrik Frystyk

July 1994

[https://www.w3.org/People/  
Frystyk/thesis/WWW.html](https://www.w3.org/People/Frystyk/thesis/WWW.html)



“Hypermedia is defined by the presence of application control information embedded within, or as a layer above, the presentation of information.”

Roy Fielding  
Architectural Styles and  
the Design of Network-based Software Architectures

# HATEOAS

The forgotten part of REST



"I'M GOING TO  
CREATE A  
RESTFUL API..."

"USING JSON..."

"HERE ARE MY  
API DOCS"

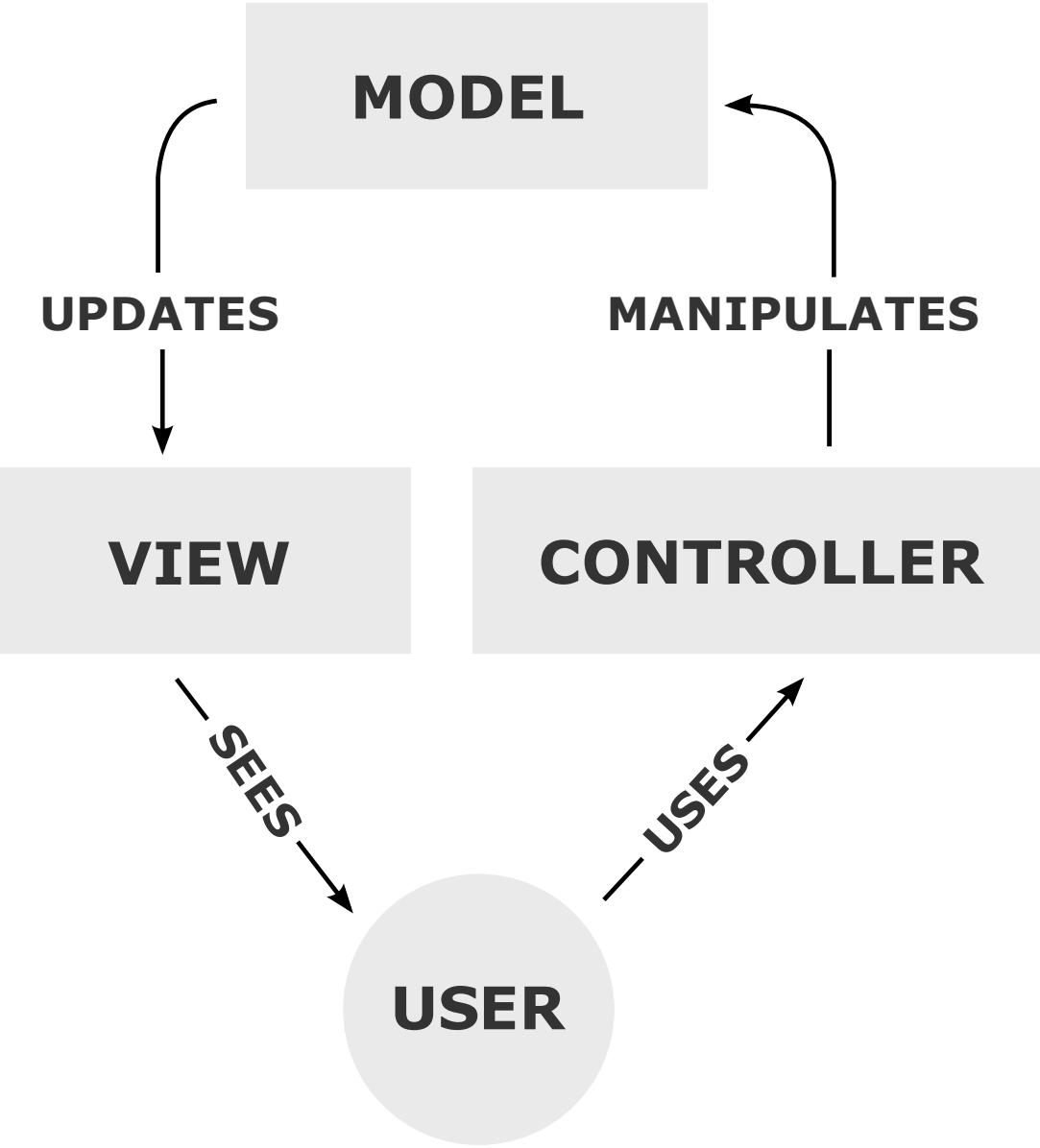


# Creating your own Client

Just use  
the Browser  
as Client

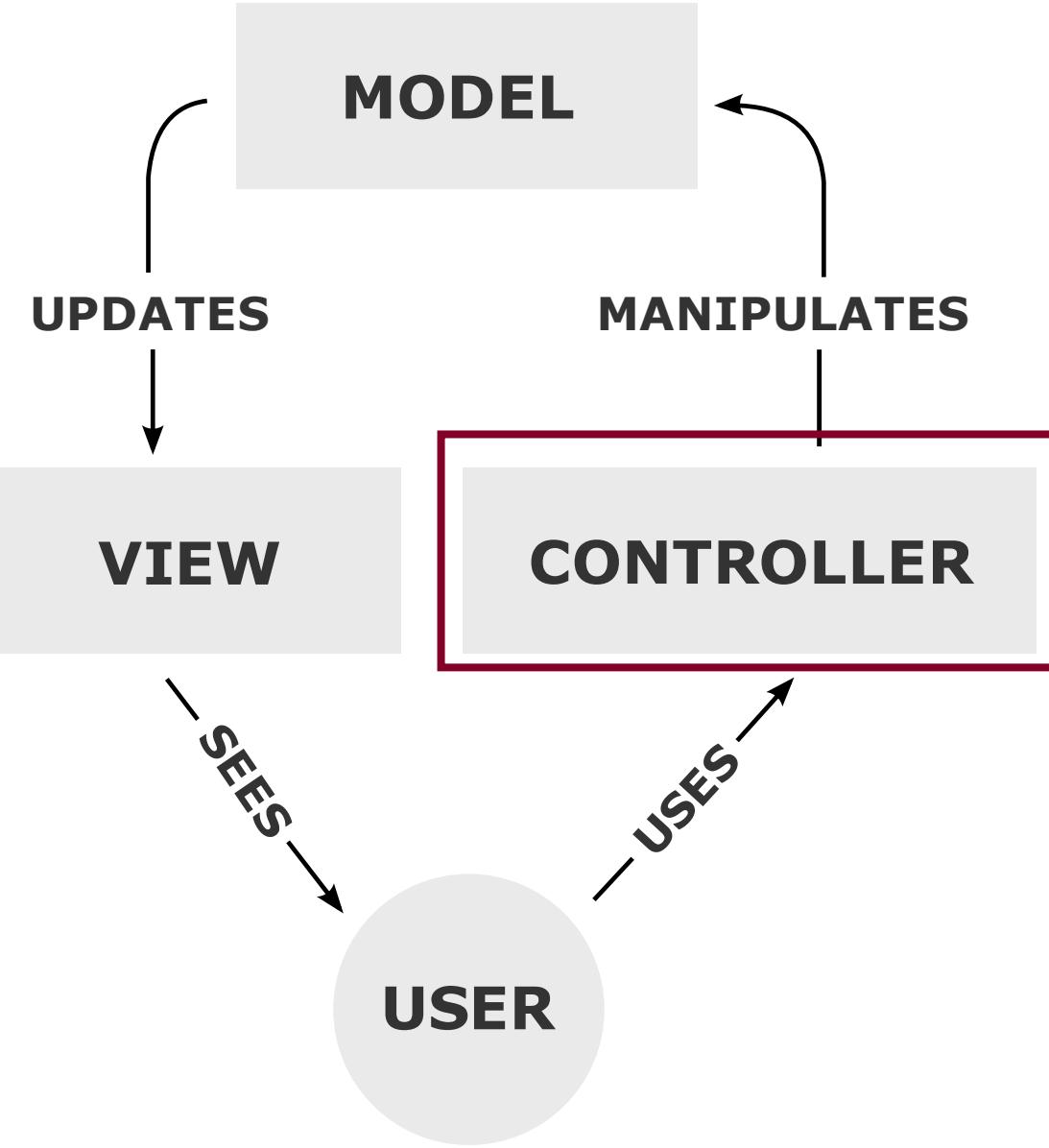


```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```



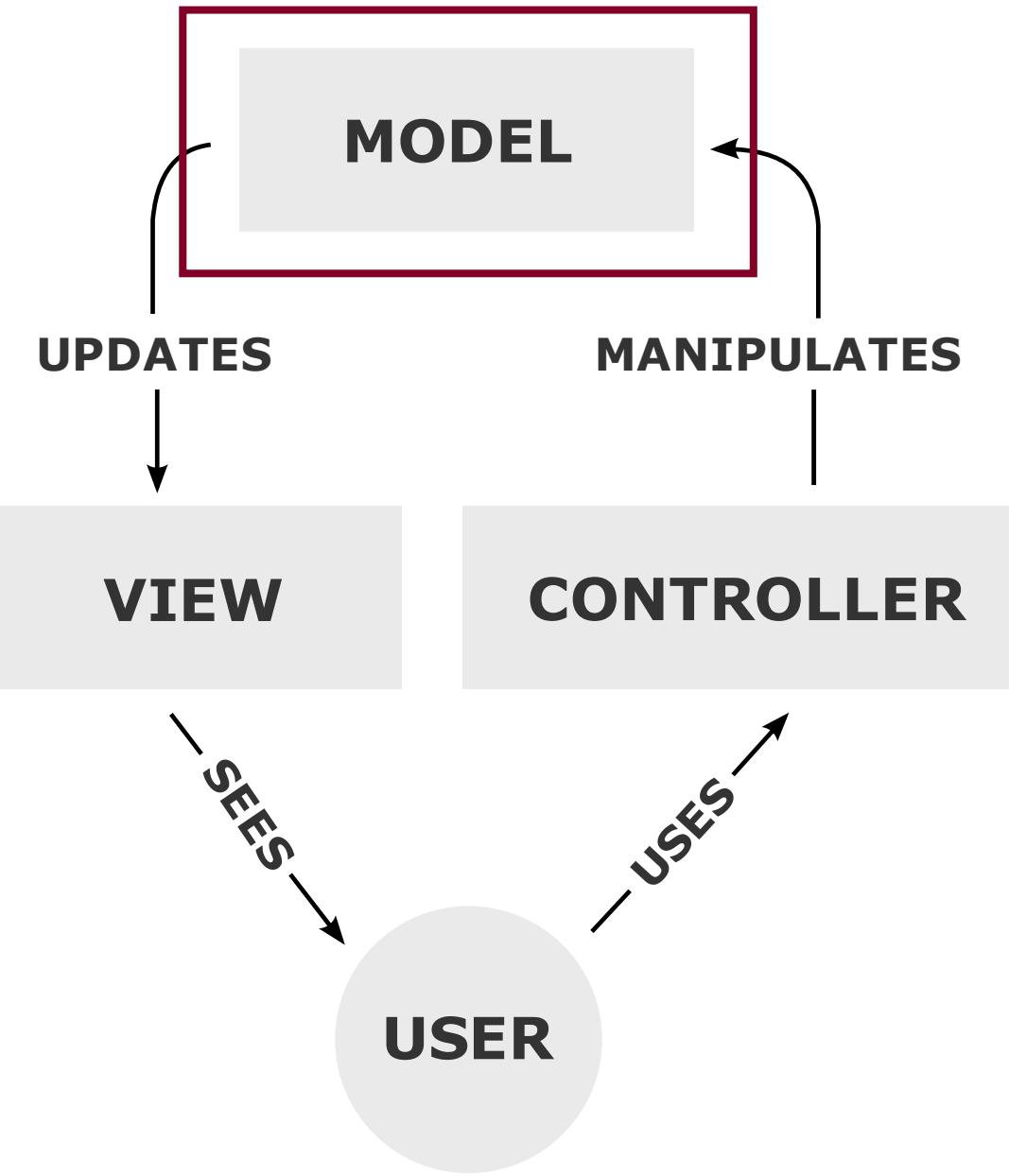


```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```



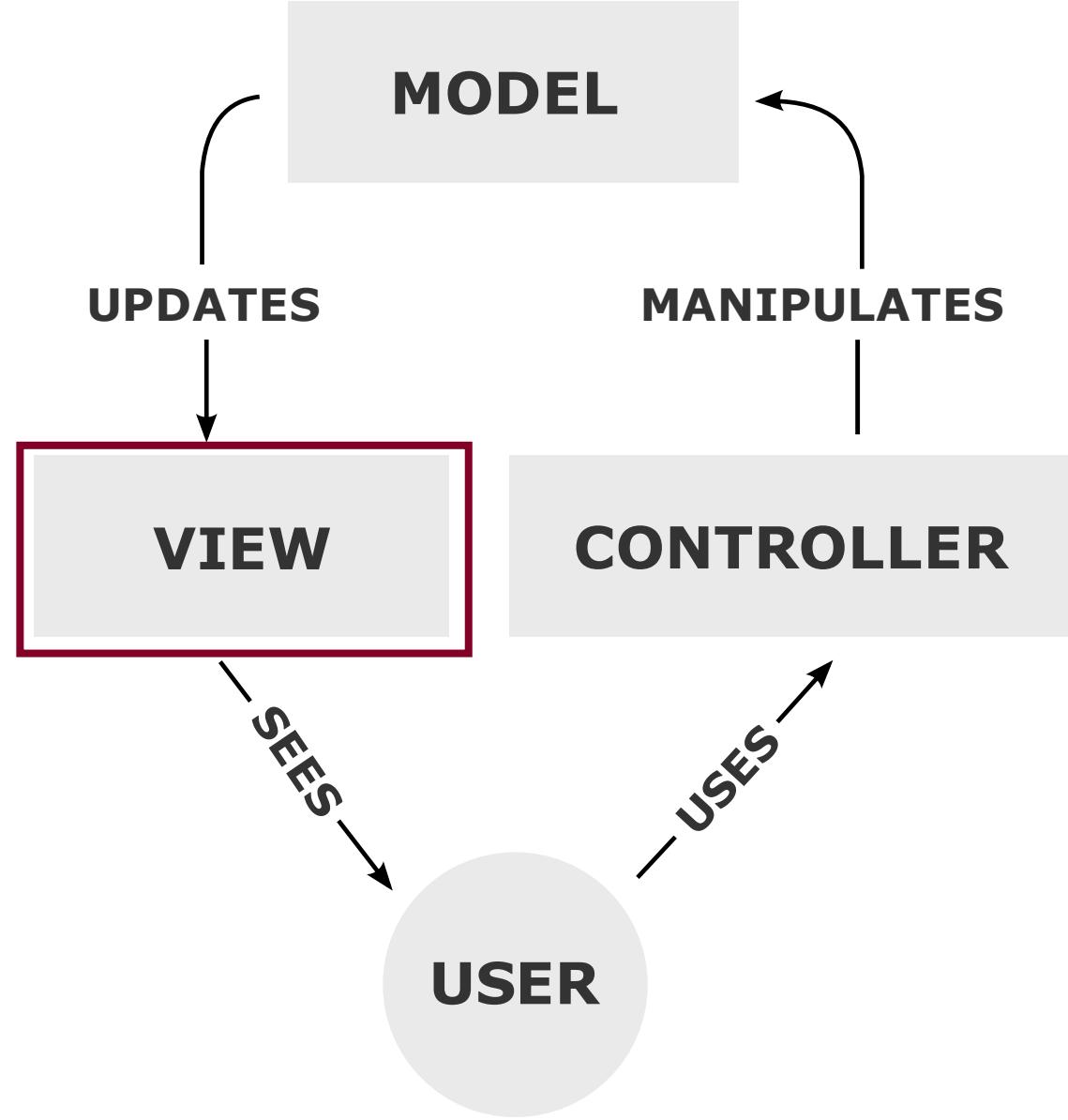


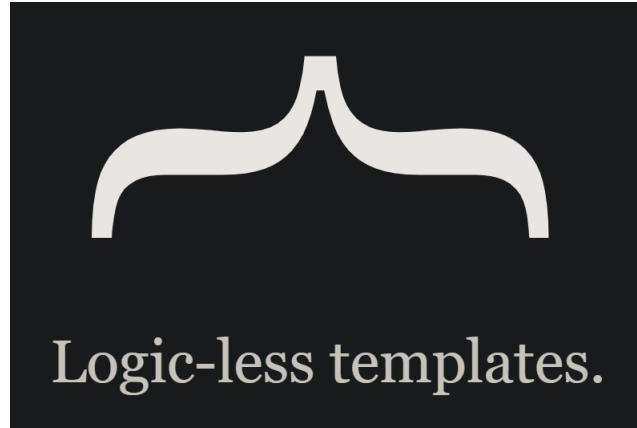
```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9     return "greeting"
10 }
11 }
```





```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9     return "greeting"
10 }
11 }
```







*Thymeleaf*



## greeting.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Getting Started: Serving Web Content</title>
</head>
<body>
    <p th:text="'Hello, ' + ${name} + '!' ">
</body>
</html>
```

● ● ● greeting.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Getting Started: Serving Web Content</title>
</head>
<body>
    <p th:text="'Hello, ' + ${name} + '!' ">
</body>
</html>
```

● ● ● greeting.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Getting Started: Serving Web Content</title>
</head>
<body>
    <p th:text="'Hello, ' + ${name} + '!''" />
</body>
</html>
```

● ● ● products.html

```
<h1>Product list</h1>
<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
  </tr>
  <tr th:each="prod : ${products}">
    <td th:text="${prod.name}">Onions</td>
    <td th:text="${prod.price}">2.41</td>
    <td th:text="${prod.inStock}? ${inStock} : ${notInStock}">yes</td>
  </tr>
</table>
```

● ● ● products.html

```
<h1>Product list</h1>
<table>
  <tr>
    <th>NAME</th>
    <th>PRICE</th>
    <th>IN STOCK</th>
  </tr>
  <tr th:each="prod : ${products}">
    <td th:text="${prod.name}">Onions</td>
    <td th:text="${prod.price}">2.41</td>
    <td th:text="${prod.inStock}? ${inStock} : ${notInStock}">yes</td>
  </tr>
</table>
```



● ● ● page.jte

```
@import org.example.Page
@param Page page

<head>
    <meta name="description"
          content="${page.getDescription( )}">
    <title>${page.getTitle( )}</title>
</head>
```



page.jte

```
@import org.example.Page  
@param Page page
```

```
<head>  
    <meta name="description"  
          content="${page.getDescription( )}">  
    <title>${page.getTitle( )}</title>  
</head>
```



## page.jte

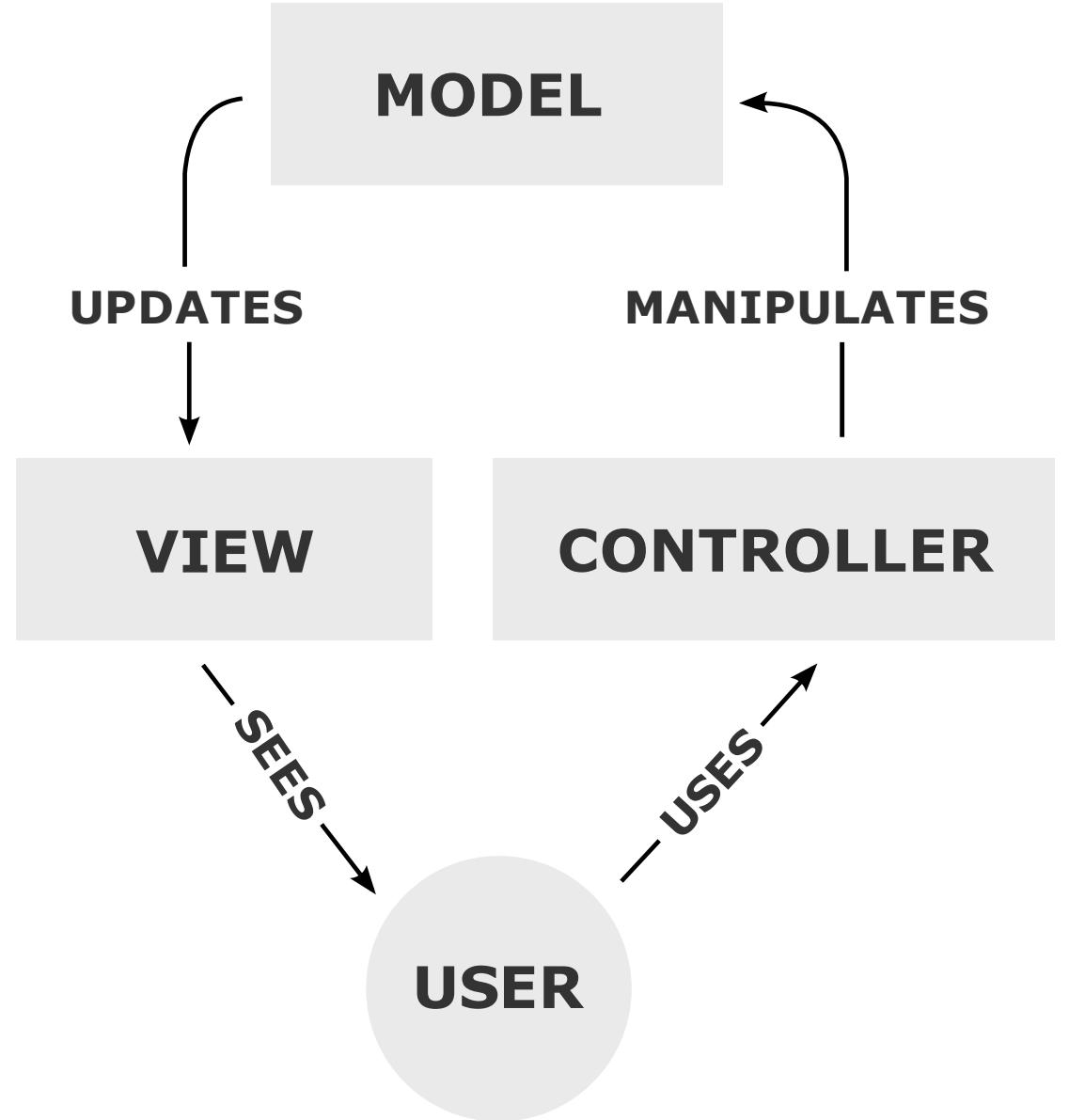
```
@import org.example.Page
@param Page page

<head>
    @if(page.getDescription( ) != null)
        <meta name="description" content="${page.getDescription( )}">
    @endif
    <title>${page.getTitle( )}</title>
</head>
<body>
    <h1>${page.getTitle( )}</h1>
    <p>Welcome to my example page!</p>
</body>
```



## personlist.jte

```
1 @import de.tschuehly.jteviewcomponentdemo.core.Person
2 @import java.util.List
3 @param List<Person> personList
4
5 <table>
6     <thead>
7         <tr>
8             <td>Id</td>
9             <td>Name</td>
10            <td>LastName</td>
11        </tr>
12    </thead>
13    <tbody>
14        @for(Person person : personList)
15            <tr>
16                <th>${person.getId( )}</th>
17                <th>${person.getName( )}</th>
18                <th>${person.getLastname( )}</th>
19            </tr>
20        @endfor
21    </tbody>
22 </table>
```







# spring® ViewComponent

[github.com/tschuehly/spring-view-component](https://github.com/tschuehly/spring-view-component)

# Template and Controller are tightly coupled

## GreetingController.kt

```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```

## greeting.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Getting Started: Serving Web Content</title>
</head>
<body>
    th:text="Hello, ' + ${name} + '!" />
</body>
</html>
```



```
1 // HomeViewComponent.kt
2 @ViewComponent
3 class HomeViewComponent(
4     private val exampleService: ExampleService
5 ) {
6     fun render() = JteViewContext(
7         "helloWorld" toProperty exampleService.getCoffee(),
8     )
9 }
```



```
1 // HomeViewComponent.jte
2 @param String helloWorld
3 <div>${helloWorld}</div>
```



```
1 // Router.kt
2 @Controller
3 class Router(
4     val homeViewComponent: HomeViewComponent
5 ) {
6     @GetMapping("/")
7     fun homeComponent() = homeViewComponent.render()
8 }
```



localhost:8080

# Watch Coffee + Software with Josh Long!



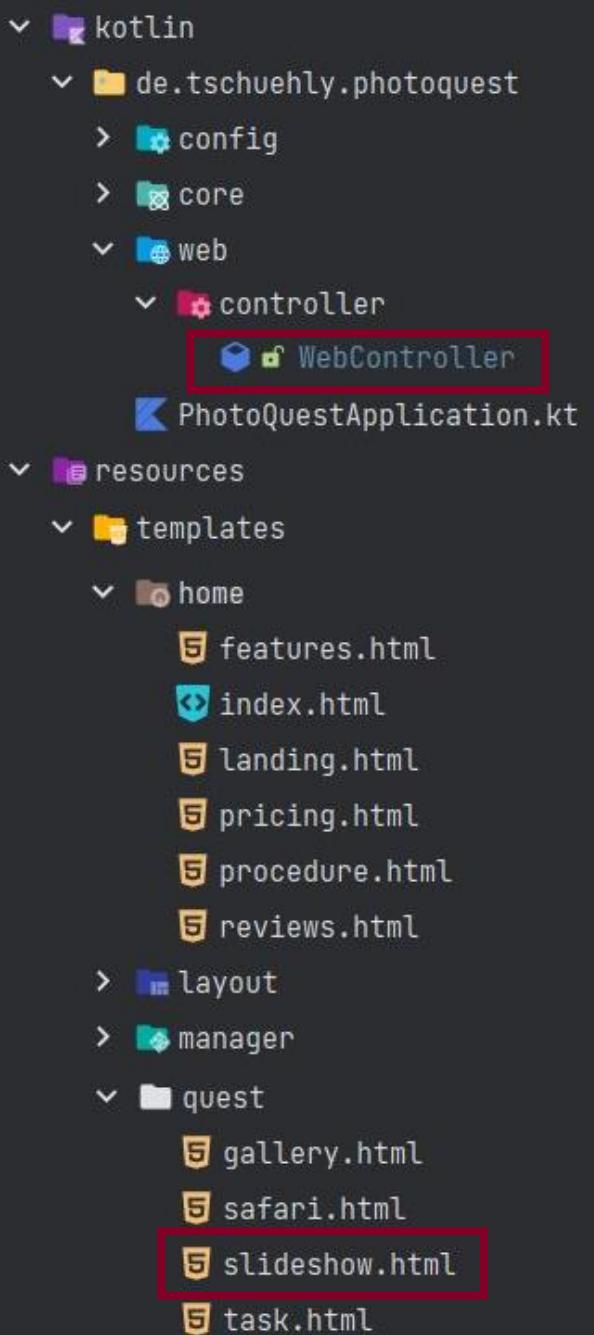
```
1 @ViewComponent
2 class HomeViewComponent(
3     private val parameterViewComponent: ParameterViewComponent
4 ) {
5     fun render() = JteViewContext(
6         "parameterViewComponent" toProperty
7         parameterViewComponent.render("Hello World")
8     )
}
```

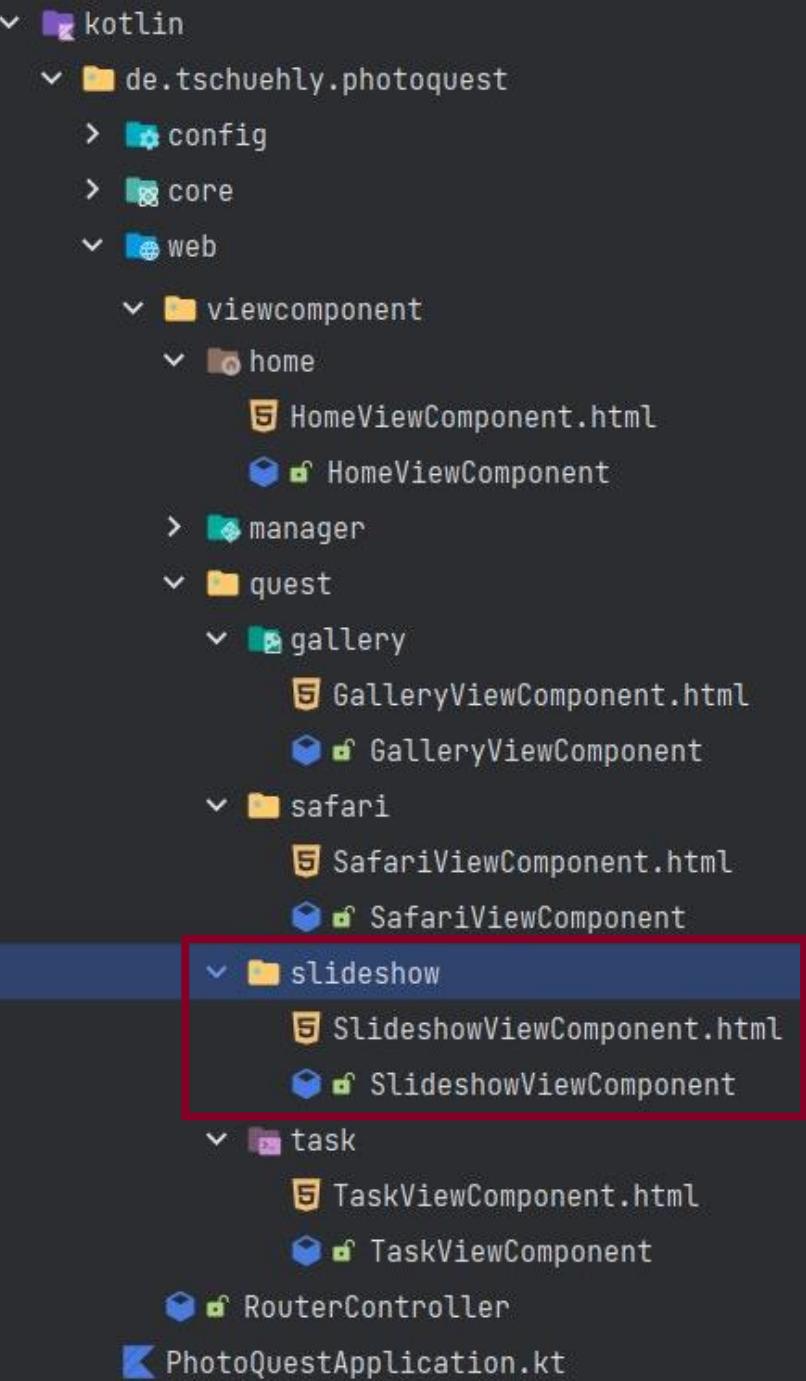
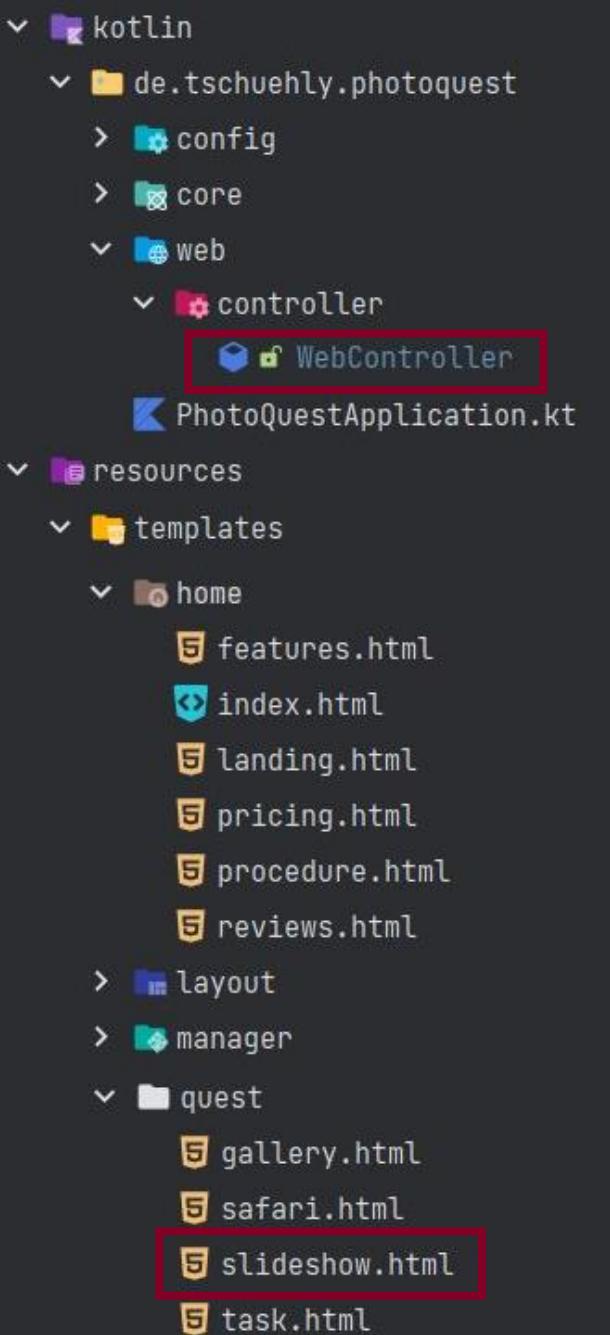


```
1 @import de.tschuehly.spring.viewcomponent.jte.JteViewContext
2 @param JteViewContext parameterViewComponent
3
4 ${parameterViewComponent}
```



```
1 @ViewComponent
2 class ParameterViewComponent {
3     fun render(message: String?) = ViewContext(
4         "message" toProperty message ?: throw Exception("You need to pass in a message"))
5     )
6 }
```





#### 8.2.4. Example Form Submission: Questionnaire Form

Consider the following document:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<title>Sample of HTML Form Submission</title>
<H1>Sample Questionnaire</H1>
<P>Please fill out this questionnaire:
<FORM METHOD="POST" ACTION="http://www.w3.org/sample">
<P>Your name: <INPUT NAME="name" size="48">
<P>Male <INPUT NAME="gender" TYPE=RADIO VALUE="male">
<P>Female <INPUT NAME="gender" TYPE=RADIO VALUE="female">
<P>Number in family: <INPUT NAME="family" TYPE=text>
<P>Cities in which you maintain a residence:
<UL>
<LI>Kent <INPUT NAME="city" TYPE=checkbox VALUE="kent">
<LI>Miami <INPUT NAME="city" TYPE=checkbox VALUE="miami">
<LI>Other <TEXTAREA NAME="other" cols=48 rows=4></textarea>
</UL>
Nickname: <INPUT NAME="nickname" SIZE="42">
<P>Thank you for responding to this questionnaire.
<P><INPUT TYPE=SUBMIT> <INPUT TYPE=RESET>
</FORM>
```

#### Document type

RFC **Historic**

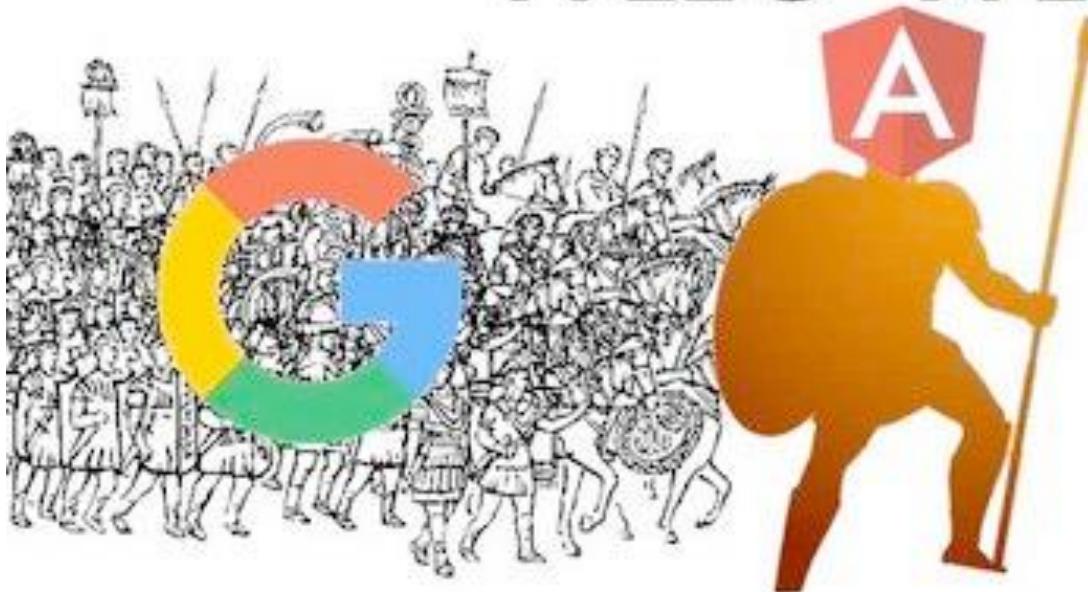
November 1995

[Report errata](#)

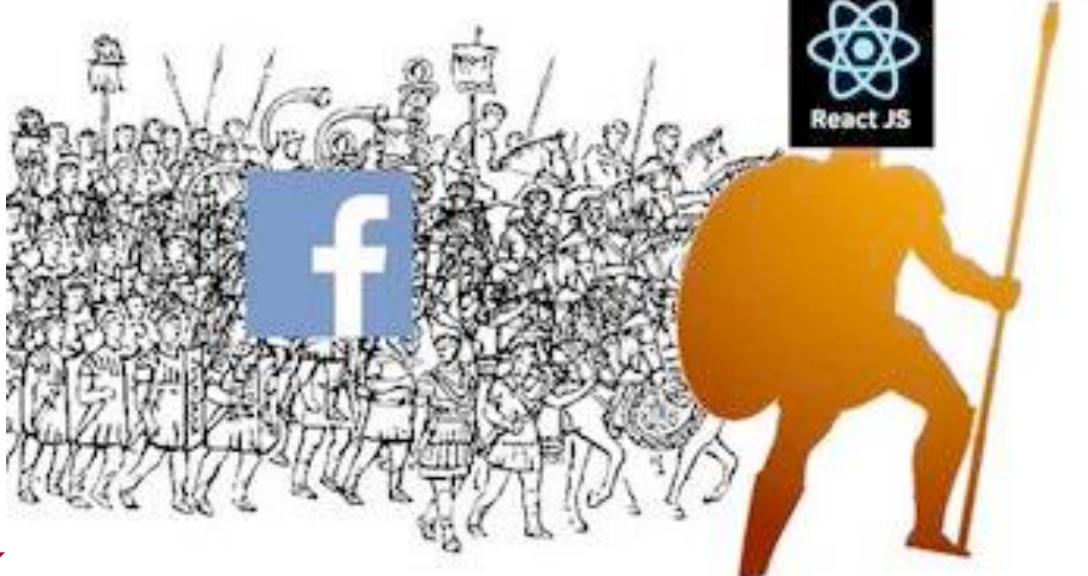
Obsoleted by [RFC 2854](#)

Was [draft-ietf-html-spec](#)

# WHO WILL WIN?



VS.



"u guys should use  
hypermedia"



a lunatic in  
montana and his  
internet friends



# </> htmx

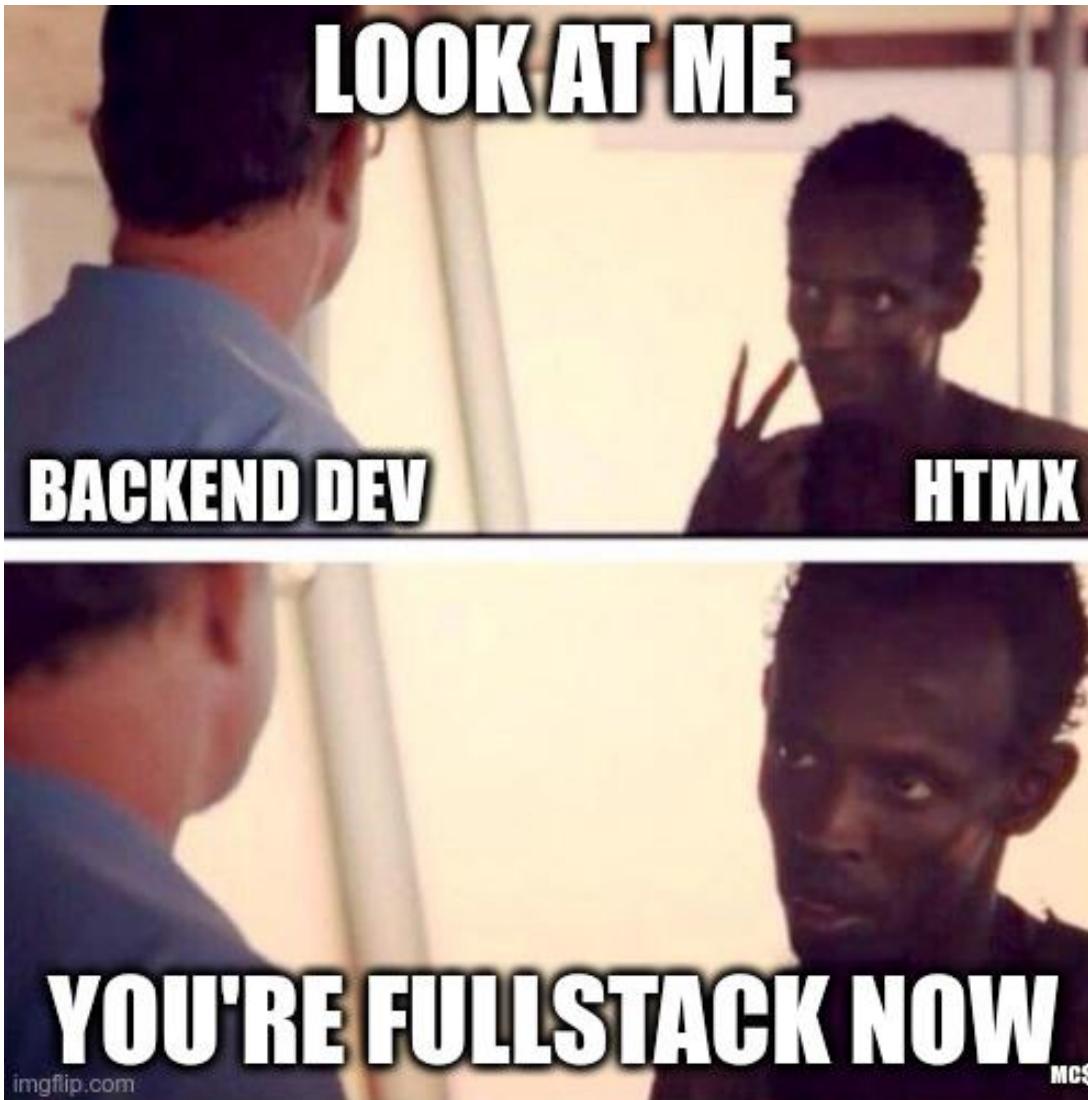
*high power tools for HTML*



```
<a href="/blog">Blog</a>
```



```
<button hx-post="/clicked"
         hx-trigger="click"
         hx-target="#parent-div"
         hx-swap="outerHTML" >
    Click Me!
</button>
```

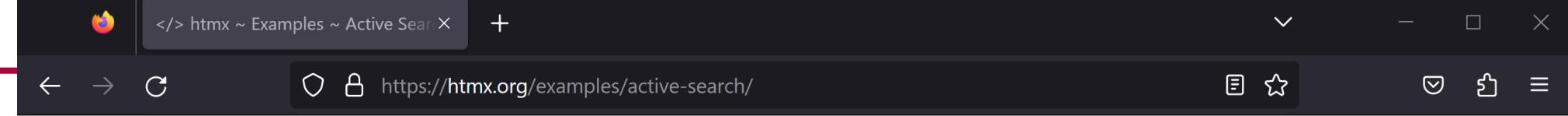


# haiku

*javascript fatigue:  
longing for a hypertext  
already in hand*



```
1
2 <input type="search" name="search"
  placeholder="Begin Typing To Search Users..." 
3   hx-post="/search"
4   hx-trigger="keyup changed delay:500ms,
  search"
5   hx-target="#search-results">
6 <div id="search-results"></div>
```



Server Requests ↓ Hide

**Initial State**

**HTML**

```
<h3>
  Search Contacts
  <span class="htmx-indicator">
     Searching...
  </span>
</h3>

<input class="form-control" type="search"
      name="search" placeholder="Begin Typing To Search Users..."
      hx-post="/search"
      hx-trigger="keyup changed delay:500ms, search"
      hx-target="#search-results"
      hx-indicator=".htmx-indicator">
```



C PPL School of Freedom, Equal Op... X +

Discussion in public session ( 93 amendments filed )

Find a keyword

Authors Articles · 11 Spells

★ Favorites 0

Text order | Latest filed

27 amendments correspond to your criteria - Reset

ART. ADD. BEFORE ARTICLE 1

n ° 82 IS Marco's monique + 11 signatories Withdrawn  
**Av. art. 1st** The school of the Third Republic was born in the context of a military defeat, in order to streng...

ARTICLE 1

n ° 83 IS Marco's monique + 11 signatories 2 identical amendments ●● Rejected  
**Art. 1st** DELETION Article 1 of the bill aims to experiment for a period of 5 years, the possibility of establi...

ART. ADD. AFTER ARTICLE 1

n ° 71 CRCE Pierre Ouzoulias + 2 signatories Adopted  
**Ap. art. 1st** Article L 912-1-1 of the Education Code provides in its first paragraph: « The teaching freedom ...

ART. ADD. AFTER ARTICLE 2

n ° 2 rect. 2 LR Alexandra Borchio Fontimp + 22 signatories Adopted  
**Ap. art. 2** The law « confirming compliance with the Principles of the Republic » adopted in August 2021 ...

n ° 45 LR Stéphane Piednoir + 26 signatories Adopted  
**Ap. art. 2** The purpose of monitoring family instruction is to verify that the reason for this home instructio...

Export my selection of amendments in one click and without errors !

Export in .xlsx 27

PPL School of Freedom, Equal Opportunities and Secularity

Article 10 See the 11 amendments ★ Follow the article

- 1 The education code is thus modified :
- 2 1 ° The third paragraph of article L. 111-1 is supplemented by a sentence thus written: « Persons participating in the public education service are also required to respect these values. » ;
- 3 2 ° After the first paragraph of article L. 141-5-1, a paragraph is inserted as follows :
- 4 « The same prohibition applies to persons who participate, including during school trips, in activities related to teaching in or outside establishments, organized by these schools and local public educational establishments. »

Article 11 See the 9 amendments ★ Follow the article

- 1 After article L. 111-2 of the education code, an article L. 111-2-1 is inserted as follows :  
« Art. L. 111-2-1. – Students in primary schools, colleges and high schools wear uniform clothing within their school. The characteristics of this dress, including the choice of a uniform sports outfit, are specified by the internal regulations of each establishment. The obligation mentioned in this paragraph does not apply to shows, including rehearsals, played by students and related events.

Follow the folder

Feedback icon

# Software as a Service Startup „Contexte“



What happened to the team?

	Django + React	Django + htmx
Browser-only devs	1	0
Server-only devs	2	0
Full-stack devs	1	3

## ✓ Component Rendertime

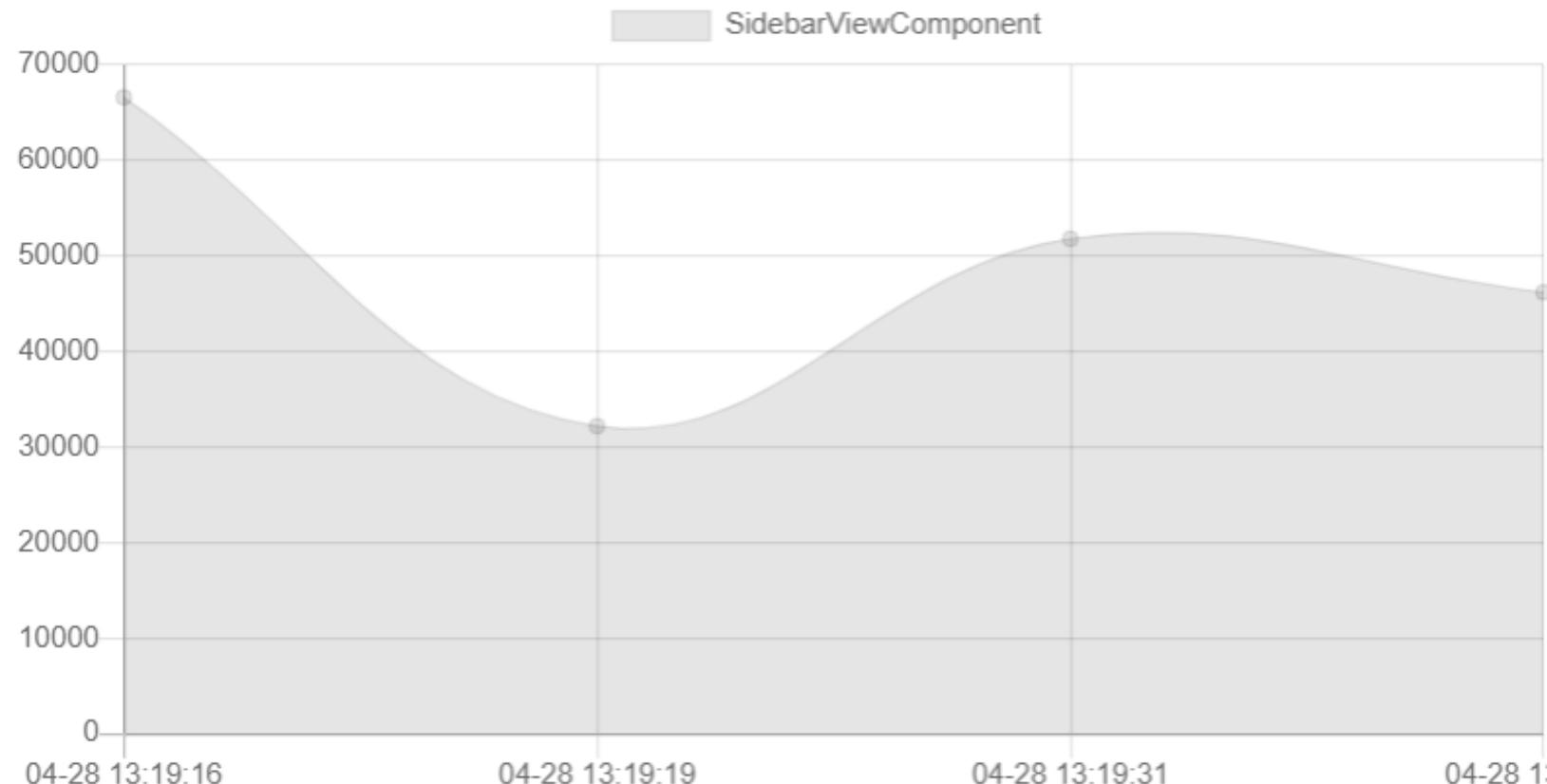
DashboardViewComponent

SidebarViewComponent

HeaderViewComponent

ChartViewComponent

TableViewComponent





```
1 @ViewComponent
2 class AnalyticsChartViewComponent(
3     private val analyticsService: AnalyticsService
4 ) {
5     fun render(name: String) = ViewContext(
6         "chartDetails" toProperty ChartViewComponent.ChartDetails(
7             "line",
8             400,
9             200,
10            name,
11            analyticsService.getTimeStampsFor(name),
12            analyticsService.getComputeTimesFor(name)
13        )
14    )
15 }
```

## ● ● ● DashboardViewComponent.kt

```
1 @ViewComponent
2 class DashboardViewComponent(
3     private val analyticsService: AnalyticsService
4 ) {
5     fun render() = ViewContext(
6         "componentLabelList" toProperty analyticsService.getLabelList(),
7     )
8 }
```

## ● ● ● DashboardViewComponent.html

```
1 <button th:each="label, index : ${componentLabelList}"
2         th:text="${label}"
3         th:hx-get="@{/lineChartComponent(name=${label})}"
4         th:hx-trigger="${index.first}? 'load,click'"
5         hx-target="#lineChart"></button>
6 <div id="lineChart">
7 </div>
```



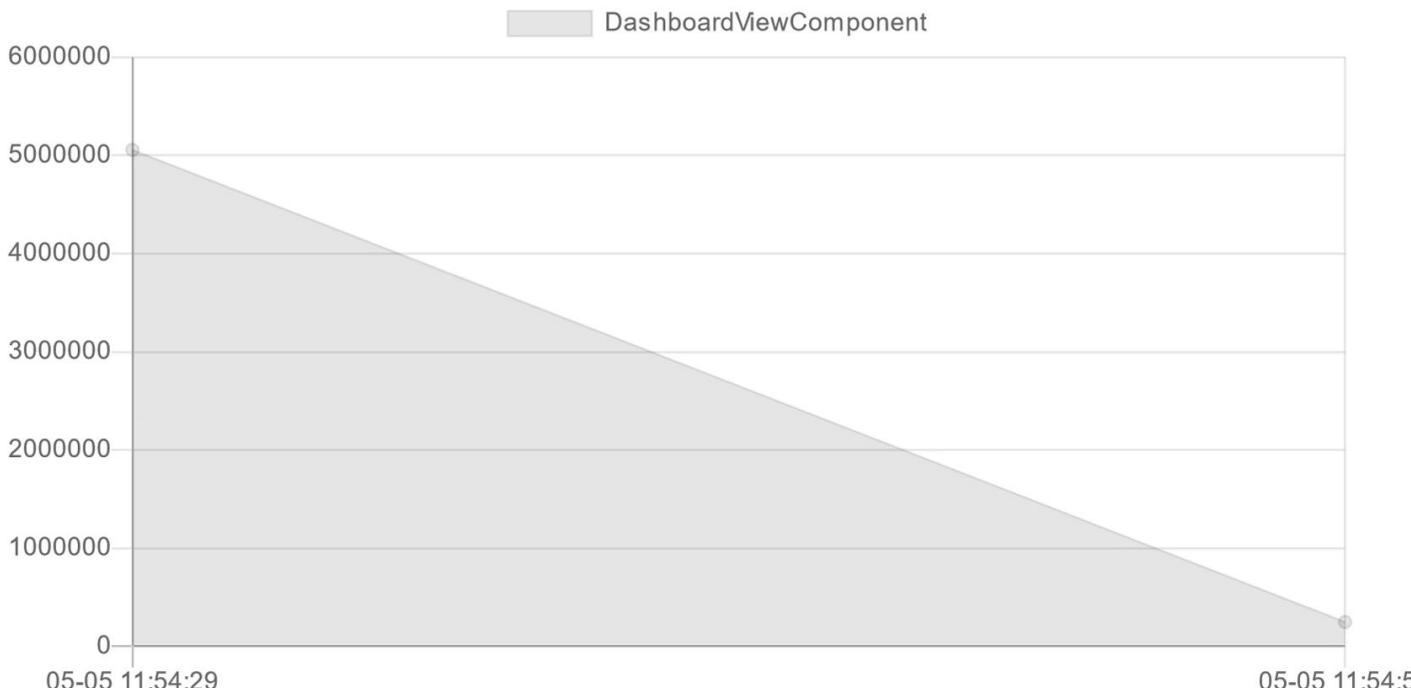
```
1 @Controller
2 class ComponentController(
3     private val analyticsChartViewComponent: AnalyticsChartViewComponent
4 ) {
5     @GetMapping("/lineChartComponent")
6     fun chartComponent(
7         @RequestParam name: String
8     ) = analyticsChartViewComponent.render(name)
9 }
```



# ADMIN

[+ New Report](#)[Dashboard](#)[Tables](#)[Forms](#)[Tabbed Content](#)[Calendar](#)

## ✓ Component Rendertime

[DashboardViewComponent](#)[SidebarViewComponent](#)[HeaderViewComponent](#)[ChartViewComponent](#)[TableViewComponent](#)

## ☰ Person Form

First Name

Your First Name

Last Name

Your Last Name

Phone Number

Your Telephone Number

Birth Date

13.05.2023



Email

Your Email

Submit

The screenshot shows a Java IDE interface with three code files:

- PersonFormViewComponent.html**: A Thymeleaf template file containing HTML for a form to input First Name and Last Name.
- PersonController.kt**: A Spring Controller class with a savePerson() method that handles a PUT request to save a person.
- PersonFormViewComponent.kt**: A Kotlin ViewComponent class that implements the render() method to produce a ViewContext, and contains a PersonFormDTO class definition.

The **PersonFormViewComponent.kt** file is highlighted with a red border. The code within it is:

```
import ...

@ViewComponent
class PersonFormViewComponent {
    fun render() = ViewContext(...contextAttributes:
        "personPut" toProperty ApiConfig.Person.put
    )

    class PersonFormDTO(
        val id: Int?,
        val firstName: String,
        val lastName: String,
        val phoneNumber: String,
        val email: String,
        val birthDate: LocalDate
    ) {
        fun convertToPerson() = Person(
            this.id,
            this.firstName,
            this.lastName,
            this.phoneNumber,
            this.email,
            this.birthDate
        )
    }
}
```

PersonFormViewComponent.html

```
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <form th:hx-put="${personPut}">
4     <label>
5         First Name
6         <input placeholder="Your First Name" type="text">
7     </label>
8     <label>
9         Last Name
10        <input placeholder="Your Last Name" type="text">
11    </label>
```

PersonController.kt

```
11 @Controller
12 class PersonController(
13     private val personService: PersonService,
14     private val toastViewComponent: ToastViewComponent
15 ) {
16
17     @PostMapping(ApiConfig.Person.put)
18     fun savePerson(
19         personFormDTO: PersonFormViewComponent.PersonFormDTO,
20         response: HttpServletResponse
21     ): ViewContext {
22         personService.savePerson(personFormDTO.convertToPerson())
23         return toastViewComponent.render(
24             message: "Successfully saved Person",
25             durationMs: 2000
26         )
27     }
28 }
```

PersonFormViewComponent.kt

```
3 > import ...
4
5 @ViewComponent
6 class PersonFormViewComponent {
7     fun render() = ViewContext( ...contextAttributes:
8         "personPut" toProperty ApiConfig.Person.put
9     )
10
11 class PersonFormDTO(
12     val id: Int?,
13     val firstName: String,
14     val lastName: String,
15     val phoneNumber: String,
16     val email: String,
17     val birthDate: LocalDate
18 ) {
19     fun convertToPerson() = Person(
20         this.id,
21         this.firstName,
22         this.lastName,
23         this.phoneNumber,
24         this.email,
25         this.birthDate
26     )
27 }
28 }
```

5 PersonFormViewComponent.html ×

```
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<form th:hx-put="${personPut}">
    <label>
        First Name
        <input placeholder="Your First Name" type="text">
    </label>
    <label>
        Last Name
        <input placeholder="Your Last Name" type="text">
    </label>
```

! 2 ! 1 ! 1 ↑ ↓ 3

PersonFormViewComponent.kt x

```
@ViewComponent
class PersonFormViewComponent {
    fun render() = ViewContext(...contextAttributes:
        "personPut" toProperty ApiConfig.Person.put
    )
}

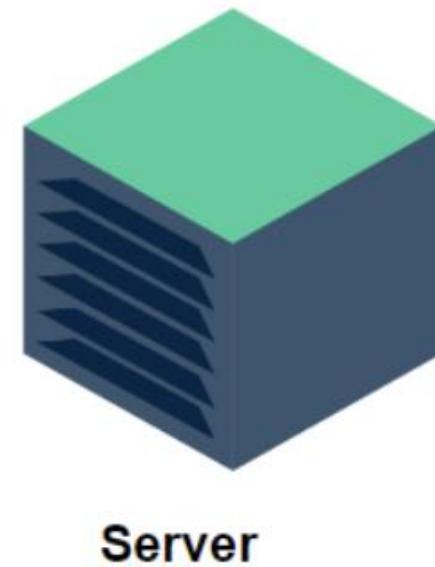
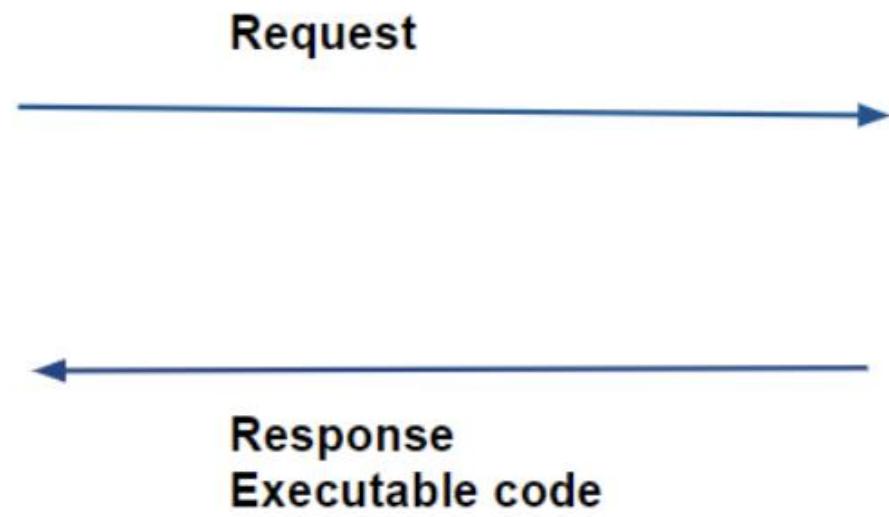
class PersonFormDTO(
    val id: Int?,
    val firstName: String,
    val lastName: String,
    val phoneNumber: String,
    val email: String,
    val birthDate: LocalDate
) {
    fun convertToPerson() = Person(
        this.id,
        this.firstName,
        this.lastName,
        this.phoneNumber,
        this.email,
        this.birthDate
    )
}
```

PersonController.kt

```
11 @Controller
12 class PersonController(
13     private val personService: PersonService,
14     private val toastViewComponent: ToastViewComponent
15 ) {
16
17     @PutMapping(ApiConfig.Person.put)
18     fun savePerson(
19         personFormDTO: PersonFormViewComponent.PersonFormDTO,
20         response: HttpServletResponse
21     ): ViewContext {
22         personService.savePerson(personFormDTO.convertToPerson())
23         return toastViewComponent.render(
24             message: "Successfully saved Person",
25             durationMs: 2000
26         )
27     }
28 }
```



**Client**



**Server**



## ToastViewComponent.html

```
1 <div id="toast-element">
2   <script th:inline="javascript" >
3     setTimeout(() =>
4       document.getElementById('toast-element').style.display = 'none',
5       [[$duration]])
6   )
7   </script>
8
9   <span th:text="${message}"></span>
10  <button onclick="document.getElementById('toast-element').style.display = 'none'">
11    X
12  </button>
13 </div>
```



# Alpine.js



## ADMIN

+ New Report

Dashboard

Tables

Forms

Tabbed Content

Calendar

## Alpine Tabs Content

AlpineTab1

AlpineTab2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.



## AlpineTabViewComponent.kt

```
1 @ViewComponent
2 class AlpineTabViewComponent {
3     fun render(tabs: List<Tab>) = ViewContext(
4         "headerList" toProperty tabs.map { it.tabHeader },
5         "contentList" toProperty tabs.map { it.tabContent }
6     )
7
8     class Tab(
9         val tabHeader: String,
10        val tabContent: String
11    )
12 }
```



## AlpineTabViewComponent.html

```
1 <div x-data="{ openTab: 0 }">
2   <ul>
3     <li th:each="header, el : ${headerList}">
4       <button th:text="${header}"
5           th:x-on:click="'openTab = '+ ${el.index}">
6         </button>
7       </li>
8     </ul>
9
10    <div th:each="content, el : ${contentList}"
11        th:x-show="'openTab === ' + ${el.index}"
12        th:text="${content}">
13      </div>
14  </div>
```



## AlpineTabViewComponent.html

```
1 <div x-data="{ openTab: 0 }">
2   <ul>
3     <li th:each="header, el : ${headerList}">
4       <button th:text="${header}"
5           th:x-on:click="'openTab = '+ ${el.index}">
6         </button>
7       </li>
8     </ul>
9
10    <div th:each="content, el : ${contentList}"
11        th:x-show="'openTab === ' + ${el.index}"
12        th:text="${content}">
13      </div>
14  </div>
```



## AlpineTabViewComponent.html

```
1 <div x-data="{ openTab: 0 }">
2   <ul>
3     <li th:each="header, el : ${headerList}">
4       <button th:text="${header}"
5           th:x-on:click="'openTab = '+ ${el.index}">
6         </button>
7       </li>
8     </ul>
9
10    <div th:each="content, el : ${contentList}"
11        th:x-show="'openTab === ' + ${el.index}"
12        th:text="${content}">
13      </div>
14 </div>
```



## AlpineTabViewComponent.html

```
1 <div x-data="{ openTab: 0 }">
2   <ul>
3     <li th:each="header, el : ${headerList}">
4       <button th:text="${header}"
5           th:x-on:click="'openTab = '+ ${el.index}">
6         </button>
7       </li>
8     </ul>
9
10    <div th:each="content, el : ${contentList}"
11        th:x-show="'openTab === ' + ${el.index}"
12        th:text="${content}">
13      </div>
14  </div>
```



```
<div x-data="{ openTab: 0 }">
  <ul>
    <li x-on:click="openTab = 0">
      <button>AlpineTab1</button>
    </li>
    <li x-on:click="openTab = 1">
      <button>AlpineTab2</button>
    </li>
  </ul>
  <div>
    <div x-show="openTab === 0">
      Lorem ipsum dolor sit amet, consetetur sadipscing
    </div>
    <div x-show="openTab === 1" style="display: none;">
      At vero eos et accusam et justo duo dolores
    </div>
  </div>
</div>
```



# spring® ViewComponent



# Alpine.js

# supabase Security spring® Boot Starter

[github.com/tschuehly/supabase-security-spring-boot-starter](https://github.com/tschuehly/supabase-security-spring-boot-starter)



# supabase



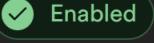
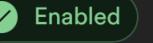
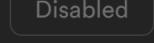
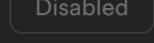
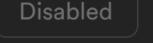
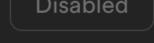
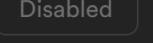
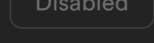
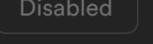
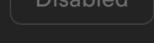
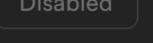
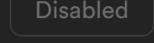
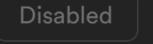
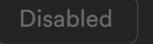
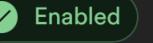
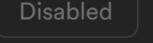
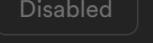
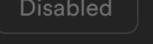
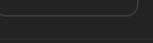
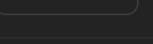
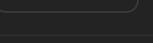
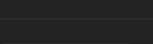
**Hosted PostgreSQL**



**Hosted GoTrue  
Authentication Server**

## Auth Providers

Authenticate your users through a suite of providers and login methods

▼  Email	 Enabled	 Disabled
▼  Phone	 Disabled	 Enabled
▼  SAML 2.0	 Disabled	 Disabled
▼  Apple	 Disabled	 Disabled
▼  Azure	 Disabled	 Disabled
▼  Bitbucket	 Disabled	 Disabled
▼  Discord	 Disabled	 Disabled
▼  Facebook	 Disabled	 Disabled
▼  GitHub	 Disabled	 Disabled
▼  GitLab	 Disabled	
▼  Google	 Enabled	
▼  Keycloak	 Disabled	
▼  LinkedIn	 Disabled	
▼  Notion	 Disabled	
▼  Twitch	 Disabled	
▼  Twitter	 Disabled	
▼  Slack	 Disabled	
▼  Spotify	 Disabled	
▼  WorkOS	 Disabled	
▼  Zoom	 Disabled	



```
1 supabase:  
2   projectId: yourProjectId  
3   anonKey: ${SUPABASE_ANON_KEY}  
4   databasePassword: ${SUPABASE_DATABASE_PW}  
5   jwtSecret: ${SUPABASE_JWT_SECRET}
```



```
1 supabase:  
2   public:  
3     get:  
4       - "/"  
5       - "/resetPassword"  
6       - "/unauthenticated"  
7       - "/unauthorized"  
8     post:  
9       - "/api/user/register"  
10      - "/api/user/login"  
11      - "/api/user/jwt"  
12      - "/api/user/sendPasswordResetEmail"
```



```
1 <script>
2   if (window.location.hash.startsWith("#access_token")) {
3     htmx.ajax('POST', '/api/user/jwt', {target: '#body', swap: 'outerHTML'})
4       .then(window.location.hash = "")
5   }
6 </script>
```



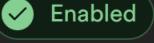
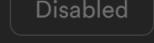
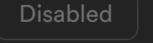
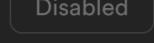
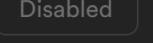
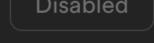
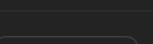
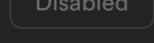
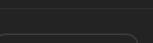
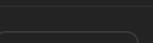
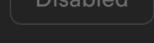
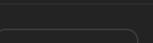
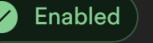
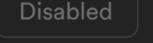
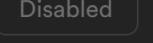
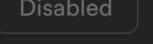
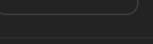
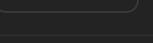
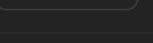
```
1 <form>
2   <label>Email:
3     <input type="text" name="email"/>
4   </label>
5   <label>Password:
6     <input type="password" name="password"/>
7   </label>
8   <button hx-post="/api/user/register">Submit</button>
9 </form>
```



```
1 <form>
2   <label>Email:
3     <input type="text" name="email"/>
4   </label>
5   <label>Password:
6     <input type="password" name="password"/>
7   </label>
8   <button hx-post="/api/user/login">Submit</button>
9 </form>
```

## Auth Providers

Authenticate your users through a suite of providers and login methods

▼  Email	 Enabled	 Disabled
▼  Phone	 Disabled	
▼  SAML 2.0	 Disabled	 Disabled
▼  Apple	 Disabled	 Disabled
▼  Azure	 Disabled	 Disabled
▼  Bitbucket	 Disabled	 Disabled
▼  Discord	 Disabled	 Disabled
▼  Facebook	 Disabled	 Disabled
▼  GitHub	 Disabled	 Disabled
▼  GitLab		 Disabled
▼  Google	 Enabled	
▼  KeyCloak		 Disabled
▼  LinkedIn		 Disabled
▼  Notion		 Disabled
▼  Twitch		 Disabled
▼  Twitter		 Disabled
▼  Slack		 Disabled
▼  Spotify		 Disabled
▼  WorkOS		 Disabled
▼  Zoom		 Disabled



```
1 <a href="https://<projectId>.supabase.co/auth/v1/authorize?provider=google">Sign In with Google</a>
```



```
1 supabase:  
2   roles:  
3     admin:  
4       get:  
5         - "/admin/**"  
6     user:  
7       post:  
8         - "/user-feature-1/**"
```



```
1 curl -X PUT --location "http://localhost:8080/api/user/setRoles" \
2 -H "Cookie: JWT=service_role_secret" \
3 -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" \
4 -d "userId=381c6358-22dd-4681-81e3-c79846117511&roles=user"
```



```
1 @Service
2 class UserManagementService(
3     private val supabaseService: SupabaseUserService
4 ) {
5     fun setRolesForUser(){
6         supabaseService.setRoles("service_role_secret", "1234", listOf("ADMIN"))
7     }
8 }
```



```
1 update auth.users
2 set role = 'service_role'
3 where email = 'test@example.com';
```



```
1 <form>
2     <label>User Id
3         <input name="userId" type="text">
4     </label>
5     <label>Admin Role
6         <input name="roles" type="checkbox" value="admin"/>
7     </label>
8     <label>User Role
9         <input name="roles" type="checkbox" value="user"/>
10    </label>
11    <button hx-put="/api/user/setRoles">Submit</button>
12 </form>
```



```
1 @ControllerAdvice
2 class ExceptionController(
3     private val toastViewComponent: ToastViewComponent
4 ) {
5     @ExceptionHandler(InvalidLoginCredentialsException::class)
6     fun handleInvalidLoginCredentialsException(exception: InvalidLoginCredentialsException): ViewContext {
7         return toastViewComponent.render(
8             "Your credentials are invalid",
9             10000
10        )
11    }
12 }
```

# ADMIN

[\*\*+ New Report\*\*](#) [Dashboard](#) [Tables](#) [Forms](#) [Tabbed Content](#) [Calendar](#)

## Login

Email:

tschuehly@outlook.com

Password:

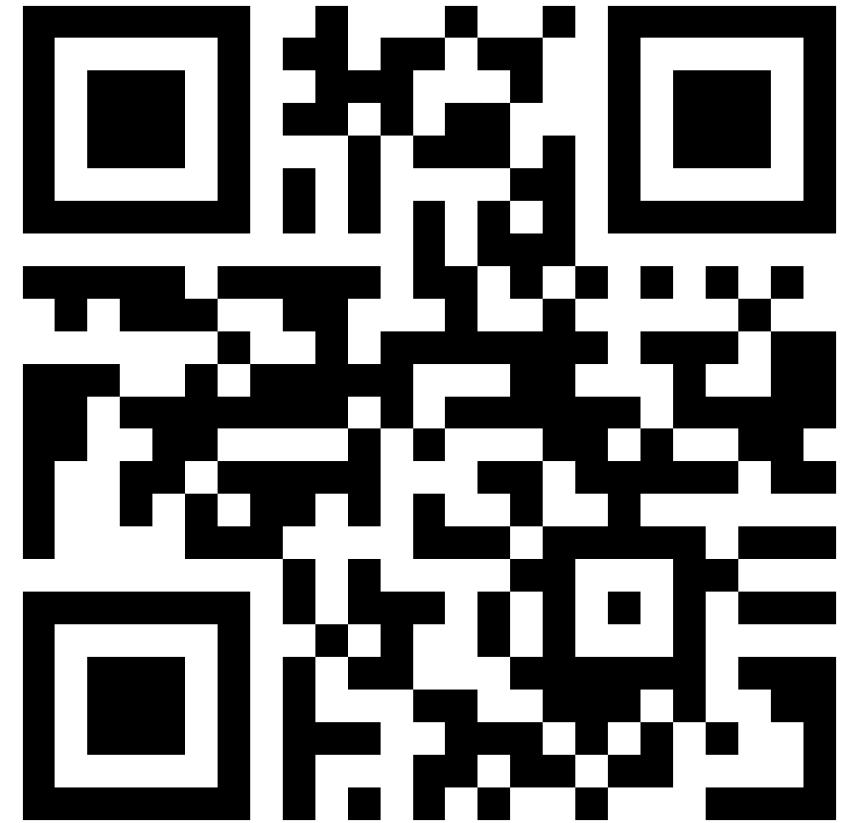
••••••••

[Submit](#)Template by [David Grzyb](#).

Your credentials are invalid

---

[fullstack.tschuehly.de](https://fullstack.tschuehly.de)



[github.com/tschuehly/rapid-  
fullstack-demo](https://github.com/tschuehly/rapid-fullstack-demo)



**I WANT YOU  
TO CONTRIBUTE  
AND CONNECT**

**TSCHUEHLY.DE**

# Standard HTMX

PersonFormViewComponent.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xml...
3 <!--@thymesVar id="person" type="de.tschuehly.rapidfu...
4
5 <form th:hx-put="${ApiConfig.Person.put}"
6     class="p-10 bg-white rounded shadow-xl"
7     th:object="${person}>
8     <div class="hidden">
9         <label class="block text-sm text-gray-600">
10            ID
11            <input aria-label="ID" class="w-full px-5 py-3" type="text" th:field="*{id}">
12        </label>
13    </div>
14    <div class="">
15        <label class="block text-sm text-gray-600">
16            First Name
17            <input aria-label="Name" class="w-full px-5 py-3" placeholder="Your First Name" type="text" th:field="*{firstName}">
18        </label>
19    </div>
20
21 </div>
22
```

PersonController.kt

```
2
3     > import ...
10
11 @Controller
12 class PersonController(
13     private val personService: PersonService,
14     private val toastViewComponent: ToastViewComponent
15 ) {
16
17     @PutMapping(ApiConfig.Person.put)
18     fun savePerson(
19         personFormDTO: PersonFormViewComponent.PersonFormDTO,
20         response: HttpServletResponse
21     ): ViewContext {
22         personService.savePerson(personFormDTO.convertToPerson())
23         response.addHeader( name: "HX-Retarget", value: "#toast")
24         return toastViewComponent.render(
25             message: "Successfully saved Person",
26             durationMs: 2000
27         )
28     }
29 }
```

# ViewAction

5 PersonFormViewComponent.html ×

4

5 <form view:action="createNewPerson(form)"

6   class="p-10 bg-white rounded shadow-xl"

7   th:object="\${person}">

8   

9     <label class="block text-sm text-gray-600">

10     ID

11     <input aria-label="ID" class="w-full px-5 py-2" type="text" th:field="\*{id}">

12   </label>

13 </div>

14 <div class="">

15   <label class="block text-sm text-gray-600">

16     First Name

17     <input aria-label="Name" class="w-full px-5 py-2" placeholder="Your First Name" type="text" th:field="\*{name}">

18 </label>

19

20

21 @ViewComponent

22 class PersonFormViewComponent(

23   private val personService: PersonService

24 ) {

25   fun render(personId: Int?) = ViewContext( ...contextAttributes:

26     "personPost" toProperty ApiConfig.Person.put,

27     "person" toProperty getPerson(personId)

28 }

29

30 @ViewAction

31 fun createNewPerson(form: PersonFormDTO): ViewContext {

32   val savedPerson = personService.savePerson(form.convertToPerson())

33   return render(savedPerson.id)

34 }

35

36

mimacom

RAPID SERVER-SIDE FULL STACK WEB DEVELOPMENT WITH VIEWCOMPONENTS AND HTMX

tschuehly.de @tschuehly

2023-05-18

101