
THOMAS SCHUEHLY

Joyful server-side full stack web development with Spring ViewComponent

9th August 2023

Thanks to JavaLand for sponsoring this talk!



Thomas Schühly



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

Would you describe yourself as a Software Engineer?

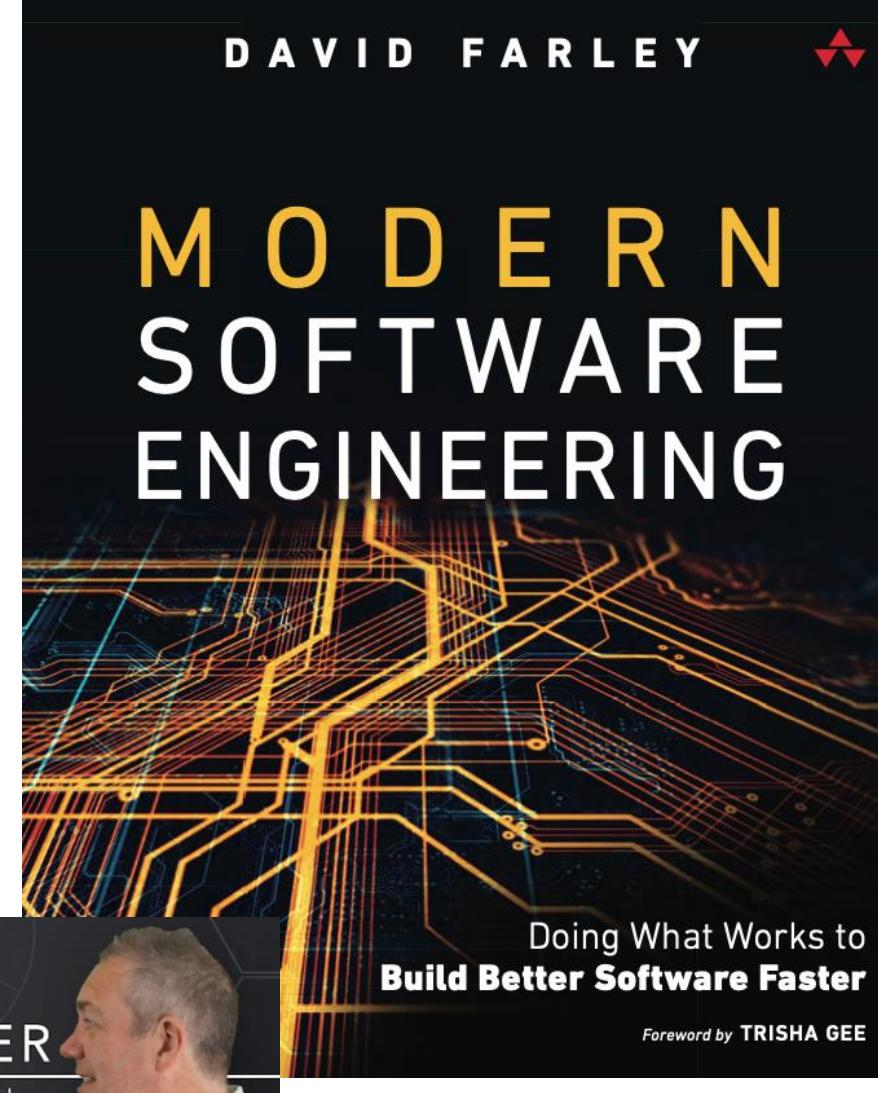
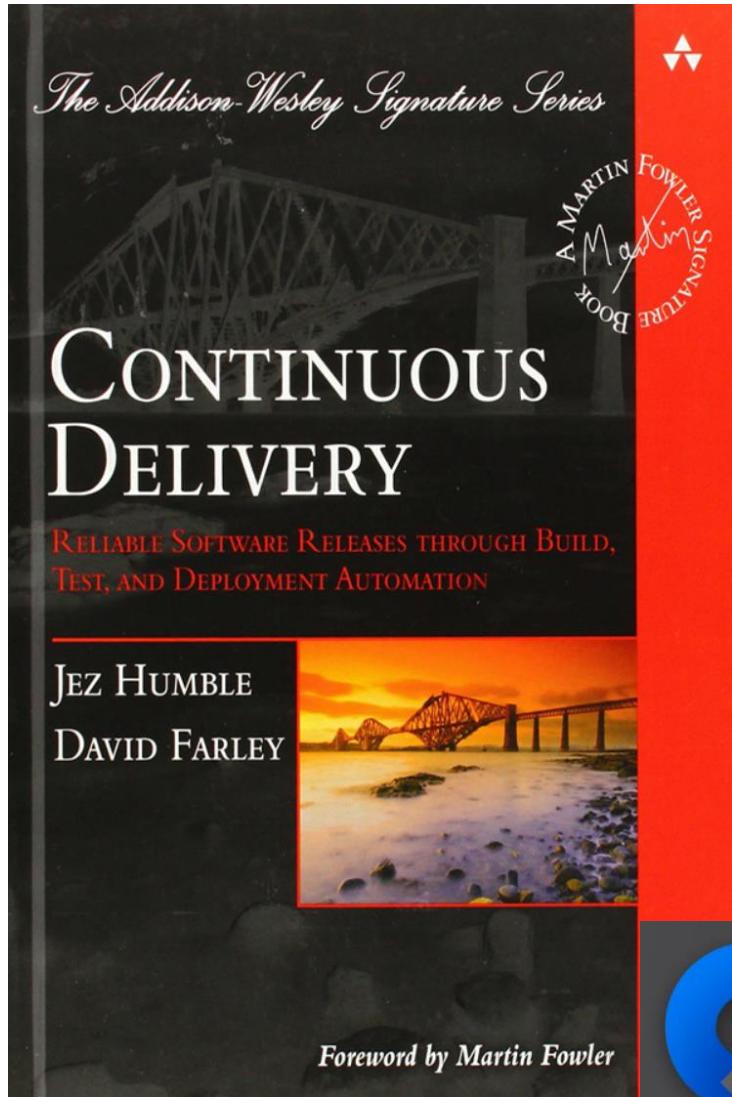
Software Engineering

Software Engineering



J AuthServerApplication.java

```
@SpringBootApplication()
public class AuthServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(AuthServerApplication.class, args);
    }
}
```



BETTER SOFTWARE FASTER

New videos every Wednesday with Dave Farley

SPONSORS

EQUAL EXPERTS roost.ai Tricentis TransFLICC SLEUTH

A portrait of David Farley from the chest up, wearing a white shirt, smiling slightly.

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?

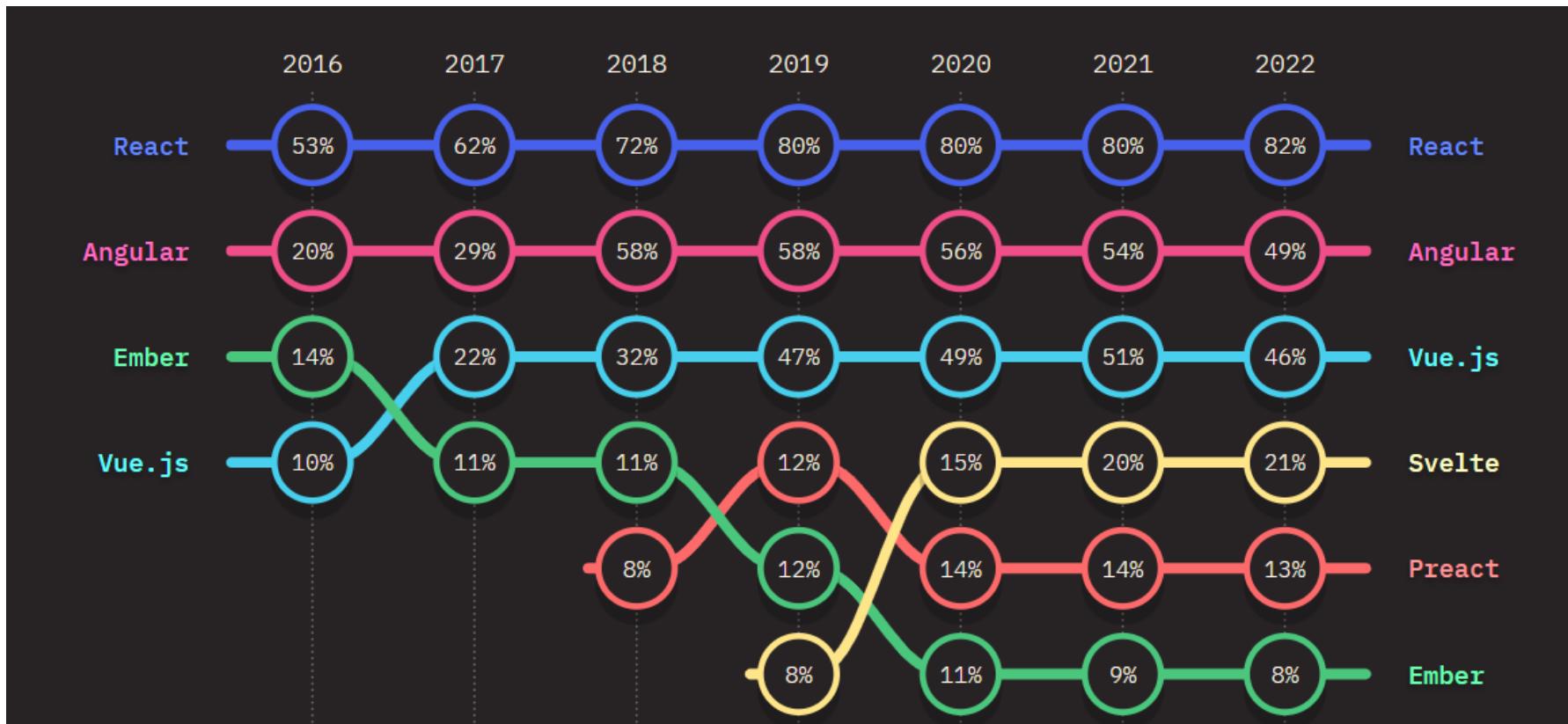
“The systems that modern engineers create are increasingly complex and sophisticated, [..] modern software engineering, needs to focus us on managing that complexity”

Dave Farley

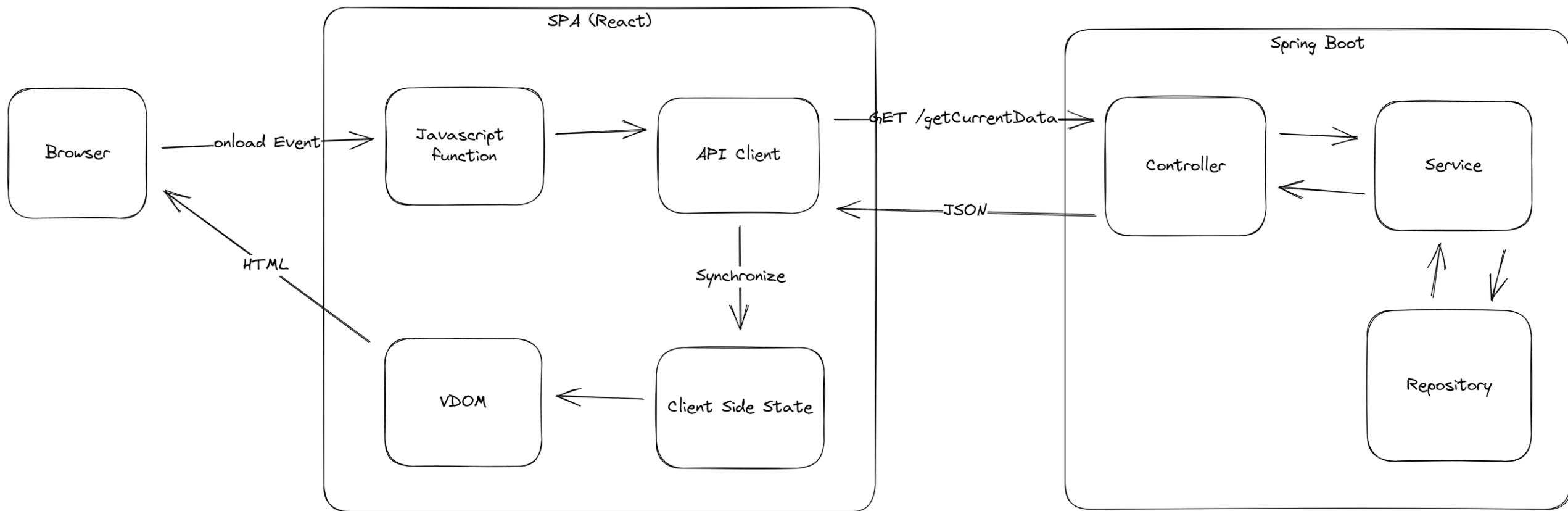
What is Modern Software Engineering?



FRONT-END FRAMEWORK 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?**



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.”

Tom MacWright

Second-guessing the modern web

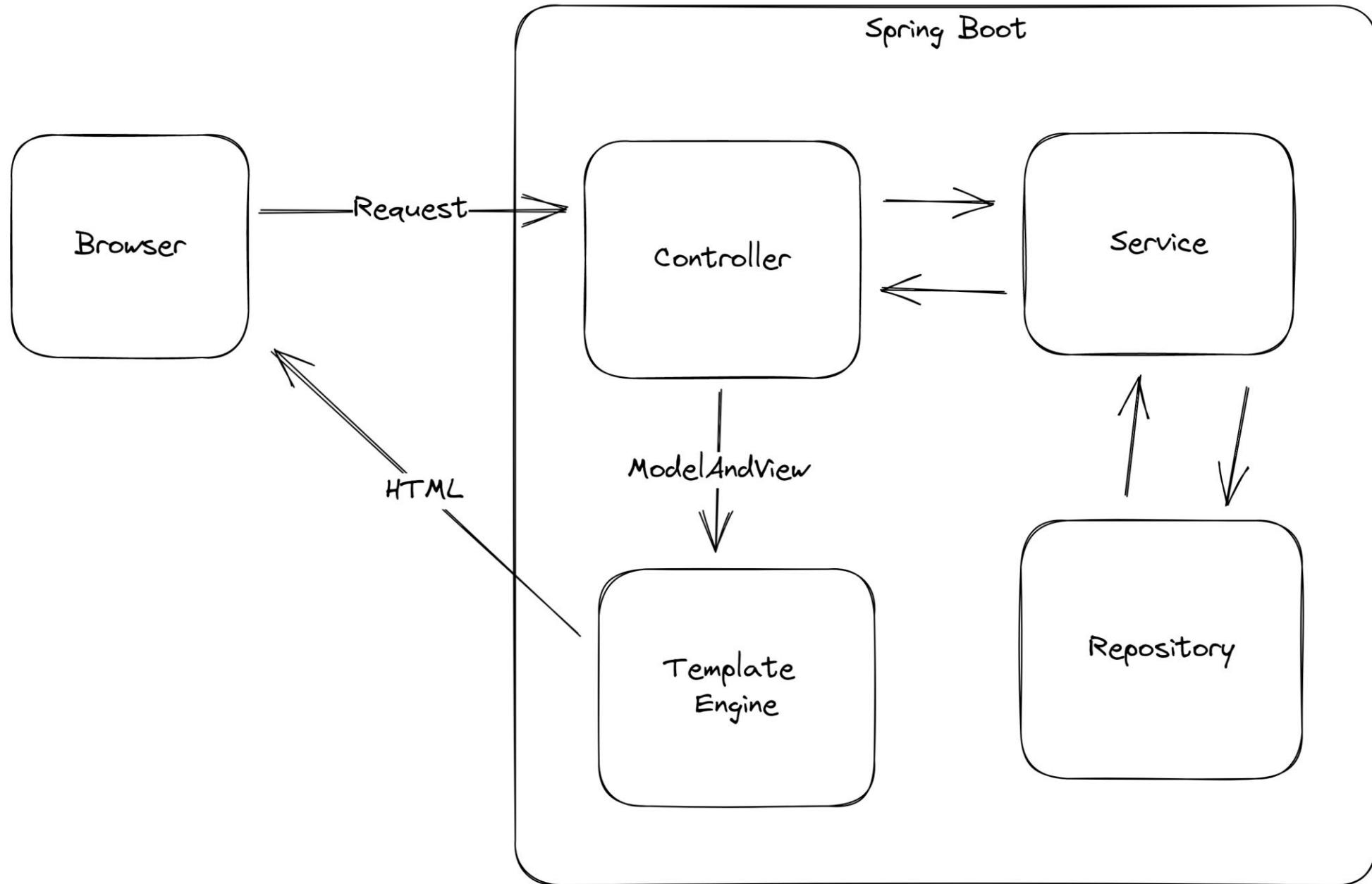
<https://macwright.com/2020/05/10/spa-fatiguehtml>

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

thoughtworks Technology Radar Vol. 27

www.thoughtworks.com/content/dam/thoughtworks/documents/radar/2022/10/tr_technology_radar_vol_27_en.pdf

Multi Page Application



**DB TO
JSON TO
JS TO HTML**

**DB
TO HTML**

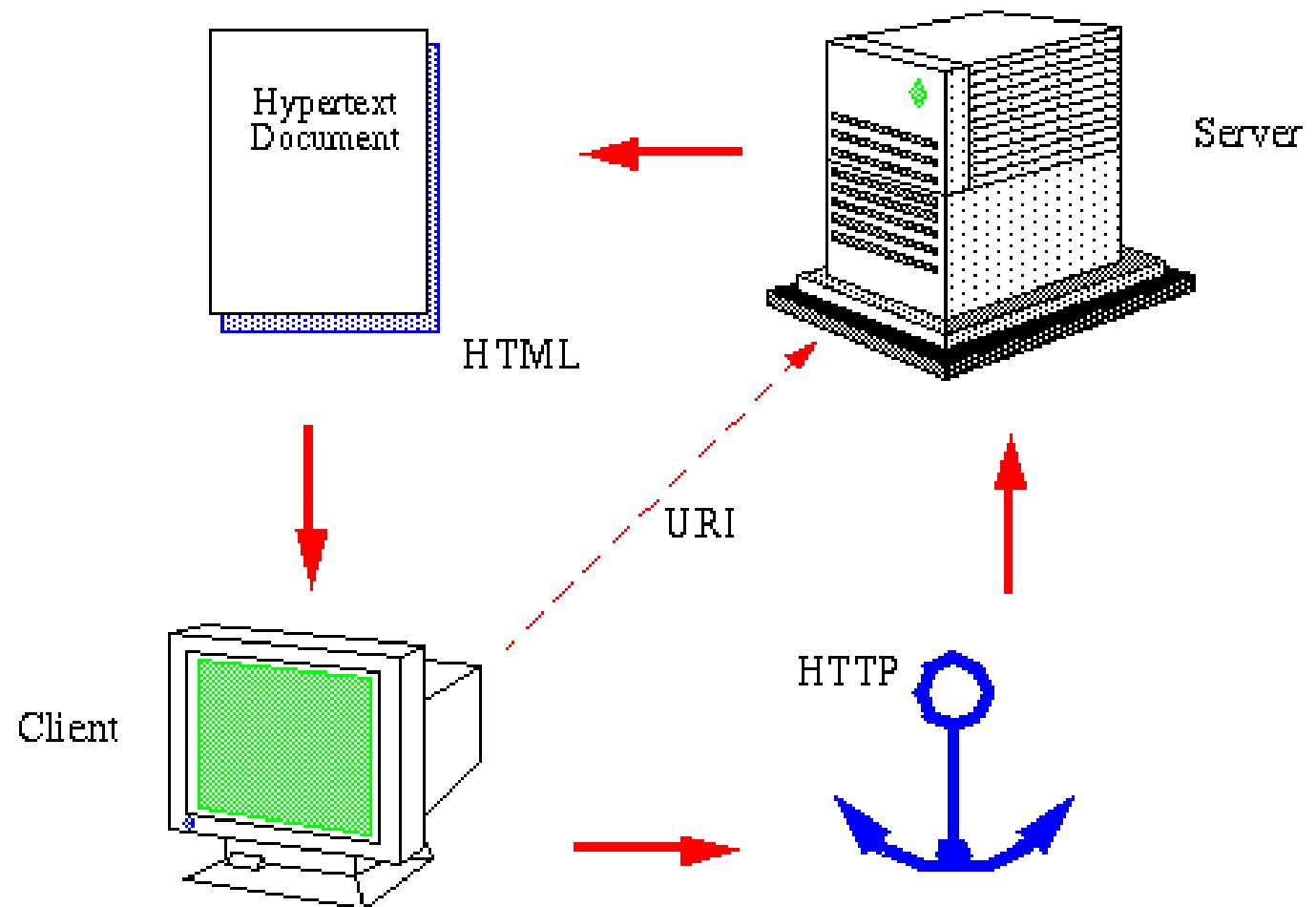


The World Wide Web

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.

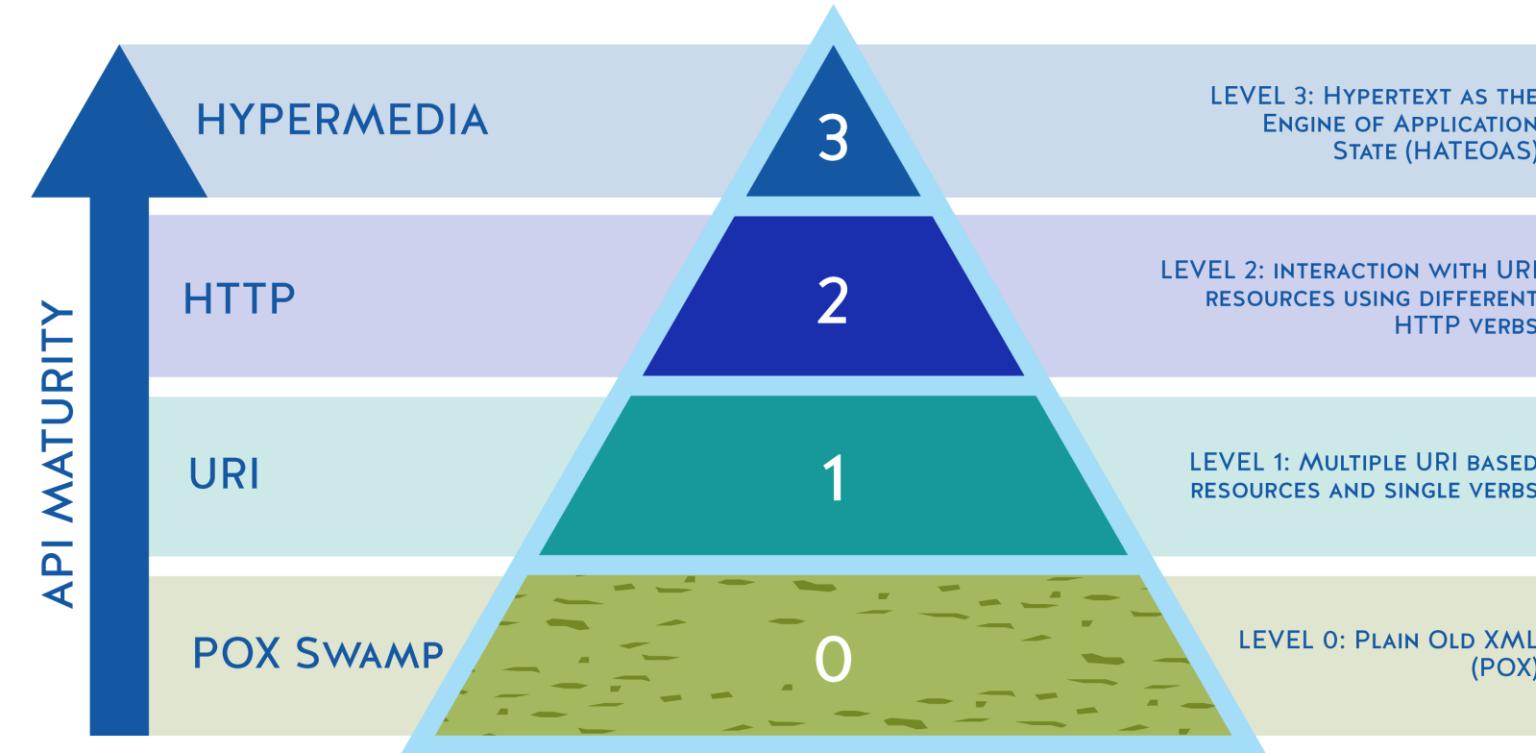
Roy Fielding

Architectural Styles and
the Design of Network-based Software Architectures

THE FORGOTTEN PART OF REST

HATEOAS

RICHARDSON MATURITY MODEL



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



"USING JSON..."



"HERE ARE MY
API DOCS"

```
1 {
2   "_embedded": {
3     "orderList": [
4       {
5         "orderId": "001A",
6         "price": 150,
7         "quantity": 25,
8         "_links": {
9           "self": {
10             "href": "http://localhost:8080/spring-security-rest/api/customers/10A/001A"
11           }
12         }
13       },
14       {
15         "orderId": "002A",
16         "price": 250,
17         "quantity": 15,
18         "_links": {
19           "self": {
20             "href": "http://localhost:8080/spring-security-rest/api/customers/10A/002A"
21           }
22         }
23       }
24     ],
25     "_links": {
26       "self": {
27         "href": "http://localhost:8080/spring-security-rest/api/customers/10A/orders"
28       }
29     }
30   }
31 }
```



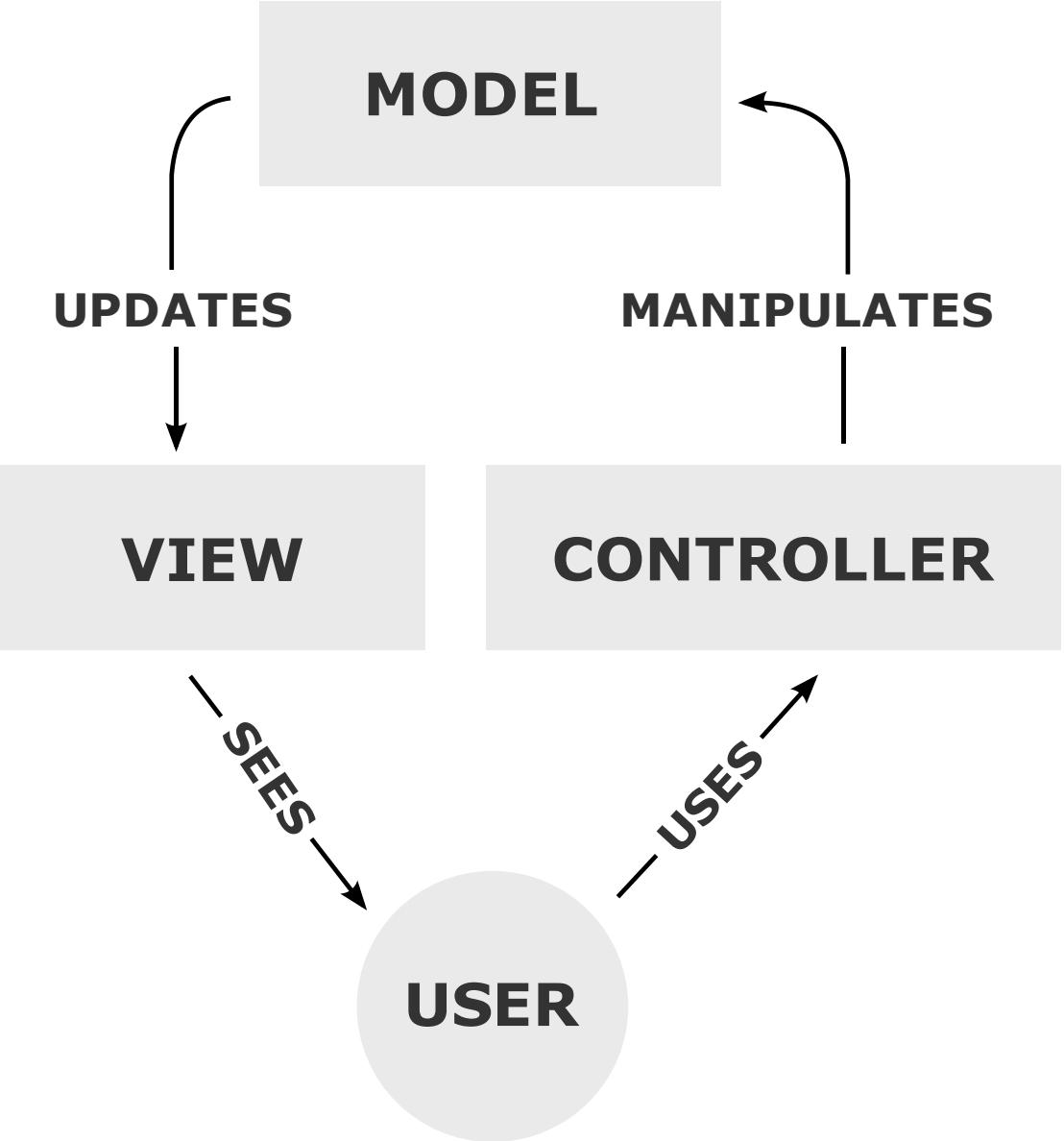
Creating your own Client

Just use
the Browser
as Client

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <p>This form will be submitted using the POST method:
6   </p>
7 <form action="/action_page.php" target="_blank"
8   method="post">
9   <input type="text" name="fname" value="John">
10  <input type="text" name="lname" value="Doe">
11  <input type="submit" value="Submit">
12
13 </form>
14
15 </body>
16
17 </html>
```

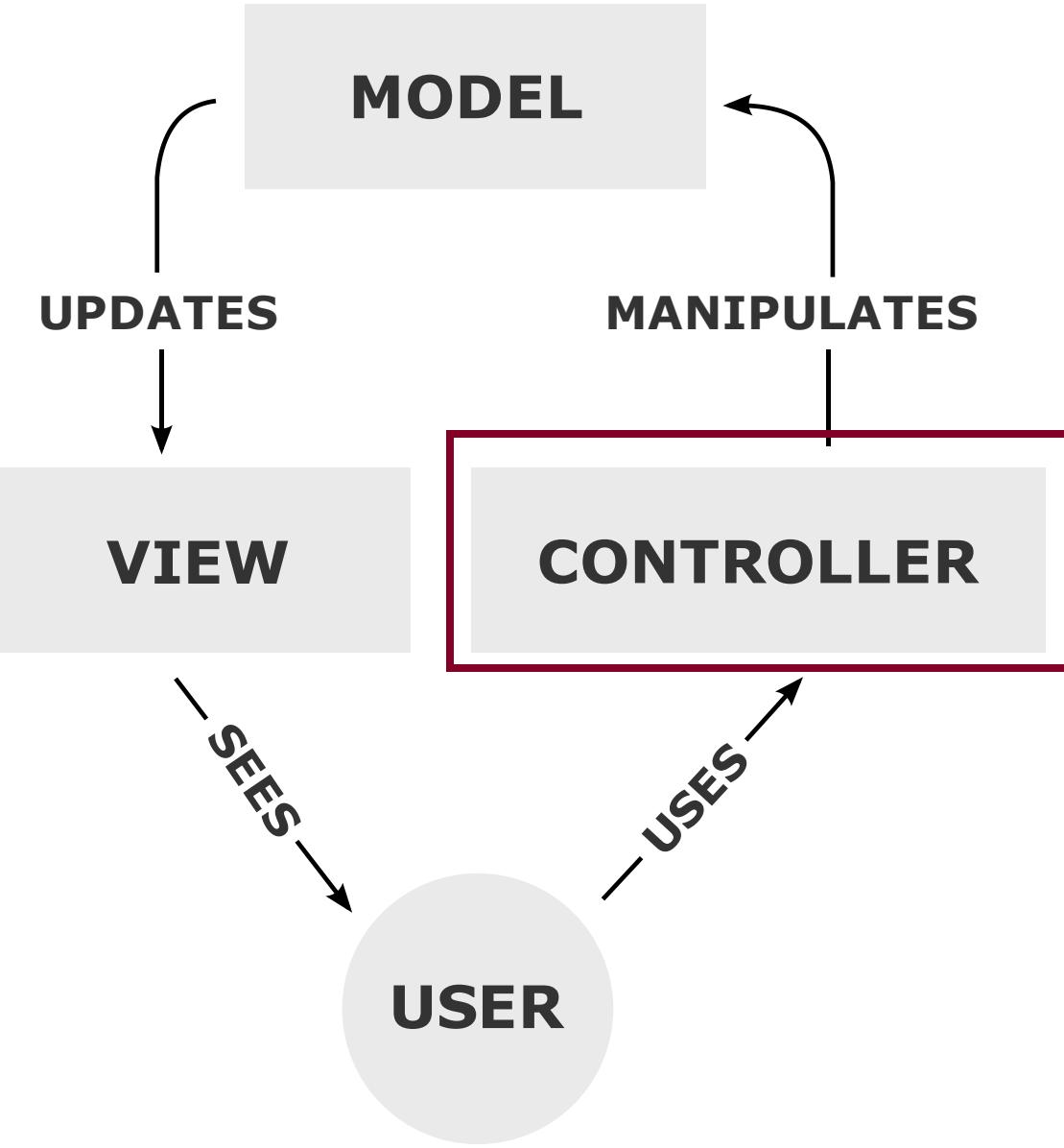


```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```



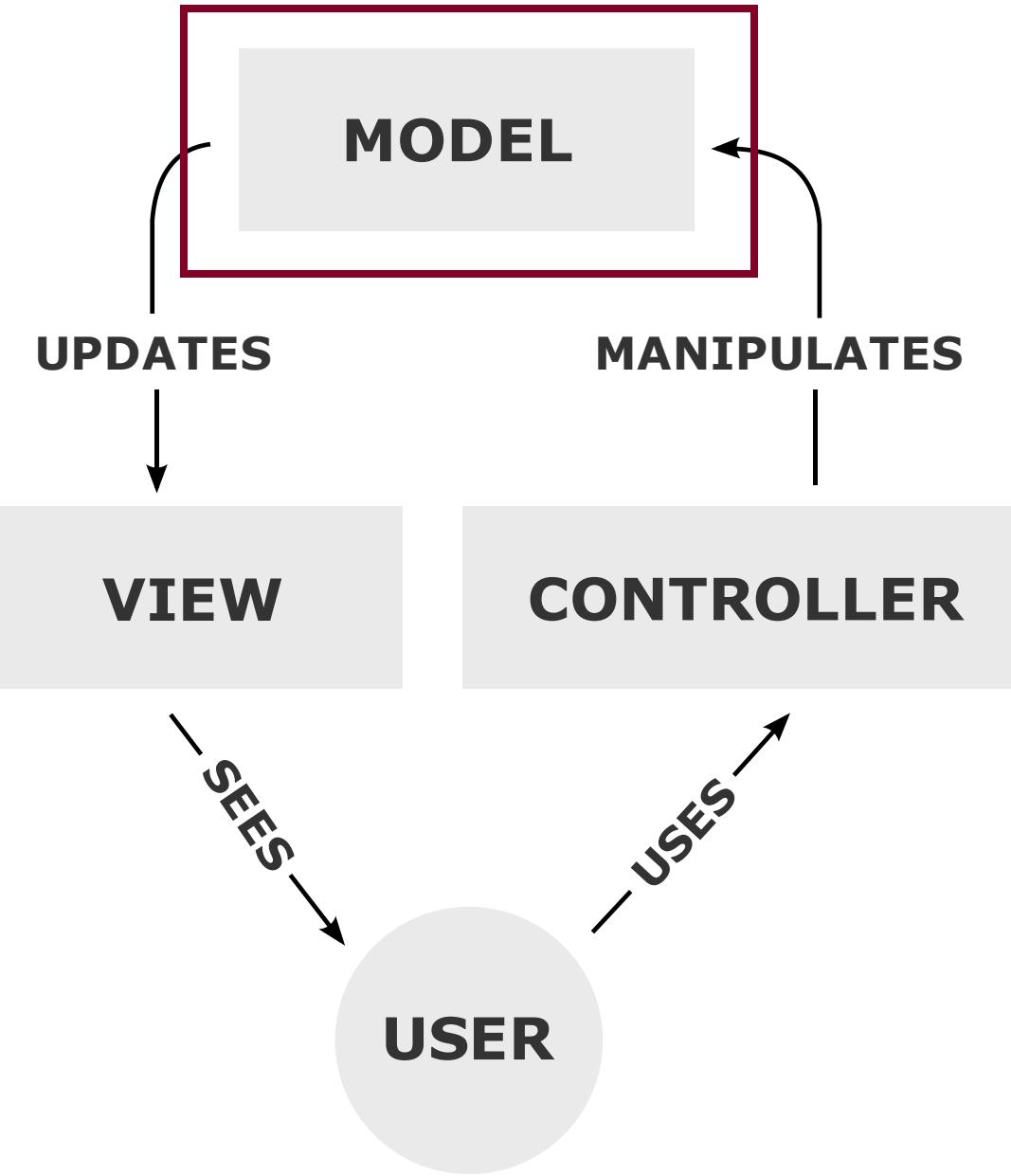


```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```



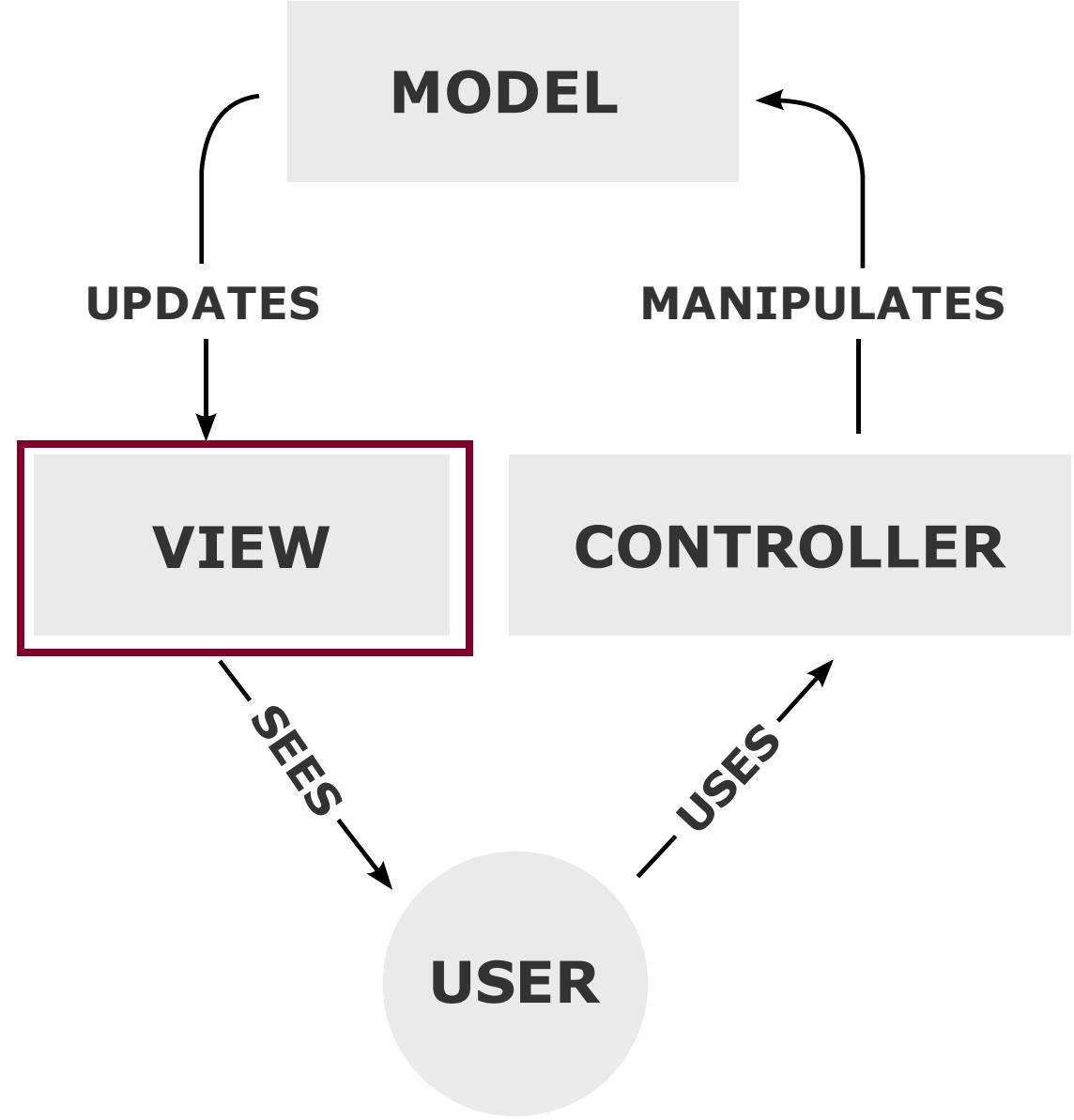


```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9     return "greeting"
10 }
11 }
```

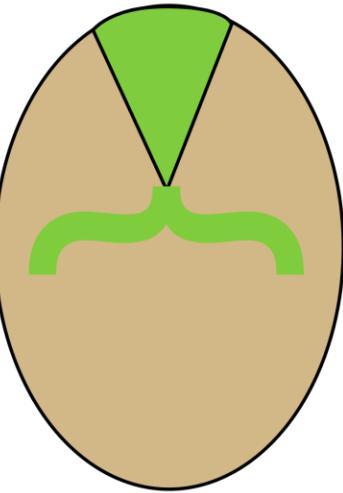




```
1 @Controller
2 class GreetingController {
3
4     @GetMapping("/greeting")
5     fun home(model: Model): String{
6
7         model.addAttribute("name", "SpringIO")
8
9         return "greeting"
10    }
11 }
```



jstachio



Logic-less templates.



Thymeleaf





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

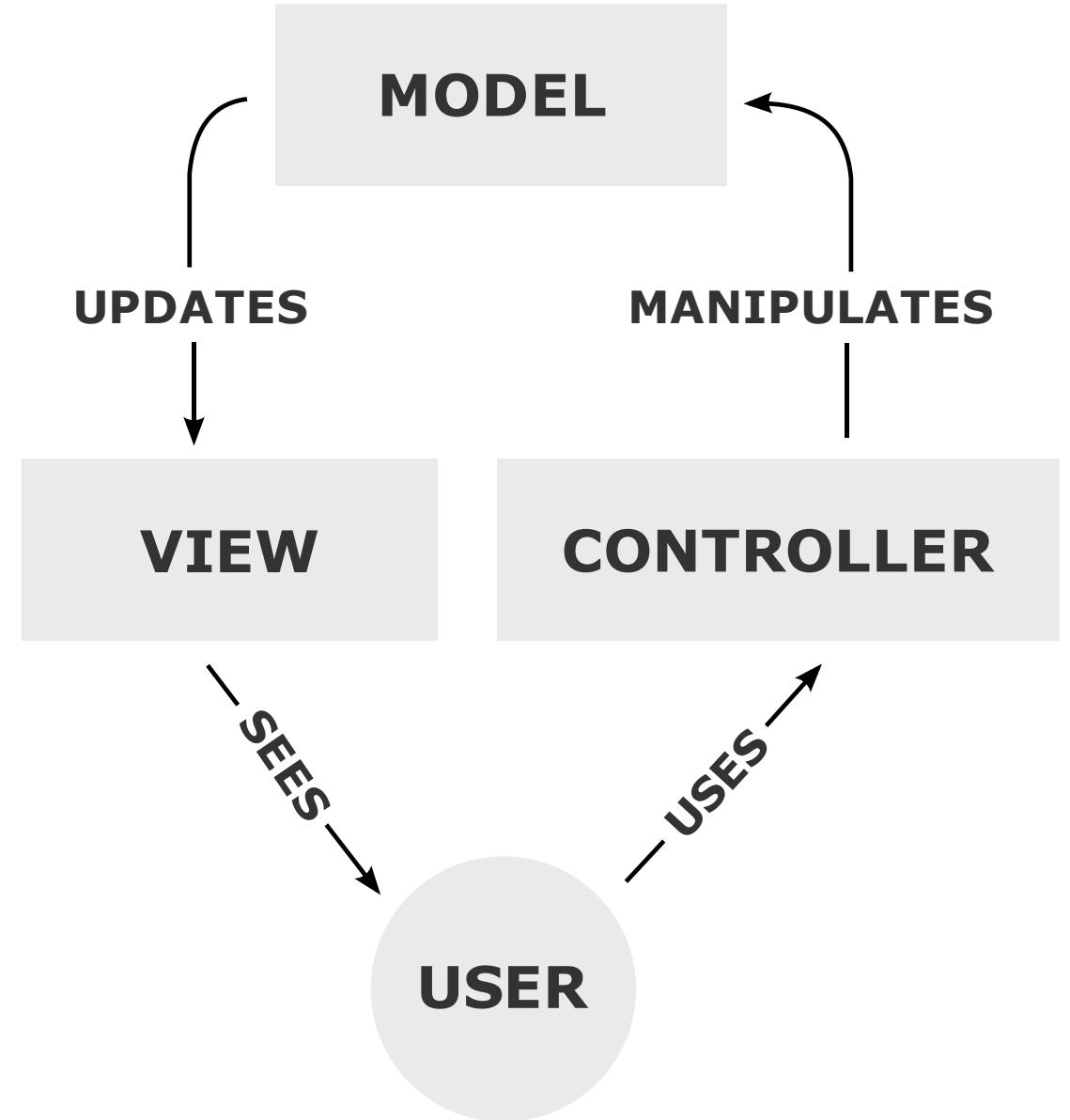
```
@import org.example.Page
@param Page page

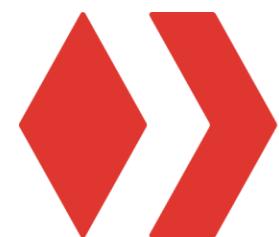
<head>
    @if(page.getDescription( ) != null)
        <meta name="description" content="${page.getDescription( )}">
    @endif
    <title>${page.getTitle( )}</title>
</head>
<body>
    <h1>${page.getTitle( )}</h1>
    <p>Welcome to my example page!</p>
</body>
```



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>
```





ViewComponent



spring® ViewComponent

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() = ViewContext(
7         "helloWorld" toProperty exampleService.getCoffee(),
8     )
9 }
```

```
1 // HomeViewComponent.html
2 <div th:text="${helloWorld}"></div>
```

```
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() = ViewContext(
6         "parameterViewComponent" toProperty parameterViewComponent.render("Hello World")
7     )
8 }
```



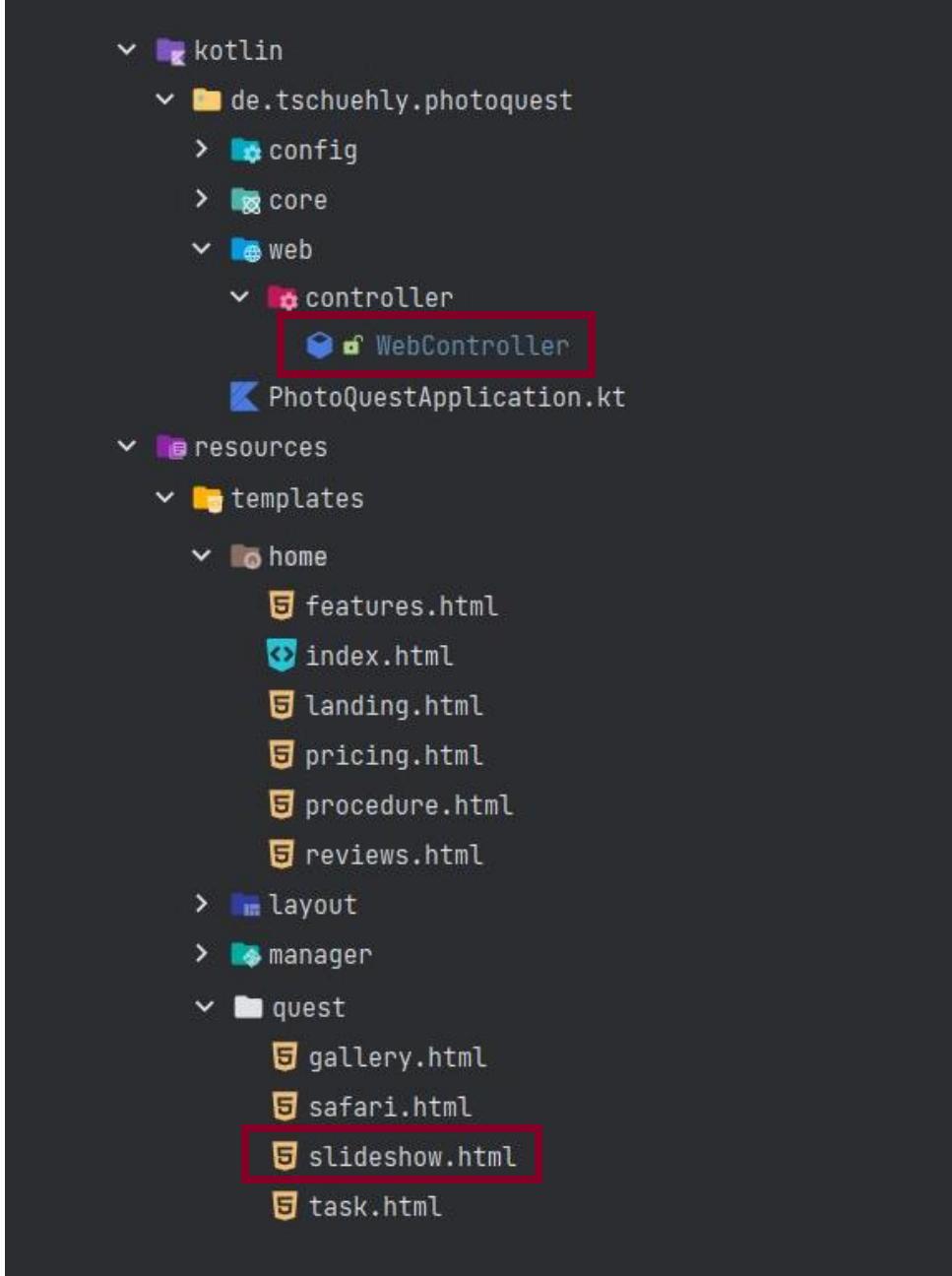
```
1 <div view:component="${parameterViewComponent}"></div>
```

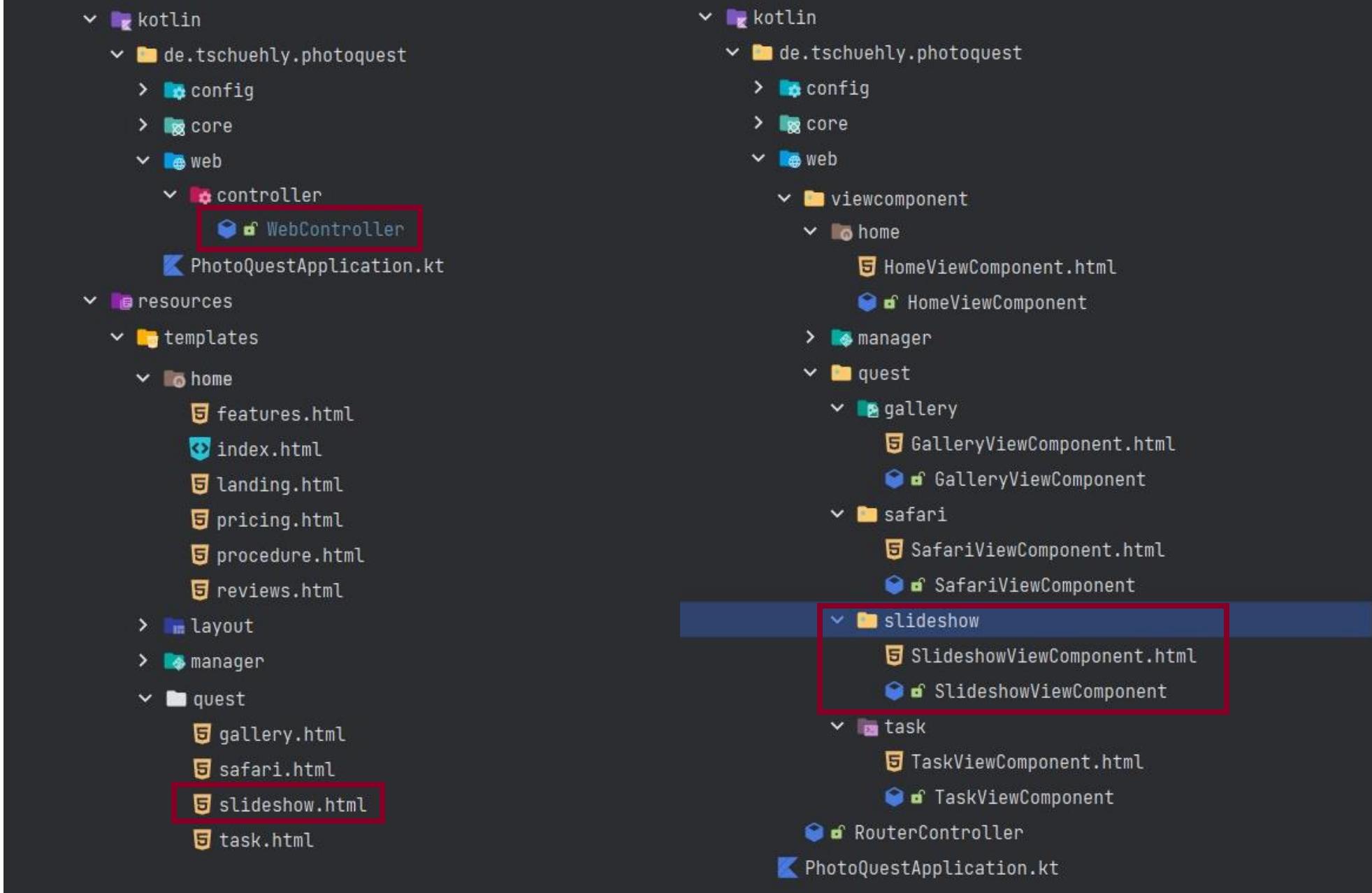


```
1 @import de.tschuehly.spring.viewcomponent.jte.ViewContext
2 @param ViewContext parameterViewComponent
3
4 <div>${parameterViewComponent}</div>
```



```
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**

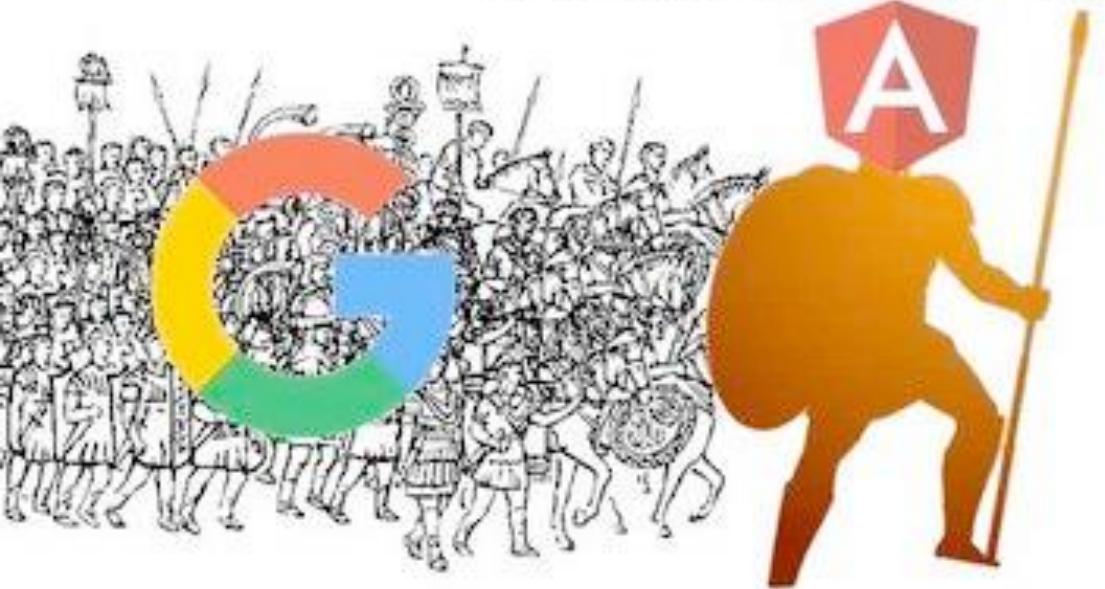
November 1995

[Report errata](#)

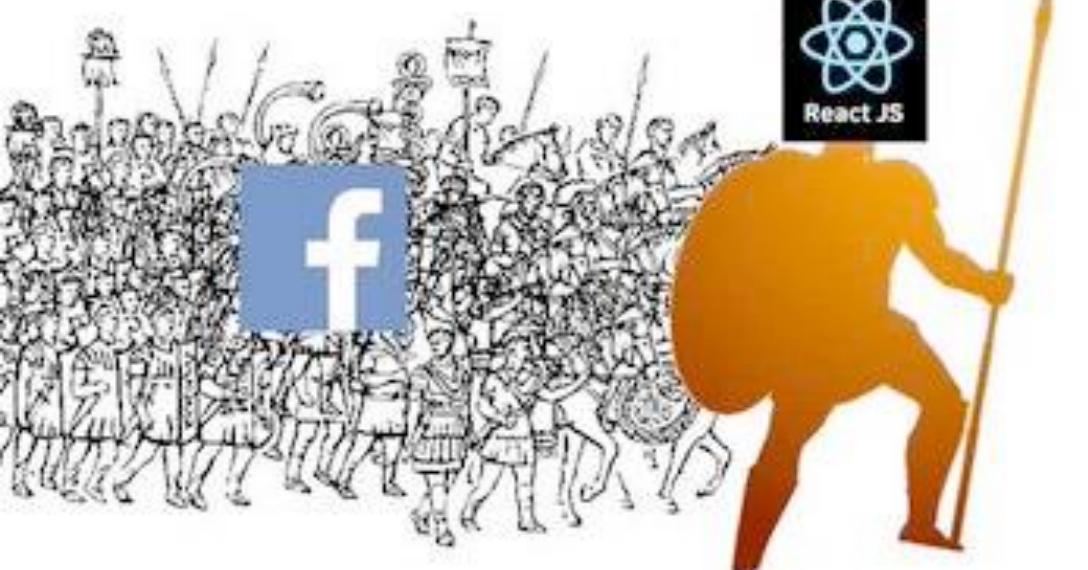
Obsoleted by [RFC 2854](#)

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>
```

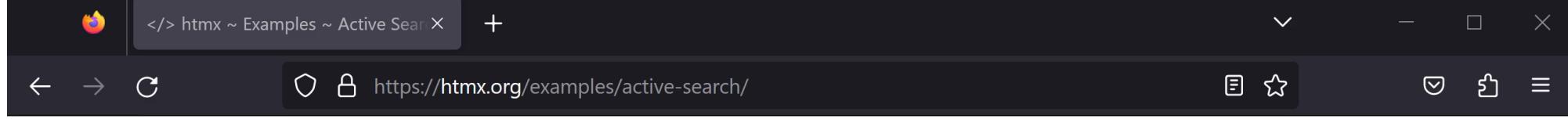


haiku

*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

Follow the folder

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 :

2 « 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.

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

Follow the folder

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 :

2 « 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.

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

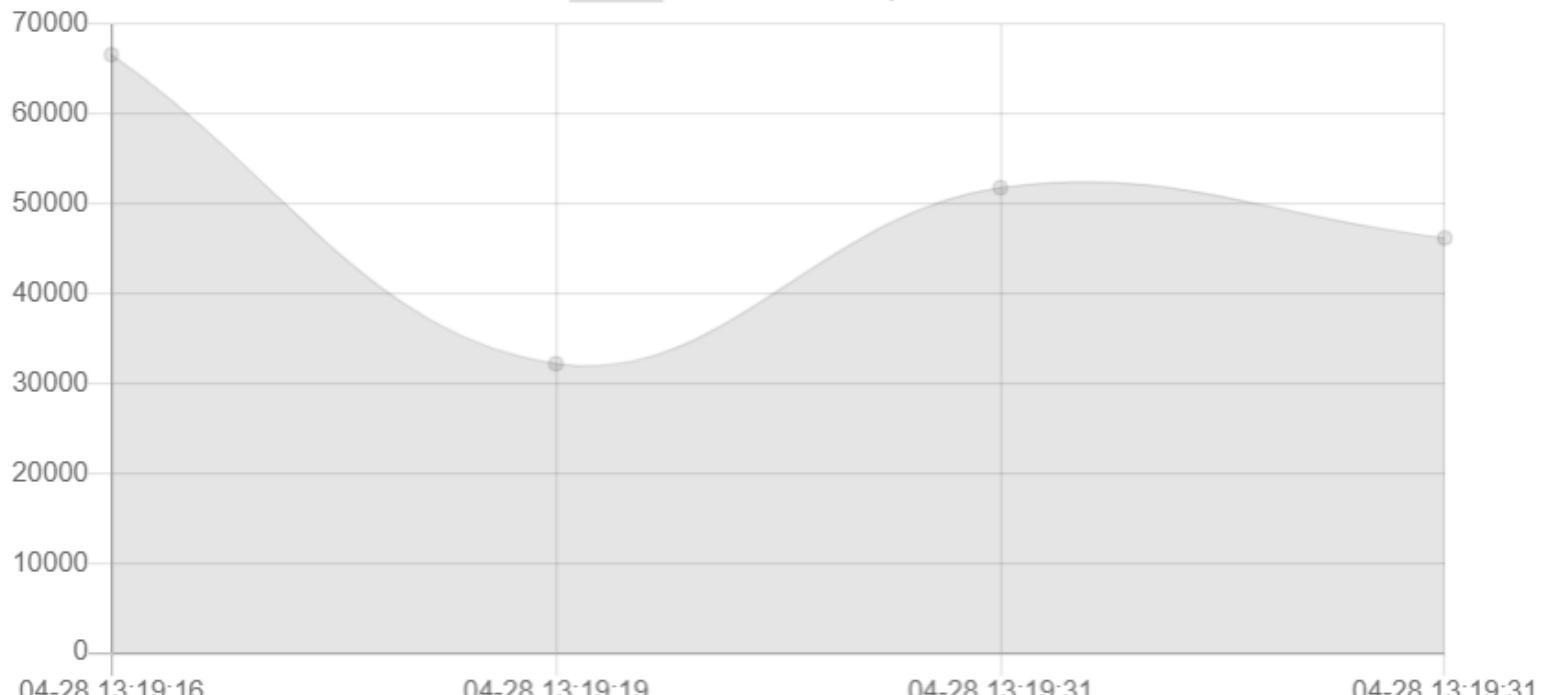
SidebarViewComponent

HeaderViewComponent

ChartViewComponent

TableViewComponent

SidebarViewComponent





```
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 }
```

```
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">
6 </button>
7 <div class="p-6 bg-white" id="lineChart">
8 </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

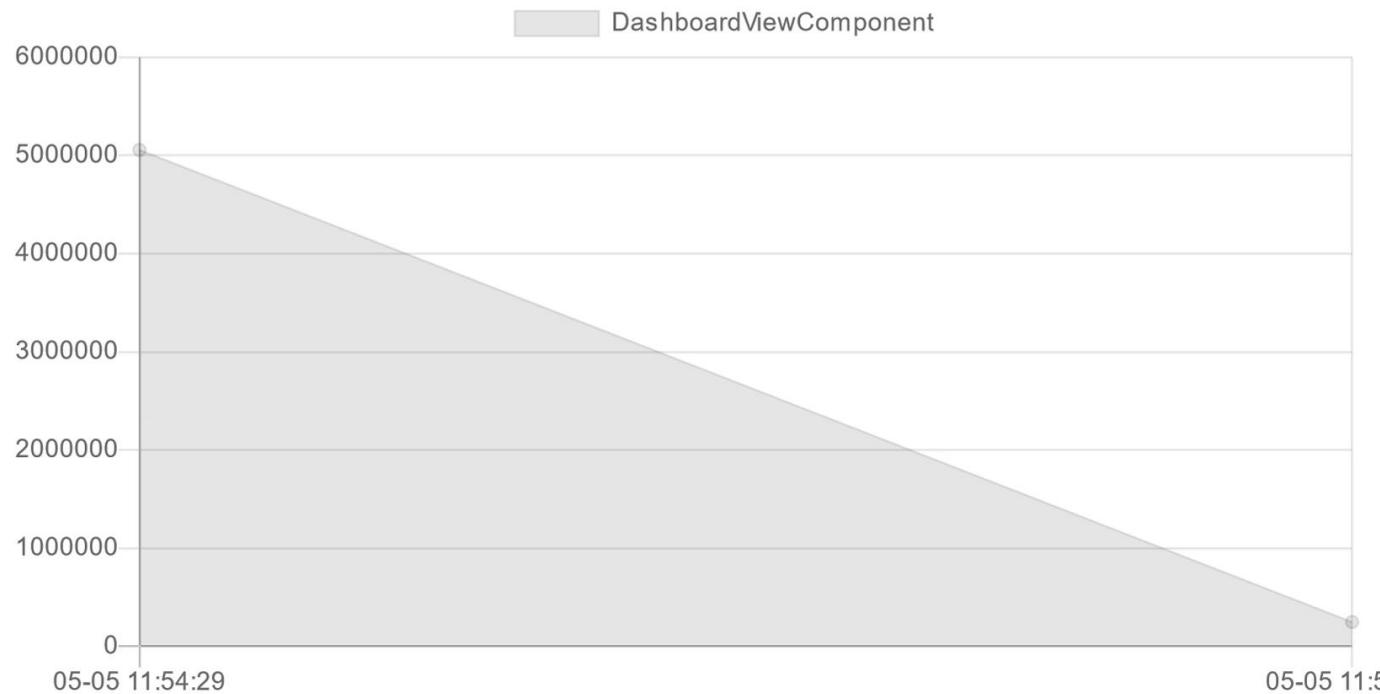
DashboardViewComponent

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

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     @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 }
```

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 }
```

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 ×

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

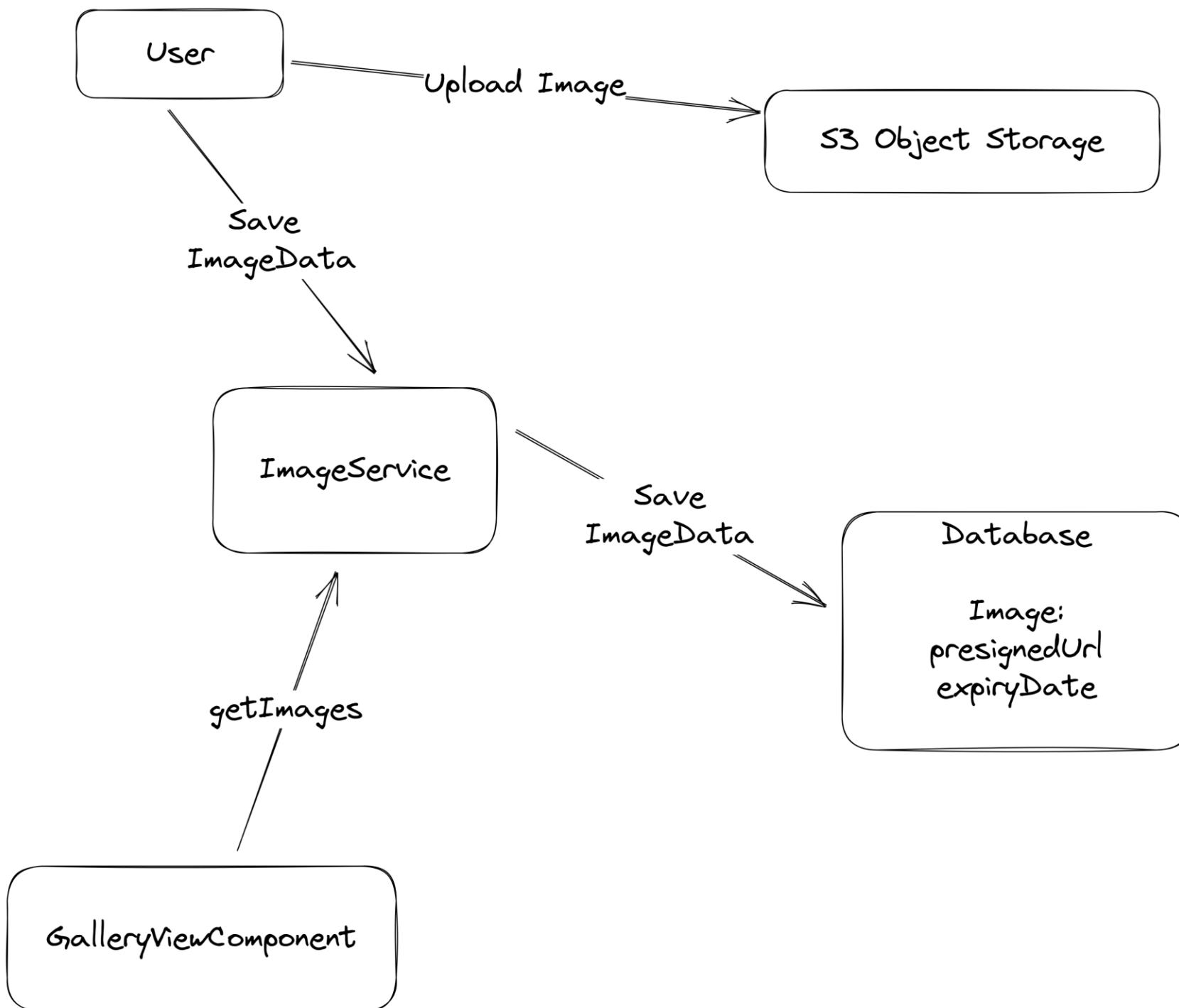
class PersonFormDTO(
    val id: Int?,
    val firstName: String,
    val lastName: String,
    val phoneNumber: String,
    val email: String,
    val birthDate: LocalDate
) {
    fun convertToPerson() = Person(
        this.id,
        this.firstName,
        this.lastName,
        this.phoneNumber,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

Bildergalerie





Bildergalerie





ImagePreviewComponent.html

```
1 <div th:id="|imagePreview-${image.id}|">
2   <div th:if="${isExpired}"
3     th:hx-get="/galerie/refresh-preview/${image.id}|"
4     th:hx-target="#imagePreview-' +${image.id}"
5     hx-trigger="revealed"
6     hx-swap="outerHTML" >
7   </div>
8   <div th:if="${!isExpired}" x-data="{showImage: false}">
9     
10  </div>
11 </div>
```



```
1 @ViewComponent
2 class ImagePreviewViewComponent(private val imageService: ImageService) {
3     fun render(imagePreview: ImagePreview, refreshImage: Boolean): ViewContext {
4         if (refreshImage == false) {
5             return ViewContext(
6                 "isExpired" toProperty imagePreview.downloadUrlExpiry.isExpired(),
7                 "image" toProperty imagePreview
8             )
9         }
10        val image = imagePreview.id?.let { gameId ->
11            imageService.getImageAndRefreshDownloadUrl(gameId)
12        } ?: throw Exception("imagePreview.id is null")
13        return ViewContext(
14            "isExpired" toProperty image.downloadUrlExpiry.isExpired(),
15            "image" toProperty image
16        )
17    }
18 }
```

Bildergalerie





```
@ViewComponent
class GalleryViewComponent(
    private val imageService: ImageService,
    private val imagePreviewViewComponent: ImagePreviewViewComponent
) {

    fun render(subDomain: String): ViewContext {
        val imageList = imageService.getImagesBySubdomain(subDomain)
        val imagePreviewList = imageList.map {
            imagePreviewViewComponent.render(ImagePreview(it), false)
        }
        return ViewContext(
            "imagePreviewList" toProperty imagePreviewList,
        )
    }
}
```



GalleryViewComponent.html

```
1 <th:block th:each="imagePreview : ${imagePreviewList}">
2   <div view:component="${imagePreview}">
3
4   </div>
5 </th:block>
```



```
1 @Controller
2 class QuestPageController(
3     private val galleryViewComponent: GalleryViewComponent,
4     private val imagePreviewViewComponent: ImagePreviewViewComponent,
5 ) {
6     @GetMapping("/galerie")
7     fun gallery(
8         request: HttpServletRequest,
9     ): ViewContext {
10         return galleryViewComponent.render(request.getSubDomain())
11     }
12
13     @GetMapping("/galerie/refresh-preview/{imageId}")
14     fun refreshGalleryImage(@PathVariable imageUrl: Long): ViewContext {
15         return imagePreviewViewComponent.render(
16             ImagePreview(imageId), true
17         )
18     }
19 }
```

Bildergalerie



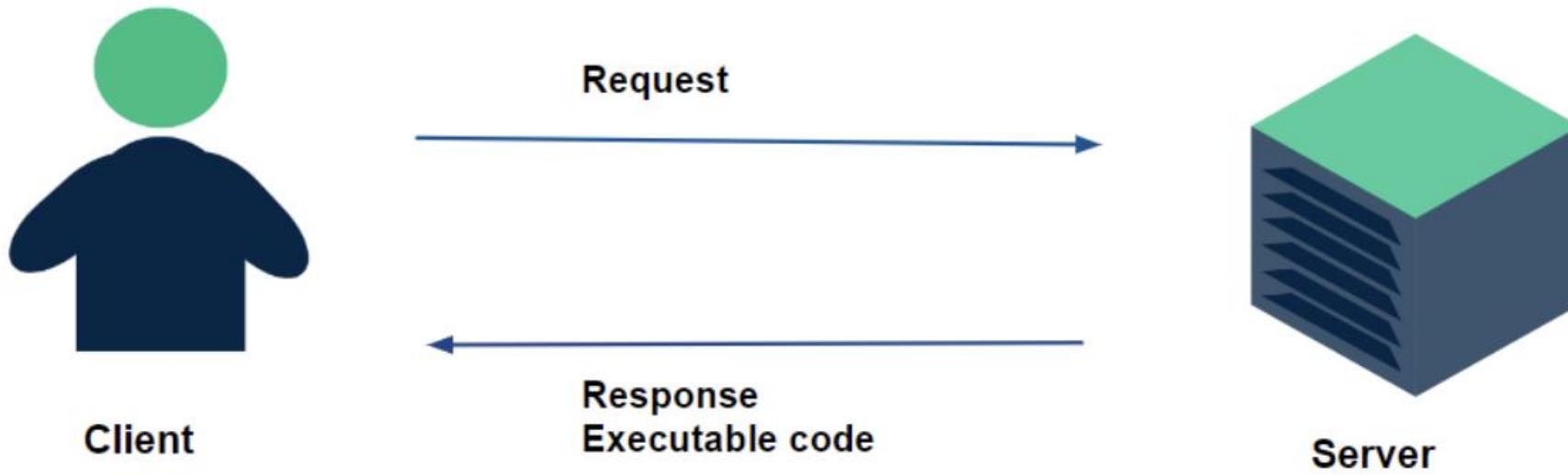
Elements Console Sources Network Performance Memory Application > 1

Preserve log Disable cache No throttling Invert Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm
 Has blocked cookies Blocked Requests 3rd-party requests
 Use large request rows Group by frame
 Show overview Capture screenshots

10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms

Recording network activity...
Perform a request or hit **Ctrl + R** to record the reload.
[Learn more](#)

Console Issues Network conditions





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

The screenshot shows a web browser window with the title "Tabbed View". The address bar displays "localhost:8080/tabbed". The main content area is titled "Alpine Tabs Content". It features a tab navigation with "AlpineTab1" and "AlpineTab2". The "AlpineTab1" tab is active, showing the following placeholder text:
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut
labore et dolore magna aliquyam erat, sed diam voluptua.

ADMIN

+ New Report

Dashboard

Tables

Forms

Tabbed Content

Calendar

localhost:8080/tabbed

Alpine Tabs Content

AlpineTab1 AlpineTab2

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>
```



+ 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.*





LayoutViewComponent.html

```
<body>
  <div view:component="${sidebarViewComponent}"></div>
  <div>
    <div view:component="${headerViewComponent}"></div>
    <div>
      <div view:component="${nestedViewComponent}"></div>
    </div>
    <footer>
      Template by <a>David Grzyb</a>.
    </footer>
  </div>
</body>
```



```
@ViewComponent
class LayoutViewComponent(
  private val sidebarViewComponent: SidebarViewComponent,
  private val headerViewComponent: HeaderViewComponent
) {
  fun render(
    activeTab: SidebarViewComponent.ActiveTab?,
    nestedViewComponent: ViewContext
) = ViewContext(
  "sidebarViewComponent" toProperty sidebarViewComponent.render(activeTab),
  "headerViewComponent" toProperty headerViewComponent.render(activeTab),
  "nestedViewComponent" toProperty nestedViewComponent
)
}
```



```
<th:block th:each="tab : ${activeTabsArray}">
    <a th:href="${tab.href}"
        th:classappend="${tab.equals(activeTab) ?
            'active-nav-link' :
            'opacity-75 hover:opacity-100'}">
        <i class="fas" th:classappend="${tab.icon}"></i>
        <span th:text="${tab.displayName}"></span>
    </a>
</th:block>
```

```
@ViewComponent
class SidebarViewComponent {
    fun render(activeTab: ActiveTab?) = ViewContext(
        "activeTab" toProperty activeTab,
        "activeTabsArray" toProperty ActiveTab.values()
    )
}

enum class ActiveTab(val displayName: String, val icon: String, val href: String) {
    Dashboard("Dashboard", "fa-tachometer-alt", dashboardHref),
    Tables("Tables", "fa-table", tablesHref),
    Forms("Forms", "fa-align-left", formsHref),
    TabbedContent("Tabbed Content", "fa-tablet-alt", tabbedHref),
    Calendar("Calendar", "fa-calendar", calendarHref),
}

companion object {
    const val dashboardHref = "/"
    const val tablesHref = "/tables"
    const val formsHref = "/forms"
    const val tabbedHref = "/tabbed"
    const val calendarHref = "/calendar"
}
```



```
<th:block th:each="tab : ${activeTabsArray}">
    <a th:href="${tab.href}"
        th:classappend="${tab.equals(activeTab) ?
            'active-nav-link' :
            'opacity-75 hover:opacity-100'}">
        <i class="fas" th:classappend="${tab.icon}"></i>
        <span th:text="${tab.displayName}"></span>
    </a>
</th:block>
```

```
@ViewComponent
class SidebarViewComponent {
    fun render(activeTab: ActiveTab?) = ViewContext(
        "activeTab" toProperty activeTab,
        "activeTabsArray" toProperty ActiveTab.values()
    )
}

enum class ActiveTab(val displayName: String, val icon: String, val href: String) {
    Dashboard("Dashboard", "fa-tachometer-alt", dashboardHref),
    Tables("Tables", "fa-table", tablesHref),
    Forms("Forms", "fa-align-left", formsHref),
    TabbedContent("Tabbed Content", "fa-tablet-alt", tabbedHref),
    Calendar("Calendar", "fa-calendar", calendarHref),
}

companion object {
    const val dashboardHref = "/"
    const val tablesHref = "/tables"
    const val formsHref = "/forms"
    const val tabbedHref = "/tabbed"
    const val calendarHref = "/calendar"
}
```



```
<th:block th:each="tab : ${activeTabsArray}">
    <a th:href="${tab.href}"
        th:classappend="${tab.equals(activeTab) ?
            'active-nav-link' :
            'opacity-75 hover:opacity-100'}">
        <i class="fas" th:classappend="${tab.icon}"></i>
        <span th:text="${tab.displayName}"></span>
    </a>
</th:block>
```

```
@ViewComponent
class SidebarViewComponent {
    fun render(activeTab: ActiveTab?) = ViewContext(
        "activeTab" toProperty activeTab,
        "activeTabsArray" toProperty ActiveTab.values()
    )
    enum class ActiveTab(val displayName: String, val icon: String, val href: String) {
        Dashboard("Dashboard", "fa-tachometer-alt", dashboardHref),
        Tables("Tables", "fa-table", tablesHref),
        Forms("Forms", "fa-align-left", formsHref),
        TabbedContent("Tabbed Content", "fa-tablet-alt", tabbedHref),
        Calendar("Calendar", "fa-calendar", calendarHref),
    }
    companion object {
        const val dashboardHref = "/"
        const val tablesHref = "/tables"
        const val formsHref = "/forms"
        const val tabbedHref = "/tabbed"
        const val calendarHref = "/calendar"
    }
}
```

```
fun render(  
    activeTab: SidebarViewComponent.ActiveTab?,  
    nestedViewComponent: ViewContext  
) = ViewContext(  
    "sidebarViewComponent" toProperty sidebarViewComponent.render(activeTab),  
    "headerViewComponent" toProperty headerViewComponent.render(activeTab),  
    "nestedViewComponent" toProperty nestedViewComponent  
)
```



```
@Controller  
class PageController(  
    private val dashboardViewComponent: DashboardViewComponent,  
    private val layoutViewComponent: LayoutViewComponent,  
) {  
    @GetMapping(SidebarViewComponent.dashboardHref)  
    fun home() = layoutViewComponent.render(  
        ActiveTab.Dashboard,  
        dashboardViewComponent.render()  
    )  
}
```



spring® ViewComponent



Alpine.js



ActionViewComponent.html

```
<div>
  <button view:action="countUp">
    Default ViewAction [GET]
  </button>
  <h3 th:text="${counter}"></h3>
</div>
```



```
@ViewComponent
class ActionViewComponent {
  var counter = 0
  fun render() = ViewContext(
    "counter" toProperty counter,
  )
  @GetViewAction
  fun countUp(): ViewContext {
    counter += 1
    return render()
  }
}
```



```
<div>
  <button view:action="countUp">
    Default ViewAction [GET]
  </button>
  <h3 th:text="${counter}"></h3>
</div>
```



```
<div id="actionviewcomponent">
  <h2>ViewAction Get CountUp</h2>
  <button hx-get="/actionviewcomponent/countup"
          hx-target="#actionviewcomponent"
          hx-swap="outerHTML">
    Default ViewAction [GET]
  </button>
</div>
```



```
@ViewComponent
class ActionViewComponent(
    val exampleService: ExampleService
) {

    fun render() = ViewContext(
        "itemList" toProperty exampleService.someData,
    )

    @PostViewAction("/customPath/addItemAction")
    fun addItem(actionFormDTO: ActionFormDTO): ViewContext{
        exampleService.addItemToList(actionFormDTO.item)
        return render()
    }

    @DeleteViewAction
    fun deleteItem(id: Int): ViewContext {
        exampleService.deleteItem(id)
        return render()
    }
}
```



ActionViewComponent.html

```
<form view:action="addItem">
    <input type="text" name="item">
    <button type="submit">Save Item</button>
</form>

<div th:each="item : ${itemList}">
    <span th:text="${item.value}"></span>
    <button th:view:action="|deleteItem?id=${item.key}|">
        Delete Item [[${item.key}]]
    </button>
</div>
```



```
@ViewComponent
class ActionViewComponent(
    val exampleService: ExampleService
) {

    fun render() = ViewContext(
        "itemList" toProperty exampleService.someData,
    )

    @PostViewAction("/customPath/addItemAction")
    fun addItem(actionFormDTO: ActionFormDTO): ViewContext{
        exampleService.addItemToList(actionFormDTO.item)
        return render()
    }

    @DeleteViewAction
    fun deleteItem(id: Int): ViewContext {
        exampleService.deleteItem(id)
        return render()
    }
}
```



ActionViewComponent.html

```
<form view:action="addItem">
    <input type="text" name="item">
    <button type="submit">Save Item</button>
</form>

<div th:each="item : ${itemList}">
    <span th:text="${item.value}"></span>
    <button th:view:action="|deleteItem?id=${item.key}|">
        Delete Item [[${item.key}]]
    </button>
</div>
```



```
@ViewComponent
class ActionViewComponent(
    val exampleService: ExampleService
) {

    fun render() = ViewContext(
        "itemList" toProperty exampleService.someData,
    )

    @PostViewAction("/customPath/addItemAction")
    fun addItem(actionFormDTO: ActionFormDTO): ViewContext{
        exampleService.addItemToList(actionFormDTO.item)
        return render()
    }

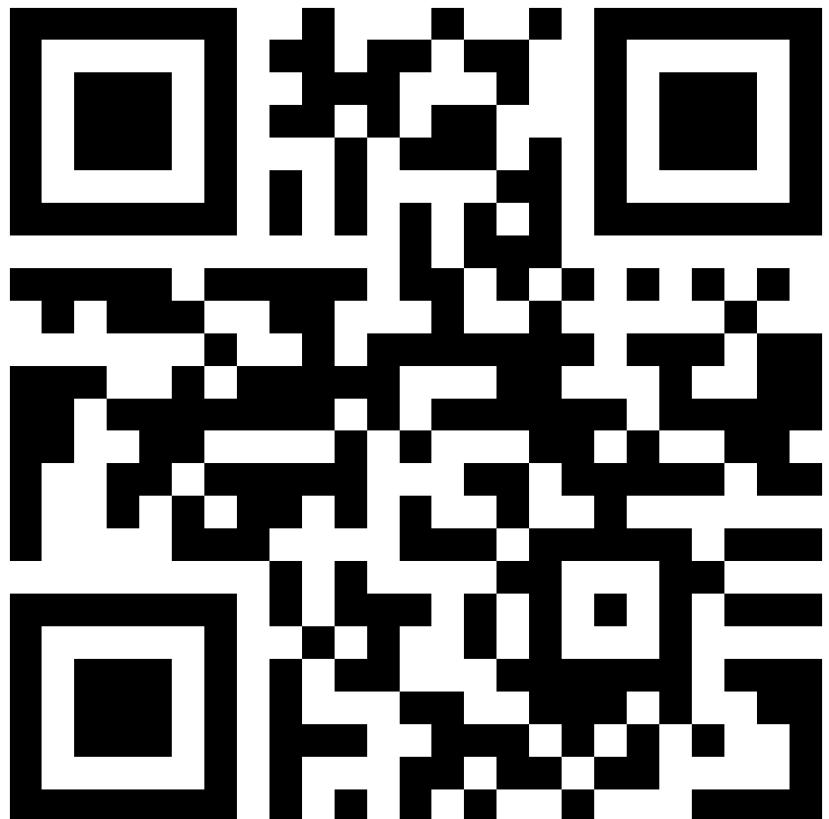
    @DeleteViewAction
    fun deleteItem(id: Int): ViewContext {
        exampleService.deleteItem(id)
        return render()
    }
}
```



ActionViewComponent.html

```
<form view:action="addItem">
    <input type="text" name="item">
    <button type="submit">Save Item</button>
</form>
<div th:each="item : ${itemList}">
    <span th:text="${item.value}"></span>
    <button th:view:action="|deleteItem?id=${item.key}|">
        Delete Item [[${item.key}]]
    </button>
</div>
```

fullstack.tschuehly.de





**I WANT YOU
TO CONTRIBUTE
AND CONNECT**

TSCHUEHLY.DE

supabase Security spring® Boot Starter

github.com/tschuehly/supabase-security-spring-boot-starter



```
<th:block th:each="tab : ${activeTabsArray}">
    <a th:href="${tab.href}"
        th:classappend="${tab.equals(activeTab) ?
            'active-nav-link' :
            'opacity-75 hover:opacity-100'}">
        <i class="fas" th:classappend="${tab.icon}"></i>
        <span th:text="${tab.displayName}"></span> ●●●
    </a>
</th:block>
```

```
@ViewComponent
class SidebarViewComponent {
    fun render(activeTab: ActiveTab?) = ViewContext(
        "activeTab" toProperty activeTab,
        "activeTabsArray" toProperty ActiveTab.values()
    )
    enum class ActiveTab(val displayName: String, val icon: String, val href: String) {
        Dashboard("Dashboard", "fa-tachometer-alt", dashboardHref),
        Tables("Tables", "fa-table", tablesHref),
        Forms("Forms", "fa-align-left", formsHref),
        TabbedContent("Tabbed Content", "fa-tablet-alt", tabbedHref),
        Calendar("Calendar", "fa-calendar", calendarHref),
    }
    companion object {
        const val dashboardHref = "/"
        const val tablesHref = "/tables"
        const val formsHref = "/forms"
        const val tabbedHref = "/tabbed"
        const val calendarHref = "/calendar"
    }
}
```