~~LIVEVIEW~~: BEST PRACTICES FOR BUILDING REAL-TIME INTERACTIVE INTERFACES

herminiotorres
herminiotorres

simplebet™

AGENDA

1. Foundation
2. Single Source of Truth (SSOT)
3. Async updates a LiveComponent
4. Context
5. Scoped CSS

DISCLAIMER

Context and Scoped CSS using Surface-UI

FOUNDATION

FOUNDATION

1. Loading *twice* on mount/3 - (demo | code)
2. Split into *two phases* before and after the socket is *connected?* - (demo | code)
3. *Prevent blocking* the LiveView process until it *die* - (demo | code)
4. Using *assign_async* - (demo | code)
5. Using *start_async* - (demo | code)

SINGLE SOURCE OF TRUTH (SSOT)

SSOT

- *Avoiding* updates for those who are not the *data owner*.
- It creates *data inconsistencies* when there are other components that depend on.
- It should *always request* who is the *owner* to perform updates.

SSOT

- Assign to the new value in the socket into a Component. demo
- code

SSOT

- Sending an internal message to the socket to assign the new value. demo
- code

ASYNC UPDATE/2 A LIVECOMPONENT

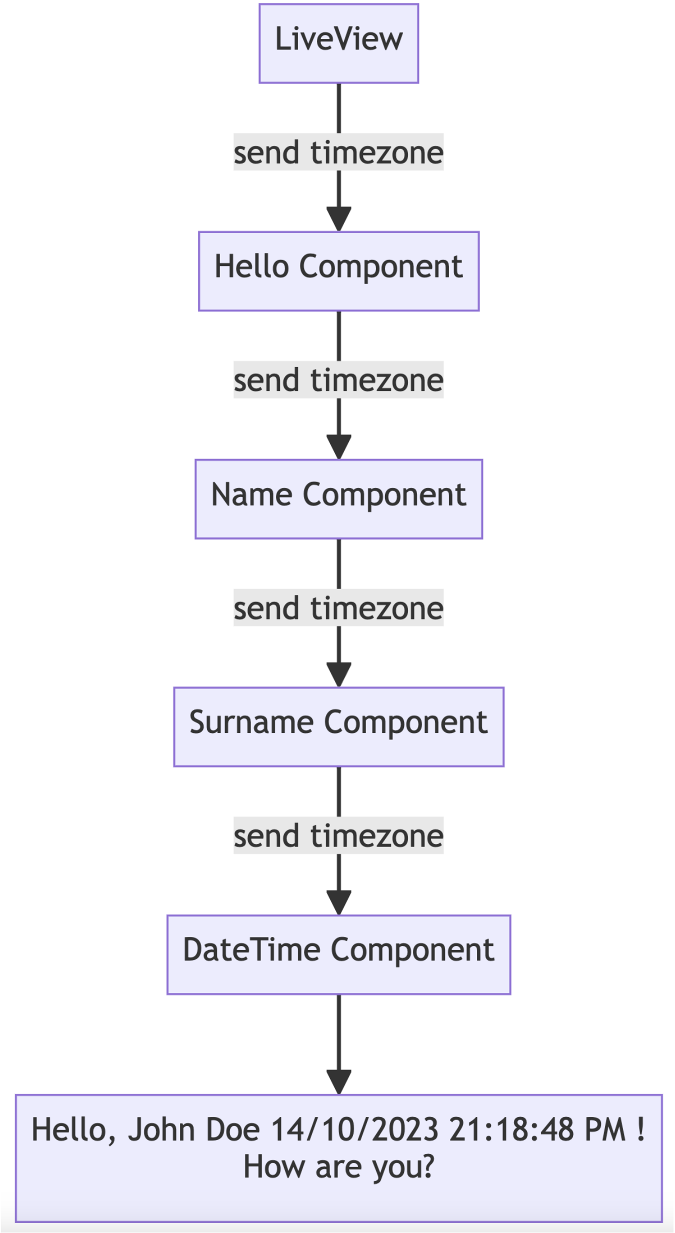
SEND_UPDATE/2

- Using *send_update/2* to update the component.
- It is useful for updating a component that *manages its own state*.
- *Leaking internal details* of the state is problematic.
- *Any change* in the shape of the state might *break* the code.
- (demo | code | code)

SEND_UPDATE/2

- The internal state should only be performed from *the component's public API*.
- (demo | code | code)

CONTEXT


LiveView
send timezone
Hello Component
send timezone
Name Component
send timezone
Surname Component
send timezone
DateTime Component
Hello, John Doe 14/10/2023 21:18:48 PM !
How are you?

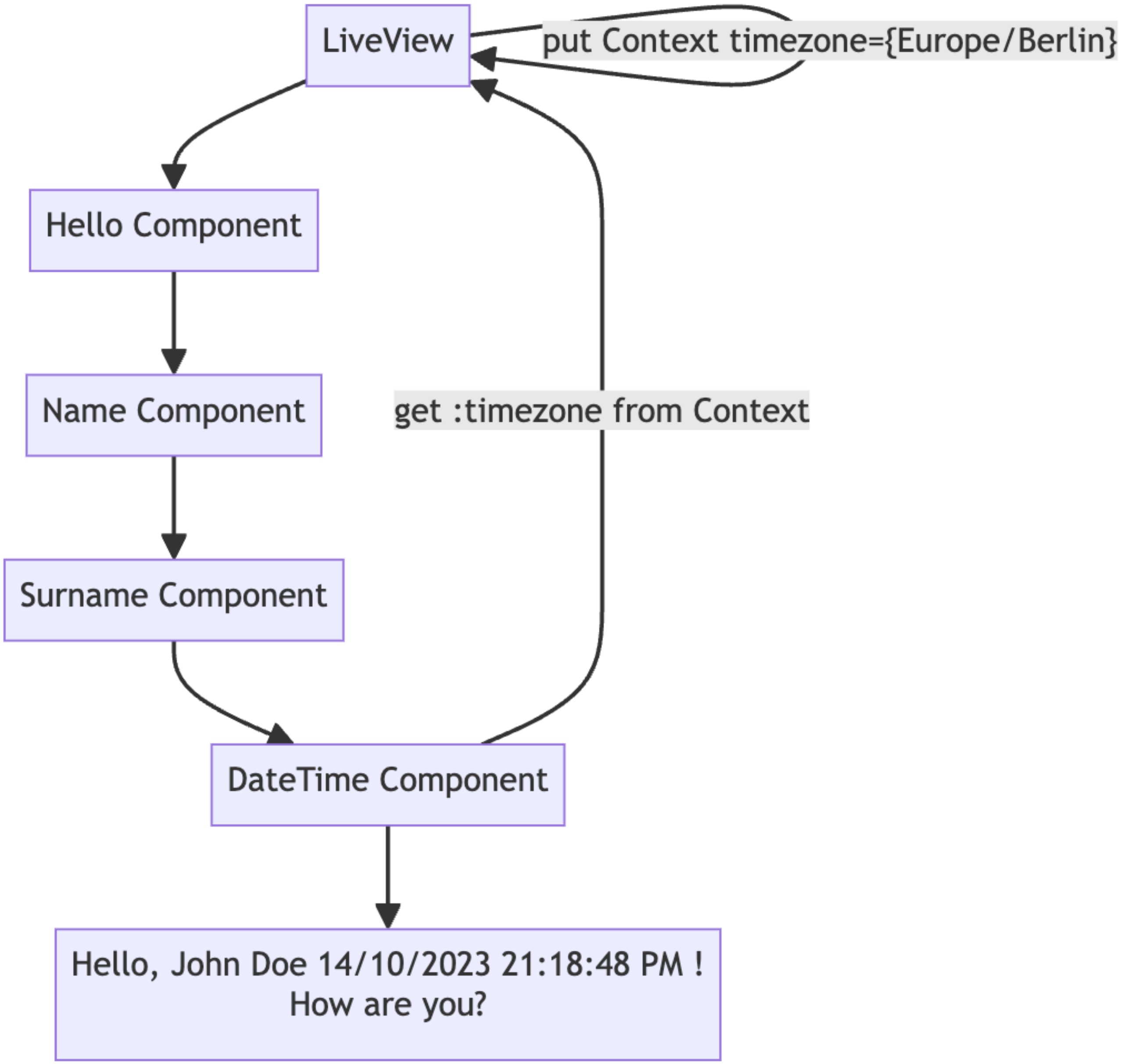
THE CONCEPT

Avoid Prop drilling.

It is the process of passing props from one to another, even when intermediate components do not use or need those props.

CONTEXT

- Select timezone to shift to some zone - demo
- Put timezone in the context - code
- Get timezone from the context - code


LiveView
put Context timezone={Europe/Berlin}
Hello Component
Name Component
get :timezone from Context
Surname Component
DateTime Component
Hello, John Doe 14/10/2023 21:18:48 PM !
How are you?

SCOPED CSS

THE CONCEPT

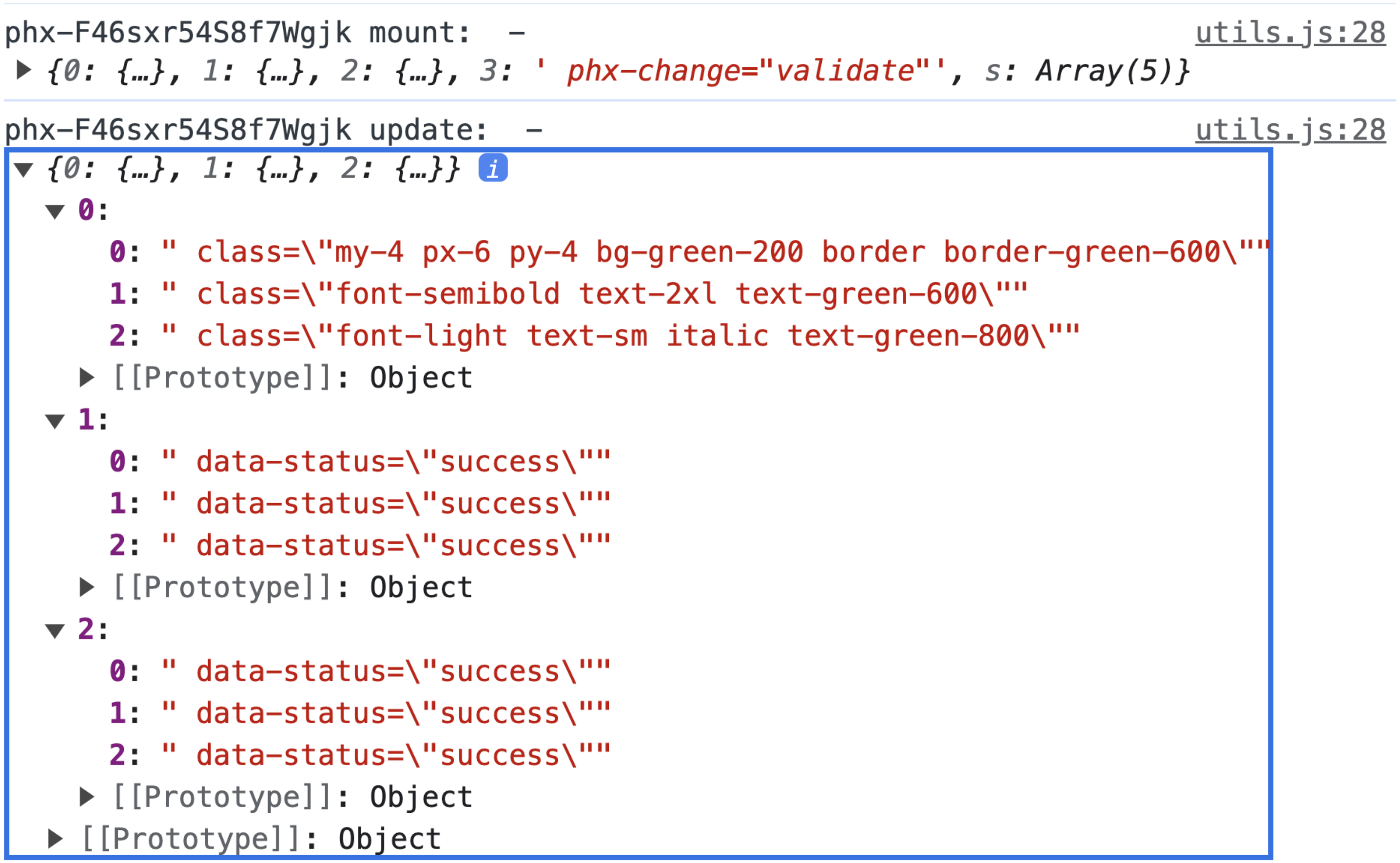
It is to prevent CSS rules defined within a component from affecting elements outside that component. This helps prevent style conflicts.

SCOPED CSS

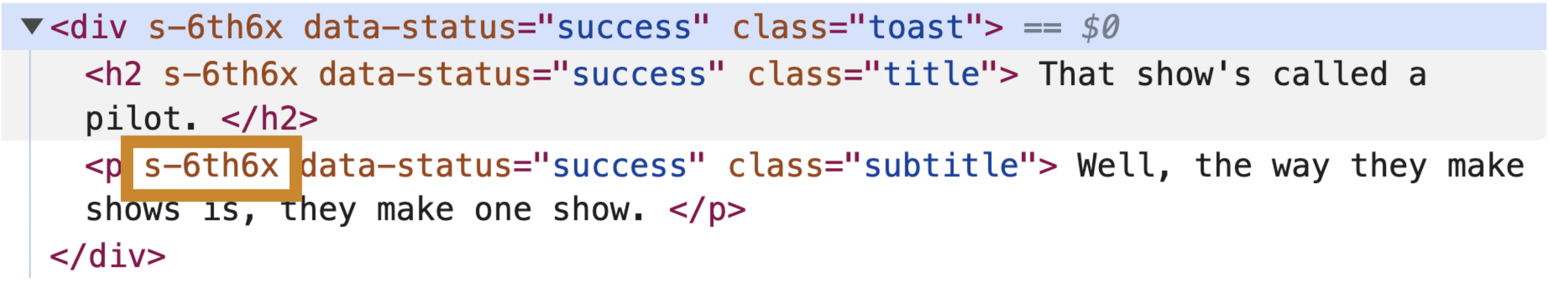
- Toast message in LiveView - demo
- Apply CSS Style in LiveView - code

SCOPED CSS

- Toast message in Surface - demo
- Apply CSS Style in Surface - code
- Scope CSS Style in Surface - code
- Scope CSS Style in Surface - code



```
phx-F46sxr54S8f7Wgjk mount:  -                                        utils.js:28
▶ {0: {…}, 1: {…}, 2: {…}, 3: ' phx-change="validate"', s: Array(5)}
phx-F46sxr54S8f7Wgjk update:  -                                       utils.js:28
▼ {0: {…}, 1: {…}, 2: {…}} i
  ▼ 0:
      0: " class=\"my-4 px-6 py-4 bg-green-200 border border-green-600\""
      1: " class=\"font-semibold text-2xl text-green-600\""
      2: " class=\"font-light text-sm italic text-green-800\""
    ▶ [[Prototype]]: Object
  ▼ 1:
      0: " data-status=\"success\""
      1: " data-status=\"success\""
      2: " data-status=\"success\""
    ▶ [[Prototype]]: Object
  ▼ 2:
      0: " data-status=\"success\""
      1: " data-status=\"success\""
      2: " data-status=\"success\""
    ▶ [[Prototype]]: Object
  ▶ [[Prototype]]: Object
```



```
▼<div s-6th6x data-status="success" class="toast"> == $0
   <h2 s-6th6x data-status="success" class="title"> That show's called a
   pilot. </h2>
   <p s-6th6x data-status="success" class="subtitle"> Well, the way they make
   shows is, they make one show. </p>
 </div>
```

OBRIGADO
DANKE
THANK YOU

herminiotorres
herminiotorres

simplebet™