

WordPress Interactivity API

for real life?!

ENQUEUE 2025

Can the **Interactivity API**
compete with  **React**?

What you'll learn today



When to pick the **Interactivity API** over other JavaScript frameworks



How to use the Interactivity API **outside of blocks**

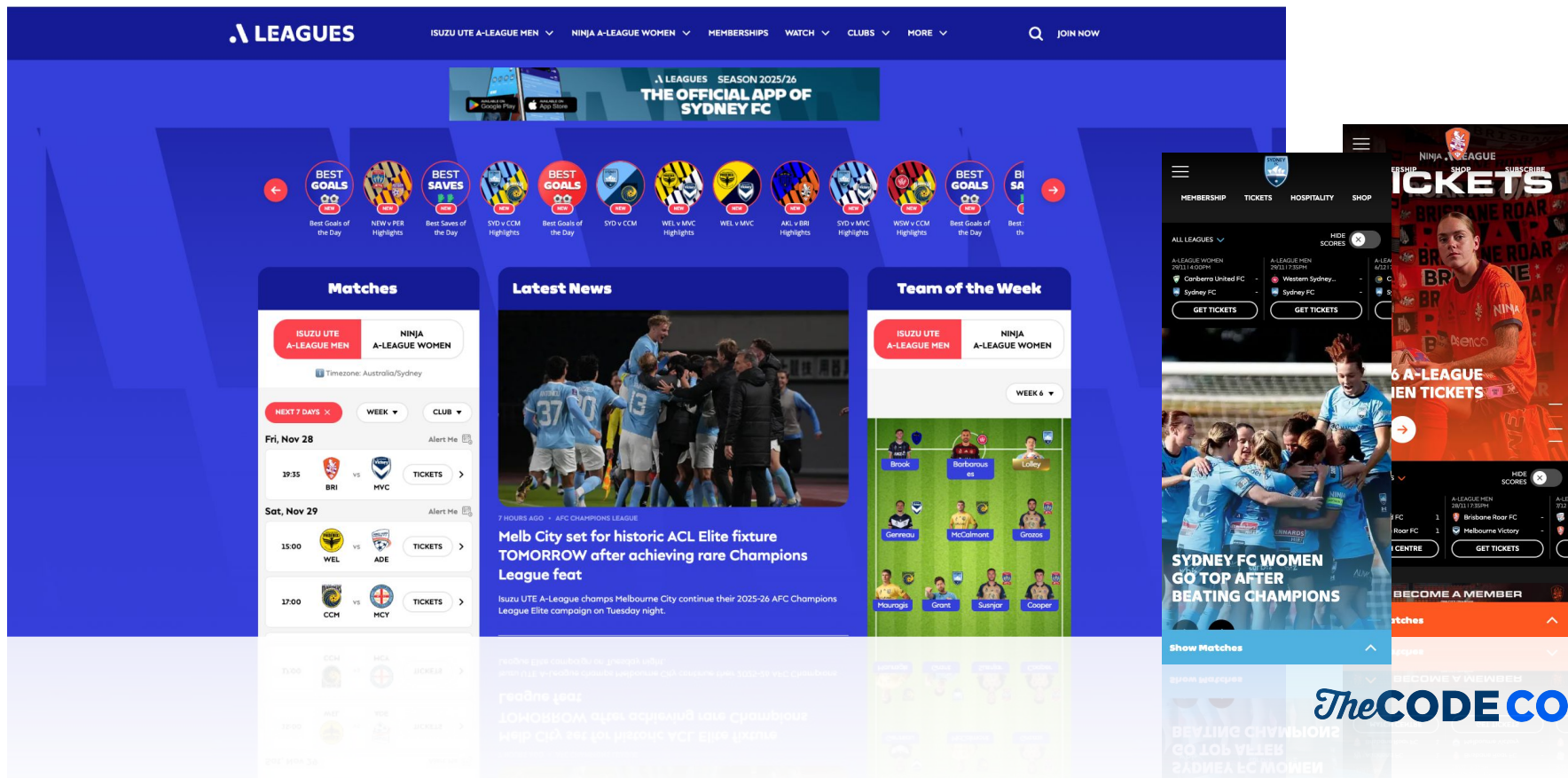


Handling **large** (+ live) data sets



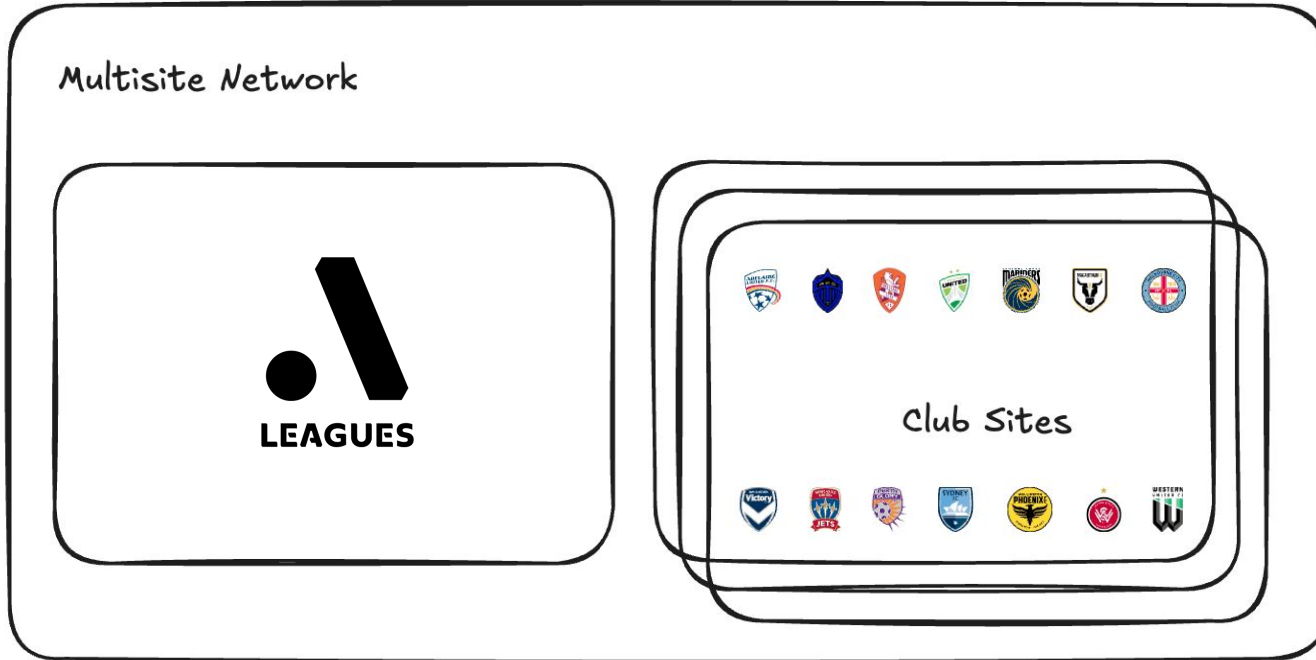
Real lessons on features such as script modules, state/context, templating, etc.

A-Leagues



TheCODECO

A-Leagues: Platform Overview



A-Leagues: Existing Frontend Stack

JS

Vanilla JS

- Tabs
- Main and Mobile Menus
- News Category Filters
- Modals
- And more...



React

- Fixtures and Results
- Match Centre
- Fixtures bar
- Article Match card embeds

Why consider the Interactivity API?

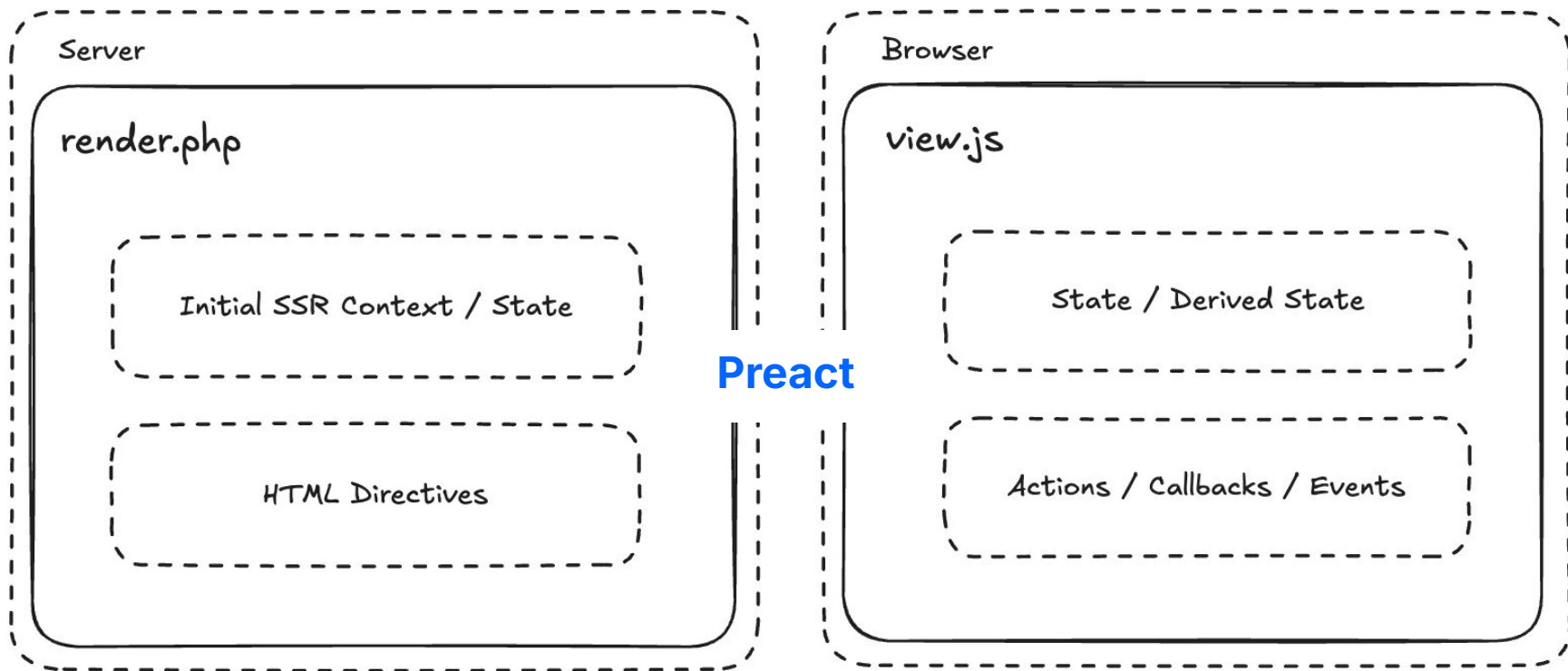
- ✓ **Lightweight** (~16KB on-page)
- ✓ **Native** to WordPress
- ✓ **Server-side** friendly
- ✓ Impressive **range**, small to large components

Quick Primer: Directives

```
<span
  data-wp-context='{ "isSelected": false }'
  data-wp-on--click="actions.toggleSelection"
  data-wp-class--selected="context.isSelected"
>
  Option 1
</span>
```

Basic example

Quick Primer: What is the Interactivity API?



Level 1: Subscribe Form



Email Subscribe Form

SUBSCRIBE TO OUR NEWSLETTER

Keep up with our latest news and events. Subscribe to our newsletter.

Enter your email address *

- ☐ I agree to the [Privacy Policy](#) of the A-Leagues. *
- ☐ I agree to receive marketing communications from the A-Leagues. *

SUBSCRIBE

2025GUIDE

The Assignment

Email Subscribe Form



Inline **validation**



SDK-based submission



Error/success **states**



Simple interactive flow

SUBSCRIBE TO OUR NEWSLETTER

Keep up with our latest news and events. Subscribe to our newsletter.

hello@example.com

☐ I agree to the [Privacy Policy](#) of the A-Leagues. *

☐ I agree to receive marketing communications from the A-Leagues. *

SUBSCRIBE

Please complete the required fields.

SUBSCRIBE TO OUR NEWSLETTER

Keep up with our latest news and events. Subscribe to our newsletter.

Thank you for subscribing to our newsletter!

Why Interactivity API Over Alpine/Vanilla JS?



Vanilla JS

Too imperative/verbose

Alpine *

Not WP-native

Interactivity API

Faster / SSR / WP-native

Subscribe Form: Code Structure

```
<form data-wp-on--submit="actions.submitForm">
  <input data-wp-bind--value="context.emailValue" />
  <p data-wp-bind--hidden="!state.isError">Error message</p>
</form>
```

partial.php

```
const { state, actions } = store("subscribe-form", {
  actions: {
    *submitForm() {
      yield submitEmailForm(value); // Via SDK.
    },
  },
});
```

index.js

[illegible]

The Fix: Script Modules - Enqueueing

Enqueuing Script Modules

```
// Old method.  
wp_enqueue_script(  
    'subscribe-form',  
    get_theme_file_uri('/build/js/subscribe.js'),  
    [ 'wp-interactivity' ] // ❌ Can't do this :(  
);
```

```
// New method.  
wp_enqueue_script_module(  
    'subscribe-form',  
    get_theme_file_uri('/build/js/subscribe.js'),  
    [ 'wp-interactivity' ] // ✅ Can do this!  
);
```

enqueues.php

The Fix: Script Modules - Webpack

```
const defaultConfig = require('@wordpress/scripts/config/webpack.config');
```

webpack.config.js

```
const baseConfig = defaultConfig[0];
```

```
const modulesConfig = defaultConfig[1];
```

```
baseConfig.entry = {  
  ...baseConfig.entry(),  
  scripts: './source/js/scripts.js',  
};
```

```
modulesConfig.entry = {  
  ...modulesConfig.entry(),  
  email_subscribe_form: './source/js/email-subscribe-form/index.js',  
};
```

```
module.exports = [baseConfig, modulesConfig];
```

Moving to Script Modules

- ✓ Add `--experimental-modules` flag to wp-scripts commands
- ✓ Update webpack config to correctly bundle your JavaScript
- ✓ Replace `wp_enqueue_script` with `wp_enqueue_script_module`

Challenge #2

wp_interactivity_state rendering method

SSR state rendered with wp_print_inline_script_tag()

```
<link rel="modulepreload" href="interactivity.js"  
      id="@wordpress/interactivity-js-modulepreload">  
  
<script type="application/json"  
      id="wp-script-module-data-@wordpress/interactivity">  
  {"state":"matches-slider":{"selectedLeague":"a-league-men", ...}}  
</script>
```

html

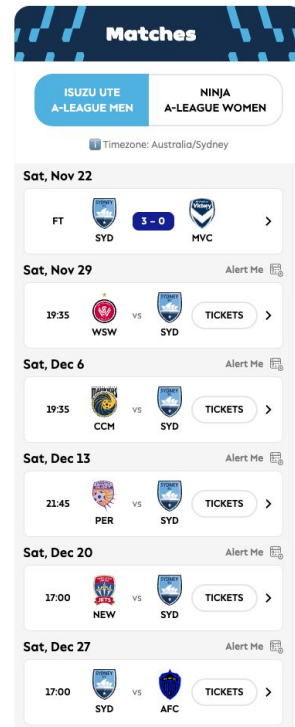
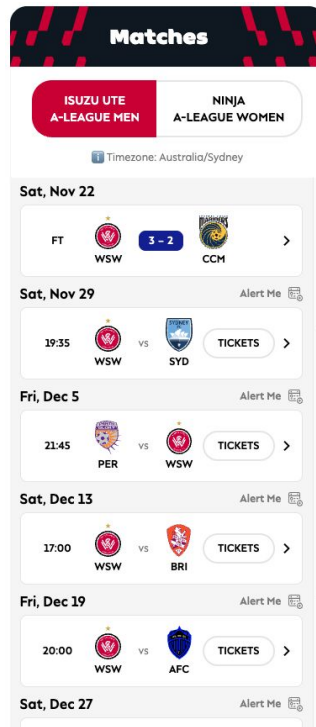
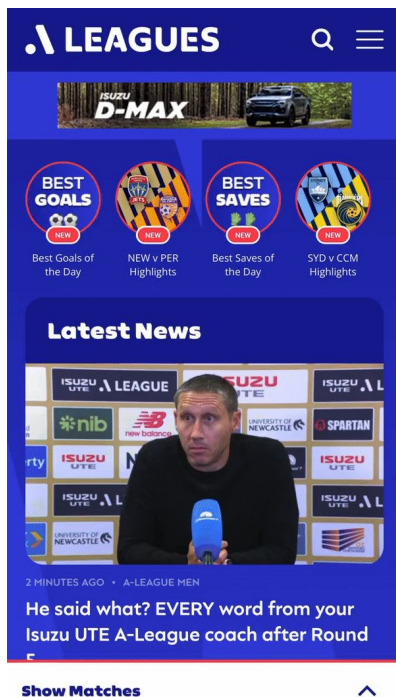
Subscribe Form Takeaways

- ✓ **Great** for small components
- ✓ Think **declaratively**
- ✓ Use blocks or **Script Modules** with Interactivity API

Level 2: Matches Slider



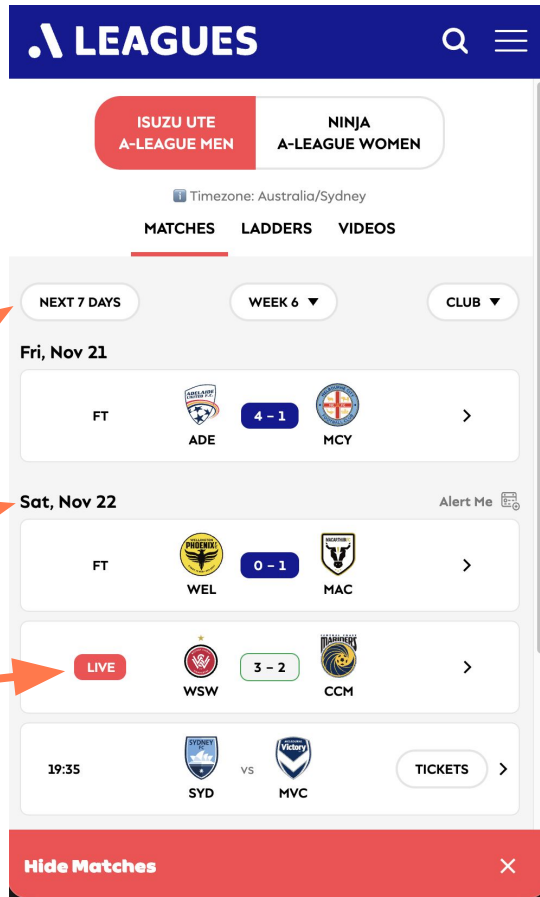
Matches Slider



The Assignment

Matches Slider

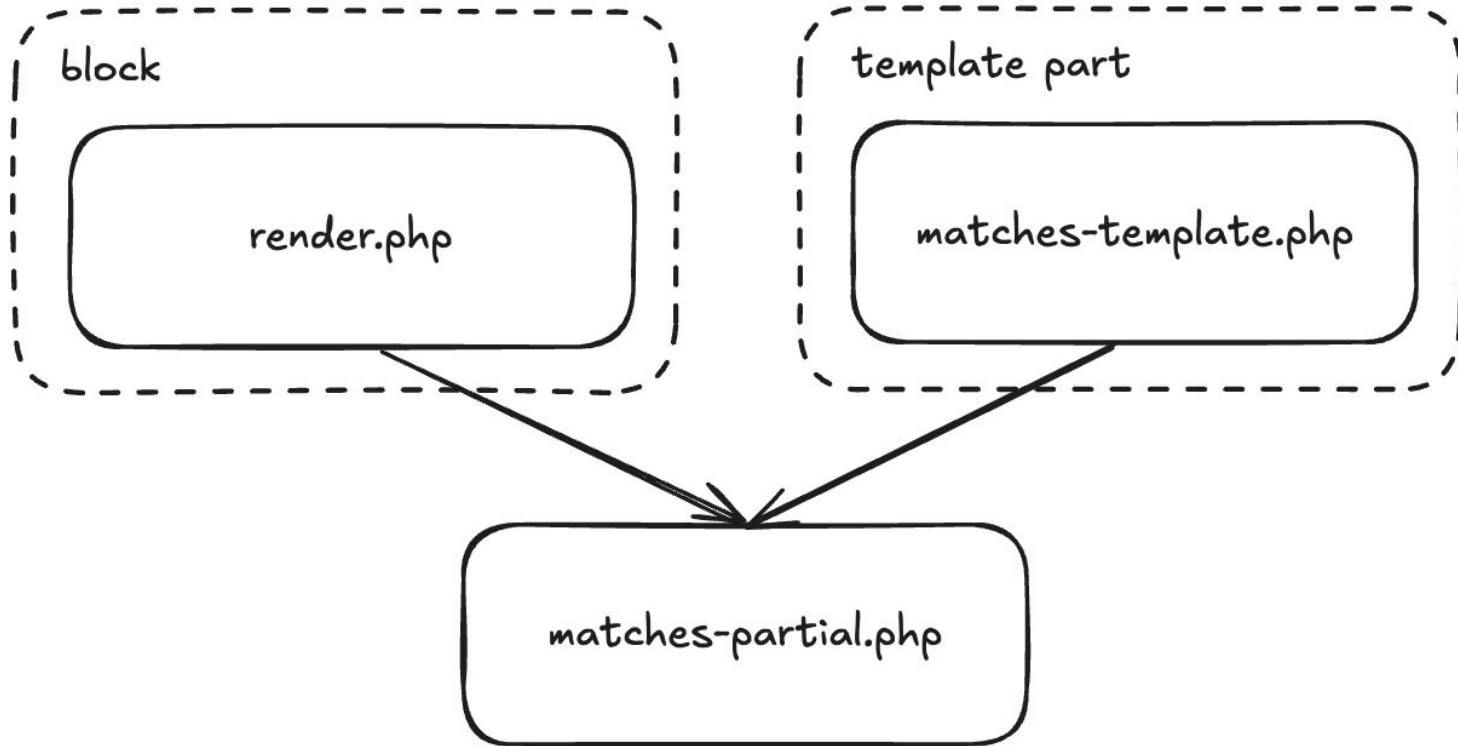
- ✓ Loads on **almost every** page
- ✓ Fetches matches via **API**
- ✓ **Filters** per competition, club, week, etc.
- ✓ **Nested** match groupings by day
- ✓ **Live updates** polled every 30s



Why Choose the Interactivity API Here?

	React	Interactivity API
SSR	✗	✓
Bundle Size	Heavy-er	Light
Shared PHP templates	✗	✓
Live Data Updates	✓	✓

Architecture: Block + Template Part



State vs Context Planning

State

```
state: {  
  isALeagueMen: true,  
  isALeagueWomen: false,  
  selectedLeague: 'a-league-men',  
  matches: [],  
  ...  
  get currentMatchWeek() {  
    const context = getContext();  
    return context.matchWeek[state.selectedLeague];  
  },  
}
```

view.js



Ideal for dynamic data



Global / shared



Accessible by other blocks



Can be derived with getters

State vs Context Planning

Context

```
wp_interactivity_data_wp_context( [                                render.php
    'selectedLeague' => 'a-league-men',
    'matchWeek'      => [
        'a-league-men' => get_current_calendar_week('men'),
        'a-league-women' => get_current_calendar_week('women'),
    ],
    'clubs'          => [
        'a-league-men' => get_club_selector_data('men'),
        'a-league-women' => get_club_selector_data('women' ),
    ],
    'isALeaguesSite' => is_current_blog_aleagues(),
] );
```



Local to DOM elements



Best for isolated UI



Useful for initial state



There is also **config**

Fetching Matches

```
const { state, actions, callbacks } = store("matches-slider", {  
  callbacks: {  
    *fetchMatches(args) {  
      state.isLoading = true;  
      const data = yield getMatchesData(...args);  
      state.hasLiveMatches = data.some(  
        (match) => match.status === 'Playing'  
      );  
      const groupedMatches = callbacks.groupMatchesByDay(data);  
      state.matches = groupedMatches;  
      state.isLoading = false;  
    },  
  },  
});
```

view.js

Templating: *wp-each* + *wp-key*

render.php

```
<ul class="matches-group-list">
  <template
    data-wp-each--group="state.matches"
    data-wp-each-key="context.group.date">
    <li data-wp-bind--data-group-index="context.group.index">
      <ul>
        <template
          data-wp-each--child="context.group.matches"
          data-wp-each-key="context.child.uid">
          <li class="match-card">
            <span class="score" data-wp-text="context.child.home_team_score"></span>
          </li>
```

Live Data Updates

```
callback.fetchLiveMatches = {  
  const fetchAndUpdate = async () => {  
    const matches = state.matches; // Loop through matches.  
    const shouldRefreshScores = compareTimeStamp(match.start_timestamp, 30, 240);  
    if (match.uuid && shouldRefreshScores) {  
      const stats = await getLiveMatchData( match.uuid );  
      clearInterval(state.liveMatchesInterval);  
      fetchAndUpdate();  
      state.liveMatchesInterval = setInterval(fetchAndUpdate, 30000);  
    }  
  };  
};
```

view.js

State Binding: *wp-bind*, *wp-class* + *wp-text*

```
<div class="match-card"                                render.php
  data-wp-class--live="callbacks.isMatchLive">

  <span data-wp-text="match.home_team_score"></span>
</div>
```



Filters

ISUZU UTE
A-LEAGUE MEN

NINJA
A-LEAGUE WOMEN

NEXT 7 DAYS ✕

WEEK ▼

CLUB ▼

```
<div data-wp-context='{league:"a-league-men"}'>  
  <span  
    data-wp-on--click="actions.setSelectedLeague"  
    data-wp-class--active="state.isSelectedLeague"  
  >  
    Isuzu Ute A-League Men  
  </span>  
</div>
```

[render.php](#)

Analytics Integration

```
actions.sendCustomEvent = function* (event, data) {  
  yield window.dataLayer.push({  
    event,  
    ...data,  
  });  
};
```

view.js



Google
Tag Manager

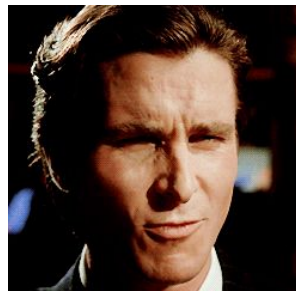
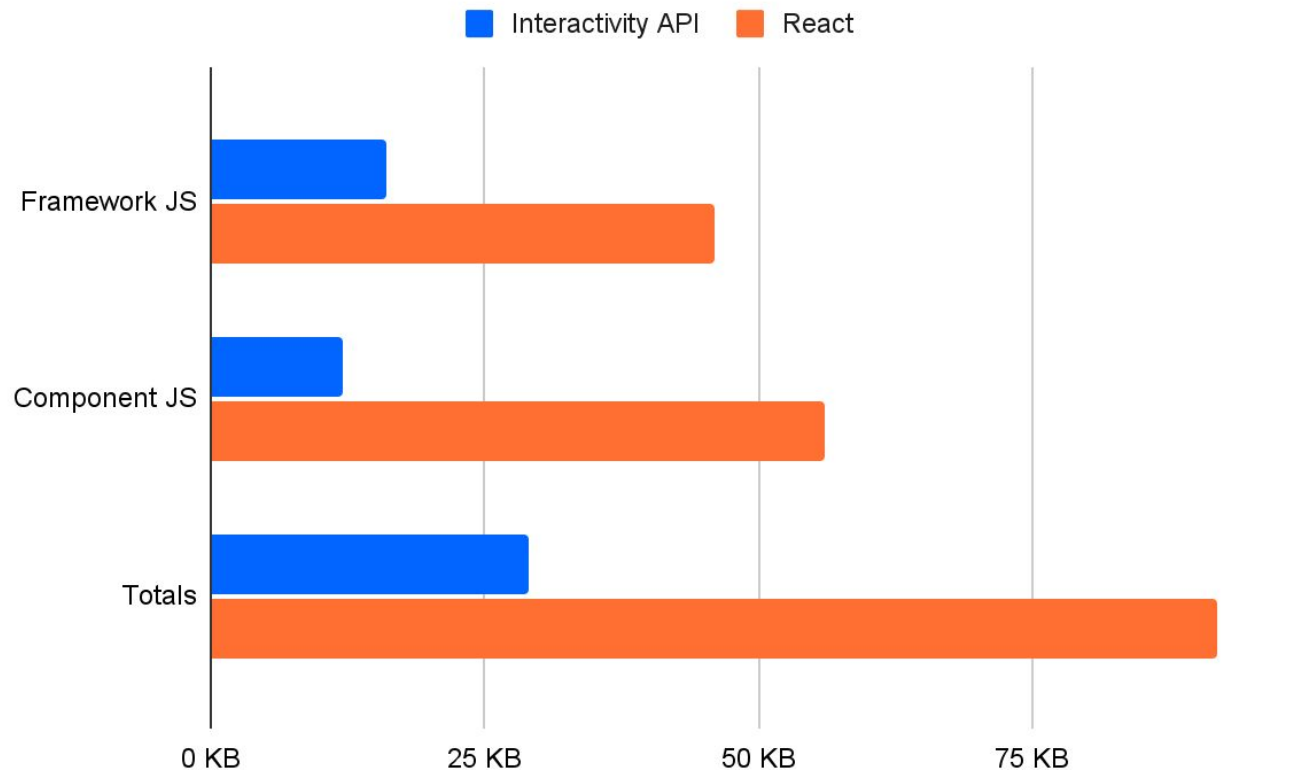


Google Analytics

*The***CODECO**

Let's see the results

Size Performance Comparison



Key Takeaways

- ✓ Works great for both small & large components
- ✓ Best inside blocks; can work outside with Script Modules
- ✓ Lightweight, fast, maintainable
- ✓ WordPress-native with SSR
- ✓ React still appropriate for large and/or complex UI

Helpful Resources 👁️👁️

- Interactivity API Reference (docs)
- WP Block Development Examples Repo
- WP Movies Demo
- 10up Guides
- Ryan Welcher
- Seth Rubenstein @ Pew Research
- Preact DevTools Chrome Extension

