High quality and efficient application development collaboration solutions

Michal Rafalski

Product owner at IQVIA / Novartis

8 common issues

01

Development takes too much time and resources

02

Architecture is not optimised

03

Customisation requires refactoring

04

Code duplicates in many projects

05

Missing documentation

06

Code is not tested properly and it is not maintainable.

07

Too short knowledge transfer even inside one team

08

Teams don’t know what others achieved

I would like to present

4 collaboration solutions to solve

8 common issues

4 collaboration solutions

01

Monorepo

Architecture to share features internally (building library in monorepo).

02

Tests

Providing stability of reusable parts with end-2-end and unit tests.

03

Registry

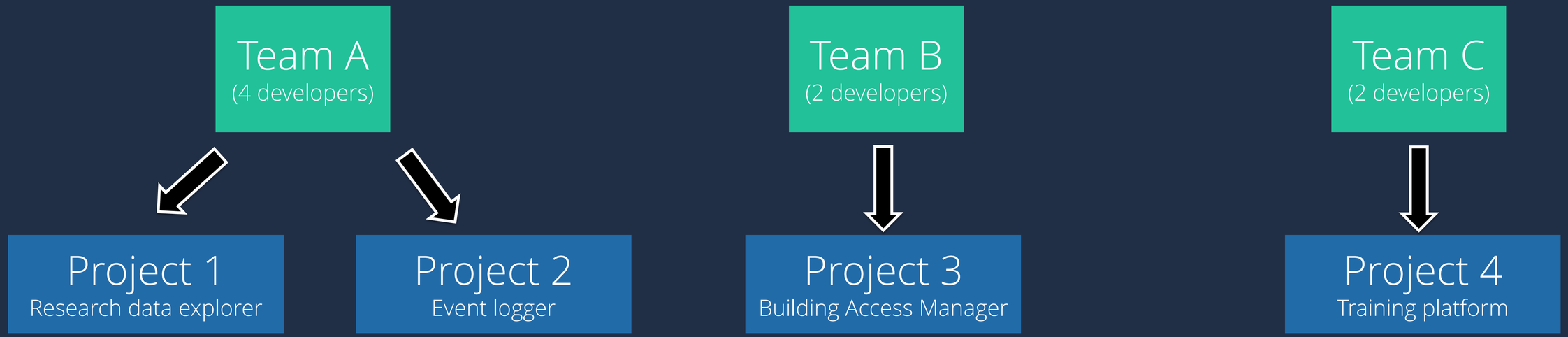
Service to share features with all developers.

04

Transparency

Share a list of features, all decisions, actions, conclusions, and knowledge with Docusaurus.

Story of 3 teams

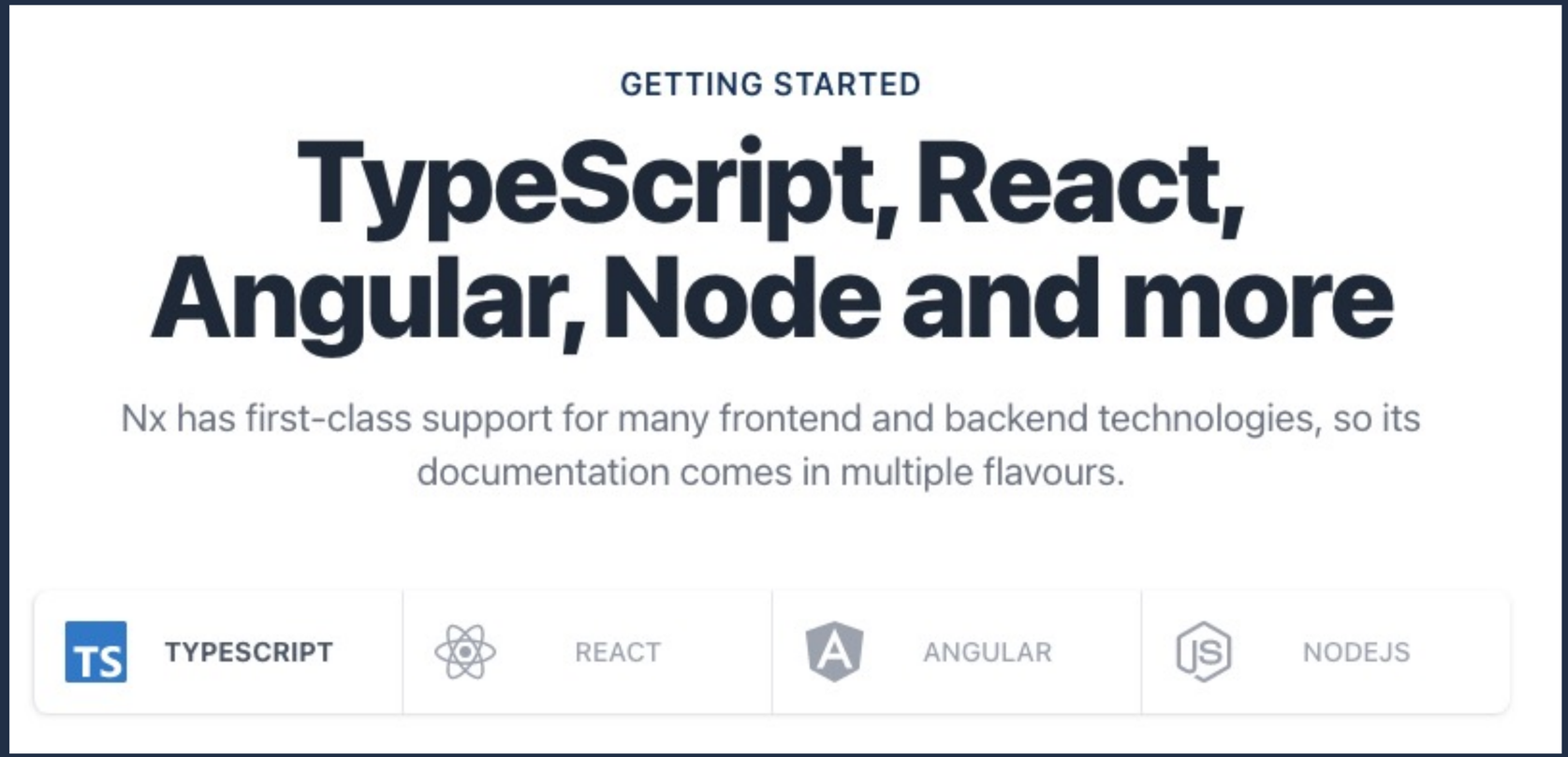


All projects require:

- Login interface
- Stable connection between UI and API
- Authentication and authorisation module
- Admin access
- Smart tables in UI
- Database on OCP
- ORM for databases
- Logs on PV
- Bundler for UI
- Navbar
- Documentation

NX workspace

https://nx.dev/



NX workspace for Team A

Project 1 UI

Library of components and features for UI

Project 1 API

Library of features UI and API

Project 2 UI

Library of features for API

Project 2 API

1 minute configuration of NX workspace

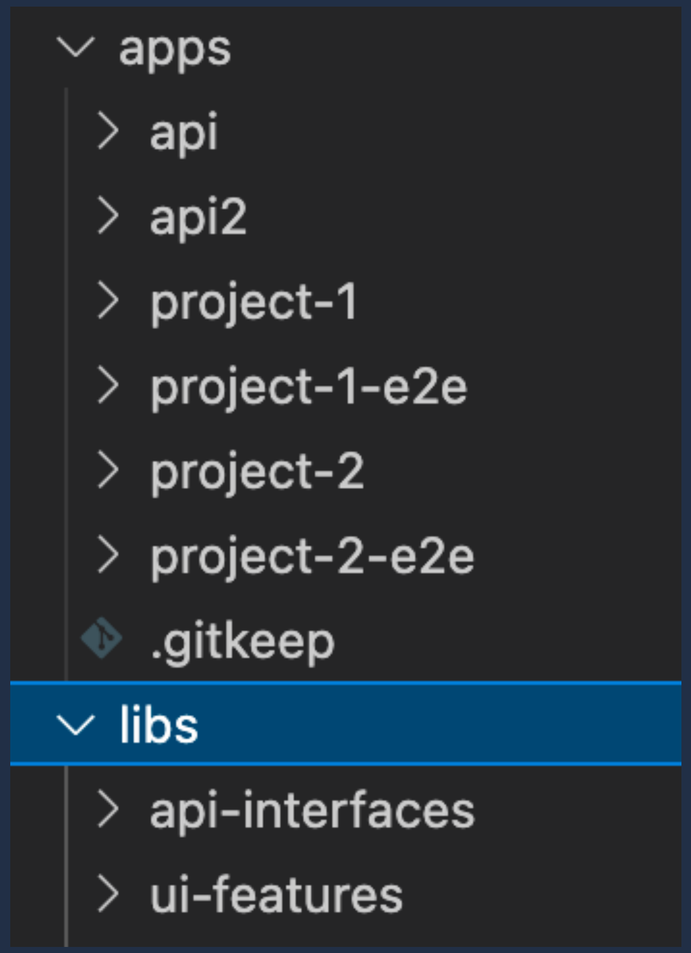
```
(base) C02Z51N3LVCJ:~ rafalmi1$ npx create-nx-workspace@14.1.7
```

yarn nx generate @nrwl/react:app project-2

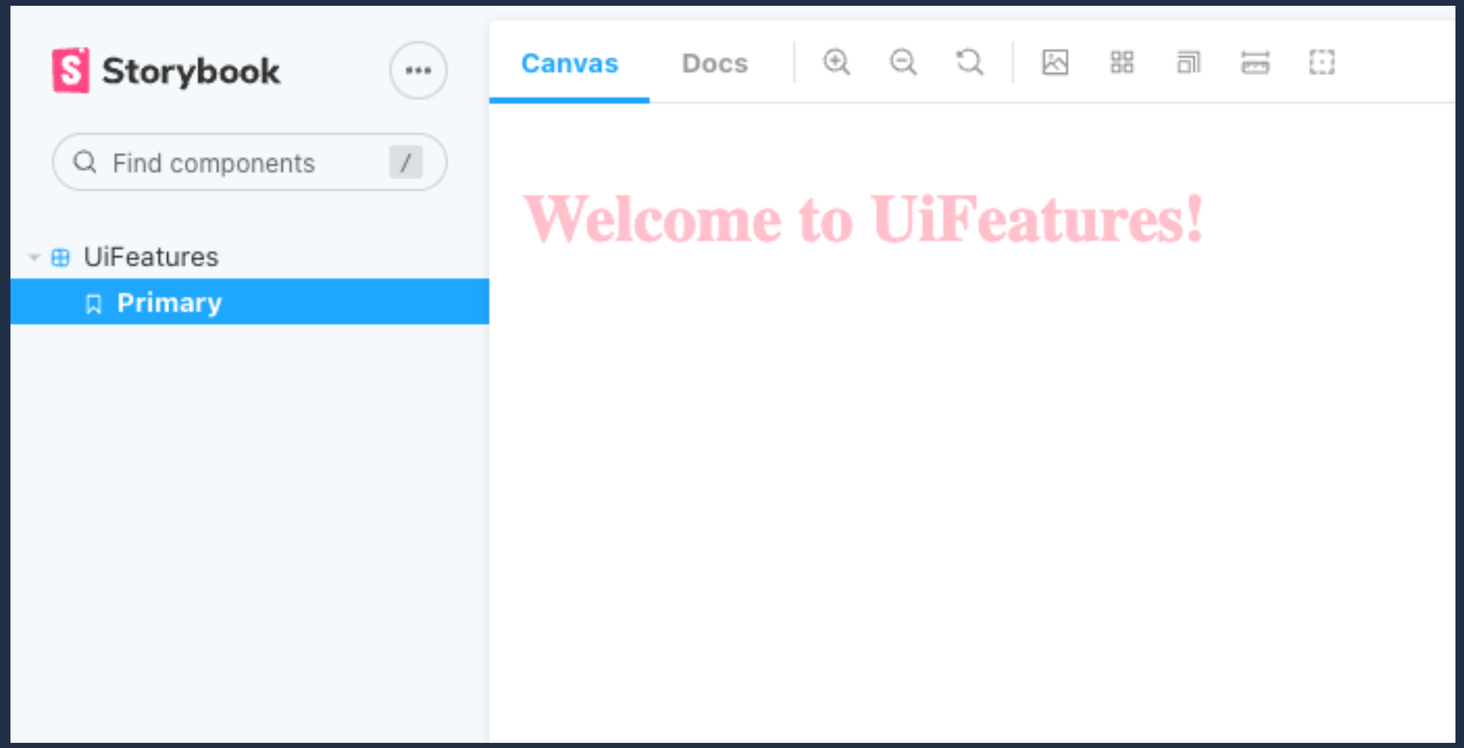
yarn nx generate @nrwl/express:app api2

yarn nx generate @nrwl/react:lib ui-features

Prepare for reusable component in NX workspace



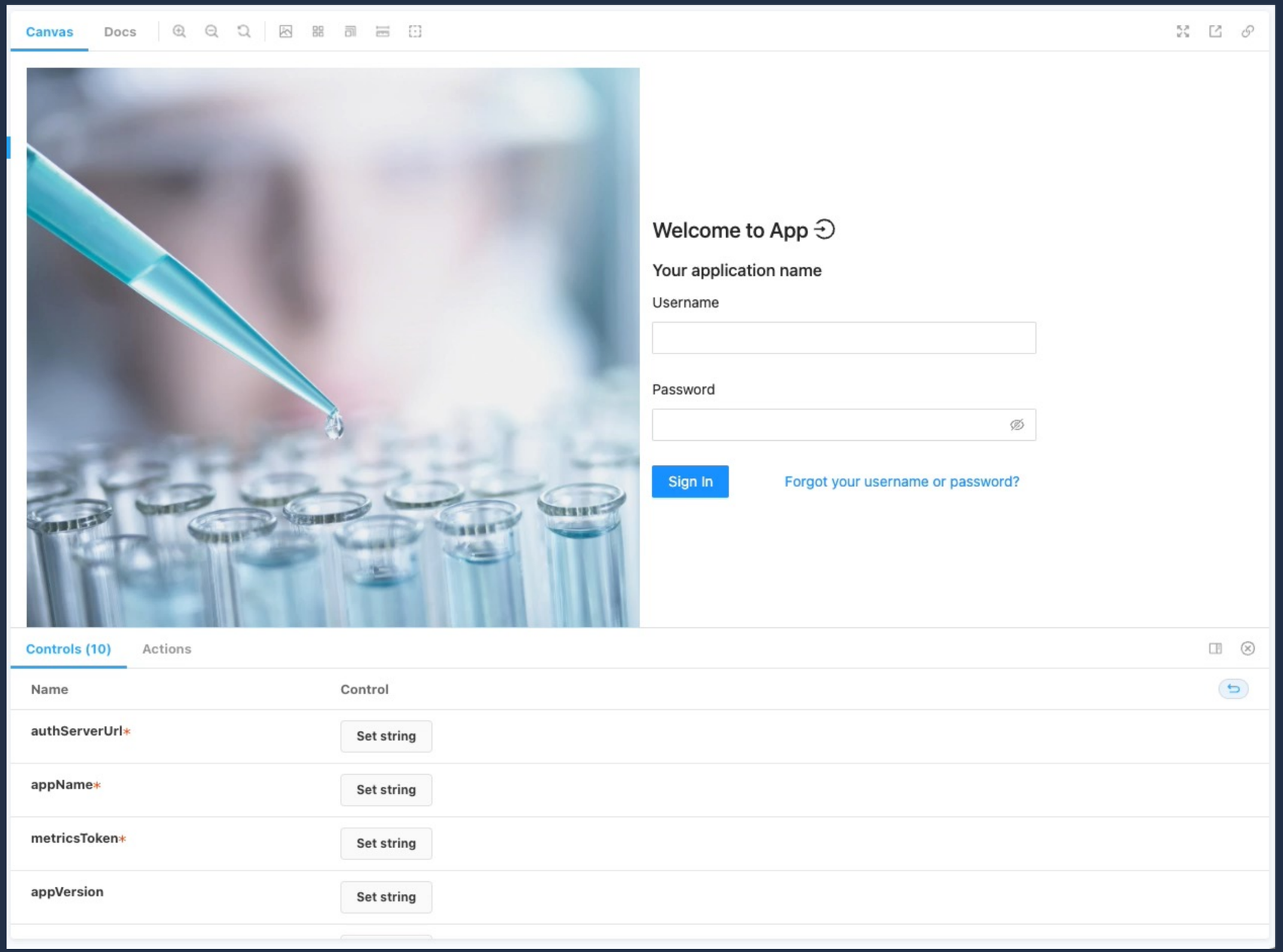

```
import { UiFeatures } from '@team-a-workspace/ui-features';
```




```
yarn add --dev @nrwl/storybook
yarn nx g @nrwl/react:storybook-configuration ui-features
yarn nx run ui-features:storybook
```

First success: in Team A Project 1 and Project 2 can use the same reusable component!

Features of Storybook



Welcome to MDR

Meta Data Repository

Username

521ID

Password

Sign In

Forgot your username or password?

Welcome to Comet

Controlled Terminology Management

Username

Password

Please enter your password

Sign In

Forgot your username or password?

Welcome to Clinical Data Explorer

Explore GPS datasets in your browser

Username

Password

Sign In

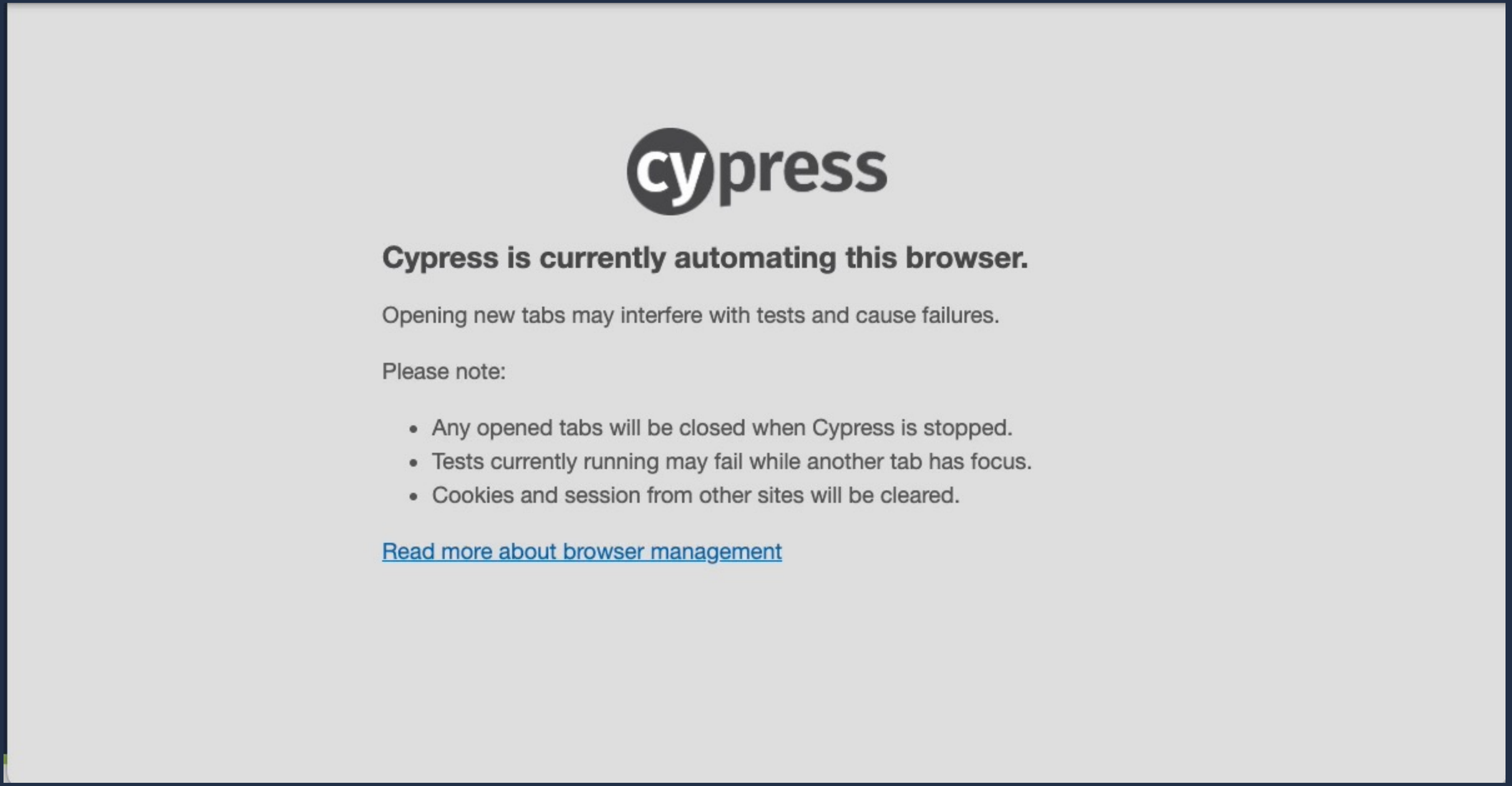
Forgot your username or password?

Building stable and maintainable components with e2e tests

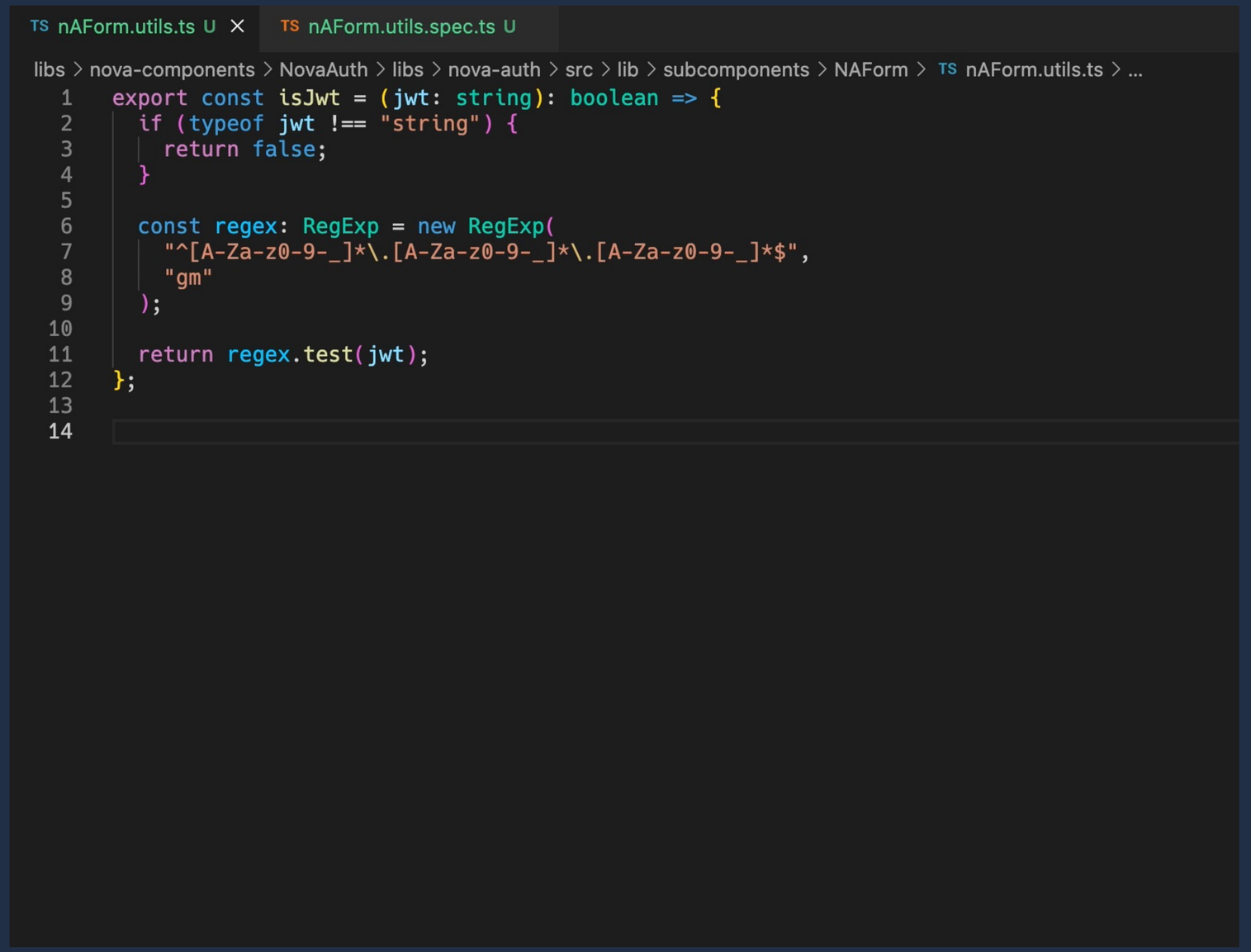
yarn nx run ui-features-e2e:e2e --watch

```
cy.get('#username').type("MyUserName");

cy.get("#username").clear();

cy.get("div").contains("Please enter your username").should('exist’)

cy.get('#username').type("MyUserName");

cy.get("div").contains("Please enter your username").should('not.exist')
```

Cypress demo

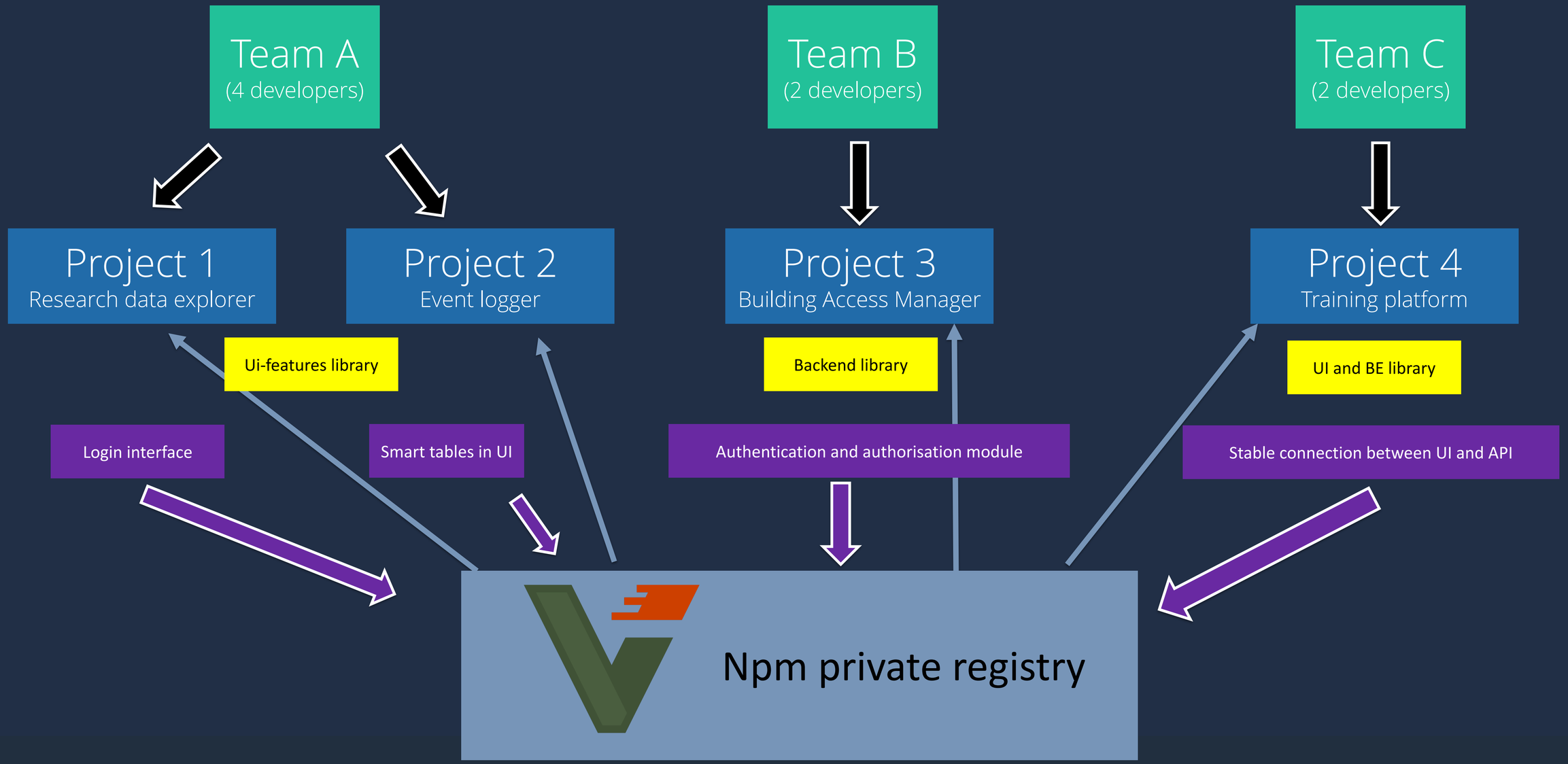


Building stable and maintainable features with unit tests

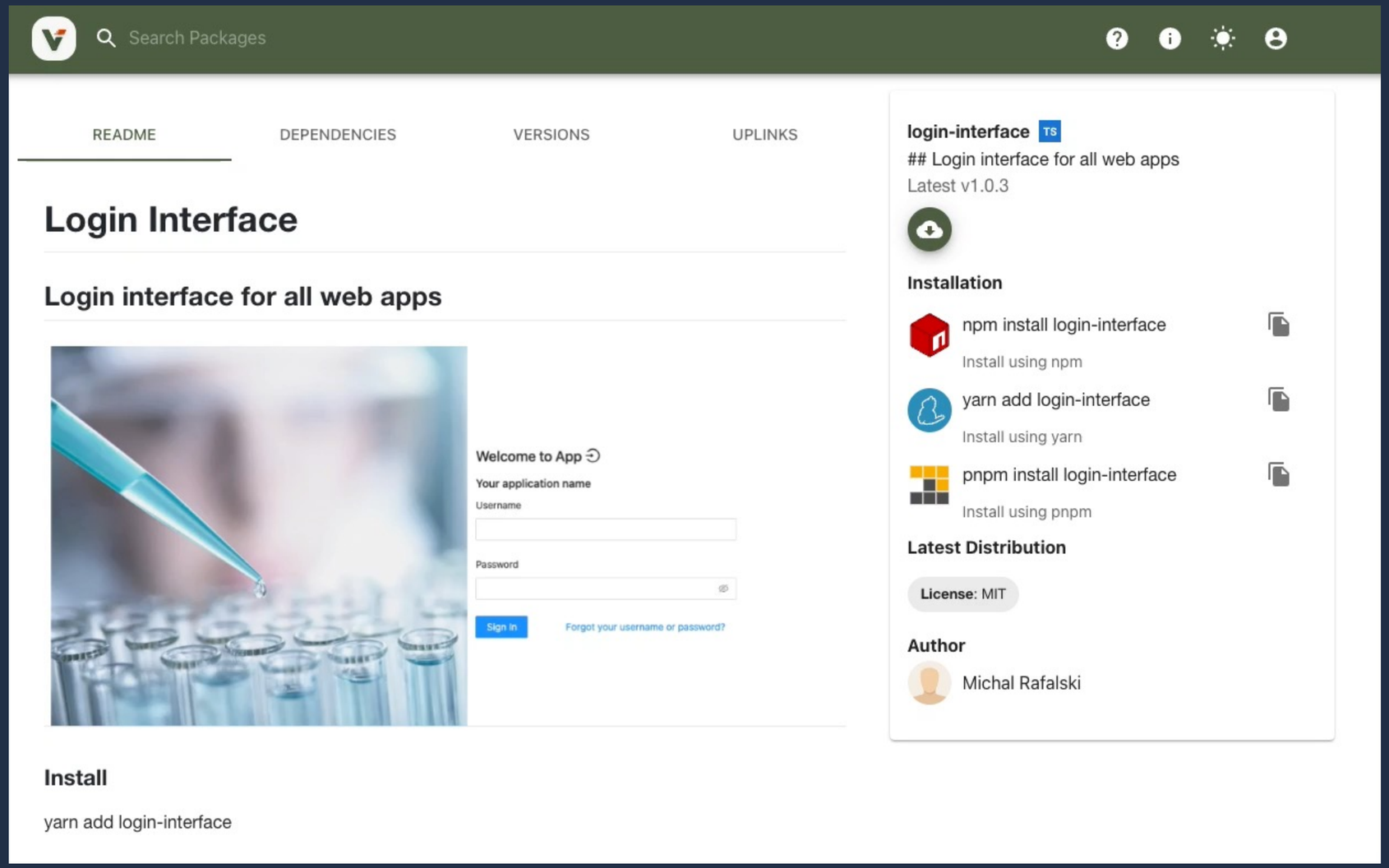


Second success: our features inside the component are stable, well tested and ready for many edge cases.

Publishing reusable features



Publishing reusable features with Verdaccio



Third success: shareable features from all groups are available for everyone. Team A can use features from team B and team C and vice versa.

Transparency

1. Present your applications to other groups
2. Document your decisions about the technology stack that others can benefit from it
3. List all features from your applications and make it accessible to others
4. Document well your research
5. Write articles to cover unique knowledge about projects you work on
6. Document logic of functions with Mermaid.js that others can understand better the implementation

Docusaurus

```
---
id: greeting
title: Hello
---

## Hello from Docusaurus

Are you ready to create the documentation site for your open source project?

### Headers

will show up on the table of contents on the upper right

So that your users will know what this page is all about without scrolling down or even without reading too m
```

Hello from Docusaurus

Are you ready to create the documentation site for your open source project?

Headers

will show up on the table of contents on the upper right

So that your users will know what this page is all about without scrolling down or even without reading too much.

````
```jsx title="/src/components/HelloCodeTitle.js"
function HelloCodeTitle(props) {
  return <h1>Hello, {props.name}</h1>;
}
```
````

/src/components/HelloCodeTitle.js

```jsx
function HelloCodeTitle(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

```
$$
I = \int_0^{2\pi} \sin(x)\,dx
$$
```

$$I = \int_0^{2\pi} \sin(x)\,dx$$

Mermaid.js

```
pie title Did you get vaccinated?
```


Fourth success: all decisions, actions, conclusions, features, and knowledge are documented.

8 solved common issues

01

Development takes too much time and resources

02

Architecture is not optimised

03

Customisation requires refactoring

04

Code duplicates in many projects

05

Missing documentation

06

Code is not tested properly and it is not maintainable.

07

Too short knowledge transfer even inside one team

08

Teams don’t know what others achieved

Thank you for your attention

Michal Rafalski
Product owner at IQVIA / Novartis