Storybook

from O Zero to 🦸 Hero



Merci à nos sponsors

"Etoile"







































Gaetan Maisse

@gaetanmaisse

Contributeur **Storybook**

Développeur Gravitee.io











Thibaud Avenier

@ThibaudAV

Contributeur **Storybook**

Développeur Gravitee.io













Typescript, Angular

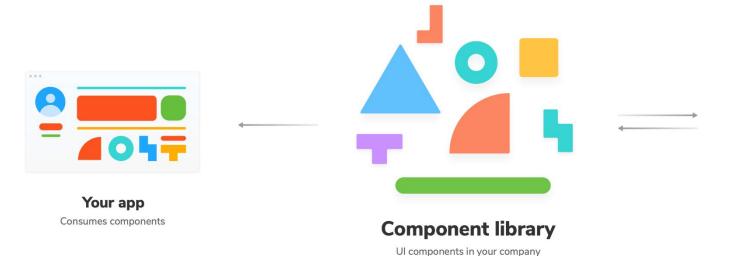


Rhums & Beers

Sommaire

- Les bases de Storybook
- A la découverte des Addons
- Jouer avec Decorators
- Augmenter nos Story avec de la Doc MDX
- Et bien plus encore: "Play function", "Storybook-ception", "..."
- Intégration dans un workflow de CI: build, validation UI et déploiement

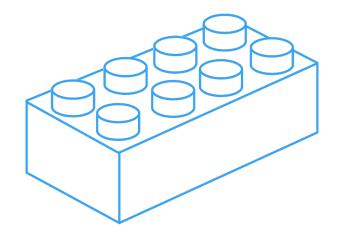
Storybook lives alongside your app





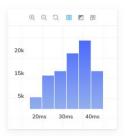
Storybook

Builds and indexes components



Modern UIs are built from components

- Fast: Standardized and interchangeable
- Durable: Modular and clearly defined inputs
- **Reusable**: No reinventing the wheel



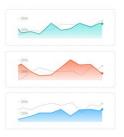
Build Uls in isolation

Implement components and pages without needing to fuss with data, APIs, or business logic.



Document use cases as stories

Save use cases as stories in plain JavaScript to revisit during development, testing, and QA.



Mock hard to reach use cases

Render components in key states that are tricky to reproduce in an app.



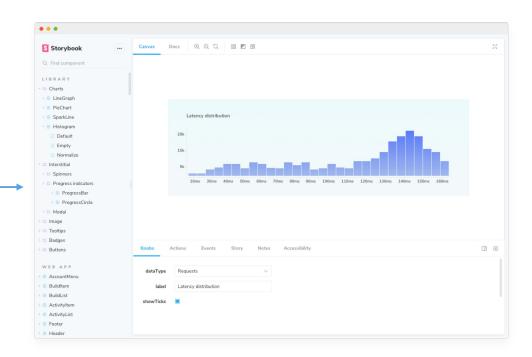
Supercharge your workflow with addons

Use addons to customize your workflow, automate testing, and integrate with your favorite tools.

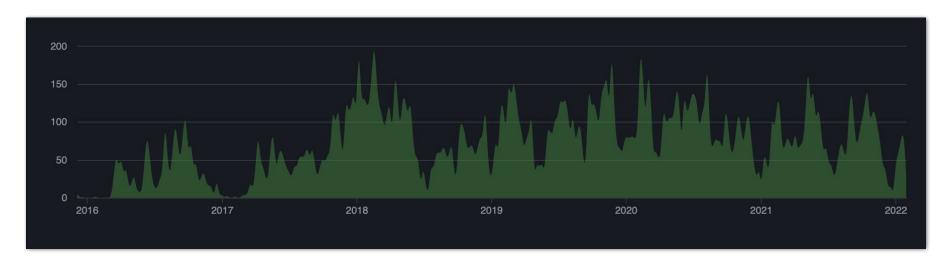








The Storybook Story



Initialement développé par une startup Sri Lankaise appelée Kadira





Installer Storybook dans votre projet existant
 npx sb init

Démarrer Storybook
 npm run storybook



Setup Storybook~15min

Retour à la théorie

Disclaimer

Tous les exemples de code seront liés à une app React

Component Story Format (v2)

- Format d'écriture d'une Story
- Standard open source basé sur les modules ES
 - 1 default export -> metadata
 - X named exports -> 1 story par export

```
1 // MyComponent.story.js/jsx
 3 import React from 'react';
 5 import { MyComponent } from './MyComponent';
7 export default {
8 title: 'Path/To/MyComponent',
    component: MyComponent,
10 };
12 export const Basic = () => <MyComponent />;
13
14 export const WithProp = () => <MyComponent prop="value" />;
15
```

1 import React from 'react'; 3 import { MyComponent } from './MyComponent'; 5 export default { 6 title: 'Path/To/MyComponent', 7 component: MyComponent, 8 decorators: [...], 9 parameters: { ... } 10 }; 12 export const Simple = () => <MyComponent />; 13 Simple.decorators = [...]; 14 Simple.parameters = { ... }; 15 Simple.storyName = 'So simple!';

Preview iframe ... Docs Storybook Canvas LIBRARY Charts Manager app -■ LineGraph ■ PieChart Latency distribution Communication ■ SparkLine 209 requests channel Last hour ☐ Last week Interstitial □ Spinners 20ms 30ms 40ms 50ms 60ms 70ms 80ms 90ms 100ms 110ms 120ms 130ms 140ms 150ms 160ms Progress indicators ■ ProgressBar ■ ProgressCircle ■ □ Modal □ Image □ Tooltips - Badges Actions Events Story Notes Knobs Accessibility □ Buttons Highlight results 20 violations WEB APP ■ AccountMenu > Ensures role attribute has an appropriate value for the element V BuildItem ■ BuildList > Ensures elements with ARIA roles have all required ARIA attributes ■ ActivityItem B > Ensures elements with an ARIA role that require child roles contain them 9 ■ ActivityList ⊞ Footer > Ensures buttons have discernible text 2 ⊞ Header

Configuration

- Configuration disponible dans le dossier .storybook
- main.ts → Configuration des stories, addons, Babel, Webpack etc
- preview.ts → Configuration du rendering des stories

- manager-head.html → Ajout de styles custom sur la UI Storybook
- preview-head.html → Ajout de CSS externe ou webfonts dans la preview des Stories

```
1 // .storybook/main.js|ts
  3 module.exports = {
     stories: ['../src/**/*.stories.@(js|mdx)'],
     addons: ['@storybook/addon-essentials'],
     features: {
       storyStoreV7: true, // Optional and not by default
     },
    staticDirs: ['../public'],
10 };
11
12
13
14
15
```

CLI

start-storybook

Démarre Storybook avec un serveur local et un mode live reload

build-storybook

Compile un Storybook statique dans le répertoire storybook-static, qui pourra ensuite être déployé

Create your first story ~30min

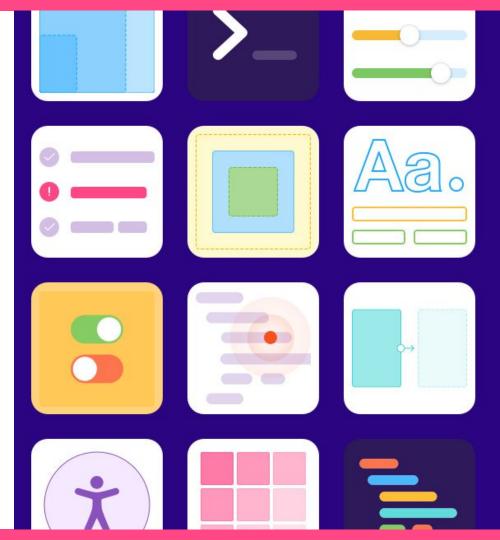
- Initialiser un composant Like/Dislike avec un template Hello world
- Créer une story associée à ce composant
- Ajouter la fonctionnalité de Like
- Ajouter la fonctionnalité de Dislike

Objectif:

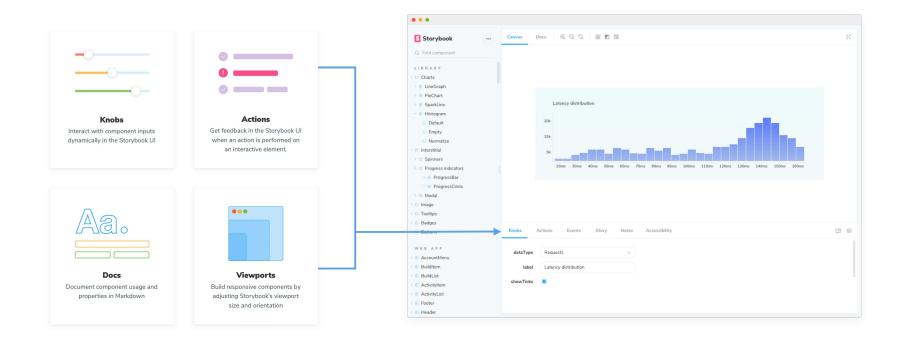


5 Likes - 6 Dislikes

Addons

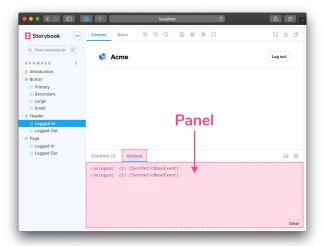


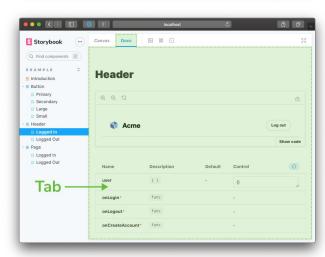
Addons

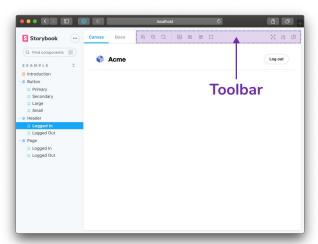


Addons

- Mécanisme de plugins
- Officiel vs Community
- Large varietes d'addon: orienté test, design system, accessibility



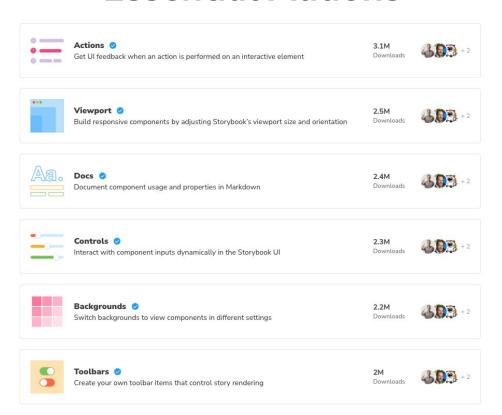




Preset ↓

```
module.exports = {
  managerWebpack: async (config, options) => {
      // Update config here
      return config;
  },
  managerBabel: async (config, options) => {
      // Update config here
      return config;
  },
  webpackFinal: async (config, options) => {
      return config;
  },
  babel: async (config, options) => {
      return config;
  },
  },
};
```

Essential Addons



Backgrounds

- Ajout d'un élément dans la Toolbar
- Permet de choisir le background d'une story
- Liste de couleurs à définir dans la configuration

Viewport

- Ajout d'un élément dans la Toolbar
- Permet de choisir la taille du container d'une story
- Soit en donnant des valeurs de hauteur x largeur en px
- Soit à partir de valeurs correspondantes à des écrans smartphone, tablette
 - o iPhone 13, Pixel 5 etc
 - ~50 options disponibles par défaut
 - Possibilité d'en ajouter

Args & Controls

- Ajout d'un Panel
- Donne la possibilité de modifier les inputs/props d'un composant
- Valeurs statiques à définir dans la Story
- Valeurs dynamiques accessibles dans l'interface de Storybook

Args & Controls

```
1 export default {
  2 component: Button,
    argTypes: {
      type: { control: "text", name: "Type" },
      label: {
      name: "Label",
         options: ["Plain", "Bold"],
      control: { type: "select", labels: { Bold: "BOLD" } },
         mapping: { Bold: <b>Bold</b> }
 10
11 }
12 };
 13
14 export const Basic = (args) => (<Button type={args.type}>
   {args.label}</Button>);
 15 Basic.args = {
 16 type: "primary",
17 };
 18
```

Toolbar & globals

- Ajout d'un élément custom dans la Toolbar
- Définition d'args globaux partagés entre toutes les Stories
- Souvent utilisé pour
 - l'internationalisation
 - le multi thème

Toolbar & globals

```
// .storybook/preview.js
 3 export const globalTypes = {
     locale: {
       name: 'Locale',
       description: 'Internationalization locale',
       defaultValue: 'en',
      toolbar: {
 9
      icon: 'globe',
10
       items:
11
        { value: 'en', right: '■', title: 'English' },
         { value: 'fr', right: '♥', title: 'Français' },
12
         { value: 'es', right: '

', title: 'Español' },
13
         { value: 'zh', right: '■', title: '中文' },
14
15
          { value: 'kr', right: '♨', title: '한국어' },
16
17
       },
18
     },
19 };
20
```



Démo

Interaction avec un composant ~30min

- Ajouter un input/prop avec le nombre de Likes/Dislikes
- Lier l'input avec un Control Storybook
- Ajouter un output/event indiquant le nombre de Likes/Dislikes après chaque click
- Lier l'output avec une Action Storybook

Objectif:

Simuler les interactions programmatiques lorsque le composant est intégré dans une app

Decorators

Decorator

- Permet de wrapper le composant utilisé dans la Story
 - Ajout de margin, border
 - Utilisation d'un composant parent
- Peut aussi mocker le context d'utilisation du composant
- La plupart des addons utilisent des decorators pour ajouter du comportement

Decorators

- Plusieurs niveaux de decorators possibles
 - Story
 - Component
 - Global
- Tous les decorators sont appliqués dans l'ordre de définition

Decorator - Implementation

- Decorator = Function
- Input: la Story "de base"
- Output : un objet correspondant à une Story
- Attaché à une Story via : "decorators": [...]

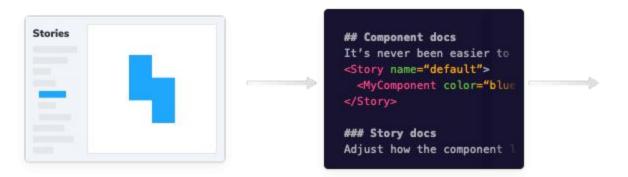
Création d'un decorator ~30min

• Créer un décorator pour ajouter des marges autour d'un composant

Objectif:



Docs + MDX











Docs

- Onglet Docs
- Regroupe toutes les stories liées à un composant
- Possibilité d'ajouter du texte de documentation/description

MDX

- MDX = Markdown + JSX
 - Syntaxe Markdown
 - o Possibilité d'intégrer des stories, components etc
- Conversion automatique MDX -> CSF

Included Doc Blocks

- Meta
- Canvas
- Story
- ArgsTable

```
1 <!-- Checkbox.stories.mdx -->
 3 import { Canvas, Meta, Story } from '@storybook/addon-docs';
 5 import { Checkbox } from './Checkbox';
 6
 7 <Meta title="MDX/Checkbox" component={Checkbox} />
 9 export const Template = (args) => <Checkbox {...args} />;
11 # Checkbox
12
13 With `MDX`, we can define a story for `Checkbox` right in the
   middle of our
14 Markdown documentation.
15
16 <Canvas>
17 <Story
18
      name="Unchecked"
19
      args={{
     label: 'Unchecked',
20
21
      }}>
22
      {Template.bind({})}
23
      </Story>
24
25
     <Story
26
      name="Checked"
27
      args={{
     label: 'Unchecked',
28
29
     checked: true,
30
      }}>
31
      {Template.bind({})}
32
      </Story>
33 </Canvas>
```



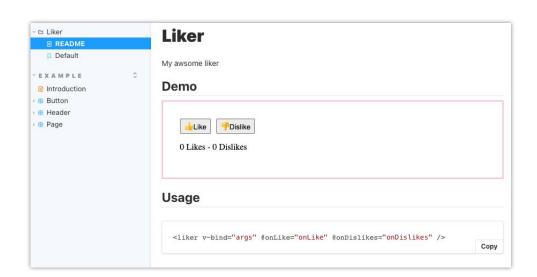
Démo



Ajout de documentation MDX ~15min

- Créer un fichier MDX et ajouter de la documentation Markdown
- Ajouter des stories avec plusieurs inputs/états différents

Objectif:



What's next?

Component Story Format (v3)

CSF 2.0 CSF 3.0

```
import React from "react";

import { Button } from "./Button";

sexport default {
   title: "Button/CSF2",
   component: Button,
   };

export const Basic = () => <Button />;

export const Primary = (args) => <Button {...args} />;

Primary.args = { primary: true };
```

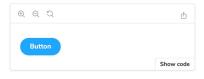
```
import { Button } from "./Button";

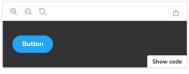
sexport default {
   component: Button,
};

export const Basic = {};

export const Primary = { args: { primary: true } };

10
11
12
13
14
```





Play function

- Permet de simuler des interactions utilisateurs
- Exécutée à l'affichage d'une story
- Possibilité d'utiliser `@storybook/testing-library`

```
1 Default.play = async () => {
2   const likeBtn = screen.getByText("Like", {
3     selector: "button",
4   });
5
6   await userEvent.click(likeBtn);
7 };
```

Addon interaction

- A mi-chemin entre interaction et tests
 - Permet d'interagir avec les éléments
 - Puis faire des expects

- Docs + examples
 - https://storybook.js.org/docs/react/essentials/interactions

```
Canvas Docs ⊕ ⊖ ⊃ № ⊞ □ ₩ □
Like Dislike
15 Likes - 14 Dislikes
Controls (2)
            Actions (29)
                         Interactions
✓ userEvent.click(within(ShadowRoot).getByText("Like"))
vuserEvent.click(within(ShadowRoot).getByText("Like"))
vuserEvent.click(within(ShadowRoot).getByText("Like"))
expect(action(onLike)).toHaveBeenCalledWith(42)
 expect(jest.fn()).toHaveBeenCalledWith(...expected)
 Expected: 42
 Received
        1: {"_constructor-name_": "CustomEvent", "isTrusted": false}
        2: {"_constructor-name_": "CustomEvent", "isTrusted": false}
        3: {"_constructor-name_": "CustomEvent", "isTrusted": false}
 Number of calls: 3
```

Storybook on-demand architecture

- Build 3x plus petit
- Chargement 4x plus rapide

```
1 // .storybook/main.js
2 module.exports = {
3  features: {
4  storyStoreV7: true,
5  }
6 };
```

Storybook-ception

- Agrège plusieurs Storybook dans un Storybook
- Fonctionne avec un Storybook local ou distant
- Exemple:
 - Faire des déclinaisons d'un Design System par Framework et les agréger



Ajout d'une play function ~15min

- Convertir les stories en CSF v3
- Ajouter une play function cliquant plusieurs fois sur le bouton Like
- Bonus: Ajouter l'addon interaction

Objectif: Simuler des interactions utilisateurs

CI

Intégration dans une Cl

- Build statique de Storybook pour chaque Pull Request
- Déploiement du site statique
- Review des Stories comme tout le reste du code
- Mise en place d'outils de snapshots testing pour éviter les régressions (addon, jest, etc)

Test non régression UI

- Utilisation de Chromatic
 - Déploiement Storybook PR + master
 - Création d'un catalogue de composants
 - Snapshot des Stories puis comparaison avec la branche de base



Storybook World

Storybook en quelques chiffres

- Monorepo avec 50 packages NPM
- ~40 000 commits
- ~ 1500 contributeurs
- Core team de 10 personnes
- Discord
- Toute contribution est la bienvenue!

Travaux en cours

- Amélioration des performances
- Extraction de tout ce qui est lié à Webpack
- Focus sur le support React, Angular, Vue, Web Components
- Interaction avec outils externes

Merci