Storybook

Storybook Overview

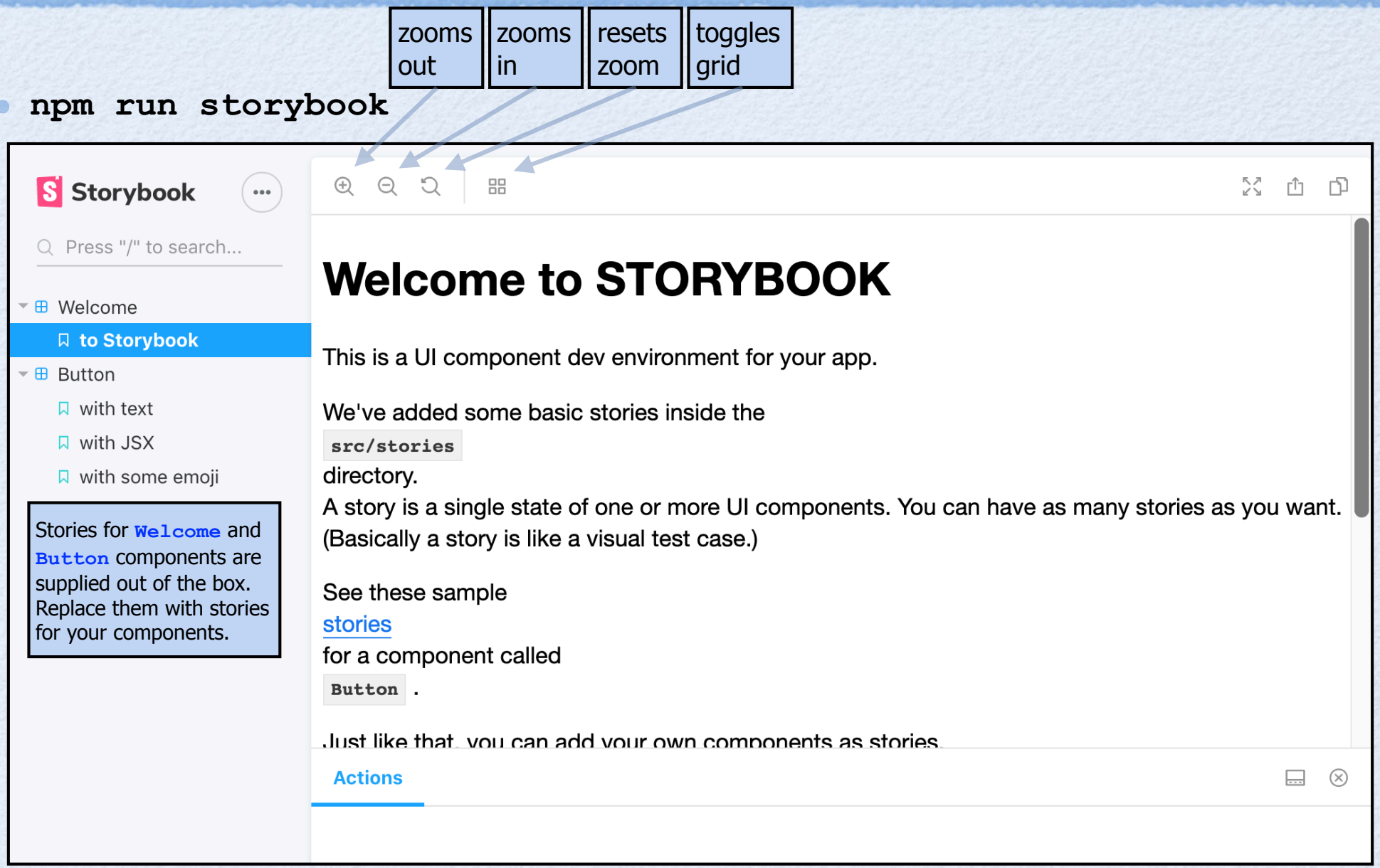
- "A UI development environment and playground for UI components"
- "Showcase components interactively in an isolated development environment"
- Works with many web UI frameworks including React, Vue, Angular, and Svelte
- Benefits
  - can implement and test components in isolation before pages that will use them are implemented
  - designers can experiment with component styling without running the app
  - can demonstrate components to customers for feedback before app is complete
  - can verify operation of components without following a long navigation sequence in the app
  - can use as "design system" documentation
- https://storybook.js.org/

Installing Storybook for Vue

- **`npx -p @storybook/cli sb init --type vue`**
- Adds **`storybook`** and **`build-storybook`** npm scripts in **`package.json`**
- Creates **`.storybook`** directory containing
  - **`config.js`**
    - modify **`config.js`** to use plugins like Vuex
    - modify to register custom components (Some components don't need this! Which do?)
  - **`addons.js`**
    - imports addons to display data received by Storybook event handlers and create links that navigate between stories
- Creates **`stories`** directory containing
  - **`index.stories.js`** - defines stories
    - it's also possible to colocate story definitions with component implementations under **`src`** directory
  - **`Welcome.js`** - an example component that provides instructions on how to add stories
  - **`MyButton.js`** - another example component

Running Storybook

- `npm run storybook`



Adding Stories ...

following slides demonstrate components in todo app

- In **`stories/index.stories.js`**

```
import {storiesOf} from '@storybook/vue';
import {action} from '@storybook/addon-actions';

import Arrow from '../src/components/Arrow';
import Todo from '../src/components/Todo';
import TodoList from '../src/components/TodoList';
```

`action` function is for mocking event handling and logging it; see examples in story for `Todo` component

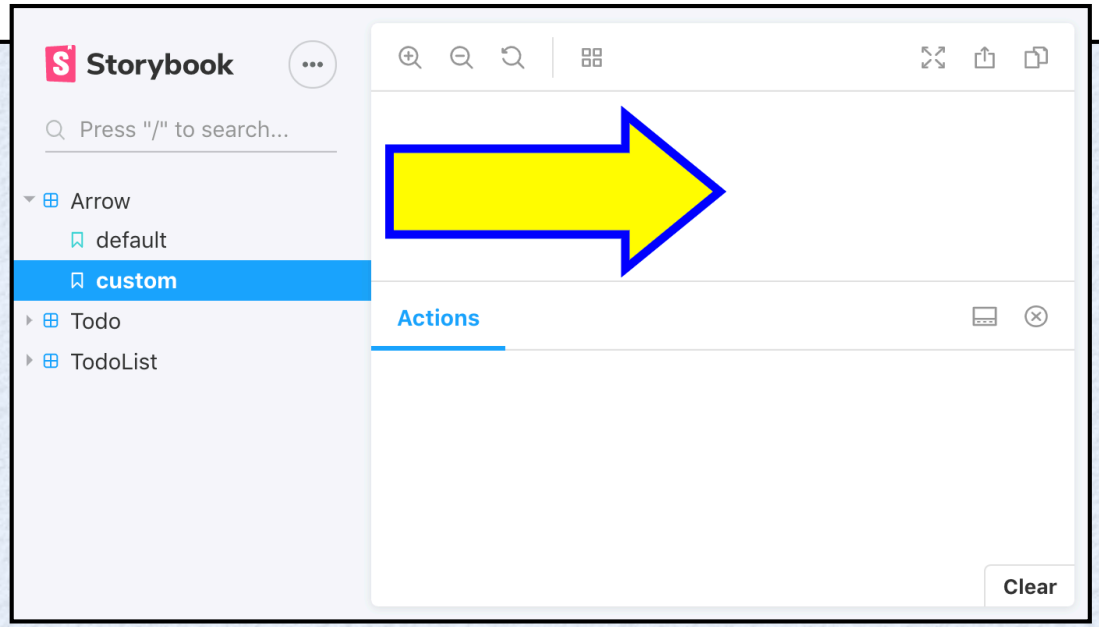
stories for imported components are on next three slides

... Adding Stories ...

- **Arrow** component stories

```
storiesOf('Arrow', module)
  .add('default', () => ({
    components: {Arrow},
    template: '<Arrow :height="100" :width="200" />'
  }))
  .add('custom', () => ({
    components: {Arrow},
    template:
      '<Arrow fill="yellow" :height="100" stroke="blue" :strokeWidth="5" :width="200" />'
  }));
```

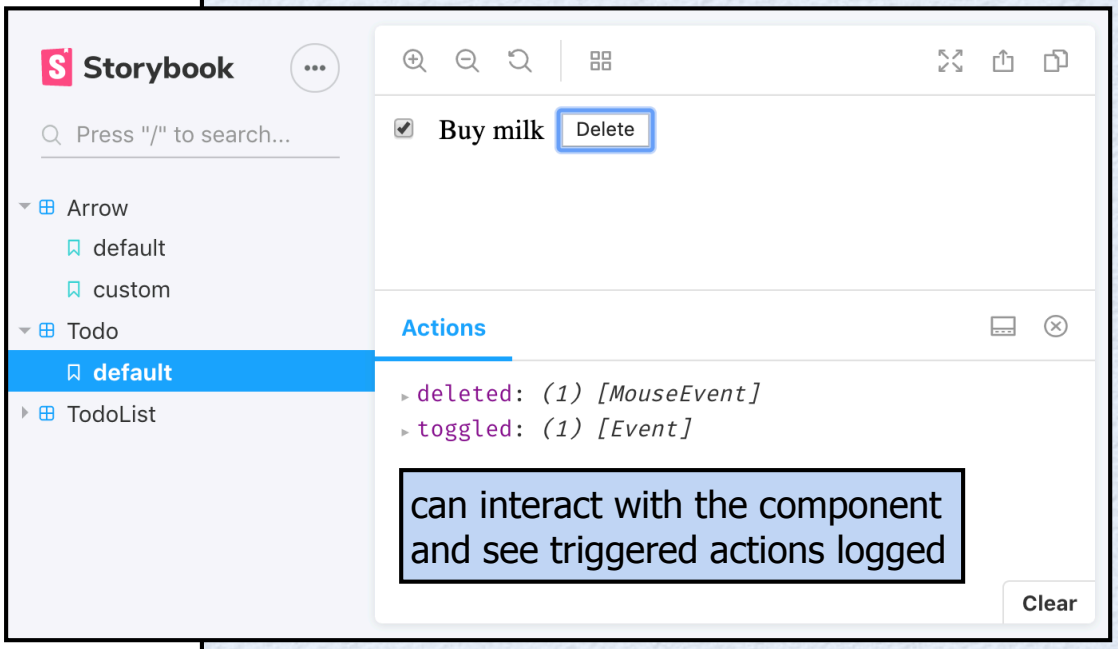
Passing `module` is necessary for "**hot module replacement**". Webpack uses this to add and remove modules at runtime without a full reload. This way everything needed by every component doesn't need to be retained in memory the entire time Storybook is running.



... Adding Stories ...

- `Todo` component story

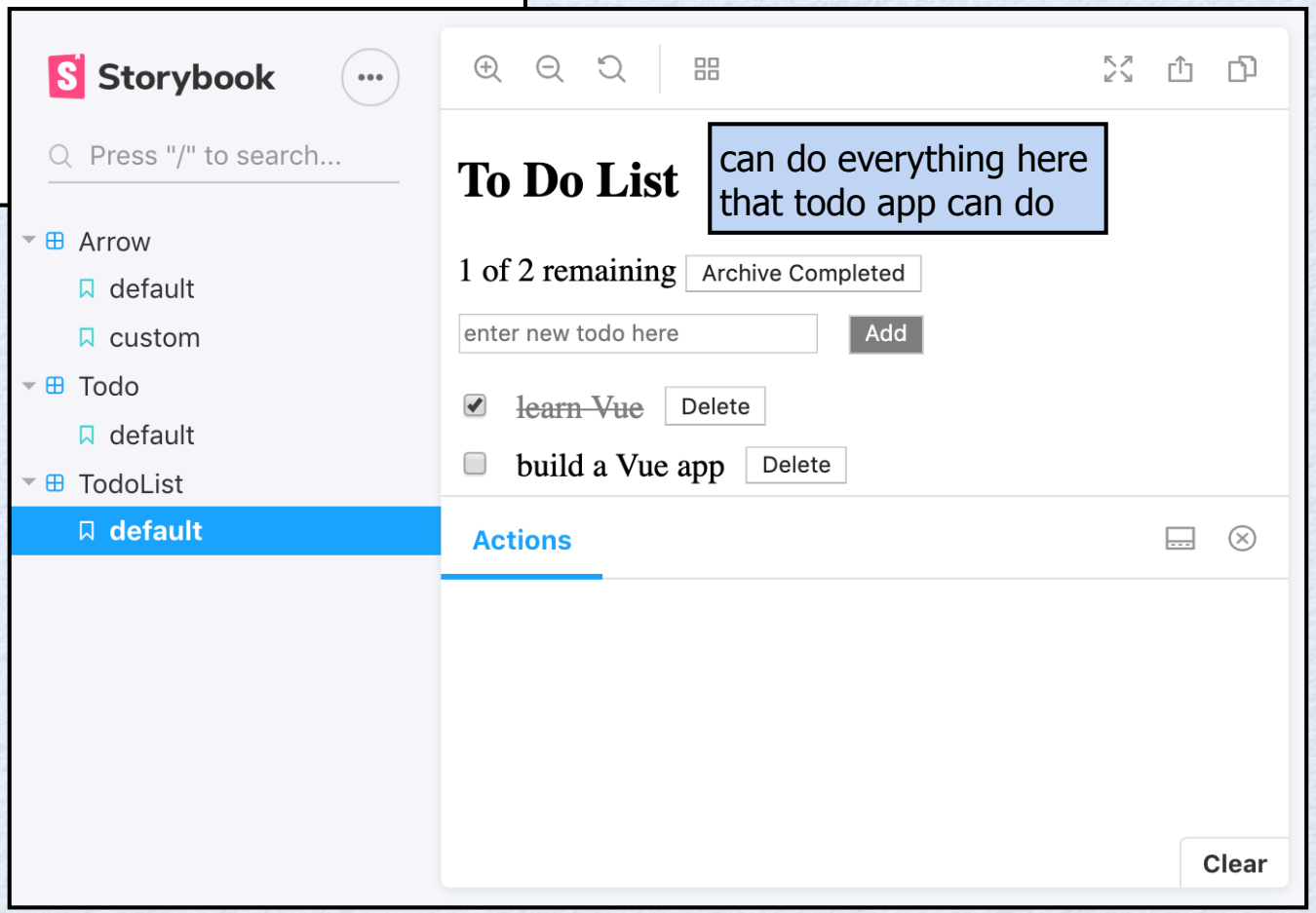
```
storiesOf('Todo', module).add('default', () => {
  return {
    components: {Todo},
    data() {
      return {
        todo: {text: 'Buy milk'}
      };
    },
    template: `<Todo
      :todo="this.todo"
      :onDeleteTodo="this.deleteAction"
      :onToggleDone="this.toggleAction"
    />`,
    methods: {
      deleteAction: action('deleted'),
      toggleAction: action('toggled')
    }
  };
});
```



... Adding Stories

- `TodoList` component story

```
storiesOf('TodoList', module).add('default', () => {
  return {
    components: {TodoList},
    template: '<TodoList />'
  };
});
```



Default CSS

- To provide default styling defined outside components being rendered
  - create **`stories/storybook.css`**

```
body {
  font-family: sans-serif;
}
```

  - import near top of **`index.stories.js`**

```
import './storybook.css';
```

Mocking Fetch

- Can create stories for components that make REST calls without actually making the calls
- When using Fetch API, one way is to replace **`fetch`** function with a no-op in **`.storybook/config.js`**

```
// Replace fetch function with a no-op so
// components that make REST calls don't fail.
window.fetch = () => {};
```

Add-ons

- Add functionality to Storybook
- Documented at https://storybook.js.org/addons/addon-gallery/
- Some only work with certain web frameworks
- Some are maintained by the Storybook team
  and others are maintained by the community
- Steps to install each differ, so check documentation

More

- There's more to learn about Storybook,
  but we've covered the basics
- For more read the official docs starting at
  https://storybook.js.org/docs/basics/introduction/