

GRASPING REACT BASICS

React is a library developed by Facebook that deals with rendering your UI in a very smart and performant way. It's gained a lot of traction due to its simplicity and proposed mindset which empowers developers to have a modular, component-based approach on their frontend architectures.

This training covers a lot of React related concepts, setting the base for understanding applications built on this technical stack. The focus will be on code examples (live coding, application building) and practicality. Feel free to consult the high-level agenda below.

INTENDED AUDIENCE

This training assumes the participants are already familiar with ES6 concepts (classes, destructuring, default parameters, arrow functions, let/const, etc.), basic JavaScript knowledge and syntax, npm (interacting with the npm ecosystem, package.json) and Git (basic commands). Thus, the aforementioned concepts will not be discussed or detailed during the training.

This training targets beginner and intermediate developers who do not have any (or have very basic) experience with React and who want to learn how to build UIs using this technology.

OUTLINE

Included below are the main topics covered in this training. The duration is 3 days.

1. What is React?
 - a. Talk about differences between frameworks and libraries
 - b. What problems is React trying to solve
 - c. Platforms that are impacted by React (Web, VR, Native)
2. What sets React apart?
 - a. Updating UI philosophy
 - b. JSX, VDOM, state, props, overall data flow, performance considerations
3. Spin up a simple React app in 3 minutes
 - a. Create React App (CRA) small intro
 - b. Installing CRA
 - c. Create a React project using CRA
 - d. Start the project
 - e. General overview of the project's structure
 - f. Pros/cons of CRA
4. JSX
 - a. A markup/templating language

- b. How JSX is converted to JavaScript
 - c. Differences from HTML
- 5. Writing our first React component
 - a. Writing using ES6 classes
 - b. Writing using functions (preferred, in the context of hooks)
- 6. VDOM (Virtual DOM)
 - a. What is VDOM?
 - b. How React employs VDOM behind the scenes
- 7. React props
 - a. What are props in React?
 - b. Immutability of props
 - c. PropTypes
- 8. Dealing with events in React
 - a. Responding to user events
 - b. List of predefined handlers in React
- 9. React state
 - a. What is state and how is it different than props?

- b. Changing the state & re-renders

10. React component lifecycle hooks

- a. Hooks that trigger when component first gets created
- b. Hooks that trigger when component's state changes
- c. Hooks that trigger when component's props change
- d. React component lifecycle hooks diagram

11. Fetching data from the server

- a. The useEffect hook
- b. Fetching data and updating the state

12. Building a puppy adoption application

- a. Initializing the app with CRA
- b. Building the component to display the list of puppies
- c. Adding in filters
- d. Add CRUD operations

13. Testing React components

- a. Introduction to Jest and Enzyme
- b. Types of rendering in Enzyme

- c. What to test in a React component
- d. Setting up tests in CRA
- e. Practice with shallow rendering and state testing

14. React architectural patterns, antipatterns and best practices

- a. Lifting state up for sharing info between components
- b. Aggressive componentization
- c. Container vs. Presentational components
- d. How the testing strategy adapts to Container vs. Presentational
- e. Local component state vs. state higher up in the chain
- f. Spreading props in render and destructuring props parameter
- g. Using React children prop
- h. Mutating DOM (anti-pattern) & when is it necessary

15. Approaching UIs by thinking in React

- a. Identifying components from UI and aggressive componentization
- b. Identifying state shape and level in component hierarchy
- c. Thinking only about state changes over time

16. Tweaking and improving the puppy adoption application

- a. Moving to Container vs. Presentational approach
- b. Split work between multiple components
- c. Introduction of React.memo, React.lazy, Suspense and their benefits

17. Recap

ABOUT THE TRAINER

Vlad Zelinschi is a pragmatic software architect, lover of the web and caffeine addict. He cares about surrounding himself with thoughtful leaders. He's the CTO of [Strongbytes](#), [Google Developer Expert](#) on Web Technologies since October 2016 and a certified trainer. He's part of [JSHeroes Community](#) and actively acting as an advisor for a couple of well known Romanian conferences such as [Codecamp](#) and [NDR](#).