

Micro Frontends with React

Audience Course Micro Frontends with React

This course is intended for React developers, frontend specialists, and solution architects who want to master building scalable applications based on a micro frontend architecture.

Prerequisites Course Micro Frontends with React

Participants should be comfortable with JavaScript, TypeScript, and React development. Experience with component-based design, build pipelines and state management is beneficial.

Realization Training Micro Frontends with React

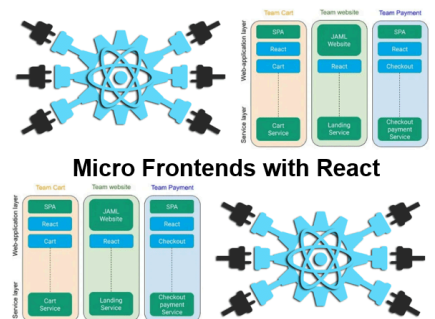
The training consists of interactive lectures combined with practical labs under the guidance of an experienced trainer. Emphasis is placed on hands-on exercises and applying the concepts in realistic project scenarios.

Micro Frontends with React Certificate

After completing the course, participants receive a certificate of participation in Micro Frontends with React.

Duration: 3 days

Price: € 2299

[Open Schedule](#)


Content Micro Frontends with React

The course Micro Frontends with React from SpiralTrain teaches you how to design modern, modular frontend solutions with React. You will practice breaking down complex applications into micro frontends, orchestrating them for smooth user experiences, and applying patterns that enable autonomous development and deployment.

Introduction

The course Micro Frontends with React begins with an overview of micro frontend concepts and their advantages over monolithic user interfaces. Design principles are discussed, with attention to scalability and independent deployments.

React Recap

Next participants review the fundamentals of React, including JSX, components, props and state, React Hooks, and event handling. Core features such as conditional rendering, routing with React Router, and the Context API are also refreshed.

Architecture

This module covers architectural patterns for React micro frontends, including component composition, container and presenter patterns, shared versus isolated state, routing across applications, and the role of a shell architecture.

Module Federation

Participants learn how to set up Webpack Module Federation for React projects. Topics include host and remote applications, dynamic imports, shared dependencies, version handling, and best practices for runtime integration.

Integration

Integration strategies are then explored, with focus on UI composition, shared navigation, cross-application routing, authentication and authorization handling, and service integration within a micro frontend orchestration.

State Management

This module examines state management challenges and solutions. Topics include local state, prop drilling, Context API usage, and Redux with Redux Toolkit for sharing and synchronizing state across micro frontends.

Deployment

Deployment aspects include CI/CD pipelines, independent releases, environment configuration, and containerization with Docker. Hosting with Kubernetes, cloud providers, rollback strategies, and monitoring options are also discussed.

Testing

Essential testing practices are introduced, including unit and integration tests with Jest and React Testing Library, end-to-end tests with Cypress, contract testing, automation, and accessibility checks.

Advanced Topics

The course concludes with advanced subjects such as authentication and authorization patterns, error boundary handling, performance optimization, caching strategies, and future trends. A hands-on project brings all concepts together.

Modules Micro Frontends with React

Module 1 : Introduction	Module 2 : React Recap	Module 3 : Architecture
Micro Frontends Overview Why React Micro Frontends Monolith vs Distributed UI Benefits and Drawbacks Key Use Cases Design Principles Team Scalability Deployment Independence Real-World Examples Future Outlook	React Fundamentals JSX Syntax Components Basics Props and State React Hooks Intro Event Handling Conditional Rendering List Rendering React Router Basics Context API	Architecture Patterns Component Composition Container and Presenters Shared vs Isolated State Routing Across Apps Cross-App Communication Lazy Loading Routes Micro Frontend Shell Error Boundaries Resilience Patterns
Module 4 : Module Federation	Module 5 : Integration	Module 6 : State Management
Webpack Federation Setup Host and Remote Apps Dynamic Imports Exposing Components Shared Dependencies Version Handling Runtime Integration Error Handling Configuration Files Best Practices	UI Composition Shared Navigation Authentication Integration Authorization Handling Service Integration Shell Architecture Cross-App Routing Micro Frontend Orchestration Performance Monitoring Testing Integration	State Management Basics Local Component State Prop Drilling Issues Context API Usage Redux Fundamentals Redux Toolkit Cross-App State Sharing State Synchronization Async State Handling Debugging Tools
Module 7 : Deployment	Module 8 : Testing	Module 9 : Advanced Topics
Deployment Strategies CI/CD Pipelines Independent Deployments Version Management Docker with React Kubernetes Deployments Cloud Hosting Options Environment Configurations Rollback Strategies Monitoring and Alerts	Unit Testing React Jest Framework React Testing Library Integration Testing End-to-End Testing Cypress Framework Contract Testing Test Automation Performance Testing Accessibility Testing	Security Concerns Authentication Patterns Authorization Patterns Performance Optimization Error Boundary Handling Caching Strategies Accessibility Concerns Future Trends Case Studies Hands-On Project