Angular Architectures for Enterprise Application Eğitimi

Eğitim Hakkında

Angular Architectures for Enterprise Application Eğitimi'nde; Manfred Steyer - Angular GDE and Trusted Collaborator in the Angular team - how large-scale enterprise Angular applications can be designed and developed. After the individual exercises, you will have a case study you can use as a template for your own projects. Also, this workshop allows you to evaluate the variety of options you can choose for your own projects.

Neler Öğreneceksiniz

- Structure for Large Applications: Monorepos, Nx and Strategic Design
- Scalable architectures: micro Frontends with Module Federation and Angular Elements
- State management patterns with NGRX and Redux
- Performance Tuning
- Reactive architectures with RxJS
- Customization and White Label Solutions
- Modern security architectures and single sign-on
- Trends and the future of Angular

Eğitim İçeriği

Structure for Large Applications: Monorepos, Nx and Strategic Design

- Plan architectures with Domain Driven Design (DDD)
- CLI workspaces and monorepos with Nx (Nrwl Extensions)
- Develop and distribute reusable npm packages
- Categorization for libraries, modules and components
- Enforce architecture constraints with access restrictions
- Build performance: Incremental builds and tests with the Build Cache
- Integration into the CI process
- Customizable libraries with advanced DI patterns and content projection
- The open/close principle in Angular

Scalable architectures: micro Frontends with Module Federation and Angular Elements

- From strategic design to micro frontends

- Pros and cons of Micro Frontends
- Monorepos vs. multiple repos
- Leverage Webpack Module Federation to load separately compiled and deployed micro frontends
- Dynamic Module Federation
- Sharing dependencies
- Dealing with different versions and version mismatches
- Communication between Micro Frontends
- Cross-framework development with Angular Elements and Web Components

State management patterns with NGRX and Redux

- The state layer and DDD
- NGRX: When to use it or not and alternatives?
- Using the Redux approach with NGRX
- Different types of states
- Building Blocks: Actions, Reducers
- Selectors and view models
- Effects and side effects
- Generate building blocks with schematics
- Manage entities with @ngrx/entities
- Practical handling of immutables
- Facades as the linchpin
- Introduce NGRX gradually
- NGRX and lazy loading of modules
- Local states with the brand new NGRX/Component store

Performance tuning

- Lazy lodging with and without a router
- Data binding performance with OnPush
- AOT and Tree Shaking
- Analyze bundles
- Build performance with the build cache and incremental compilation

Reactive architectures with RxJS

- Reactive thinking and reactive design
- Chaining/piping of operators
- Combination operators
- Higher order observables
- Implicit and Explicit Closing
- Cold and hot observables, and multicasting
- Using subjects
- Error handling
- Debugging

Customization and White Label Solutions

- Adjustments at runtime with lazy loading and module federation
- Compile-time adjustments with libraries and path mappings
- Configure libraries with forRoot and DI

Bonus: modern security architectures and single sign-on

- Connect existing identity solutions such as Active Directory
- Social login (login with Facebook, etc.)
- OAuth 2 as well as 2.1 and OpenId Connect
- JSON Web Tokens (JWT)
- Token refresh
- Single sign out
- Current recommendations of the OAuth 2 Working Group and consequences
- Tokens in the browser vs. security gateways (backend for frontends)

Bonus: Trends and the future of Angular

- A future without NgModules
- Web Components
- Lazy loading of components
- Zoneless change detection