**Developers**

**🔹 1. Core Angular Refresher**

* Angular architecture: components, modules, services
* TypeScript essentials
* Component lifecycle hooks
* Data binding and directives

**🔹 2. Routing & Navigation**

* Router setup and configuration
* Lazy loading modules
* **Route guards**:
  + CanActivate, CanDeactivate, Resolve
  + AuthGuard implementation for login protection
* Dynamic and nested routes

**🔹 3. Authentication & Authorization**

* JWT-based login flow
* Token storage (localStorage/sessionStorage)
* HTTP interceptors for token injection
* Role-based access control
* Secure route handling with guards

**🔹 4. Services & API Integration**

* Centralized HttpService with constants
* REST API consumption
* Error handling and retry logic
* Interceptors for headers and logging

**🔹 5. RxJS & Reactive Programming**

* Observables vs Promises
* Operators: map, switchMap, catchError, debounceTime
* Subjects and BehaviorSubjects
* Async pipe and memory management

**🔹 6. Forms & Validation**

* Template-driven vs Reactive forms
* FormBuilder, FormGroup, FormControl
* Custom validators
* Dynamic forms and conditional fields

**🔹 7. Component Communication**

* Input/Output decorators
* EventEmitter
* Shared services
* ViewChild and ContentChild

**🔹 8. UI & Styling**

* Angular Material / PrimeNG
* Responsive design (Flex Layout, Tailwind)
* Theming and custom components

**🔹 9. Testing**

* Unit testing with Jasmine & Karma
* E2E testing with Cypress or Playwright
* Mocking services and HTTP calls

**🔹 10. Build & Deployment**

* Angular CLI and workspace setup
* Environment configuration
* Production build optimization
* SSR with Angular Universal (optional)

**🔹 11. Advanced Topics**

* Standalone components (Angular 14+)
* Change detection strategies (OnPush)
* Dynamic component loading
* Internationalization (i18n)
* Performance tuning

**🔹 12. Integration with Java Backend**

* RESTful API design and consumption
* JWT authentication
* CORS handling
* File upload/download
* WebSocket integration (if applicable)

**🔹 13. Collaboration with Frontend Teams**

* API contracts and Swagger/OpenAPI
* Shared DTOs and models
* Versioning and backward compatibility
* CI/CD pipelines
* Git workflows and code reviews