

Reactive Programming with RxJS

Audience Course Reactive Programming with RxJS

The course Reactive Programming with RxJS is intended for JavaScript Developers who want to learn how to use the reactive RxJS library in frontend applications.

Prerequisites Course Reactive Programming with RxJS

Experience with JavaScript and Web Applications is required to participate in this course. Knowledge of how asynchronous code works helps with understanding.

Realization Training Reactive Programming with RxJS

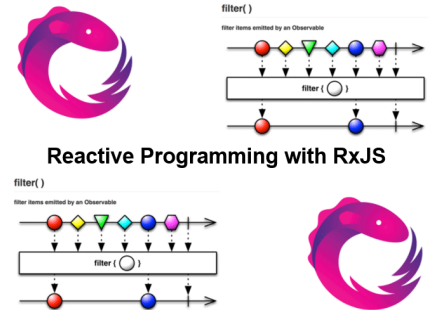
The course Reactive Programming with RxJS is a hands-on course. Theory is interchanged with demos and practical exercises and is explained with the use of presentation slides.

Certification course Reactive Programming with RxJS

After successfully completing the training, attendants will receive a certificate of participation in the course Reactive Programming with RxJS.

Duration: 1 day

Price: € 749

[Open Schedule](#)


Content Course Reactive Programming with RxJS

The course Secure C# Web Development covers how C# web applications can be optimally secured and the best practices therein. Attention is paid to the top 10 OWASP security vulnerabilities, their consequences, their prevention and also how security can be implemented programmatically.

Intro Security

The course Secure C# Web Development starts with a discussion of the main security risks as identified by the Open Worldwide Application Security Project (OWASP).

Broken Access Control

Subsequently security risks associated with access control through authentication and authorization are treated. Role Based Access Control, Access Control Lists and the implementation of Session Management are the topics that are covered.

Cryptographic Failures

Security problems with encryption are also on the course schedule. Attention is paid to problems with the use of weak keys, hard coding of secrets, insufficient verification of signatures and possible side-channel attacks.

Injection risks

An important security threat is the various forms of injection that are lurking. SQL Injection, Cross-Site Scripting and XPath injection are discussed, as well as prevention measures.

Incorrect Design

Then attention is paid to security problems that are the result of incorrect design, such as the lack of input validation and unsafe session management. Insufficient protection against Cross Site Request Forgery is treated as well.

Configuration Errors

Errors in the configuration can lead to security problems also. Various examples of this, such as the use of default credentials and weak password policies, are covered.

Obsolete Components

Security risks resulting from components that are no longer up to date, incorrectly configured or malicious packages and cryptographic weaknesses, are also on the program of the course Secure C# Web Development.

Authentication errors

Attention is also paid to common errors in authentication, such as weak password policies, overly permissive access controls and the lack of multi-factor authentication. And finally security flaws in monitoring and logging are discussed.

Modules Course Reactive Programming with RxJS

Module 1 : Intro Reactive Programming	Module 2 : RxJS Fundamentals	Module 3 : RxJS Operators
What is Reactive Programming? What is Reactive? Reactive Systems Reactive Manifesto What is Reactive Programming? Rx Library Family Streams Observing Streams Reactive Extensions Subscribing to Observables Rx Operators Asynchronous Listening Observer Pattern Why Reactive Programming? Functional Reactive Programming	RxJS Intro Observable and Observer Observer next Callback Observer error Callback Observer complete Callback Subscribe and Unsubscribe Operators in RxJS of and map Operators pipe, filter and map bufferWhen Operator Handling HTTP Requests switchMap Operator Combining Observables RxJS Subjects Unicasting and Multicasting	from Observables with from map and map Properties interval, filter and of startWith and endWith pairwise take and takeUntil pluck bufferCount and bufferTime debounceTime distinctUntilChanged tap or do delay ajax catch or catchError
Module 4 : Combination Operators		
merge and mergeAll mergeMap or flatMap Why mergeMap? switchMap Why switchMap? concatMap concat and zip scan sequenceEqual iif and share shareReplay combineAll combineLatest forkJoin withLatestFrom		