

Reactive Programmeren met RxJS

Doelgroep Cursus Reactive Programmeren met RxJS

De cursus Reactive Programmeren met RxJS is bedoeld voor JavaScript Developers die willen leren hoe je de reactive RxJS library gebruikt in frontend applicaties.

Voorkennis Cursus Reactive Programmeren met RxJS

Om aan deze cursus te kunnen deelnemen is ervaring met JavaScript en Web Applicaties vereist. Kennis van de werking van asynchrone code helpt bij de begripsvorming.

Uitvoering Training Reactive Programmeren met RxJS

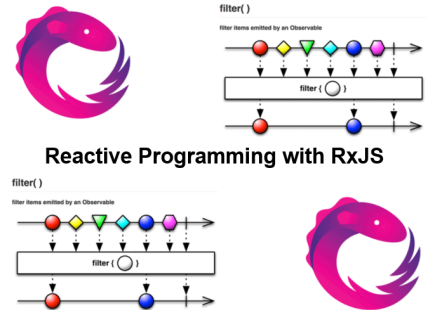
De cursus Reactive Programmeren met RxJS is een hands-on cursus. Theorie wordt afgewisseld met demos en praktische oefeningen en wordt besproken aan de hand van presentatie slides.

Certificering cursus Reactive Programmeren met RxJS

De deelnemers krijgen na het goed doorlopen van de training een certificaat van deelname aan de cursus Reactive Programmeren met RxJS.

Duur: 1 dag

Prijs: € 749

[Open Rooster](#)


Inhoud Cursus Reactive Programmeren met RxJS

De cursus Secure C# Web Development behandelt hoe C# web applicaties optimaal kunnen worden beveiligd en de best practices daarin. Aandacht wordt besteed aan de top 10 van de OWASP security vulnerabilities, hun gevolgen en ook hoe security programmatisch kan worden geïmplementeerd.

Intro Security

De cursus Secure C# Web Development gaat van start met bespreking van de voornaamste security risico's zoals vastgesteld door het Open Worldwide Application Security Project (OWASP).

Broken Access Control

Vervolgens wordt ingegaan op security risico's bij access control middels authenticatie en autorisatie. Role Based Access Control, Access Control Lists en de implementatie van Session Management zijn daarbij onderwerpen die aan de orde worden gesteld.

Cryptographic Failures

Ook security problemen met encryptie passeren de revue. Hierbij wordt aandacht besteed aan problemen bij het gebruik van weak keys, het hard coderen van secrets, het onvoldoende verifiëren van signatures en mogelijke side-channels attacks.

Injectie risico's

Een belangrijke security bedreiging vormen de diverse vormen van injectie die op de loer liggen. Hierbij worden onder andere SQL Injection, Cross-Site Scripting en XPath injection besproken evenals de maatregelen ter preventie.

Onjuist Design

Vervolgens wordt ingegaan op security problemen die het gevolg zijn van een verkeerd design zoals het gebrek aan input validatie en onveilig session management. Ook onvoldoende bescherming tegen Cross Site Request Forgery komt dan aan de orde.

Configuratie Fouten

Eveneens wordt besproken hoe fouten in de configuratie tot security problemen kunnen leiden. Diverse voorbeelden hiervan zoals het gebruik van default credentials en weak password policies komen hierbij ter sprake.

Verouderde Componenten

Ook wordt aandacht besteed security risico's die het gevolg zijn van componenten die niet meer up to date zijn, verkeerd geconfigureerde of kwaadaardige packages en cryptografische zwakheden.

Authenticatie fouten

Vervolgens komen veel voorkomende fouten bij authenticatie zoals weak password policies, overly permissive access controls en het gebrek aan multi-factor authenticatie aan bod. En tenslotte worden security gebreken bij monitoring en logging besproken.

Modules Cursus Reactive Programmeren met 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		