

KnockoutJS

Overview:

Knockout.js is a JavaScript MVVM(Model View View Model) library with increasing popularity. This library helps to create rich, responsive display and user editor interface which works directly with the web application's underlying data model. Elegant dependency tracking, Declarative bindings, Trivial extendability are some of the important features of KO. Using KO with any web application is very simple, straightforward and clean. It is very powerful in the context of dynamic UI creation

Course Objectives:

- ❖ Understanding Model-View-ViewModel pattern using Knockout.js
- ❖ Understanding Data Binding Techniques.
- ❖ Creating Custom Bindings.
- ❖ Debugging Techniques.
- ❖ Working with Forms.
- ❖

Pre-requisite / Target Audience:

- ❖ If you have ever wanted to create rich, responsive, and dynamic user interfaces without mixing a lot of script, this course will give you everything you need to accomplish that goal with Knockout.JS. Users will need basic knowledge of JavaScript and HTML.

Module 1: Introduction and First Example

- ❖ Introduction
- ❖ KO Main Features
- ❖ Benefits
- ❖ What is MVVM
- ❖ First Example

Module 2: Bindings and Observables

- ❖ Understanding Observable Property
- ❖ Reading and writing Observables
- ❖ Notify Subscribers
- ❖ Delay change notifications
- ❖ Observable Arrays

- ❖ Computer Observable

Module 2: Binding Text and Appearance

- ❖ Visible Binding
- ❖ Text Binding
- ❖ Html Binding
- ❖ CSS Binding
- ❖ Style Binding
- ❖ Attr Binding

Module 3: Control Flow and Containerless Bindings

- ❖ **if** binding
- ❖ **ifnot** binding
- ❖ **with** binding
- ❖ **foreach** binding

Module 4: Binding Form Elements

- ❖ Click Binding
- ❖ Passing data to method
- ❖ Event Binding
- ❖ Submit Biding
- ❖ HasFocus Biding
- ❖ Checked Binding
- ❖ Options and SelectedOptions Binding
- ❖ Unobstrusive Event Handling

Module 5: Template Bindings

- ❖ Understanding Template Binding
- ❖ Rendering Named Template
- ❖ Foreach option with a named template
- ❖ Using as to give an alias to foreach
- ❖ Dynamically choosing a template

Module 6: Component Bindings

Define a Component

Register a Component

Specifying a Template

Using Component without a container

Login Component

Using Template

Module 7: Custom Binding

- ❖ Writing Component
- ❖ Using Custom Component

Module 8: Handling JSON objects

- ❖ JSON ViewModel Interoperability
- ❖ Updating View Model using JSon
- ❖ KO Mapping PlugIn
- ❖ Unmapping to JavaScript Objects
- ❖ Working with Json Strings

Module 9: KO with Web API

- ❖ Example to perform CRUD operations using KO and WebAPI

At the end of the course participants will be able to

1. Use the Model-View-ViewModel software design pattern to build amazing web applications and powerful user interfaces
2. Configure your application to track the changes in the UI and update data automatically
3. Enhance the functionality of your app and make it more interactive by extending the components of your code