

Redux ile React kullanarak ASP.NET Core Geliştirmeye Giriş

React with Redux eğitimini kullanarak ASP.NET Core Geliştirmeye Giriş, katılımcılara istemci tarafında JavaScript ve arka uçta ASP.NET Core tarafından desteklenen modern bir web uygulaması oluşturmak için gerekli becerileri öğretir.

Hedefler:

- .NET Core platformunun hedeflerini ve avantajlarını anlamayı
- Uygulama mimarisi ve veri erişim teknolojisi seçimi hakkında doğru kararlar vermeyi
- REST tarzı bir mimari elde etmek için ASP.NET Core'un yönlendirme sistemini kullanmayı
- Modern bir Web API aracılığıyla verileri kullanılabilir hale getiren bir hizmet oluşturma deneyimi kazanmayı
- JavaScript paket yöneticisi kullanmayı
- Sınıflar, modüller ve ok işlevleri dahil olmak üzere yeni JavaScript dil özelliklerini anlamayı
- React'in ne olduğunu ve neden yararlı olduğunu açıklamayı
- Bir React uygulamasının temel mimarisini keşfetmeyi
- JSX ve Virtual DOM hakkında derinlemesine bir anlayış kazanmayı
- Etkileşimli arayüzler oluşturmak için React bileşenlerini kullanmayı
- Kontrollü bileşenleri kullanarak formlar oluşturun ve doğrulamayı
- Verileri okumak veya değiştirmek için HTTP çağrılarını yapmayı
- Basit ve karmaşık yönlendirmeyi yapılandırmayı
- Uygulamanın durumunu yönetmek için Redux'u kullanmayı

- React ve Redux'u birlikte kullanmayı
- React ve Redux en iyi uygulamalarını uygulamayı
- Jest ve Enzyme kullanarak React için birim testleri yazmayı

Topics:

- Giriş
 - .NET Core SDK
 - Kurulum
 - Sürüm Yönetimi
 - Komut Satırı Arayüzü (CLI)
 - Merhaba Dünya Uygulaması
 - C # Kodlaması için Visual Studio Kodunu Kullanma
 - İsteğe bağlı: C # Kodlaması için Visual Studio 2019'u Kullanma
 - İsteğe bağlı: C # Kodlaması için Mac için Visual Studio'yu Kullanma
- ASP.NET Core Uygulama Mimarisi
 - NuGet Paketleri ve Meta Paketleri
 - Uygulama Başlangıcı
 - Barındırma Ortamları
 - Ara Yazılım ve İstek İşleme Hattı
 - Hizmetler ve Bağımlılık Enjeksiyonu
- Uygulama Yapılandırması
 - Hizmetleri Yapılandırın ve Yapılandırın

- Yapılandırma Sağlayıcıları ve Kaynakları
- Yapılandırma API'si
- Options Pattern
- HTTPS, GDPR ve HTTP / 2

- Yönlendirme İsteme

- RESTful Hizmetler
- Uç Nokta Yönlendirme
- Öznitelik Tabanlı Yönlendirme
- Rota Şablonları
- Rota Kısıtlamaları

- Modeller

- Persistence Ignorance
- Nesne-İlişkisel Haritalama
- Entity Framework (EF) Çekirdeği
- Otomatikleştirici

- Kontrolörler

- Sorumluluklar
- Gereksinimler ve Kurallar
- Bağımlılıklar
- Eylem Sonuçları

- Web API'leri

- Giriş
- CRUD İşlemleri
- Kötü İstekler
- Kaynaklar Arası Kaynak Paylaşımı (CORS)
- Uygulama Durumu
 - İstemci Tarafı ve Sunucu Tarafı
 - HttpContext.Items
 - Oturum Durumu
- Hata yönetimi
 - En İyi Uygulamalar
 - HTTP Hata Durum Kodları
 - Durum Kodu Sayfaları
 - Geliştirici İstisna Sayfası
 - İstisna Filtreleri
- Logging
 - Yapılandırma
 - ILogger
 - Serilog ve Seq
- Test yapmak
 - Birim Testi
 - xUnit

- Test Kontrolörleri
- Entegrasyon Testi
- React ve Redux'a Giriş
 - React nedir?
 - React hangi sorunu çözer?
 - Gelişim Ekosistemi
 - Diğer çerçevelere karşı tepki verin
- Geliştirme araçları
 - React App proje oluşturucu oluşturun
 - React Developer Tools
 - Bir React Uygulamasını Çalıştırma ve Hata Ayıklama
 - Node.js'nin Rolü
 - React ve ReactDOM Amacı
- Fonksiyonel Bileşenler
 - Bileşenler nedir?
 - Öğe ve JSX Oluşturun
 - JSX'in faydaları
 - Parça
 - JavaScript Ok İşlevleri
 - ES2015 Modülleri
 - JSX ve İfadeler

- Veri Koleksiyonlarını Görüntüleme
- JavaScript Dizi Haritaları ve React Anahtarları
- Props ile Veri Aktarma
- PropTypes ile Props Doğrulama
- Varsayılan Sahne
- Not Kullanımı

- Sınıf Tabanlı Bileşenler

- JavaScript Sınıfları ve Uzantıları
- Durumu Yapılandırma
- Yaşam Döngüsü Yöntemleri
- Google Performans Aracı
- Olay İşleyicilerin Bağlamı
- Sınıf Özellikleri ve Sınıf Ok İşlevleri
- Sınıflarda PropTypes ve Default Props

- Hooks

- Overview of Hooks
- State Hook
- Effect Hook
- Ref Hook
- Geri Arama Kancası

- Gelişmiş Bileşenler

- Composition vs. Inheritance
- Patterns:: Uzmanlık, Kapsama ve Yüksek Dereceli Bileşenler
- Durumu Yukarı Kaldırma
- Yönlendirme Referansları
- Bağlam
- Redux
 - Uygulama Durumunu Yönetme
 - Redux'un Üç İlkesi
 - Saf İşlevler
 - Redüktör Fonksiyonları
 - İndirgeyici İşlevleri Oluşturma
 - Sevk İşlemleri
 - Aksiyon Yaratıcıları
- React'i Redux'a Bağlama
 - React-Redux ile React'i Redux'a Bağlayın
 - React-Redux Yüksek Dereceli Bileşenler
 - React-Redux Kancaları
 - Durum Seçicilerini Kullanma
 - Durum Seçicilerini Optimize Etme
- React'i ASP.NET Core REST API'ye Bağlama
 - JavaScript İncelemesi: Callbacks, Promises & Async / Await

- Fetch API'yi ASP.NET Core REST API ile kullanma
- Saga kullanarak Asenkron İşlemler ve React / Redux
- JavaScript Oluşturucular
- Sagas Yardımcıları
- Bildirim Etkileri
- Hata yönetimi
- Redux'a bağlanılıyor
- Birim Testi
 - Jest kullanma
 - Testleri ve Test Takımlarını Düzenleme
 - Testlerin Kurulumu ve Kaldırılması
 - Beklenti ile İddia Etme
 - Casusları Kullanma
 - Anlık Görüntü Testi
 - DOM Testi
 - Sığ Test
 - Kod Kapsamı Raporları Oluşturma
 - React Unit Testing
 - Redux Birim Testi
- React Yönlendirici
 - Yönlendirme nedir?

- Eyalet olarak URL
- React Yönlendirici Kancaları
- Rotaları Yapılandırma
- Sayfa Deseni
- Hatalı sayfa
- Yönlendirmeler
- Animasyonlu Geçişler
- İç İç Rotalar
- Gelişmiş Formlar
 - Formik nedir?
 - React ve Formlarla İlgili Zorluklar
 - Formik Yüksek Sipariş Bileşenleri
 - Formik Kancalar
 - Form Düzeyinde Doğrulama
 - Alan Düzeyinde Doğrulama
 - Eşzamanlı ve Eşzamansız Doğrulama
 - Form gönderme
 - Form Gönderme Aşamaları
- Doğrulama
 - Giriş
 - ASP.NET Core Kimliği

- Çerez Middleware
- Yetki
- Talep Bazlı Yetkilendirme
- React Giriş Formu
- Oturum Açma Hatası İşlemi
- Yetkilendirmeyi Yönlendirme ile Entegre Etme