React Başlangıç

React'i Deneyin

React, baştan sona aşamalı olarak benimsenmesi için tasarlandı. React'i ihtiyacınız kadar az veya daha fazla kullanabilirsiniz. React'ın tadına bakmak, basit bir HTML sayfasına biraz etkileşim katmak veya karmaşık bir React destekli uygulama başlatmak isteyip istemediğinize göre, bu bölümdeki bağlantılar başlamanıza yardımcı olacaktır.

Çevrimiçi Vakit Geçirme Alanları

React ile vakit geçirmek istiyorsanız, çevrimiçi bir kod oyun alanını kullanabilirsiniz. CodePen, CodeSandbox veya Stackblitz seçeneklerinden birisiyle bir Merhaba Dünya şablonu deneyin.

Kendi metin düzenleyicinizi kullanmayı tercih ediyorsanız bu HTML dosyasını da indirebilir, düzenleyebilir ve sisteminizde kurulu bir tarayıcı ile açabilirsiniz. Yavaş bir kod işleme dönüşümü yapar, bu yüzden bunu basit gösteriler için kullanmanızı tavsiye ederiz.

Bir Web Sitesine React Ekleyin

Bir dakika içinde bir HTML sayfasına React ekleyebilirsiniz. Daha sonra içeriğini aşamalı olarak genişletebilir veya birkaç dinamik araca dâhil edebilirsiniz.

Yeni bir React Uygulaması Oluşturun

Bir React projesi başlatırken, script etiketleri olan basit bir HTML sayfası, hâlâ en iyi seçenek olabilir. Kurulumu sadece bir dakika sürer!

Uygulamanız büyüdükçe, daha entegre bir kurulum düşünebilirsiniz. Daha büyük uygulamalar için önerdiğimiz birkaç JavaScript araç-zinciri vardır. Her biri çok az veya sıfır yapılandırma ile çalışabilir ve zengin React ekosisteminden tam olarak yararlanmanızı sağlar.

React Öğrenin

İnsanlar farklı geçmişlerden ve farklı öğrenme tarzlarıyla React'e geliyorlar. Tercih ettiğiniz yaklaşım ister teorik, ister pratik olsun, bu bölümü faydalı bulacağınızı umuyoruz

Yaparak öğrenmeyi tercih ediyorsanız, pratik öğreticimizle başlayın.

Kavramları adım adım öğrenmeyi tercih ediyorsanız, ana kavramlar rehberimizle başlayın.

Bilinmeyen herhangi bir teknoloji gibi, React de bir öğrenme eğrisine sahiptir. Pratik yaparak ve biraz da sabır ile üstesinden geleceksiniz.

İlk Örnekler

React anasayfa canlı editörü ile birkaç küçük React örneği içerir. Henüz React hakkında bir şey bilmiyorsanız bile kodlarını değiştirmeyi deneyin ve sonucu nasıl etkilediğinizi görün.

Yeni Başlayanlar İçin React

React anlatımlarının hızlı geçtiğini düşünüyorsanız, Tania Rascia ile React'e genel bakışı inceleyin. En önemli React konseptlerini ayrıntılı, acemi dostu bir şekilde sunar. İşiniz bittiğinde anlatımları bir kez daha deneyin!

Tasarımcılar İçin React

Tasarım tecrübeniz varsa, bu kaynaklar başlamanız için uygundur.

JavaScript Kaynakları

React anlatımları, JavaScript dilinde programlama konusunda biraz bilgi sahibi olduğunuzu varsaymaktadır. Uzman olmanız gerekmiyor, ancak hem React hem de JavaScript'i aynı anda öğrenmek zor.

Seviyenizi kontrol etmek için JavaScript'e genel bakış sayfasına gitmenizi öneririz. 30 dakika ile bir saat arasında sürecek ama React öğrenirken daha rahat ve özgüvenli hissedeceksiniz.

İpucu

JavaScript'te bir konuda ne zaman kafanız karışırsa, MDN ve javascript.info kontrol etmek için harika web siteleridir. Ayrıca yardım isteyebileceğiniz topluluk destek forumları da vardır.

Pratik Öğretici

Yaparak öğrenmeyi tercih ediyorsanız pratik öğreticimize göz atın. Bu derste, React ile bir Tic Tac Toe oyunu inşa ediyoruz. Oyun inşa etmediğiniz için bu kısmı atlamak isteyebilirsiniz - ama bir şans verin. Derste öğreneceğiniz teknikler herhangi bir React uygulaması oluşturmak için temeldir ve uygulamanın ileri seviyesi size daha derin bir anlayış verecektir.

Adım adım rehber

Kavramları adım adım öğrenmeyi tercih ederseniz ana kavramlar rehberi başlamak için en iyi yerdir. Her bir sonraki bölüm, önceki bölümlerde sunulan bilgilere dayanıyor, böylece ilerledikçe hiçbir şeyi kaçırmayacaksınız.

React'te Düşünmek

Pek çok React kullanıcısı React'te Düşünmek bölümünü nihayet "kafama dank etti" anı olarak belirtir. Muhtemelen React'in en eski 'oyun nasıl bitirilir rehberi' ama hala aynı derecede alakalı.

API Referansı

Bu bölüm, belirli bir React API hakkında daha fazla ayrıntı öğrenmek istediğinizde faydalıdır. Örneğin, React.Component API reference size setState()'in nasıl çalıştığını ve hangi 'yaşam döngüsü' metodunun işinize yarayacağı konusunda ayrıntılı bilgi verebilir.

Terimler sözlüğü ve SSS

Terimler sözlüğü React anlatımlarında göreceğiniz en yaygın terimlerin bir özetini içerir. AJAX isteği yapma, bileşen durumu, ve dosya yapısı dâhil olmak üzere ortak konular hakkında kısa soru ve cevaplara ayrılmış bir SSS bölümü de bulunmaktadır.

Haberdar olun

React blog, React ekibinden yapılan güncellemelerin resmi kaynağıdır. Sürüm notları veya itiraz bildirimleri dâhil olmak üzere, önemli olan her şey önce buraya gönderilecektir.

Twitter'da @reactjs hesabını da takip edebilirsiniz, ama sadece blogu okursanız önemli hiç bir şeyi kaçırmazsınız.

Her React sürümü kendi blog gönderisini hak etmiyor ama her sürüm için yapılan değişikliklerin ayrıntılarını React reposundaki CHANGELOG.md dosyasında ve Sürümler sayfasında bulabilirsiniz.

Versiyon Anlatımı

Bu bölüm her zaman React'in en son kararlı versiyonunu yansıtır. React 16'dan beri, açıklamaların eski sürümlerini ayrı bir sayfada bulabilirsiniz. Geçmiş sürümlere ait açıklamaların yayınlanma tarihinde anlık olarak görüntülendiğini ve sürekli güncellenmediğini unutmayın.

Bir Web Sitesine React Eklemek

İhtiyacınız kadar az veya daha fazla React kullanın.

React, baştan sona aşamalı olarak benimsenmesi için tasarlandı. React'i ihtiyacınız kadar az veya daha fazla kullanabilirsiniz. Belki de sadece var olan bir sayfaya biraz "interaktif parçalar" eklemek istiyorsunuz. React bileşenleri bunu yapmak için harika bir yoldur.

Web sitelerinin çoğu, tek sayfalı uygulamalar değildir ve olması da gerekmez. Hiçbir kurulum aracı olmadan sadece birkaç satır kod ile web sitenizin küçük bir bölümünde React'i deneyin. Daha sonra içeriğini aşamalı olarak genişletebilir veya sadece birkaç dinamik bileşen olarak tutabilirsiniz.

Bir Dakikada React Ekleyin

İsteğe bağlı: JSX ile React'i deneyin (ek pakete gerek yok!)

Bir Dakikada React Eklemek

Bu bölümde, mevcut bir HTML sayfasına nasıl React bileşeni ekleneceğini göstereceğiz. Kendi web sitenizle birlikte takip edebilir veya pratik yapmak için boş bir HTML dosyası oluşturabilirsiniz.

Karmaşık bir araç veya yükleme gereksinimi olmayacak. Bu bölümü tamamlamak için sadece bir internet bağlantısına ve bir dakikanıza ihtiyacınız var.

Adım 1: HTML koduna bir Div Ekleme

İlk önce, düzenlemek istediğiniz HTML sayfasını açın. React ile bir şey görüntülemek istediğiniz yeri işaretlemek için boş bir etiketi ekleyin. Örneğin:

```
<!-- ... mevcut HTML ... -->

<div id="like_button_container"></div>

<!-- ... mevcut HTML ... -->
```

Bu 'e özgün bir id HTML özelliği verdik. Bu id, daha sonra JavaScript kodundan bu 'i bulmamıza ve içinde bir React bileşeni göstermemize izin verir.

İpucu

<body> etiketinin içinde herhangi bir yere böyle bir yerleştirebilirsiniz. Tek bir sayfada istediğiniz kadar bağımsız DOM konteyneriniz olabilir. Bunlar genellikle boştur. React, DOM konteynerlerinin içindeki mevcut tüm içeriği değiştirir.

Adım 2: Script Etiketlerini Ekleyin

Daha sonra, </body> etiketini kapatmadan hemen önce HTML sayfasına üç <script> etiketi ekleyin:

```
  <!-- ... diğer HTML ... -->

  <!-- React'i yükle. -->

  <!-- Not: yayınlama için hazırlanırken, "development.js" yi "production.min.js" ile değiştirin -->

  <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>

  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>

  <!-- React bileşenimizi yükleyin. -->

  <script src="like_button.js"></script>

</body>
```

İlk iki etiket React'i yükler. Üçüncüsü, bileşen kodunuzu yükleyecektir.

Adım 3: Bir React Bileşeni Oluşturun

HTML sayfanızın yanına like_button.js adlı bir dosya oluşturun.

Bu başlangıç kodunu açın ve oluşturduğunuz dosyaya yapıştırın.

İpucu

Bu kod, LikeButton adı verilen bir React bileşenini tanımlar. Henüz anlamadıysanız endişelenmeyin. React'in yapı taşlarını daha sonra uygulamalı eğitim ve ana kavramlar rehberinde ele alacağız. Şimdilik sadece ekranda gösterelim!

Başlangıç kodundan sonra, like_button.js'in en altına iki satır ekleyin :

```
// ... yapıştırdığınız başlangıç kodu ...

const domContainer = document.querySelector('#like_button_container');

const root = ReactDOM.createRoot(domContainer);

root.render(e(LikeButton));
```

Bu üç kod satırı ilk adımda HTML'e eklediğimiz 'i bulur ve ardından içinde React bileşeni olan "Beğen" düğmesini gösterir.

Bu kadar!

Dördüncü adım yok. Web sitenize ilk React bileşenini eklediniz bile.

React'i entegre etmekle ilgili daha fazla İpucu için sonraki bölümlere göz atın.

İpucu: Bir Bileşeni Yeniden Kullanma

Genellikle, HTML sayfasındaki React bileşenlerini birden fazla yerde görüntülemek isteyebilirsiniz. "Like" düğmesini üç kez görüntüleyen ve bazı verileri ona ileten bir örnek:

Not

Bu strateji çoğunlukla, sayfanın React destekli bölümleri birbirinden izole edilirken kullanışlıdır. Bunun yerine React kodunun içinde bileşen kompozisyonu kullanmak daha kolaydır.

İpucu: Canlı Ortam İçin JavaScript'i Küçültün

Web sitenizi yayına almadan önce, küçültülmemiş JavaScript'in sayfanızı kullanıcılarınız için önemli ölçüde yavaşlatabileceğine dikkat edin.

Uygulama komut dosyalarını küçültürseniz ve yayınlamaya hazır hale getirilen HTML'in de React'in production.min.js içinde biten sürümlerini yüklediğinden emin olursanız siteniz yayına hazır olur:

```
<script src="https://unpkg.com/react@18/umd/react.production.min.js" crossorigin></script>

<script src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js" crossorigin></script>
```

JavaScript dosyalarınız için bir küçültme adımınız yoksa ayarlamanın bir yolu budur.

İsteğe bağlı: React'i JSX ile deneyin

Yukarıdaki örneklerde, yalnızca tarayıcılar tarafından doğal olarak desteklenen özelliklere itibar ettik. Bu yüzden React'e ne göstereceğini söylemek için bir JavaScript fonksiyon çağrısı kullandık:

```
const e = React.createElement;

// Bir "Like" <button>'u göster

return e(

 'button',

 { onClick: () => this.setState({ liked: true }) },

 'Like'

);
```

Bunun yerine, React'te JSX kullanma seçeneği de mevcuttur:

```
// Bir "Like" <button>'u göster

return (

 <button onClick={() => this.setState({ liked: true })}>

  Like

 </button>

);
```

Bu iki kod parçacığı eşdeğerdir. JSX tamamen isteğe bağlı olsa da, hem React'i hem de diğer kütüphaneleri kullanan birçok kişi, kullanıcı arayüzü kodu yazmak için JSX'i yararlı bulmaktadır.

Bu çevrimiçi dönüştürücüyü kullanarak JSX ile oynayabilirsiniz.

JSX'i hızlıca deneyin

JSX'i projenizde denemenin en hızlı yolu, bu <script> etiketini sayfanıza eklemektir:

```
<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
```

Şimdi JSX'i, herhangi bir <script> etikete type="text/babel" niteliğini ekleyerek kullanabilirsiniz. İşte indirebileceğiniz ve JSX ile oynayabileceğiniz örnek bir HTML dosyası.

Bu yaklaşım, öğrenmek ve basit demolar oluşturmak için iyidir. Ancak, web sitenizi yavaşlatır ve

uygulamayı yayınlamaya uygun değildir. İlerlemeye hazır olduğunuzda, bu yeni <script> etiketi ve eklediğiniz type="text/babel" özelliklerini kaldırın. Bunun yerine, bir sonraki bölümde tüm <script> etiketlerinizi otomatik olarak dönüştürmek için bir JSX ön-işleyici kuracaksınız.

Bir projeye JSX ekleme

Bir projeye JSX eklemek, bir paketleyici veya geliştirme sunucusu gibi karmaşık araçlar gerektirmez. Temel olarak, JSX eklemek bir CSS ön işleyicisi eklemek gibi bir şeydir. Tek gereksinim, bilgisayarınızda Node.js'in yüklü olması.

Terminal içinde proje klasörünüze gidin ve şu iki komutu yapıştırın:

Adım 1: npm init -y komutunu çalıştır (başarısız olursa, bu düzeltmeye bakınız.)

Step 2: npm install babel-cli@6 babel-preset-react-app@3 komutunu çalıştır

İpucu

Burada sadece JSX ön-işleyici yüklemek için npm kullanıyoruz; başka hiçbir şey için buna ihtiyacınız olmayacak. Hem React hem de uygulama kodu, <script> etiketi olarak değişiklik yapılmadan kalabilir.

Tebrikler! Projenize bir yayına hazır JSX kurulumu eklediniz bile.

JSX Ön-işleyicisini Çalıştırın

src adında bir klasör oluşturun ve bu terminal komutunu çalıştırın:

npx babel --watch src --out-dir . --presets react-app/prod

Not

npx bir harf hatası değildir. npm 5.2+ ile birlikte gelen bir paket çalıştırma aracıdır..

Eğer "You have mistakenly installed the babel package" şeklinde bir hata mesajı görürseniz, bir önceki adımı atlamış olabilirsiniz. Aynı klasör altında adımı uygulayıp tekrar deneyin.

Bitmesini beklemeyin — bu komut JSX için otomatik bir izleyici başlatır.

Şimdi JSX başlangıç kodu ile src/like_button.js adlı bir dosya oluşturursanız, izleyici tarayıcıya uygun sade JavaScript koduyla oluşturulmuş bir önişlenmiş like_button.js dosyası oluşturur. Kaynak dosyayı JSX ile düzenlediğinizde, dönüştürme otomatik olarak yeniden çalıştırılır.

Bu aynı zamanda eski tarayıcılarda çökme konusunda endişelenmeden, sınıflar (classes) gibi modern JavaScript sözdizimi özelliklerini kullanmanızı sağlar. Az önce kullandığımız araca Babel denir ve bu dokümantasyondan daha fazla bilgi edinebilirsiniz.

Yapı araçlarıyla rahat edeceğinizi fark ederseniz ve onların sizin için daha fazlasını yapmalarını isterseniz, bir sonraki bölümde en popüler ve ulaşılabilir araç serilerinden bazıları açıklanmaktadır. Yapı araçlarını

istemiyorsanız bu script etiketleri de yeterli olacaktır!

Yeni bir React Uygulaması Oluşturun

En iyi kullanıcı ve geliştirici deneyimi için tümleşik araç kullanın.

Bu sayfa, görevlere yardımcı olan bazı popüler araçları anlatır.

Birçok dosya ve bileşene ölçeklendirme.

Npm'den üçüncü parti kütüphanelerin kullanımı.

Yaygın hataların erken fark edilmesi.

JS ve CSS'in geliştirme anında canlı olarak güncellenmesi.

Çıktının canlı ortam (production) için optimize edilmesi.

Bu sayfada tavsiye edilen araçlar başlangıç için ayarlama gerektirmez.

Bir Araca İhtiyacınız Olmayabilir

Eğer yukarıda anlatılan sorunlarla karşılaşmazsanız veya henüz kendinizi JavaScript araçları kullanmak konusunda rahat hissetmiyorsanız, React'i yalın <script> etiketi ile HTML sayfasına eklemek (isterseniz JSX ile birlikte) seçeneğini aklınızda bulundurun.

Bu aynı zamanda, hâlihazırda var olan web sitesine React'i entegre etmenin en kolay yoludur. Eğer yardımcı olabileceğini düşünüyorsanız, her zaman daha büyük bir araç ekleyebilirsiniz!

Tavsiye Edilen Araçlar

React takımı öncelikli olarak şu çözümleri öneriyor:

Eğer React öğreniyorsanız veya yeni bir tek sayfa uygulama oluşturuyorsanız, Create React App kullanın.

Eğer Node.js ile sunucu tarafında işlenen bir sayfa geliştiriyorsanız, Next.js'i deneyin.

Eğer sabit içerikli bir web sitesi, geliştiriyorsanız, Gatsby'yi deneyin.

Eğer bileşen kütüphanesi geliştiriyor veya var olan bir kod temeli ile entegre ediyorsanız, Daha esnek araçlar'ı deneyin.

Create React App

Create React App, React öğrenmek için rahat bir ortamdır ve React ile yeni bir tek sayfa uygulama geliştirmeye başlamanın en iyi yoludur.

En son JavaScript özelliklerini kullanabilmeniz için geliştirme ortamınızı kurar, güzel bir geliştirici deneyimi sağlar ve uygulamanızı canlı ortam (production) için optimize eder. Bilgisayarınızda Node >= 14.0.0 ve

npm >= 5.6 sürümlerinin yüklü olması gerekir. Bir proje oluşturmak için, aşağıda yer alan komutları çalıştırın.

```
npx create-react-app my-app

cd my-app

npm start
```

Not

İlk satırdaki npx bir harf hatası değildir. — npm 5.2+ ile gelen bir paket çalıştırma aracıdır.

Create React App, backend mantığı veya veri tabanlarını idare etmez; sadece frontend geliştirme düzenini oluşturur, yani bunu istediğiniz herhangi bir backend ile kullanabilirsiniz. Arka planda, Babel ve webpack kullanır, fakat bunlar hakkında hiçbir şey bilmeniz gerekmiyor.

Ürün yayınlamaya hazır olduğunuzda, npm run build komutunu çalıştırmak build klasöründe uygulamanızın optimize edilmiş bir derlemesini oluşturur. Create React App hakkında daha fazlasını kendi README'sinden ve kullanıcı rehberinden öğrenebilirsiniz.

Next.js

Next.js, React ile statik ve sunucu tarafından işlenen uygulamalar geliştirmek için popüler ve hafif bir çatıdır. Hazır olarak stillendirme ve yönlendirme çözümleri içerir, ve sunucu ortamı olarak Node.js kullandığınızı varsayar.

Next.js'i resmi rehberinden öğrenin.

Gatsby

Gatsby, React ile statik web siteleri geliştirmenin en iyi yoludur. React bileşenlerini kullanmanıza olanak sağlar, fakat en hızlı yükleme süresini garanti etmek için önceden işlenmiş HTML ve CSS çıktılarını verir.

Gatsby'yi resmi rehberinden veya yeni başlayanlar galerisinden öğrenin.

Daha Esnek Araçlar

Aşağıdaki araçlar daha fazla esneklik ve seçenek sunmaktadır. Bunları daha tecrübeli kullanıcılar için öneriyoruz:

Neutrino; webpack'in gücünü ön ayarların sadeliği ile birleştirir, React uygulamaları ve React bileşenleri için bir ön ayar da içerir.

Nx, yerleşik olarak React, Next.js, Express ve daha fazlası için desteğe sahip, tam donanımlı bir monorepo oluşturma aracıdır.

Parcel; React ile çalışan, hızlı, ayarlama gerektirmeyen bir web uygulama paketleyicisidir.

Razzle; herhangi bir yapılandırma gerektirmeyen ancak Next.js'e göre daha fazla esneklik sunan, bir sunucu tarafında işleme çatısıdır.

Sıfırdan Bir Araç Zinciri Oluşturmak

Bir JavaScript derleme araç zinciri tipik olarak aşağıdakilerden oluşur:

Bir paket yöneticisi, Yarn veya npm gibi. Bu, uçsuz bucaksız üçüncü parti paket ekosisteminden faydalanmanıza ve bunları kolayca yüklemenize ve güncellemenize olanak sağlar.

Bir paketleyici, webpack veya Parcel gibi. Bu, modüler kod yazmanıza ve yazdığınız kodları yükleme zamanını optimize etmek için küçük parçalar halinde beraber paketlemenize olanak sağlar.

Bir derleyici Babel gibi. Bu, yazdığınız modern JavaScript kodunun eski tarayıcılarda da çalışmasını sağlar.

Eğer sıfırdan kendi JavaScript araç zincirinizi kurmayı tercih ederseniz, bazı Create React App fonksiyonelliklerini yeniden oluşturan şu rehbere bir göz atın.

Özel araç zincirinizin ürün için doğru bir şekilde kurulduğunu garanti altına almayı unutmayın.

CDN Bağlantıları

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```

Yukarıdaki sürümler yalnızca geliştirme amaçlıdır ve yayınlamaya uygun değildir. React'in küçültülmüş ve optimize edilmiş yayınlamaya uygun sürümleri şu adreste mevcuttur:

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.production.min.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js"></script>
```

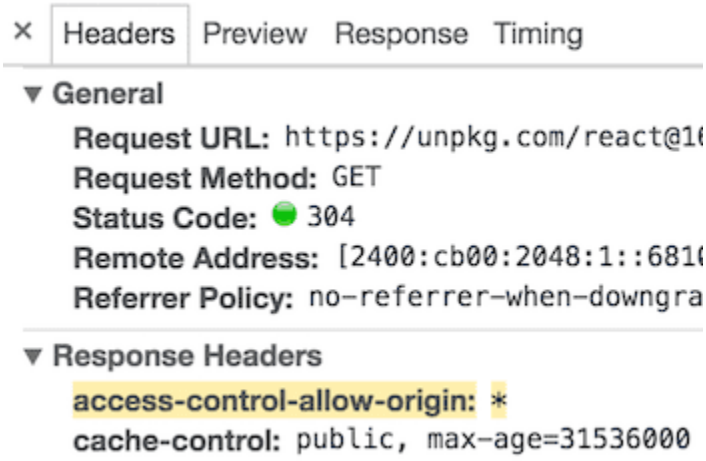
react'in ve react-dom'un belirli bir sürümünü yüklemek için, 18'i istediğiniz sürüm numarası ile değiştirin.

Neden crossorigin Özelliği?

Bir CDN'den React kullanıyorsanız, crossorigin özelliğini ayarlı tutmanızı öneririz:

```
<script crossorigin src="..."></script>
```

Ayrıca, kullandığınız CDN'nin Access-Control-Allow-Origin: * HTTP başlığını ayarladığını doğrulamanızı öneririz:



Yayın Kanalları

React, hata raporlarını dosyalamak, pull request açmak ve RFC göndermek için başarılı bir açık kaynak topluluğuna güvenir. Geri bildirimi teşvik etmek için bazen yayınlanmamış özellikler içeren özel React yapılarını paylaşırız.

Bu belge, en çok çatı, kütüphane veya geliştirici araçları üzerinde çalışan yazılımcılar için uygundur. React'i öncelikle kullanıcılara yönelik uygulamalar oluşturmak için kullanan yazılımcıların yayın öncesi kanallarımız hakkında endişelenmeleri gerekmez.

React'in yayın kanallarının her biri ayrı bir kullanım için tasarlanmıştır:

En Yeni, istikrarlı, semver React sürümleri içindir. React'i npm'den yüklediğinizde elde edeceğiniz şey budur. Bugün kullandığınız kanal bu. Bunu kullanıcılara yönelik tüm React uygulamaları için kullanın.

Sonraki, React kaynak kodu veri havuzunun ana dalını izler. Bunları bir sonraki küçük semver sürümü için aday olarak düşünebilirsiniz. React ve üçüncü taraf projeleri arasındaki entegrasyon testi için bunu kullanın.

Deneysel, kararlı sürümlerde bulunmayan deneysel API'leri ve özellikleri içerir. Bunlar ayrıca ana dalı izler, ancak ek özellik bayrakları açıktır. Gelecek özellikleri yayınlanmadan önce denemek için bunu kullanın.

Tüm sürümler npm'de yayınlanır, ancak yalnızca en yeni kullanımlar anlamsal sürüm oluşturma. Ön yayınlar (Sonraki ve Deneysel kanallarında bulunanlar), kendi içeriklerinin bir karma değerinden oluşturulan sürümlere sahiptir, örneğin İleri için 0.0.0-1022ee0ec ve Deney için 0.0.0-experimental-1022ee0ec.

Kullanıcıya yönelik uygulamalar için resmi olarak desteklenen tek yayın kanalı En Yeni kanalıdır. Sonraki ve Deneysel sürümler yalnızca test amacıyla sağlanmıştır ve En Yeni sürümler için kullandığımız semver protokolünü takip etmediklerinden dolayı sürümler arasında davranışın değişmeyeceğine dair hiçbir garanti vermiyoruz.

Ön sürümleri, istikrarlı sürümler için kullandığımız kayıt defterinde yayınlayarak, unpkg ve CodeSandbox gibi npm iş akışını destekleyen birçok araçtan yararlanabiliyoruz. .

En Yeni Kanalı

En Yeni, istikrarlı React sürümleri için kullanılan kanaldır. Npm'deki latest etiketine karşılık gelir. Gerçek kullanıcılara gönderilen tüm React uygulamaları için önerilen kanaldır.

Hangi kanalı kullanmanız gerektiğinden emin değilseniz, kullanmanız gereken kanal En Yeni kanalıdır. Bir React yazılımcısıysanız, zaten kullandığınız budur.

Güncellemelerin son derece istikrarlı olmasını bekleyebilirsiniz. Sürümler semantik versiyonlama şemasını takip eder. Sürüm oluşturma politikamız sayfasından istikrarlılık ve aşamalı geçiş taahhüdümüz hakkında daha fazla bilgi edininiz.

Sonraki Kanalı

Sonraki kanal, React kaynak kodu ana dalını izleyen bir yayın öncesi kanaldır. Bir Sonraki kanaldaki yayın öncesi sürümleri, En Son kanal için yayın adayları olarak kullanıyoruz. İleri kanalını , daha sık güncellenen En Yeni kanalının üstkümesi olarak düşünebilirsiniz.

En son yayınlanan Sonraki sürüm ile en Yeni sürüm arasındaki fark, iki küçük dönem sürümü arasında bulacağınız farkla yaklaşık olarak aynı değere sahiptir. Ancak, Sonraki kanalı, semantik sürümlemeye uymaz. Sonraki kanalda birbirini izleyen sürümler arasında zaman zaman değişiklik yapılması beklenmelidir.

Sonraki kanalındaki sürümler npm'de next etiketi ile yayınlanır. Sürümler, yapının kendi içeriğinin bir karma değerinden oluşturulur; 0.0.0-1022ee0ec.

Sonraki Kanalını Entegrasyon Testi için Kullanma

Sonraki kanalı, React ve diğer projeler arasındaki entegrasyon testini desteklemek için tasarlanmıştır.

React'teki tüm değişiklikler kamuya açıklanmadan önce kapsamlı dahili testlerden geçer. Bununla birlikte, React ekosisteminde kullanılan sayısız ortam ve konfigürasyon vardır ve her birine karşı test etmemiz mümkün değildir.

Üçüncü taraf React çatısının, kitaplığının, geliştirici aracının veya benzer altyapı tipi bir projenin sahibiyseniz, test paketinizi en yeni sürümlere karşı düzenli olarak çalıştırarak, React'i kullanıcılarınız ve tüm React topluluğu için istikrarlı tutmamıza yardımcı olabilirsiniz. İlgileniyorsanız şu adımları izleyiniz:

Tercih ettiğiniz sürekli entegrasyon platformunu kullanarak bir cron işi oluşturun. Cron işleri hem CircleCI hem de Travis CI tarafından desteklenmektedir.

Cron işinde, npm'de next etiketini kullanarak React paketlerinizi Sonraki kanalındaki en son React sürümüne güncelleyin. Npm cli kullanarak:

```
npm update react@next react-dom@next
```

Ya da yarn ile:

```
yarn upgrade react@next react-dom@next
```

Test paketinizi güncellenmiş paketlere karşı çalıştırın.

Her şey geçerse harika! Projenizin bir sonraki küçük React sürümüyle çalışmasını bekleyebilirsiniz.

Beklenmedik bir şekilde herhangi bir şeyin çalışmaması durumunda, lütfen bir sorun bildirerek bizimle iletişime geçiniz.

Next.js bu iş akışını kullanan projelerden birisidir. (Kelime oyunu yok! Gerçekten!) Örnek olarak CircleCI yapılandırması'na başvurabilirsiniz.

Deneysel Kanal

Sonraki kanalı gibi, Deneysel kanal da React deposunun ana dalını izleyen bir yayın öncesi kanaldır. Sonraki'nin aksine, Deneysel sürümler, geniş bir sürüme hazır olmayan ek özellikler ve API'ler içerir.

Genellikle Sonraki kanalına yönelik bir güncellemeye, Deneysel kanalına karşılık gelen bir güncelleme eşlik eder. Aynı kaynak revizyonuna dayanırlar, ancak farklı bir özellik bayrağı seti kullanılarak oluşturulmuştur.

Deneysel sürümler, Sonraki ve En Yeni sürümlerden daha farklı olabilir. Deneysel sürümleri kullanıcılara yönelik uygulamalarda kullanmayınız. Deney kanalındaki sürümler arasında sık sık değişiklik yapılmasını beklemelisiniz.

Deneysel sürümler npm'de experimental etiketi ile yayınlanır. Sürümler, yapının kendi içeriğinin bir karma değerinden oluşturulur, örneğin; 0.0.0-experimental-68053d940-20210623.

Deneysel Bir Sürümde Neler Oluyor?

Deneysel özellikler, daha geniş bir kitleye sunulmaya hazır olmayan özelliklerdir ve sonuçlandırılmadan önce büyük ölçüde değişebilirler. Bazı deneyler asla sonuçlandırılamaz — deney yapmamızın nedeni, önerilen değişikliklerin uygulanabilirliğini test etmektir.

Örneğin, Hooks'u duyurduğumuzda Deneysel kanal mevcut olsaydı, Hooks'u En Yeni kanalında kullanıma sunmadan haftalar önce Deneysel kanalında yayınlardık.

Denemeye karşı entegrasyon testleri yapmayı değerli bulabilirsiniz. Bu size kalmış. Ancak, Deneysel kanalının Sonraki kanalından daha az istikrarlı olduğunu unutmayınız. Deneysel sürümler arasında herhangi bir istikrarı garanti etmiyoruz.