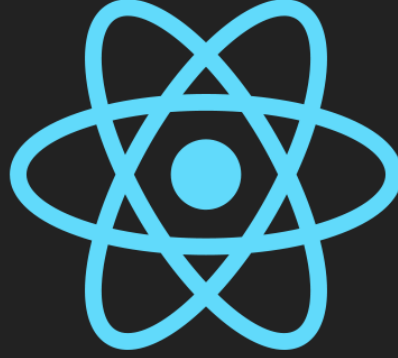


# React Native Windows Kurulumu



React Native'in ne olduğunu [bu yazımızda](#) anlattık. Fakat merak edip yeni başlayan Windows kullanıcıları için kurulumu kanserden beter olabiliyor. Bu yazımızda ise size o insanı çileden çıkartan kurulumu nasıl kolaylıkla halledebileceğinizi, hangi aracın neden gerekli olduğunu anlatacağız. Uzun olduğuna bakmayın, adım adım uyguladığınız takdirde sorunsuzca kurulumu gerçekleştirebileceksiniz.

Gelin başlayalım.

## Gerekli Araçlar

*NodeJS*

*Java JDK*

*Android Studio*

*Genymotion*

*VS Code / Atom / WebStorm vs ..*

## NODE JS

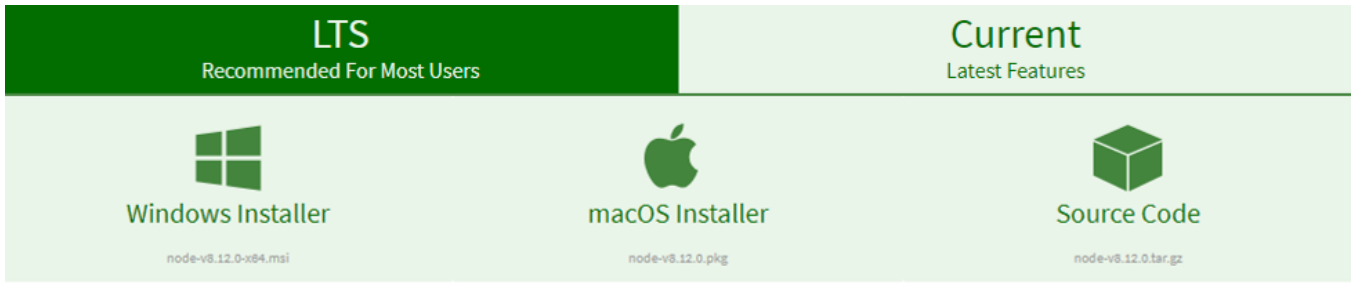


1.2K



10





Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x86/x64)

Linux Binaries (ARM)

Source Code

|                     |        |
|---------------------|--------|
| 32-bit              | 64-bit |
| 32-bit              | 64-bit |
| 64-bit              |        |
| 64-bit              |        |
| 32-bit              | 64-bit |
| ARMv6               | ARMv7  |
| node-v8.12.0.tar.gz |        |












<https://nodejs.org/en/download/>

NodeJS'i kendi sitesinden size uyumlu versiyonunu indirerek kurulumu yapabilirsiniz. Herhangi bir ayarı değiştirmenize gerek yok.

*Neden gerekli?: Hem NodeJS'in npm (node package modules) desteği sayesinde React Native'i ve diğer geliştiricilerin ürettiği paketleri yükleyebileceğimiz hem de geliştireceğimiz projelerin server üzerinde çalışmasını sağlayacağız.*

## JAVA JDK

**Java SE Development Kit 8u251**This software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

| Product / File Description          | File Size | Download  |
|-------------------------------------|-----------|---|
| Linux ARM 32 Hard Float ABI         | 72.87 MB  |  <a href="#">jdk-8u251-linux-arm32-vfp-hflt.tar.gz</a> |
| Linux ARM 64 Hard Float ABI         | 69.77 MB  |  <a href="#">jdk-8u251-linux-arm64-vfp-hflt.tar.gz</a> |
| Linux x86 RPM Package               | 171.71 MB |  <a href="#">jdk-8u251-linux-i586.rpm</a>              |
| Linux x64 Compressed Archive        | 186.09 MB |  <a href="#">jdk-8u251-linux-x64.tar.gz</a>            |
| macOS x64                           | 254.78 MB |  <a href="#">jdk-8u251-macosx-x64.dmg</a>              |
| Solaris SPARC 64-bit (SVR4 package) | 125.19 MB |  <a href="#">jdk-8u251-solaris-sparcv9.tar.Z</a>       |
| Solaris SPARC 64-bit                | 88.16 MB  |  <a href="#">jdk-8u251-solaris-sparcv9.tar.gz</a>      |
| Solaris x64 (SVR4 package)          | 133.64 MB |  <a href="#">jdk-8u251-solaris-x64.tar.Z</a>           |
| Solaris x64                         | 91.9 MB   |  <a href="#">jdk-8u251-solaris-x64.tar.gz</a>          |
| Windows x86                         | 201.17 MB |  <a href="#">jdk-8u251-windows-i586.exe</a>            |
| Windows x64                         | 211.54 MB |  <a href="#">jdk-8u251-windows-x64.exe</a>            |

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

Aynı şekilde yukarıdaki linkten JDK'nın size uyumlu versiyonunu indirip kurabilirsiniz. React Native Java'nın her sürümüyle uyumlu çalışmayabiliyor. Şu anlık uyumlu JDK sürümü jdk8, indirip kurabilirsiniz. Kurulumda spesifik bir değişiklik yapmanıza da gerek yok.

***Neden gerekli?:** Windows ortamında sadece Android (APK) geliştirebiliyoruz ve bildiğiniz üzere Android JAVA ile geliştiriliyor. React Native ile her proje oluştururken, çalıştırırken ve APK formatına çevirirken projemiz arkaplanda JAVA ile derleniyor.*

## ANDROID STUDIO



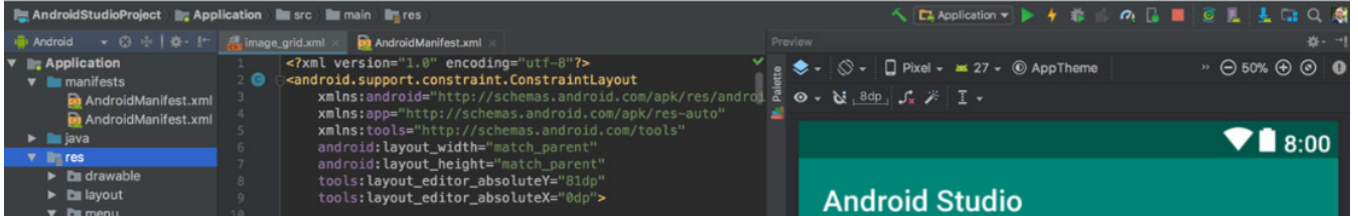
Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

3.2.1 for Windows 64-bit (927 MB)

DOWNLOAD OPTIONS

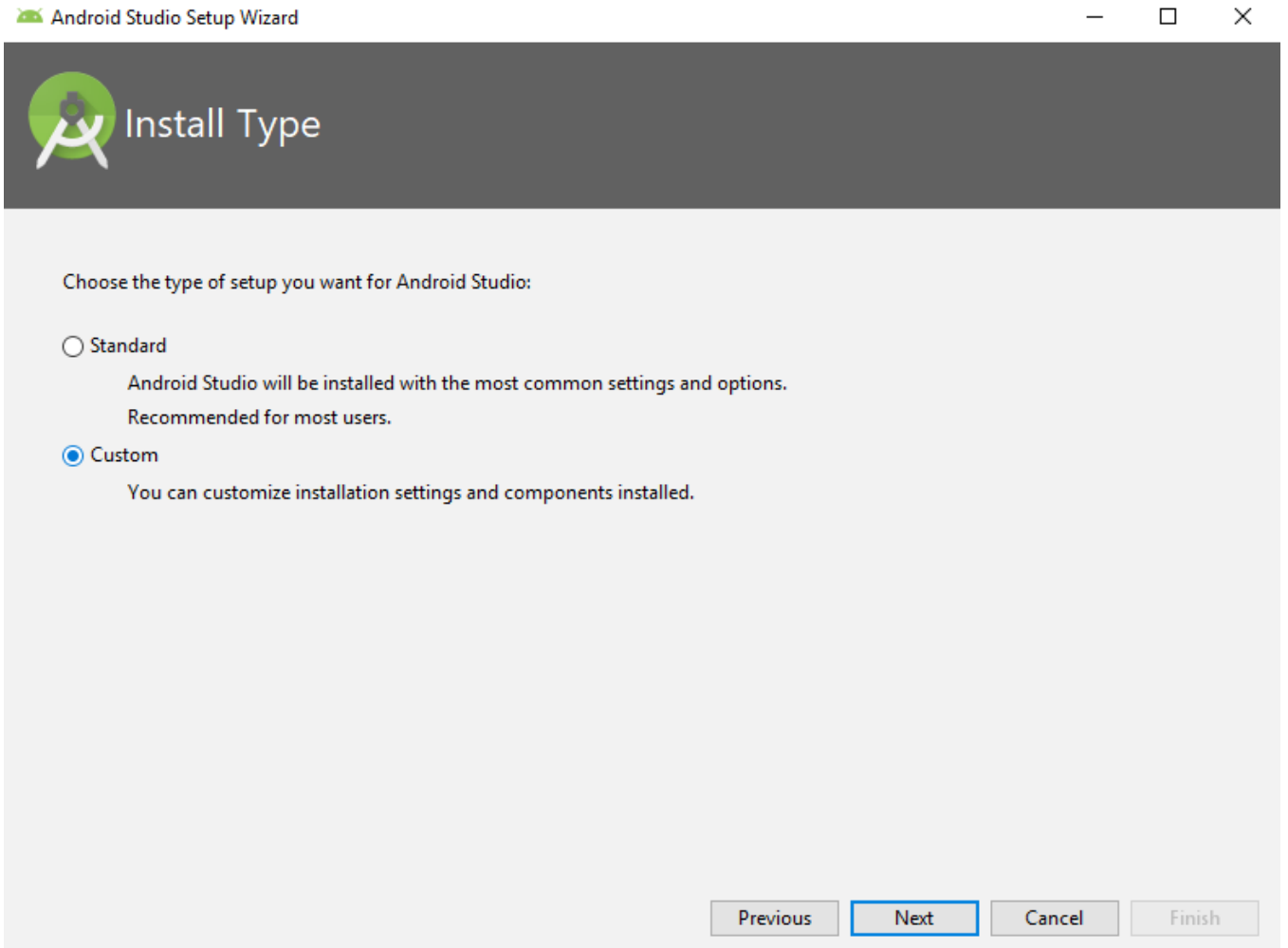
RELEASE NOTES



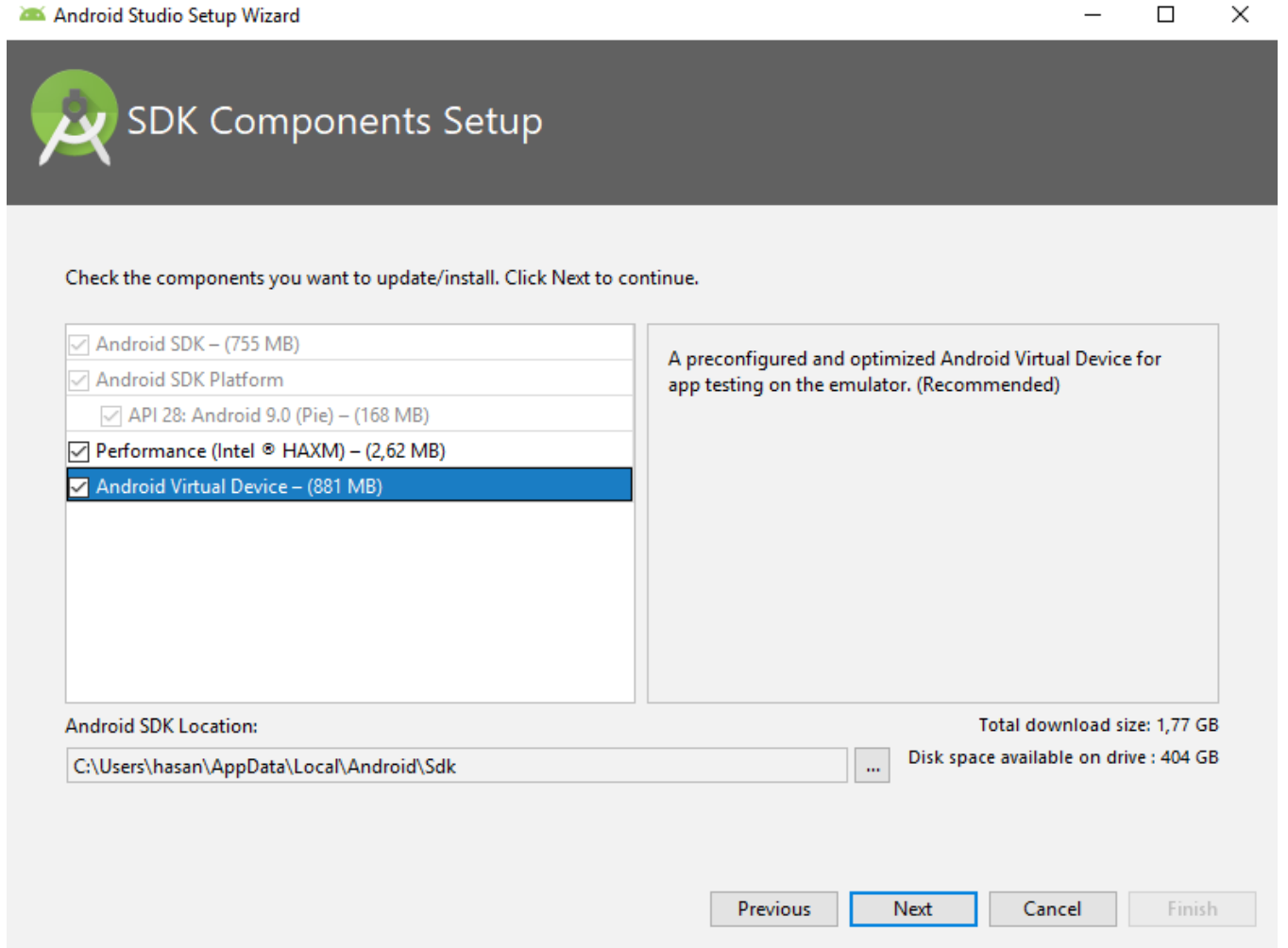
<https://developer.android.com/studio/>

Burası en önemli kısım. Önce yukarıdaki linkten Android Studio'yu indirelim.

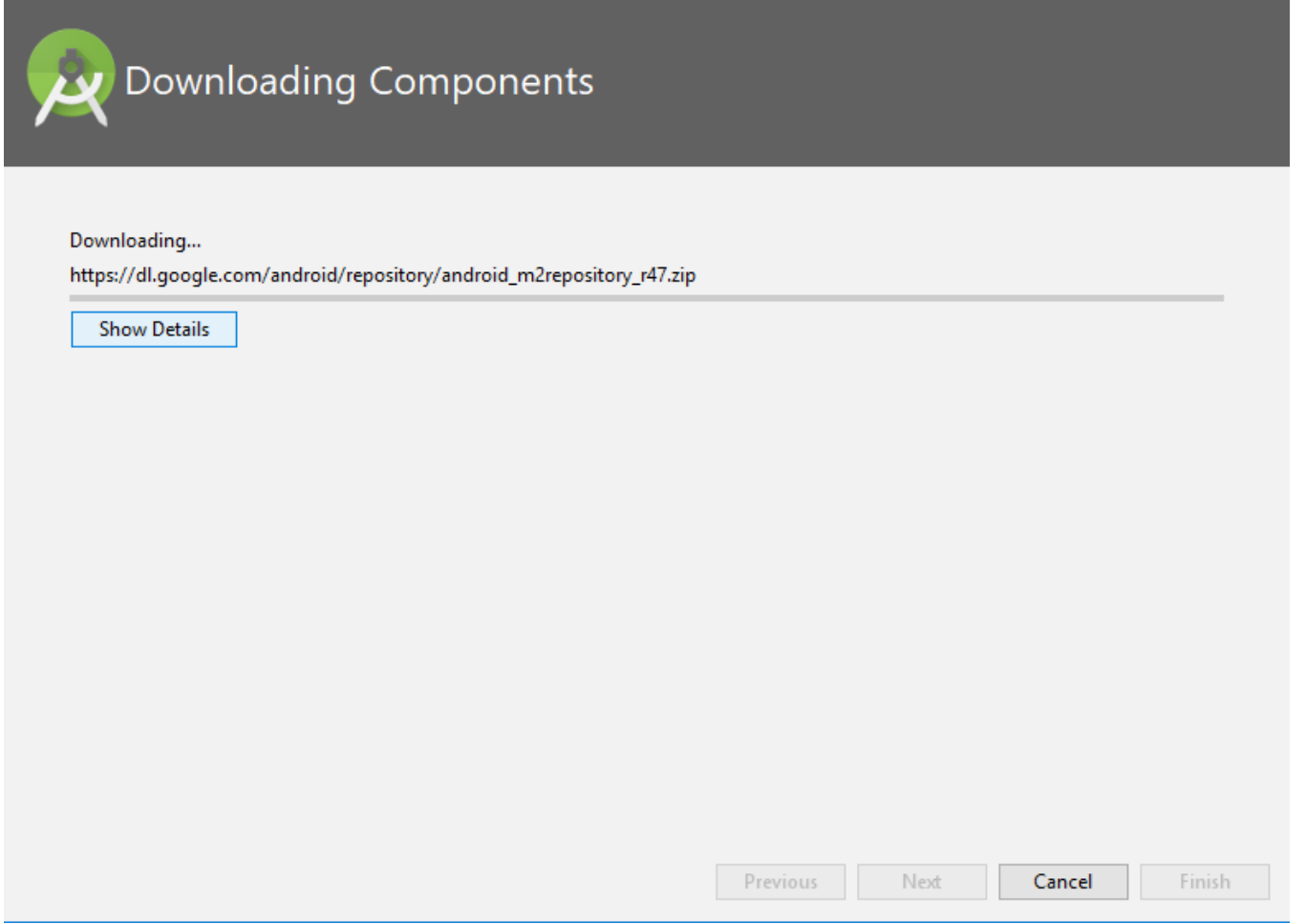
İndirilen kurulum dosyasını çalıştırdıktan sonra Setup Wizard ekranı gelecektir. "Next" diyerek ilerleyelim ve aşağıdaki ekrana kadar gelelim.



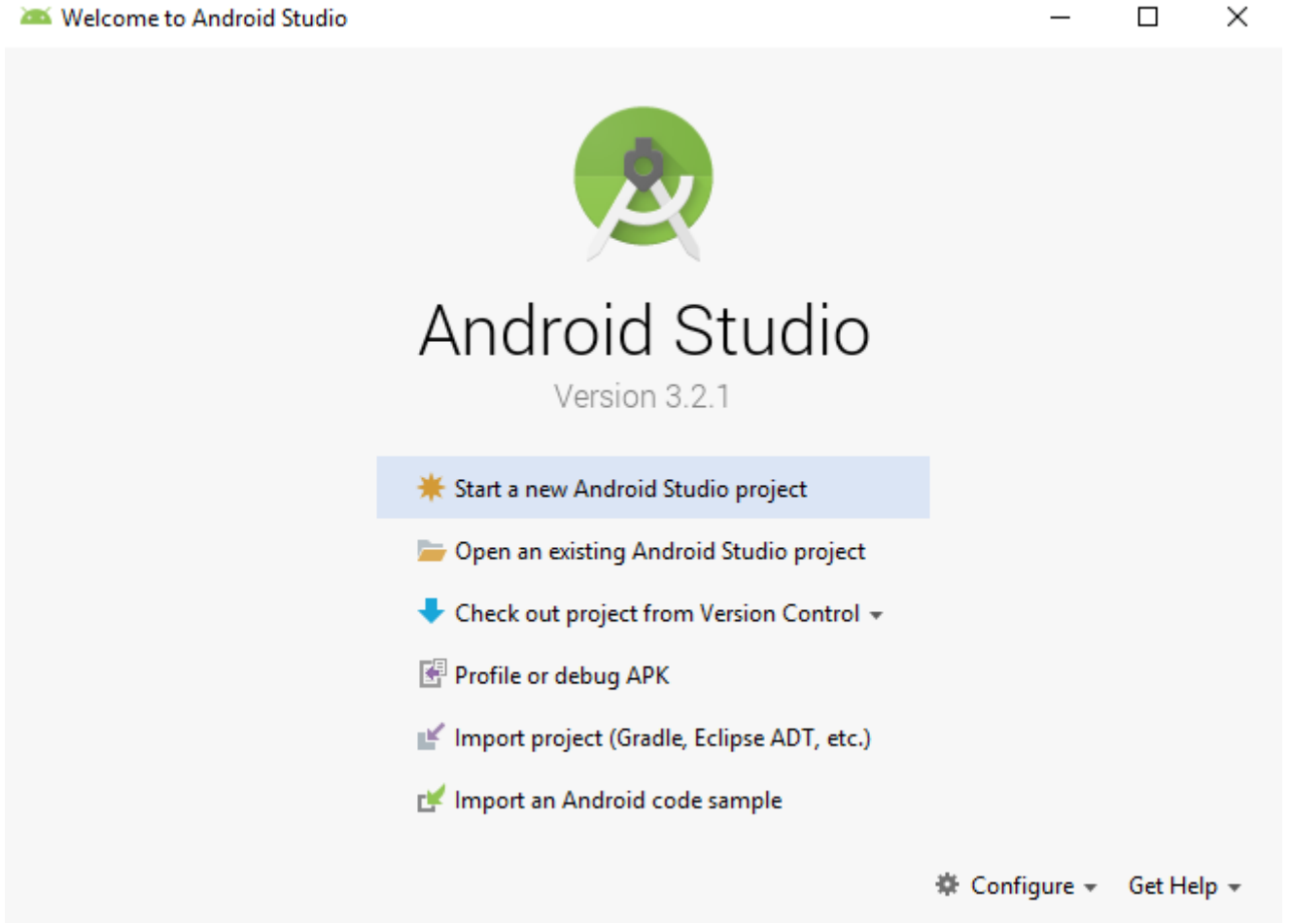
Bu ekranda “Custom” seçeneğini seçip ilerliyoruz.



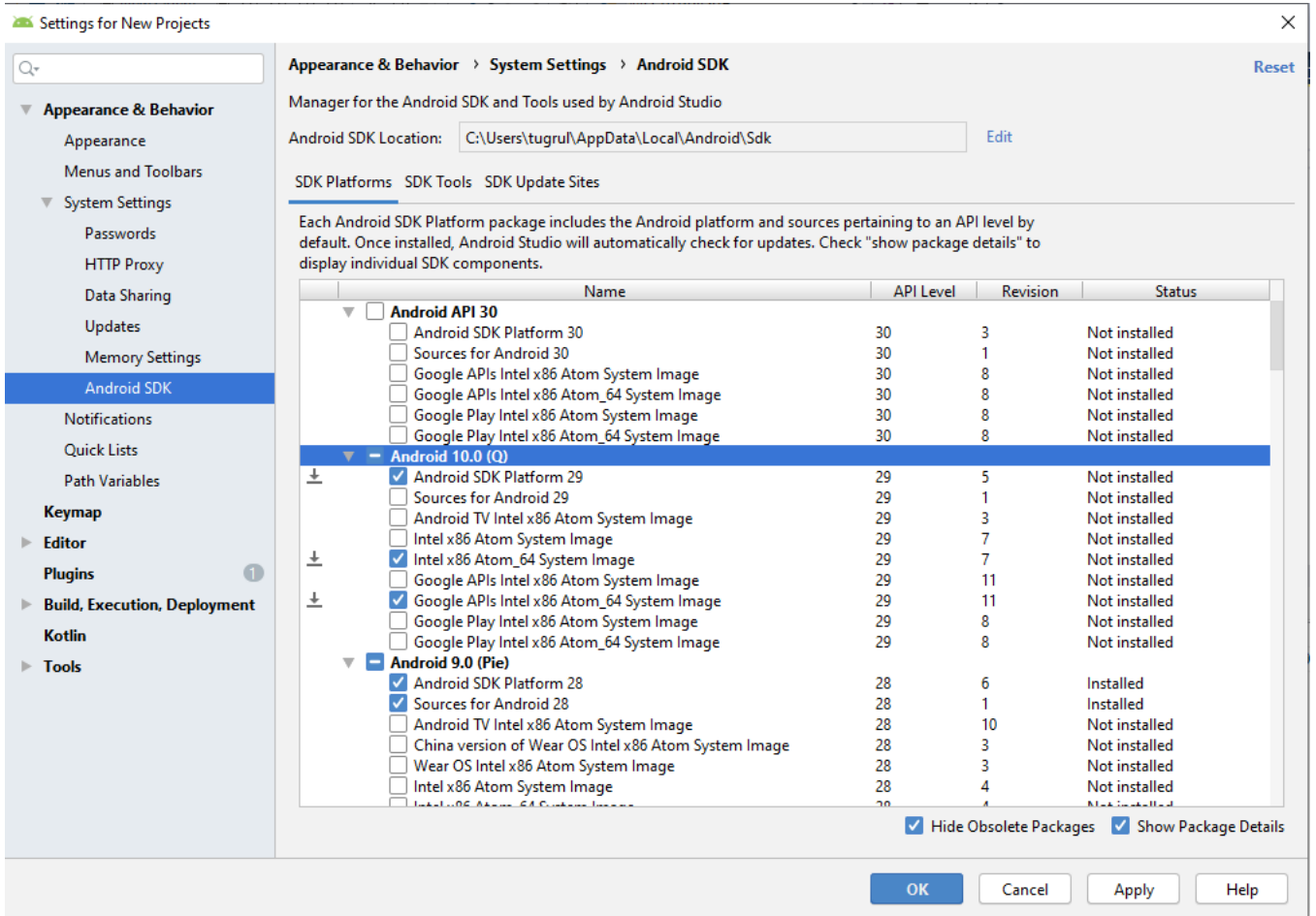
Sizin ekranınızda “Perfonmance (Intel® HAXM)” ve “Android Virtual Device” seçili olmayabilir. Onları seçerek Next diyelim.



Karşımıza bu yükleme ekranı gelecektir. Uzun sürebilir, bekleyelim.



Yüklemeden sonra Android Studio açılacaktır, açılmaz ise siz çalıştırın. Açılan ekranda sağ alt köşedeki “Configure” seçeneğine tıklıyoruz ve “SDK Manager” seçeneğini seçiyoruz.

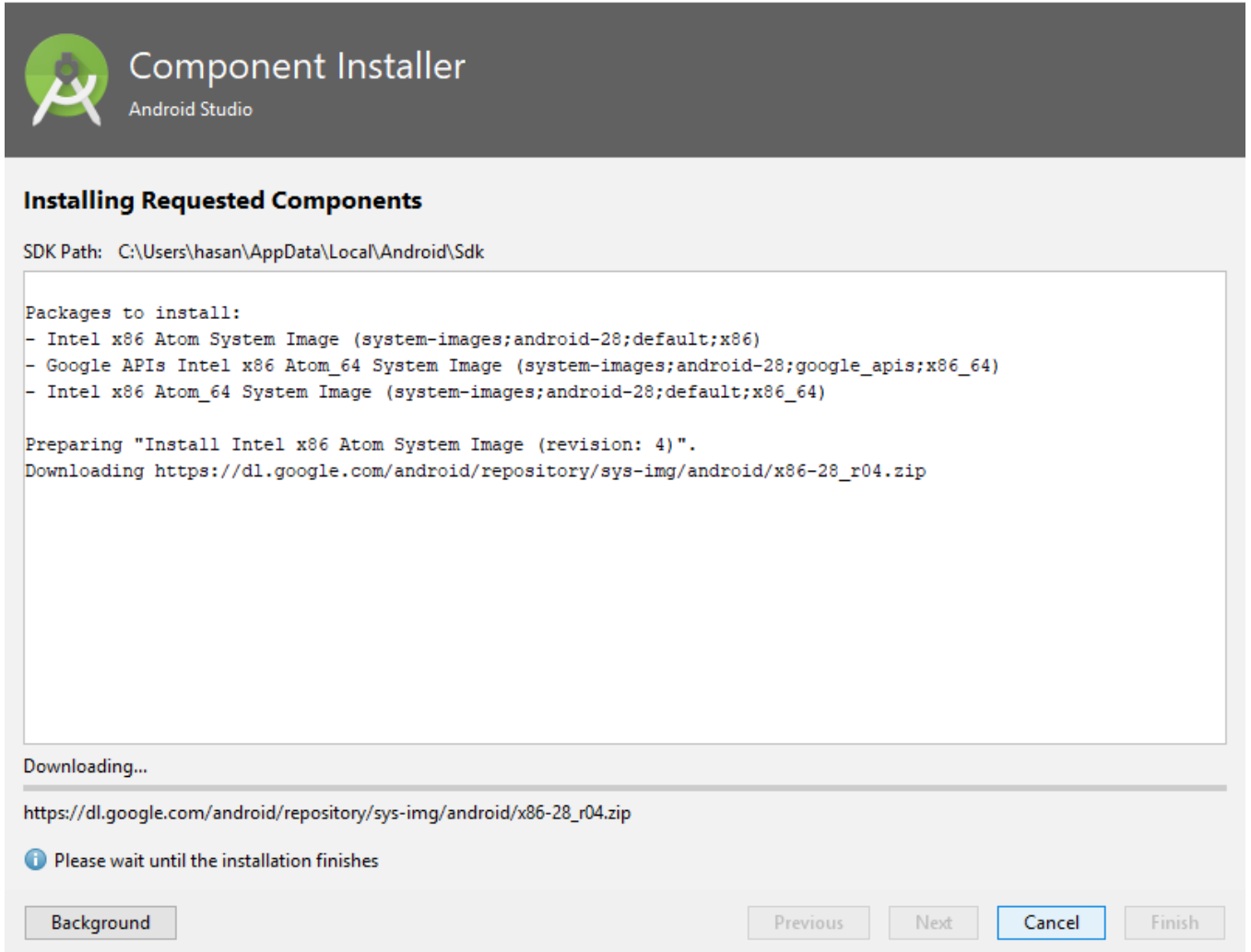


Açılan pencerede sağ alttaki “Show Package Details” seçeneğini seçerek yüklenebilir paketlerin detaylarını görebiliriz. Ekrandaki seçili paketleri seçerek (*sadece Android 10 Q altındakileri seçmeniz yeterli*) “Next” diyelim.

Android SDK konumunu açılan pencerede yukarıda bulabilirsiniz. Bu konumu not edelim, ihtiyacımız olacak.

*Not: (React Native güncel olarak Android 10 “Q” SDK 29 ile çalışmaktadır.)*





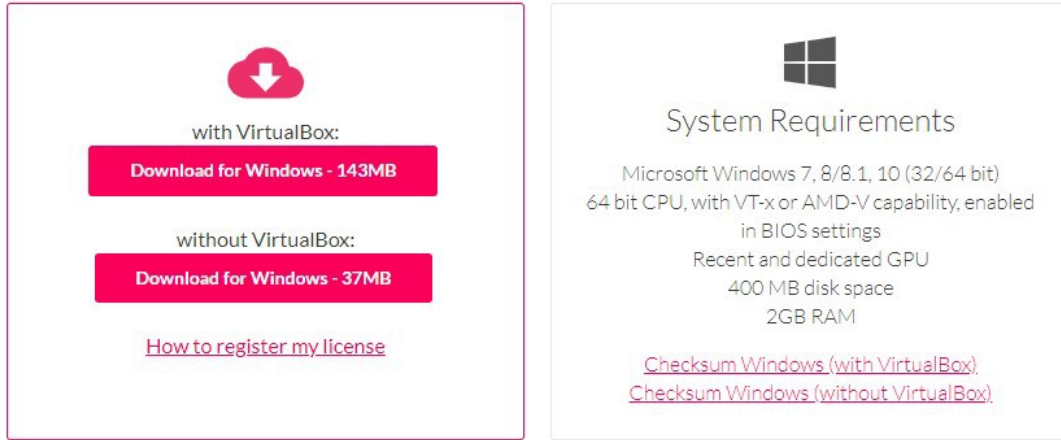
Karşımıza paket yükleme ekranı gelecektir. Bu yükleme önceki yükleme ekranına kıyasla çok daha uzun sürebilir.

Bu adımdan sonra Android Studio kurulumu bitmiş demektir.

*Neden gerekli?: Hem sanal cihazımızı daha optimum çalıştırmak, hem geliştireceğimiz uygulamaları deploy etmek, hem de uygulamamızda native değişiklikler yapmak için Android Studio'muzu kullanmamız gerekecek.*

## GENYMOTION

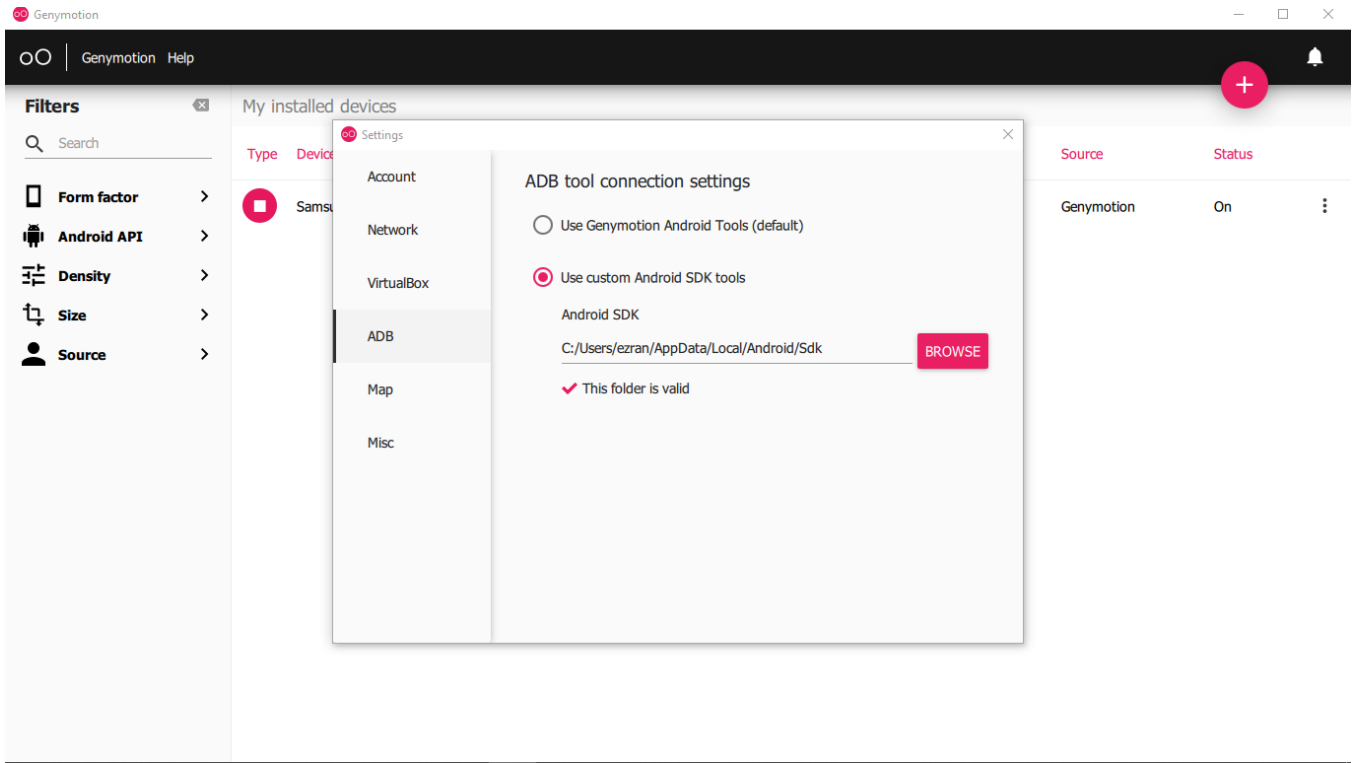
## Download Genymotion 3.0.0



Öncelikle bir Genymotion hesabı açıp giriş yapalım. Sonrasında [bu linkten](#) “with VirtualBox” seçeneği ile indirip kurulumu tamamlayalım. Bu seçenek sisteminize Oracle VB kuracaktır.

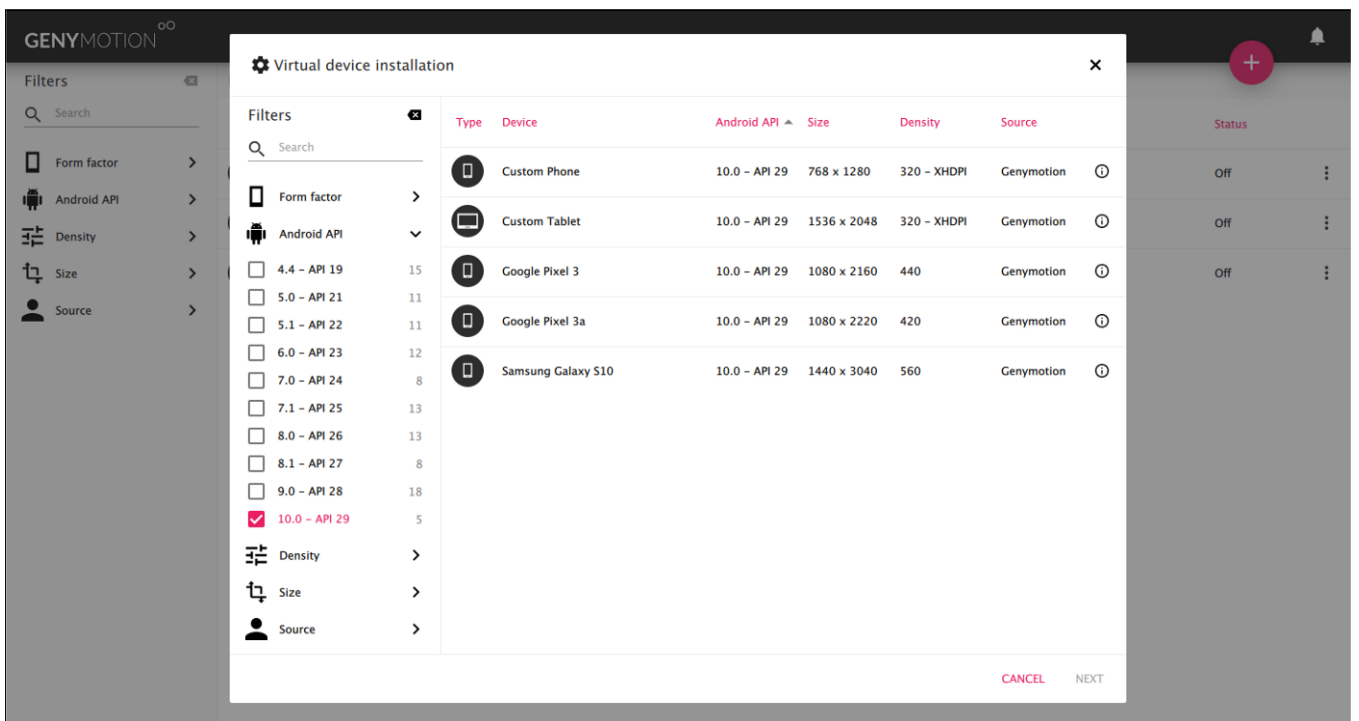
Ek olarak VB derleyicisini Android Studio seçmemiz gerekecek, bu daha performanslı derleme imkanı sunacaktır. Bunun için Genymotion uygulamamızı başlatalım, hesabımıza giriş yapalım ve sol üstten **Genymotion > Settings** seçeneğinden ADB’yi seçerek Android Studio’yu kurduğumuz yoldaki Sdk klasörünü belirtmemiz yeterli.

SDK konumunu not almadıysanız Android Studio’yu açıp karşılama ekranında **Configure > SDK Manager** penceresinde yukarıda bulabiliriz.

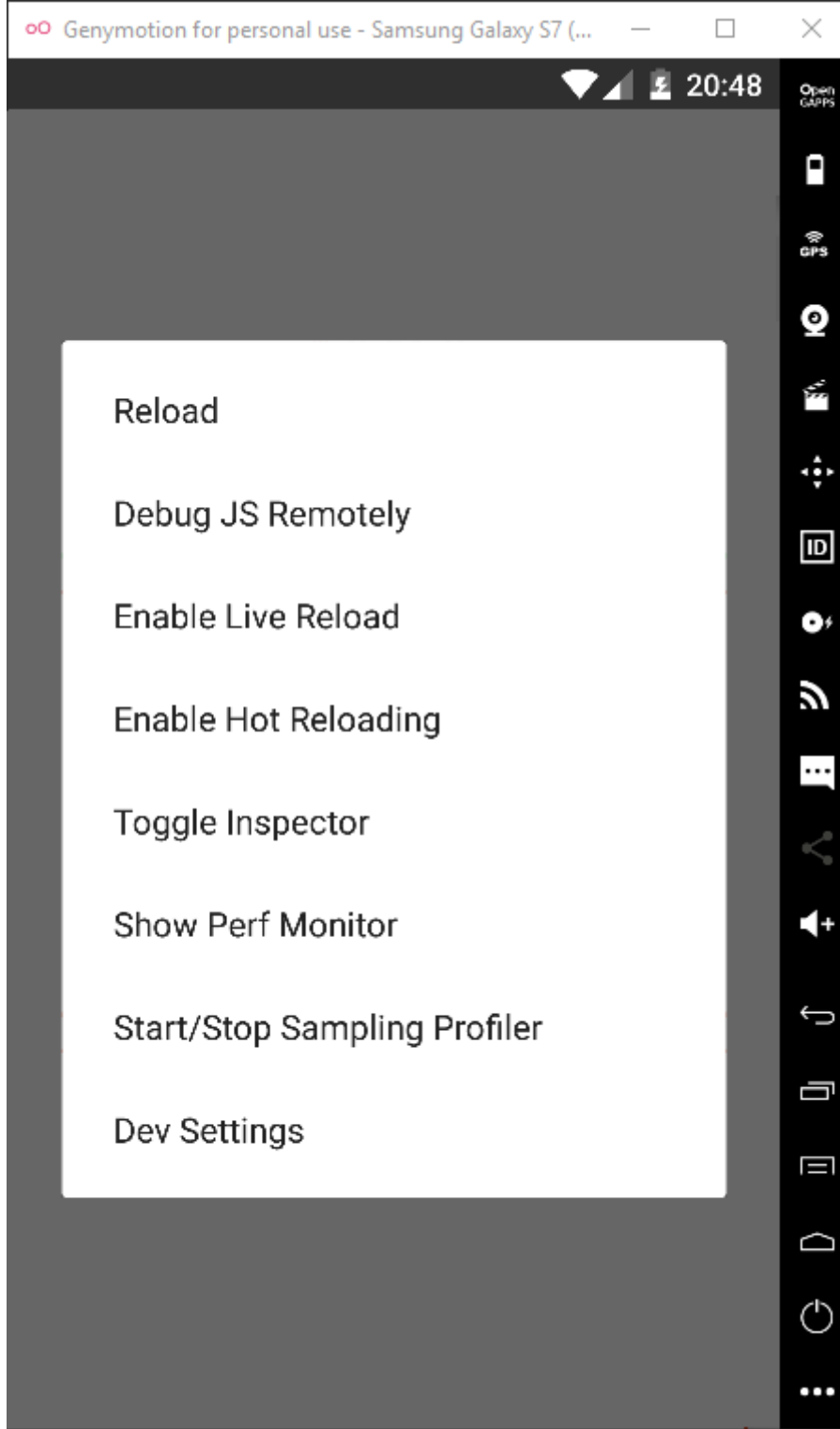


Konum doğru ise onay durumu belirecektir.

Şimdi ise bir Android cihazı kurmamız gerek. Kuracağımız cihazın Android Studio'da indirdiğimiz SDK ile uyumlu olmasına dikkat edelim. Biz SDK 29 indirdiğimiz için onunla uyumlu bir cihaz seçeceğiz. Sol taraftaki bölümden API filtremizi yapabiliriz. İçinden herhangi bir cihazı seçip Next diyerek direkt olarak yükleyebiliriz. Ben örnek olması açısından Samsung Galaxy S10 seçmişim. Siz isterseniz farklı bir cihaz seçebilirsiniz.



Cihazınızı yükledikten sonra ana ekranınıza eklenecektir. Start diyerek başlatalım ve açılmasını bekleyelim. Her React Native projesini çalıştıracığımız zaman öncelikle sanal cihazımızı başlatacak daha sonra projemizi derleyeceğiz. Sanal cihazımızı başlatıp bir proje oluşturduktan sonra yapılan değişiklikleri görüntülemek için “Developer Menu”den “Reload” yapmamız yeterli olacaktır.



**Debug JS Remotely** seçeneği projemizi derlerken Console desteği ile debug etmemize yarayan bir özellik. **Enable Live Reload** proje kodlarımızda bir değişiklik

yapıp kaydettiğimizde otomatik olarak cihazı yeniler. **Enable Hot Reload** ise kodlarda değişiklik yapıldığında kaydetmeden direkt olarak cihazı yeniler.

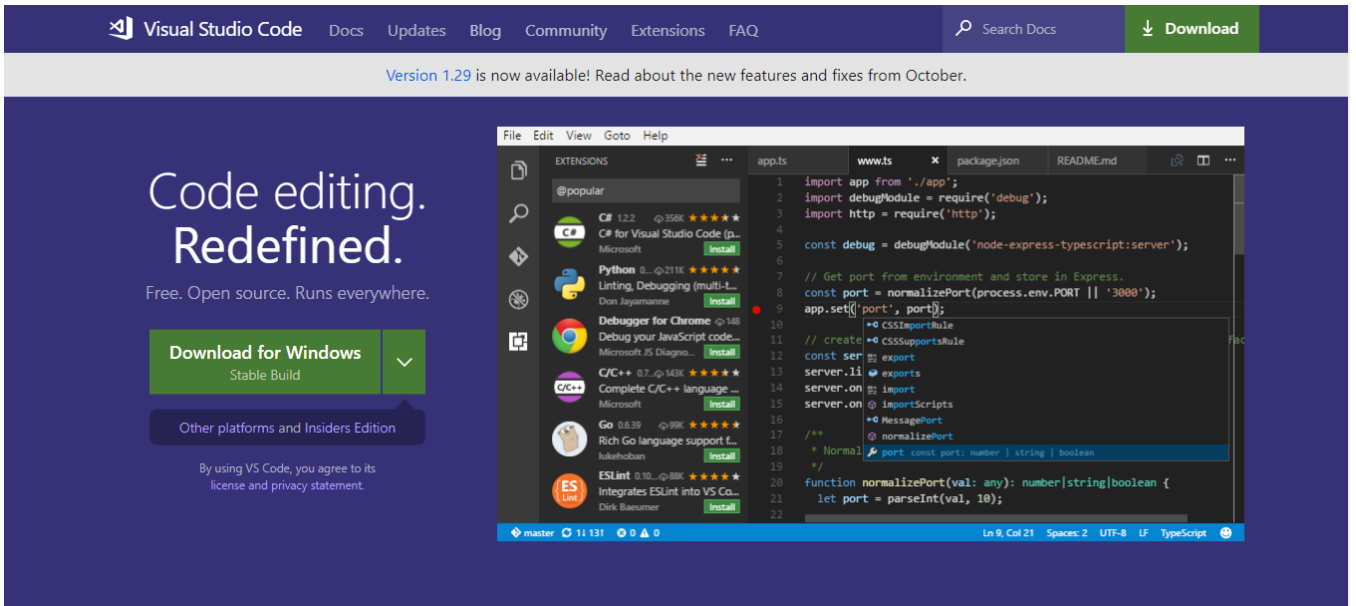
**ÖNEMLİ:** Eğer Genymotion “Virtual Engine Error” ya da “Virtualization Error” hatası verirse bilgisayarınızın BIOS ayarlarından “Virtualization Technology” seçeneğini aktifleştirmeniz gerekmektedir. Bu seçenek bilgisayarınıza sanal makine kurma izni verecektir.

Örnek BIOS ekranını aşağıda ki gibidir, sizde farklı isimler yazabilir fakat genel anlamda adı “Virtualization” olarak geçer.

| Advanced   |   |
|--|---|
| Advanced Processor Configuration   | Item Specific Help  |
| CPU Mismatch Detection: [Enabled]  | When enabled, a VMM (Virtual Machine Monitor) can utilize the additional hardware capabilities provided by Vanderpool Technology. |
| Core Multi-Processing: [Enabled]   |   |
| Processor Power Management: [Disabled]   |   |
| Intel(R) Virtualization Technology [Enabled]   |   |
| Execute Disable Bit: [Enabled]   |   |
| Adjacent Cache Line Prefetch: [Disabled]   | If this option is changed, a Power Off-On sequence will be applied on the next boot.  |
| Hardware Prefetch: [Disabled]  |   |
| Direct Cache Access [Disabled]   |   |
| Set Max Ext CPUID = 3 [Disabled]   |   |
|  |   |
| F1 Info ↑↓ Select Item -/+ Change Values F9 Setup Defaults<br>Esc Exit + Select Menu Enter Select ► Sub-Menu F10 Save and Exit |   |

*Neden gerekli?:* Geliştireceğimiz projeleri Genymotion sayesinde kolayca derleyebilir, debug edebiliriz. Android Studio'nun kendi sanal cihazı bir çok alanda sorun çıkartabiliyor.

## TEXT EDITOR (VS CODE)



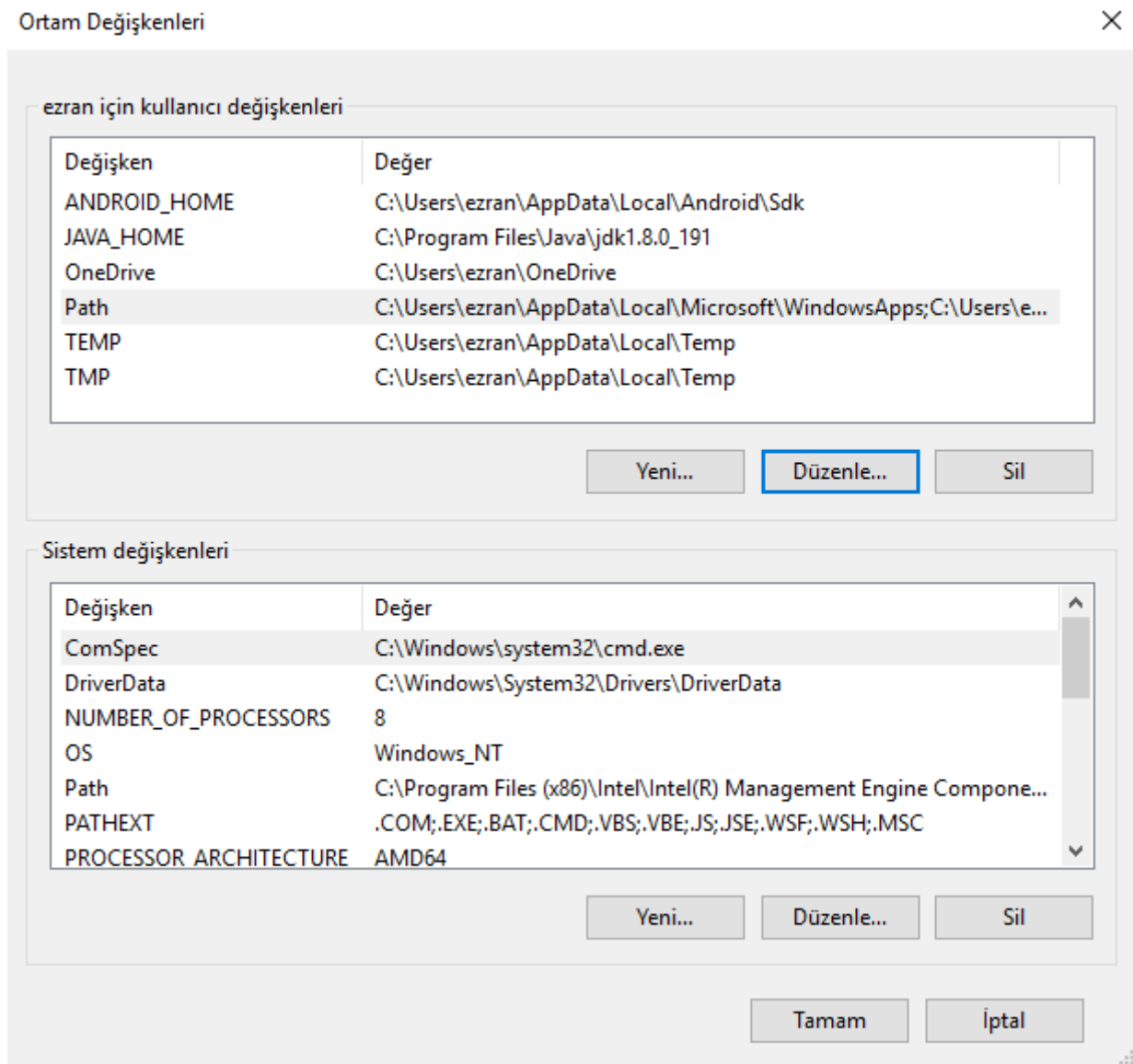
<https://code.visualstudio.com/>

Projelerimizi derlerken Visual Studio'daki gibi hazır bir proje üzerinden geliştirip aynı ortamda anlık olarak derlemeyeceğiz. Projenin kaynak kodlarını düzenleyip sanal cihazda çalıştıracacağız ve bunun için bir text editor gerekli. Ben VS Code'u kullanmayı tercih ediyorum çünkü çoklu terminal desteği, arayüzü ve lightweight olmasından ötürü (*ElectronJS*) kullanımını kolay buluyorum. Siz alternatif olarak Atom, Webstorm, Brackets ya da istediğiniz herhangi bir editor'ü seçebilirsiniz.

## ORTAM DEĞİŞKENLERİ

Ortam araçlarımızın arka planda sorunsuzca çalışması ve sadece konsoldan rahatça kurulumumuzu yapmak için yollarımızı belirlememiz gerekli.

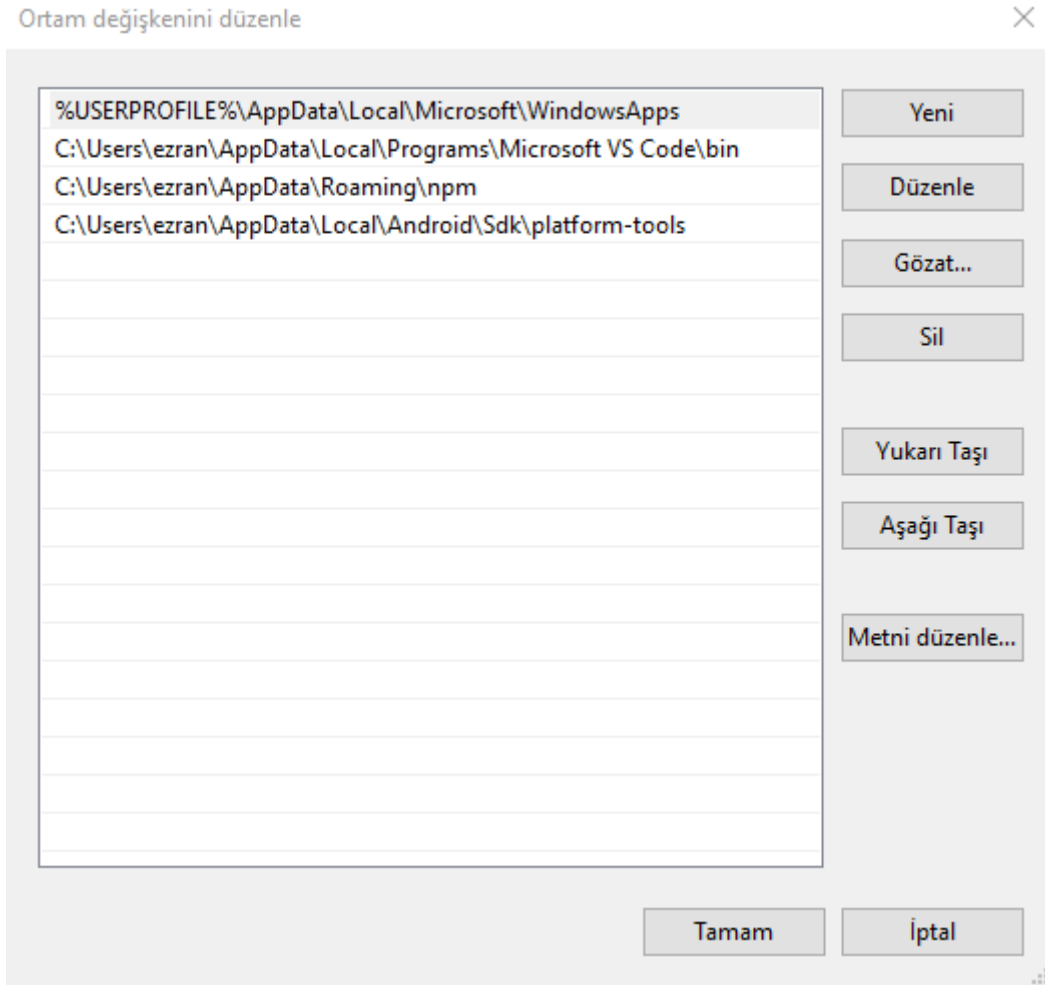
Bunun için öncelikle Bilgisayar'ımıza sağ tıklayıp "Özellikler" diyoruz ve açılan pencerede sol bölümdeki "Gelişmiş sistem ayarları" seçeneğine tıklıyoruz. Karşımıza çıkan pencerede alt bölümdeki "Ortam Değişkenleri" seçeneğine tıklıyoruz.



Daha önce Android projesi geliştirmişsek “JAVA\_HOME” değişkeni otomatik olarak kurulmuş olabilir. Değil ise “JAVA\_HOME” ve “ANDROID\_HOME” değerini üstteki “... için kullanıcı değişkenleri” bölümünde “Yeni” seçeneğine tıklayarak ekleyelim. Yol uzantısını karşısındaki yola göre seçebilirsiniz siz.

Buna ek olarak yine üst bölümdeki “Path” seçeneğine tıklayıp “Düzenle”yi seçelim ve aşağıda, en alttaki seçenekte belirtilen “platform-tools” yolunu ekleyelim.





Bu seçenek sanal cihaz ya da gerçek cihaz bağladığımızda aktif olup olmadığını kontrol etmemize yarayacaktır.

### BAĞLI CİHAZLARI KONTROL ETME (meraklısına)

Madem bahsettik nasıl olduğunu da gösterelim;

Terminalden `adb devices` komutunu girdiğimizde o anda bilgisayar üzerinden yönetilebilecek aktif cihazların adresini ve durumunu gösterir.

```
C:\Users\ezran>adb devices
List of devices attached
192.168.0.101:5555    device
```

Örneğin bende şu anda açık olan emülatörün bilgileri bu şekilde görüntüleniyor. Sanal cihazımın kullandığı IP adresini ve dinlediği port bilgilerine erişebiliyor, durumunu (*device, booting, disconnecting .. vs*) izleyebiliyorum.



```
C:\Users\ezran>adb devices
List of devices attached
192.168.0.101:5555    device
a98dbe82990         device
```

Eğer bilgisayara gerçek bir Android cihaz bağlarsam da devices komutunda emülatör ayrı sanal cihaz ayrı adreslerle bu görseldeki gibi görüntülenecektir.

Elbette kullanacağımız tek komut bu değil. Genymotion'da yapılan değişiklikleri emülatörde açılan menüden “Reload” ile cihazı yenileyerek görüntülüyorduk. Gerçek cihazda ise o menüyü açmak için şu komutu giriyoruz;

```
adb shell input keyevent 82
```

Bu sayede gerçek cihazımızda da “Developer Menu”ye erişebiliyoruz. Eğer birden fazla cihaz aynı anda bağlıysa ve tek bir tanesine komut gönderecekseniz;

```
adb -s <CIHAZ ADRESİ> shell <KOMUT>
```

Şeklinde komutları gönderebiliyoruz. Örneğin “7f1c864e” adresli cihazımıza developer menu komutu göndereceksek;

```
adb -s 7f1c864e shell input keyevent 82
```

şeklinde komutu girmemiz yeterli olacaktır. Eğer bir emülatör bir de gerçek cihazınız bağlı ise cihaz adresi yerine emülatör için -e gerçek cihaz için -d yazabilirsiniz.

Bana developer menu olayı yetmez, daha fazla komut istiyorum diyorsanız şu linke göz atabilirisiniz;

<https://stackoverflow.com/questions/7789826/adb-shell-input-events>

## Proje Oluşturma

React Native artık 0.60 versiyonundan sonra Node paketleri içerisinde default olarak gelmekte. Bu sebepten ötürü artık “**react-native-cli**”ye ihtiyaç duymadan “**npx**” kullanarak proje oluşturabiliyor, react-native komutlarını girebiliyoruz.

Kurulumumuz gerçekleştikten sonra sistem üzerinde projelerimizi saklayacağımız bir klasör oluşturup komut satırından o klasör içine ulaşalım. Tavsiyem mevcut konumda bir klasör yaratmanızdır. (*Kısayoldan klasör yaratmak için **mkdir** klasörAdı komutunu yazmanız yeterli olacaktır.*) Çok uzun bir konum seçmemeye ve mümkünse konum adresinin içerisinde Türkçe karakter kullanmamaya çalışın. Problem yaratabiliyor.

---

*Örneğin; C:\Users\kullanıcıAdı\Projects*

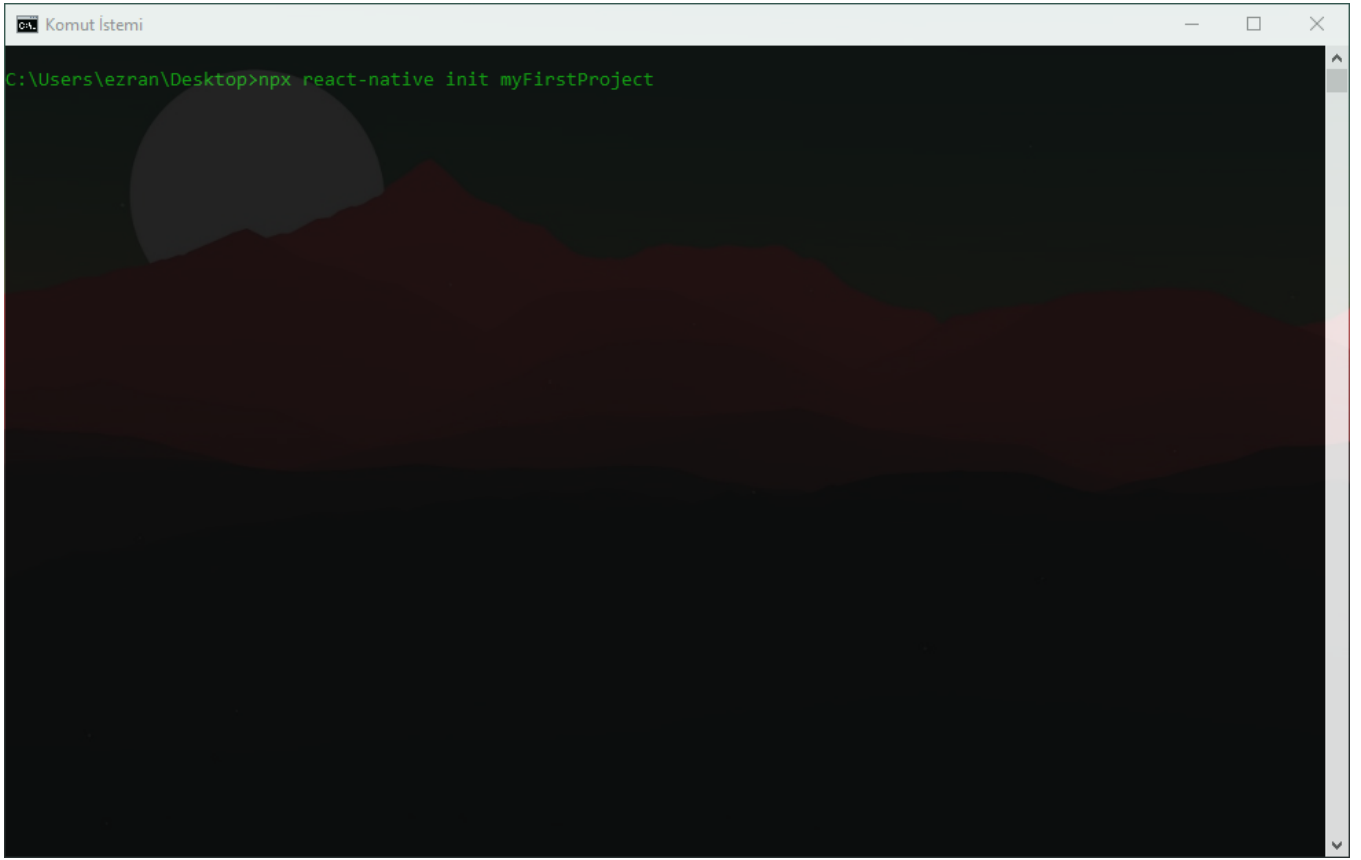
---

**Not:** Komut satırından herhangi bir yola ulaşmak için o yol öncesinde “**cd**” yazmanız yeterli. ( *cd C:\Users\kullanıcıAdı\Projects* )

Artık proje yaratabiliriz. Bunun için aşağıdaki komutu girmemiz yeterli olacaktır.

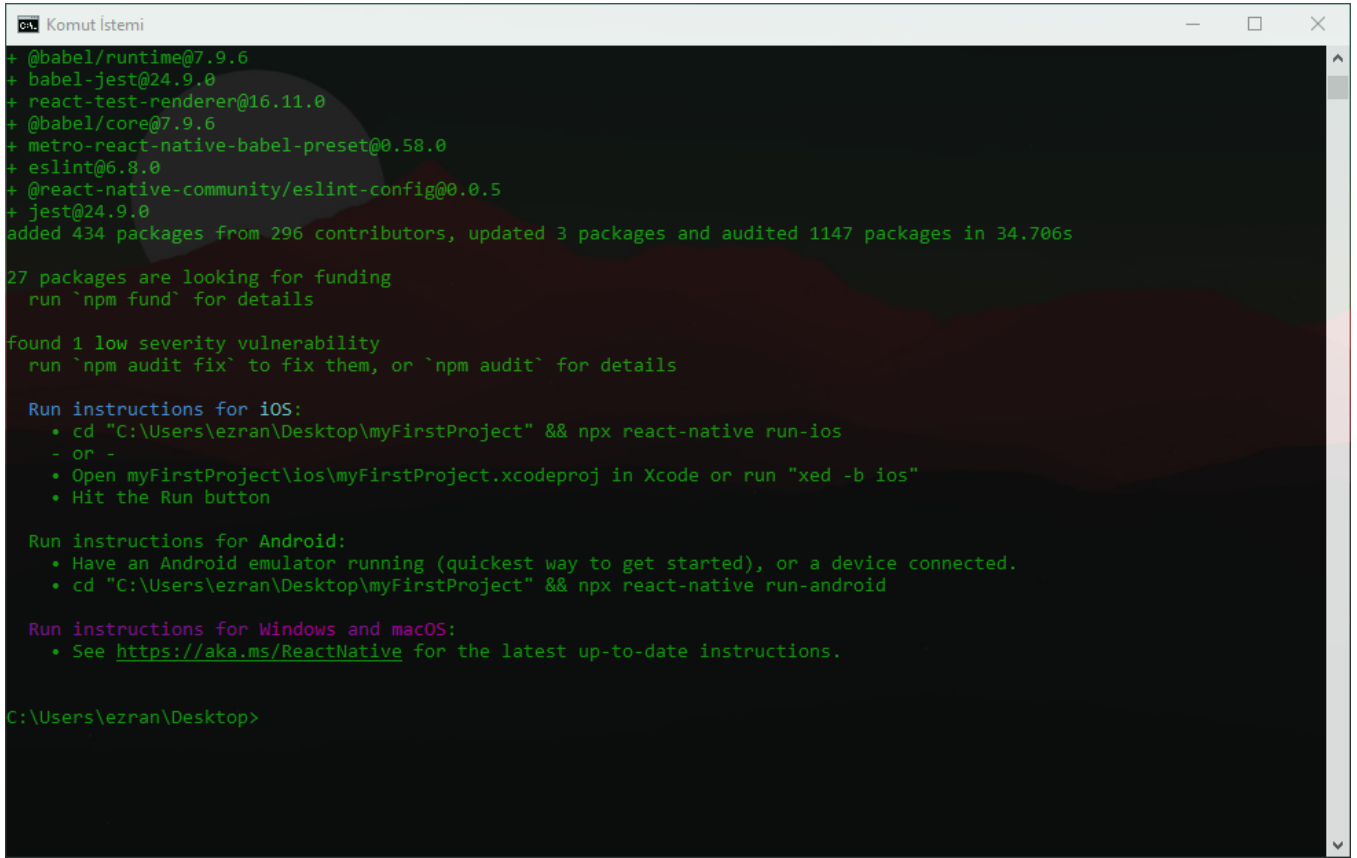
```
npx react-native init myFirstProject
```

**ÖNEMLİ NOT:** Eğer VS Code’un kendi terminalini kullanıyorsanız terminal tipini “**cmd**” olarak seçmeniz gerek. Default olarak “**powershell**” gelebiliyor ve powershell terminali proje derlerken ve oluştururken sorun çıkarabiliyor. Bu değişikliği VS Code’daki terminal penceresinin sağ üstünde terminal adı yazan bölümden yapabilirsiniz.



```
ca Komut İstemi
C:\Users\ezran\Desktop>npx react-native init myFirstProject
```

***Not:** İlk defa proje kurulumu yapacağımız için bu işlem ve proje derleme işlemi vakit alabilir fakat daha sonraki işlemlerde bu kadar beklemeyeceksiniz. Nedenini merak ediyorsanız anlatayım; Siz ilk defa bir proje oluşturduğunuzda 600'den fazla npm paketi indirilir ve işletim sistemi bu dosyaları versiyonları ile beraber belleğine kaydeder. Aynı şekilde projenizi derlediğinizde Java derlemeleri işlenir ve hafızaya atılır. Bu işleme “caching” denir. Siz daha sonra yeni bir proje oluşturmak ya da derlemek istediğinizde işletim sisteminiz gerekli dosyaları tekrardan oluşturmak yerine kaydettiği dosyaları (yani cachleri) kullanır. Eğer güncelleme gereken paketler ya da derlenmesi gereken bir Java dosyası varsa gerekli işlemleri gerçekleştirir.*



```
Komut İstemi
+ @babel/runtime@7.9.6
+ babel-jest@24.9.0
+ react-test-renderer@16.11.0
+ @babel/core@7.9.6
+ metro-react-native-babel-preset@0.58.0
+ eslint@6.8.0
+ @react-native-community/eslint-config@0.0.5
+ jest@24.9.0
added 434 packages from 296 contributors, updated 3 packages and audited 1147 packages in 34.706s

27 packages are looking for funding
  run `npm fund` for details

found 1 low severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

Run instructions for iOS:
  • cd "C:\Users\ezran\Desktop\myFirstProject" && npx react-native run-ios
  - or -
  • Open myFirstProject\ios\myFirstProject.xcodeproj in Xcode or run "xed -b ios"
  • Hit the Run button

Run instructions for Android:
  • Have an Android emulator running (quickest way to get started), or a device connected.
  • cd "C:\Users\ezran\Desktop\myFirstProject" && npx react-native run-android

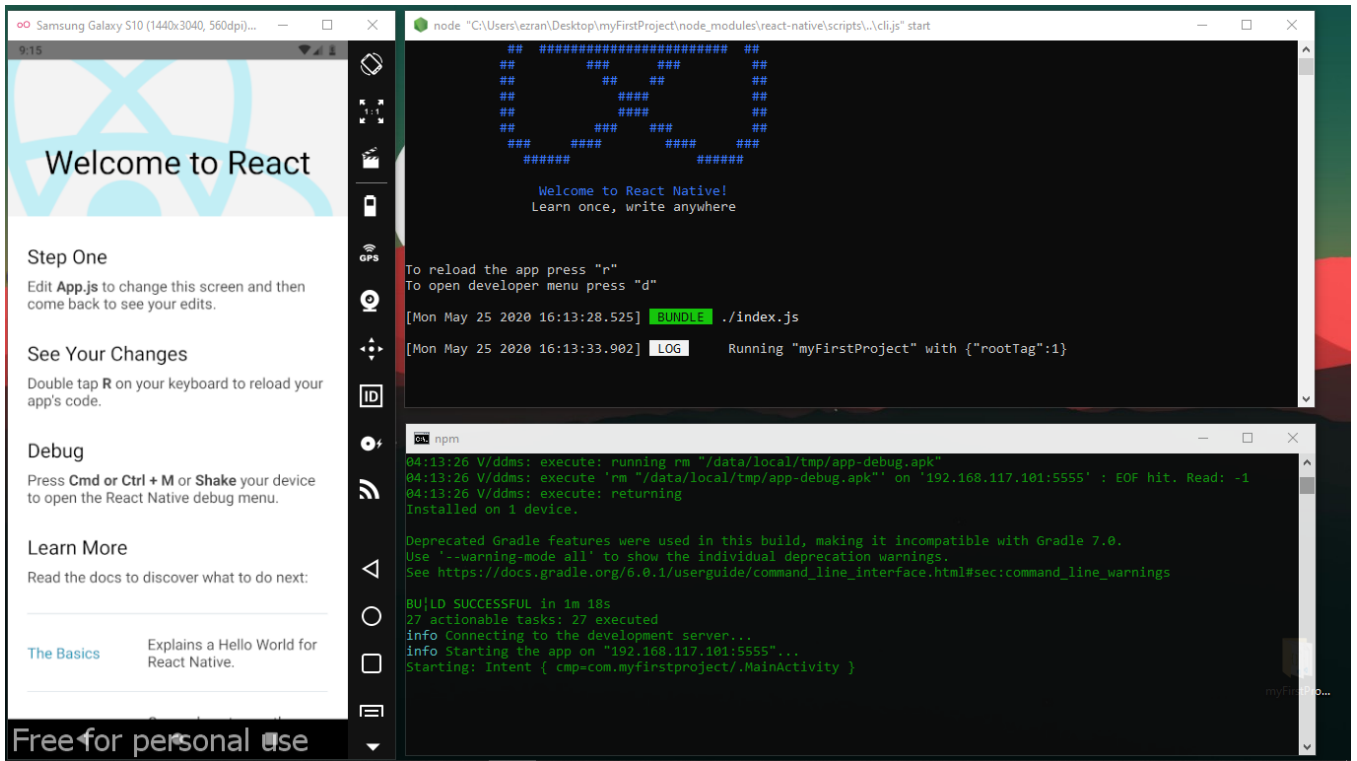
Run instructions for Windows and macOS:
  • See https://aka.ms/ReactNative for the latest up-to-date instructions.

C:\Users\ezran\Desktop>
```

Projemizi yarattıktan sonra Genymotion'umuzu açıp sanal cihazımızı çalıştıralım. Cihazımız açıldıktan sonra komut yolumuzdan proje klasörümüze erişelim ve Android derlemesi için komutunu girelim.

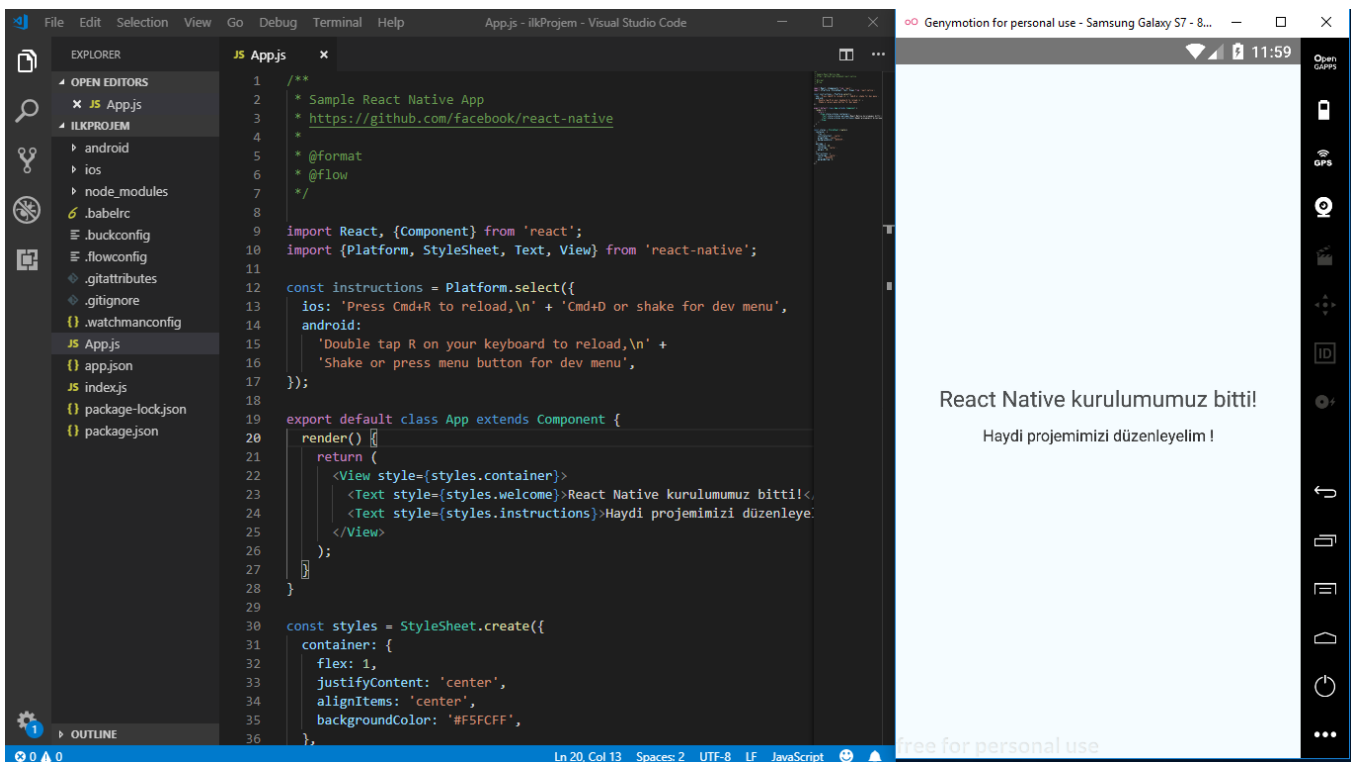
```
npx react-native run-android
```

Komutumuz çalışırken NodeJS Server'ı da çalıştıracak ve uygulamamız devreye girecektir. Başarılı bir yüklemeden sonra uygulamamız aşağıdaki gibi görünecektir.



Text Editor'ümüz ile proje klasörümüzü açarak App.js dosyası üzerinden düzenlemelere başlayabiliriz.

*Not: Text Editor'de komutlarda değişiklik yaptıktan sonra sanal cihazda iki kere R tuşuna basarak güncelleyebilirsiniz.*



Ve React Native kurulumumuzu bitirdik. Artık dilediğimiz gibi projelerimizi yaratabilir, istediğimiz uygulamayı geliştirebiliriz.

Herkese kolay gelsin, React Native dünyasına hoşgeldiniz :)

***Not:** Unutmayalım, projemizi çalıştırmak için önce emülatörümüzü (Genymotion) çalıştırıp daha sonra terminalden “.. run-android” seçeneğiniz seçiyoruz.*

***Not:** VS Code kullananlar Terminal'i (Ctrl + “) açıp “cmd”yi seçerek sadece VS Code üzerinden proje yaratabilir ya da derleyebilirler.*