



Bower

Emre Can ÖZTAŞ

www.emrecanoztas.com

Kapak Tasarımı: Emre Can ÖZTAŞ

Bower

Book Version: 1.0
NodeJS Version: 6.9.5
npm Version: 3.10.10
Git Version: 2.11.1
Bower Version: 1.8.0

Yazan: Emre Can ÖZTAŞ

Kitap Hakkında

Bu kitap; Bower'i anlatmak için yazılmıştır. Kitabı takip edebilmeniz için: temel web bilginiz ve az da olsa komut ekranı kullanabilme yeteneğinizin olması beklenmektedir. Konuların anlatımı basit düzeyde tutulmuştur. Yeni başlayanlar ve ileri düzey geliştiriciler kitabı rahatlıkla takip edebilir.

Memnun kalmanız dileğiyle.

Yazar Hakkında

Emre Can ÖZTAŞ, yazılım geliştirici, web girişimcisi, eğitmen ve yazardır. Çeşitli dillerde yazılım geliştirmektedir. Kendisine ait olan web projeleri vardır ve web girişimcilik alanında çalışmaları sürdürmektedir. Çeşitli alanlarda birebir veya grup olarak özel eğitimler vermektedir. Genellikle öğrencileri ondan memnundur. Bunun yarısına; bilgisi ve tecrübesi dahilinde Türkçe kaynağa destek ve öğrenmeye istekli olan herkesin yararlanabilmesi için e-kitap'lar yazmaktadır. Yazdıklarından herhangi bir ücret talep etmemiştir ve kazanmamıştır.

Yazar hakkında daha fazlasını merak ederseniz; kendisi www.emrecanoztas.com adresinde çeşitli konularda yazılar yazmakta ve paylaşımlarda bulunmaktadır. Aynı zamanda kitap hakkındaki; görüş, öneri ve yanıtları da bu adrese iletebilirsiniz.

İçindekiler

BÖLÜM 1: Başlarken.....	2
1.1 Bower Nedir?.....	2
1.2 Bower Kurulumu.....	2
BÖLÜM 2: Temel Kullanım.....	6
2.1 search.....	6
2.2 info.....	10
2.3 install.....	11
2.4 uninstall.....	13
2.5 update.....	14
2.6 list.....	15
BÖLÜM 3: .bowerrc.....	16
BÖLÜM 4: bower.json.....	18
4.1 init.....	18
4.2 save.....	21
4.3 Paketleri Yükleme.....	23
BÖLÜM 5: Proje Yapısı Oluşturma.....	26
BÖLÜM 6: Paket Yayınlama.....	29
6.1 Register.....	29
6.2 Unregister.....	30
BÖLÜM 7: Referanslar.....	31
7.1 Diğer Komutlar.....	31
7.1.1 cache.....	31
7.1.2 home.....	33
7.1.3 link.....	33
7.1.4 lookup.....	34
7.1.5 prune.....	35
7.2 Seçenekler.....	35
7.2.1 force.....	35
7.2.2 json.....	35
7.2.3 loglevel.....	36
7.2.4 offline.....	36
7.2.5 quit.....	36
7.2.6 silent.....	36
7.2.7 verbose.....	36
7.2.8 allow-root.....	36
KAYNAKLAR.....	38

BÖLÜM 1: Başlarken

Bu ilk bölümde; Bower nedir? Ne için Kullanılır? Bower kurulumu ve kitabın genelinde Bower kullanımı nasıldır gibi sorularının cevaplarını aramaya çalışacağız. Bu bir e – kitap. Sanırım oldukça kısa bir e – kitap olacak. Çünkü anlatabileceklerimiz oldukça sınırlı.

Hazırsanız, o halde hemen başlayalım!

1.1 Bower Nedir?

Web programlama da yaptıklarımız tipik olarak aşağıdaki gibidir:

1. Kullanılacak; Library ve Framework’lerin adreslerine git ve teker teker indir.
2. Klasörleri dizinlere çıkart.
3. Klasör içlerindeki gereksiz dosyaları temizle.
4. İndirilen kaynakları kullanmaya başla!

Ne kadar çok şeyle uğraşıyoruz değil mi? Peki bu işlerle uğraşmak yerine; ilk üç adımı otomatik olarak yapabilir miyiz? Evet, yapabiliriz!

Bower yukarıdaki adımları otomatize ederek bizler için sunar. Dolayısıyla buradan çıkacak sonuç: Bower bir paket yönetim aracıdır. Belirttiğiniz; library ve framework’leri gider bulur, indirir ve sizin kullanımınıza açar. Çok güzel değil mi?

Zaman geçtikçe; geliştiricilerin işleri oldukça kolaylaşıyor. Son zamanlar da oldukça sıklıkla kullandığım Bower, hakikaten oldukça kaliteli ve işlerimi daha kısa sürede gerçekleştirmeme yardım ediyor.

Bower’in kullanımı çok basit. Bower sayesinde daha profesyonel bir geliştirici olabilirsiniz. Bence en azından bir kere deneyin. Kesinlikle vazgeçemeyeceksiniz!

1.2 Bower Kurulumu

Bower, NodeJS üzerinde çalışan bir araç. Ayrıca, paketleri de Github üzerinden temin ediyor. Zaten şuan geliştirilen library veya framework’ler genelde Github üzerinde tutuluyor ve geliştiriliyor. İşte Bower de Github üzerinde bulunan dosyaların listesini tutuyor ve istenilen dosyaları Github üzerinden indiriyor. Dolayısıyla buradan çıkan sonuç: Bower için NodeJS, npm (NodeJS Package Manager) ve Git’in bilgisayarınız da kurulu olması gerekiyor.

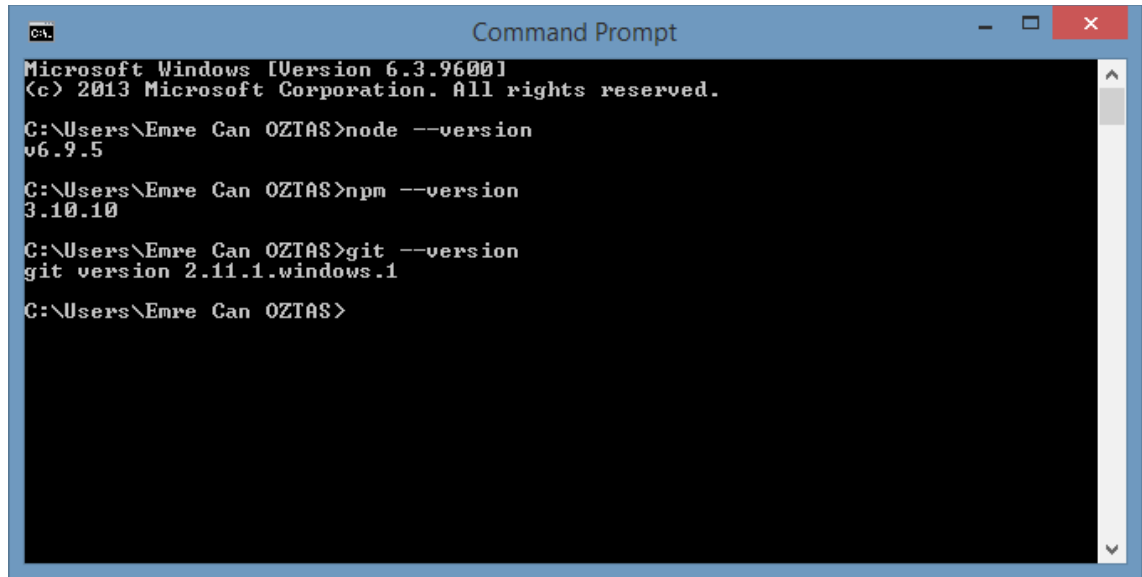
NodeJS, temini için aşağıdaki sayfayı ziyaret edebilirsiniz.

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

Git, temini için de aşağıdaki sayfayı ziyaret edebilirsiniz.

<https://git-scm.com/>

Kullandığım NodeJS, npm ve Git sürümleri de aşağıdaki gibidir.



```
C:\>
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Emre Can OZTAS>node --version
v6.9.5

C:\Users\Emre Can OZTAS>npm --version
3.10.10

C:\Users\Emre Can OZTAS>git --version
git version 2.11.1.windows.1

C:\Users\Emre Can OZTAS>
```

Her şey tamam olduğuna göre; Bower kurulumuna geçebiliriz. Bunun için cmd (Command Prompt, Terminal veya Console) açalım ve aşağıdaki komutu yazalım.

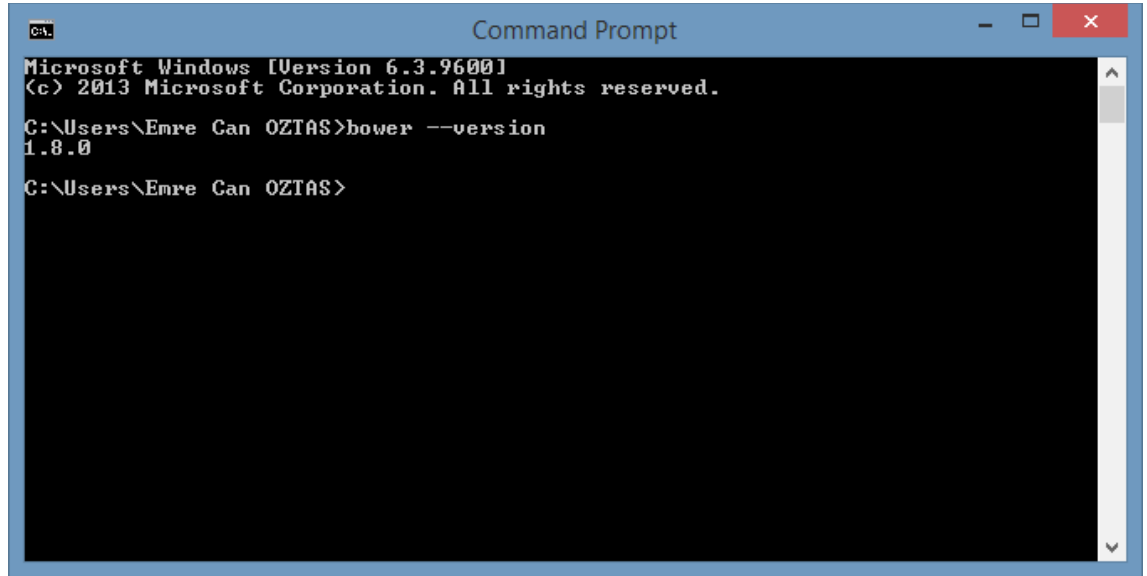
```
$ npm install -g bower
```

Yukarıdaki komuttan sonra herhangi bir hata mesajı ile karşılaşmamış isek; Bower başarılı bir şekilde kurulacaktır.

Bower kurulumunu da aşağıdaki komut ile kontrol edebiliriz.

```
$ bower -version
# veya
$ bower --version
```

Örneğin; benim ekran çıktım aşağıdaki gibidir.



```
C:\>
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Emre Can OZTAS>bower --version
1.8.0
C:\Users\Emre Can OZTAS>
```

Bower, komutları hakkında yardım almak için de aşağıdaki komutu yazmamız yeterli.

```
$ bower help
# veya
$ bower -help
# veya
$ bower --help
# veya
$ bower -h
```



```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Emre Can OZTAS>bower --help

Usage:
  bower <command> [<args>] [<options>]

Commands:
  cache      Manage bower cache
  help       Display help information about Bower
  home       Opens a package homepage into your favorite browser
  info       Info of a particular package
  init       Interactively create a bower.json file
  install    Install a package locally
  link       Symlink a package folder
  list       List local packages - and possible updates
  login      Authenticate with GitHub and store credentials
  lookup     Look up a single package URL by name
  prune      Removes local extraneous packages
  register   Register a package
  search     Search for packages by name
  update     Update a local package
  uninstall  Remove a local package
  unregister Remove a package from the registry
  version    Bump a package version

Options:
  -f, --force      Makes various commands more forceful
  -j, --json       Output consumable JSON
  -l, --loglevel   What level of logs to report
  -o, --offline    Do not hit the network
  -q, --quiet      Only output important information
  -s, --silent     Do not output anything, besides errors
  -U, --verbose    Makes output more verbose
  --allow-root     Allows running commands as root
  -v, --version    Output Bower version
  --no-color      Disable colors

See 'bower help <command>' for more information on a specific command.

C:\Users\Emre Can OZTAS>
```

Yukarıdaki listelenmiş komutlar hakkında daha detaylı bilgi almak için yine -help komutunu kullanabilirsiniz. Örneğin; -install komutu hakkında bilgi almak için aşağıdaki komutu yazmamız yeterli olacaktır.

```
$ bower help install
# veya
$ bower -help install
# veya
$ bower --help install
# veya
$ bower -h install
```

Bower kurulumunu da tamamladığımıza göre; artık Bower'i kullanmaya daha doğrusu sağladığı avantajlardan yararlanmaya başlayabiliriz.

BÖLÜM 2: Temel Kullanım

Bu bölümde; Bower'ın temel veya basit kullanımı üzerinde duracağız. İlerleyen bölümlerde detaylara ineceğiz. Ne kadar detaylara ineceğimizi de bekleyip görelim diyelim ve hemen konumuza başlayalım!

2.1 search

-search yani “ara” komutu ile Bower üzerinden paket araması yapabiliriz.

Komutumuzun kullanımı aşağıdaki gibidir.

```
$ bower search package_name
```

Yukarıdaki kullanımda da gördüğünüz gibi kullanımı oldukça basit.

Örnek olması açısından şimdi jQuery'i arayalım. Bakalım Bower bize ne tepkiler verecek?

```
$ bower search jquery
```

Yukarıdaki komutumuzdan sonra ekran çıktımıza bakalım.

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Emre Can OZTAS>bower search jquery
Search results:

jQuery https://github.com/jquery/jquery.git
jquery https://github.com/jquery/jquery-dist.git
jquery.x https://github.com/jljlabs/jquery.x.git
jt_jquery https://github.com/vicanso/jt_jquery.git
jquery.Q https://github.com/jsbuzz/jquery.Q.git
jquery-m https://github.com/meetup/jquery.git
jquery.j2d https://github.com/fsggs/jquery.j2d.git
jquery.hx https://github.com/millennialmedia/jquery.hx.git
jquery-ts https://github.com/andraaspar/jquery-ts.git
jquery-tm https://github.com/trymore/jquery-tm.git
jquery.gk https://github.com/ezoapp/jquery.gk.git
jquery-jec https://github.com/RaYell/jquery-jec.git
no-jquery https://github.com/Merott/no-jquery.git
hg-jquery https://github.com/HackerGacho/hg-jquery.git
dt-jquery https://github.com/jedmao/dt-jquery.git
jquery-fb https://github.com/amitab/jqueryFB.git
sw-jquery https://github.com/sowirepo/jquery1.8.3.git
jquery-ui https://github.com/components/jqueryui.git
jquery.up https://github.com/tonekk/jquery.up.git
jquery-py https://github.com/seandong/jquery-py.git
Rx-jQuery https://github.com/Reactive-Extensions/rxjs-jquery.git
jquery.ui https://github.com/jquery/jquery-ui.git
jquery.os https://github.com/suprMax/jquery.os.git
jquery-ol https://github.com/jordanmarkov/jquery-ol.git
jquery.ua https://github.com/cloudcome/jquery.ua.git
hn.jquery https://github.com/loi-chuanoi/jquery.git
jquery2 https://github.com/iki/jquery2.git
jqueryyp https://github.com/Ensigthen/jqueryyp.git
jquery-rss https://github.com/sdepold/jquery-rss.git
jquery.bem https://github.com/zenwalker/jquery-bem.git

C:\Users\Emre Can OZTAS>
```

Görüldüğü gibi jQuery adında bir çok proje dizini listeleniyor. Bu bulunan projelerin dizinlerinin uzantılarına baktığımız zaman .git uzantılarını görüyoruz. Yani; Bower Github üzerindeki dosyalara bakıyor. Burada aklınıza şöyle bir şey gelmiş olabilir: “Benim Github’taki projelerimi de bulabilir mi?”. Hayır bulamaz! Çünkü projelerinizi Bower’e eklemek zorundasınız. Biz bu konudan bahsetmeyeceğiz lakin size nasıl ekleneceğine dair yolu göstereceğiz.

-search komutunu aklınıza gelebilecek herhangi library veya kütüphaneyi aramak için kullanabilirsiniz. Örneğin; Bootstrap için Immm... vazgeçtim, nereye kadar Bootstrap (!) biz UIkit’e bakalım.

```
Command Prompt

C:\Users\Enre Can OZTAS>bower search uikit
Search results:

uikit https://github.com/uikit/uikit.git
mw-uikit https://github.com/mwaylabs/uikit.git
cf-uikit https://bitbucket.org/cureforward/cf-uikit.git
gsd-uikit https://github.com/julien-sarazin/gsd-uikit.git
uikit-rtl https://github.com/Oudmane/uikit-rtl.git
dev_uikit https://github.com/uikit/uikit.git
jdds-uikit https://github.com/juiceystuff/jdds-uikit.git
wappa-uikit https://github.com/wappamobile/wappa-uikit.git
laxar-uikit https://github.com/LaxarJS/dist-laxar-uikit.git
shelfi-uikit https://github.com/shelfi/bower-uikit.git
mzcore-uikit https://github.com/menduz/mzcore-uikit.git
vuikit https://github.com/vuikit/vuikit.git
angular-uikit https://github.com/orestisioakeimidis/angular-uikit.git
backbone.uikit https://github.com/vash15/backbone.uikit.git
uikit-extended https://github.com/JeremiahGeronimo/uikit-extended.git
availability-uikit https://github.com/Availability/availability-uikit.git
protosite.uikit https://github.com/BPanchenko/protosite.uikit.git
datatables-uikit-crud-plugin https://github.com/jampajeen/datatables-uikit-c
rud-plugin.git
def-ui https://github.com/16code/def-uikit.git

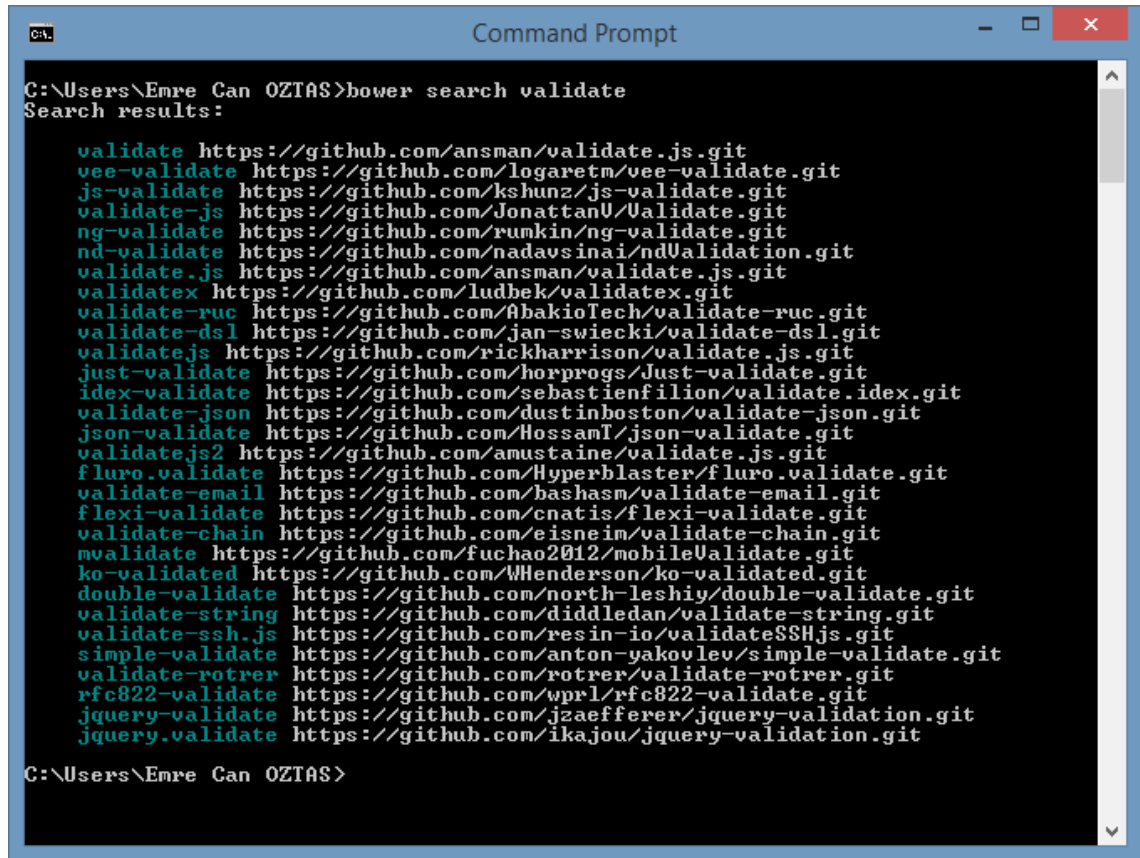
C:\Users\Enre Can OZTAS>
```

Görüldüğü gibi UIkit adındaki tüm dosyaları listeledi.

Bower sadece adını bildiğiniz library veya framework'leri aramaz. Bunun yerine tag yani etiket vererekte arama işlemini gerçekleştirebilirsiniz. Örneğin; -validate etiketiyle arama işlemini gerçekleştirelim.

```
$ bower search validate
```

Yukarıdaki arama işlemimizin sonucu da aşağıdaki gibi olacaktır.



```
C:\Users\Emre Can OZTAS>bower search validate
Search results:

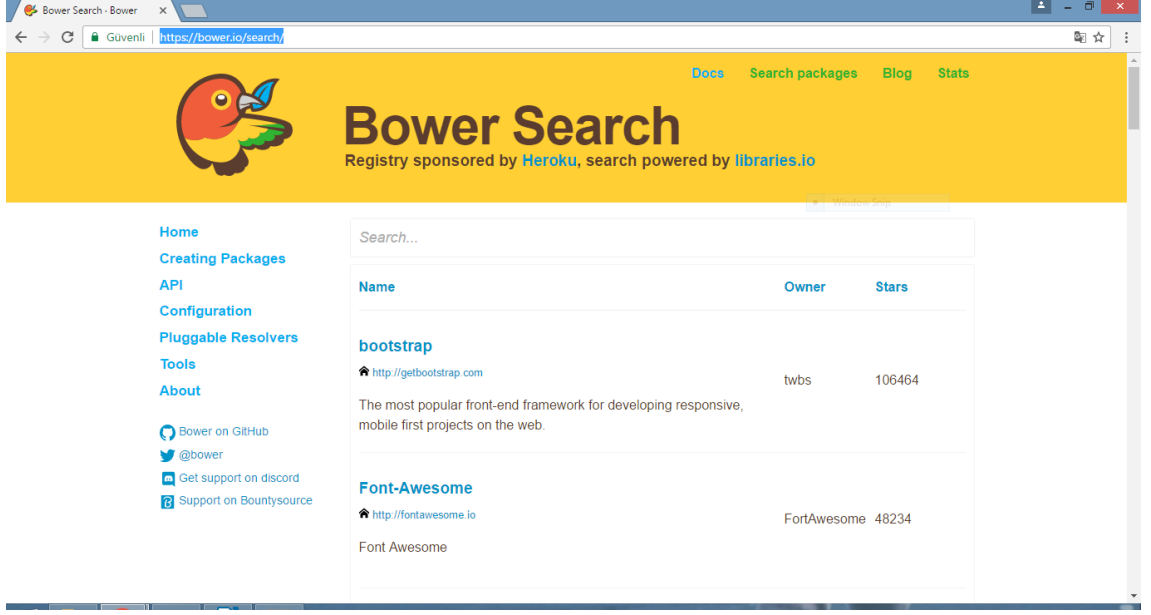
validate https://github.com/ansman/validate.js.git
vee-validate https://github.com/logaretm/vee-validate.git
js-validate https://github.com/kshunz/js-validate.git
validate-js https://github.com/JonattanU/Validate.git
ng-validate https://github.com/rumkin/ng-validate.git
nd-validate https://github.com/nadavsinai/ndValidation.git
validate.js https://github.com/ansman/validate.js.git
validateex https://github.com/ludbek/validateex.git
validate-ruc https://github.com/AbakioTech/validate-ruc.git
validate-dsl https://github.com/jan-swiecki/validate-dsl.git
validatejs https://github.com/rickharrison/validate.js.git
just-validate https://github.com/horprogs/Just-validate.git
idex-validate https://github.com/sebastienfilion/validate.idex.git
validate-json https://github.com/dustinboston/validate-json.git
json-validate https://github.com/HossanI/json-validate.git
validatejs2 https://github.com/amustaine/validate.js.git
fluro.validate https://github.com/Hyperblaster/fluro.validate.git
validate-email https://github.com/bashasm/validate-email.git
flexi-validate https://github.com/cnatis/flexi-validate.git
validate-chain https://github.com/eisneim/validate-chain.git
mvalidate https://github.com/fuchao2012/mobileValidate.git
ko-validated https://github.com/WHenderson/ko-validated.git
double-validate https://github.com/north-leshiy/double-validate.git
validate-string https://github.com/diddledan/validate-string.git
validate-ssh.js https://github.com/resin-io/validateSSHjs.git
simple-validate https://github.com/anton-yakovlev/simple-validate.git
validate-rottrer https://github.com/rottrer/validate-rottrer.git
rfc822-validate https://github.com/wprl/rfc822-validate.git
jquery-validate https://github.com/jzaefferer/jquery-validation.git
jquery.validate https://github.com/ikajou/jquery-validation.git

C:\Users\Emre Can OZTAS>
```

Yukarıdaki ekran çıktısında da görüldüğü gibi proje dizinleri listelendi. Birazdan bu dizinlerdeki paketlerin nasıl kurulacağına bakacağız! Şimdilik arama işlemi nasıl yapılır, bunun üzerinde durduk.

Komut ekranı yerine; Bower'in web sayfasından da paket araması yapabilirsiniz.

<https://bower.io/search/>



2.2 info

Bir önceki başlığımızda: paket ismine ve etikete göre arama işlemi nasıl gerçekleştirilir, buna değindik. Şimdi bu paketler hakkında nasıl bilgi edinebiliriz anlamaya çalışalım. Öyle ya her paketi indirip içeriklerini kontrol edemeyiz!

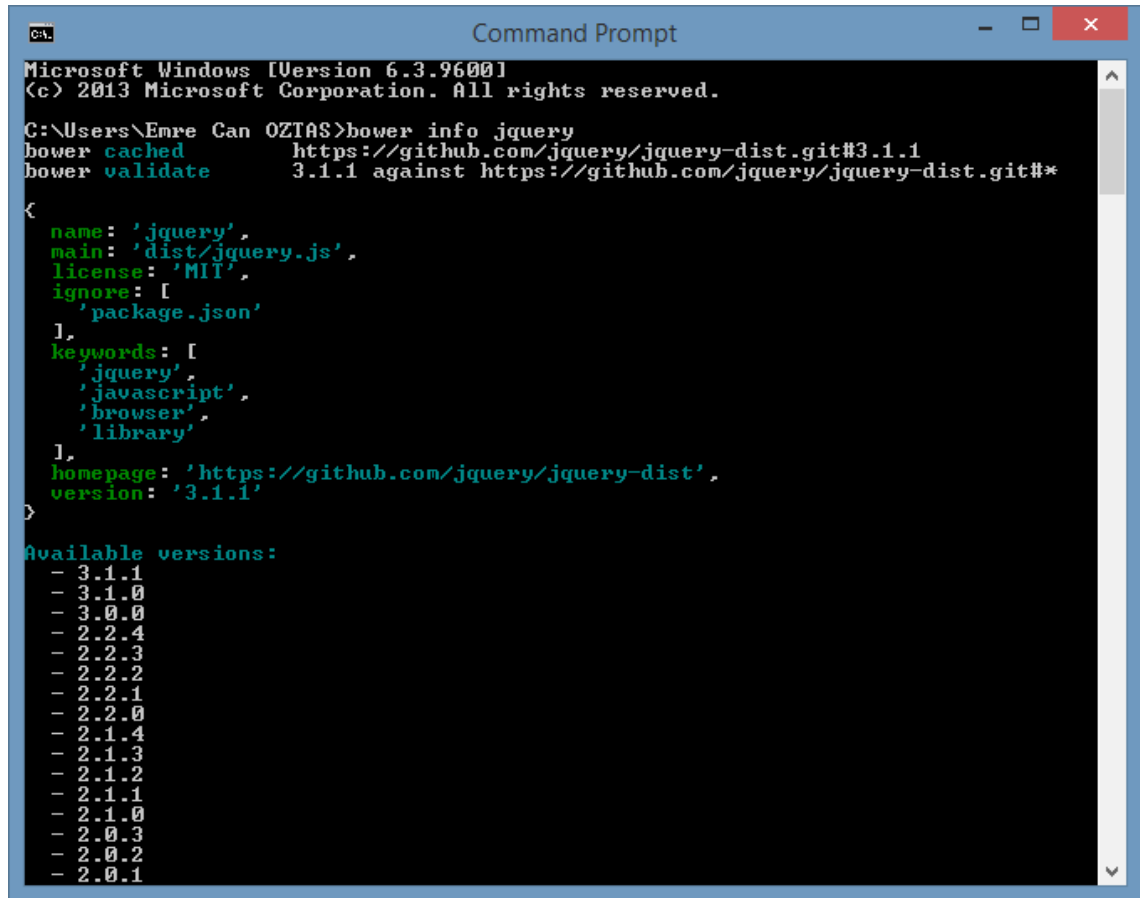
-info komutunun kullanımı aşağıdaki gibidir.

```
$ bower info package_name
```

Örnek olması açısından jQuery paketine bakalım.

```
$ bower info jquery
```

Ekran çıktımız da aşağıdaki gibi olacaktır.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Enre Can OZTAS>bower info jquery
bower cached      https://github.com/jquery/jquery-dist.git#3.1.1
bower validate    3.1.1 against https://github.com/jquery/jquery-dist.git#*

{
  name: 'jquery',
  main: 'dist/jquery.js',
  license: 'MIT',
  ignore: [
    'package.json'
  ],
  keywords: [
    'jquery',
    'javascript',
    'browser',
    'library'
  ],
  homepage: 'https://github.com/jquery/jquery-dist',
  version: '3.1.1'
}

Available versions:
- 3.1.1
- 3.1.0
- 3.0.0
- 2.2.4
- 2.2.3
- 2.2.2
- 2.2.1
- 2.2.0
- 2.1.4
- 2.1.3
- 2.1.2
- 2.1.1
- 2.1.0
- 2.0.3
- 2.0.2
- 2.0.1
```

Görüldüğü gibi; Bower jQuery'i paketini inceledi ve bize sonucu iletti. Bu bilgiler: bower.json dosyasında yazan bilgiler. bower.json dosyası hakkında ilerleyen bölümlerde bolca konuşacağız.

Yukarıdaki içeriği incelediğimizde: jQuery'in MIT lisansına sahip olduğunu, keyword'leri: -jquery, -javascript, -browser ve -library, güncel versiyonunu ve önceki versiyonlarının listesini görebiliyoruz. Bu versiyon numaraları önemli. İlerleyen kısımda değineceğiz. Çünkü istediğimiz herhangi bir sürümü indirebiliriz.

2.3 install

-install komutu, Bower aracılığıyla; istediğimiz paketin indirilmesini sağlar.

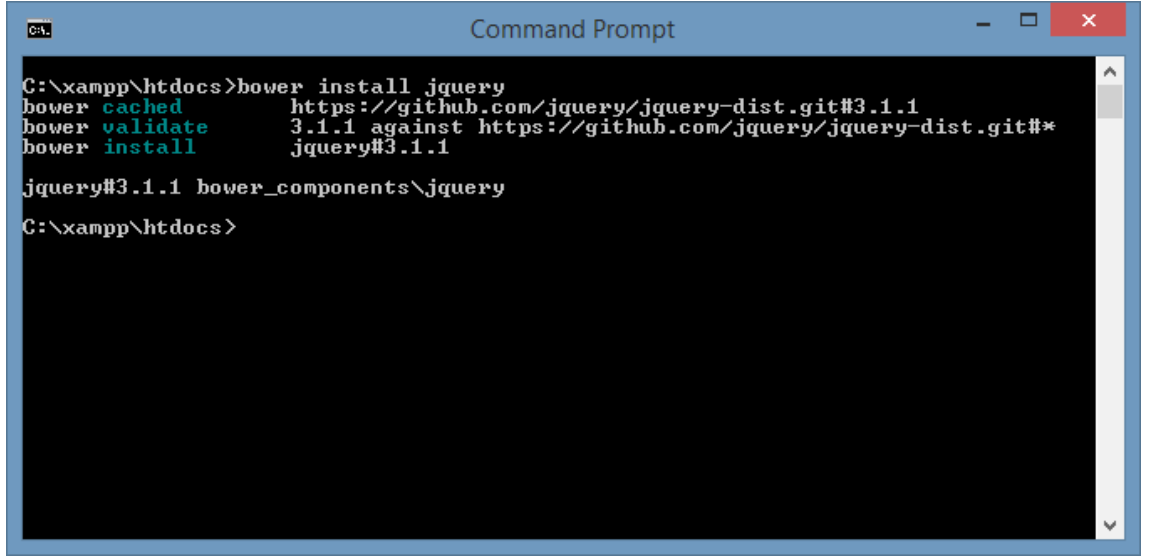
-install komutunun kullanımı aşağıdaki gibidir.

```
$ bower install package_name
```

Örneğin; jQuery paketini indirmek istediğimiz de aşağıdaki komutu yazmamız yeterlidir.

```
$ bower install jquery
```

Yukarıdaki komuttan sonra jQuery indirilecektir. Örnek olması açısından jQuery’i indirelim.

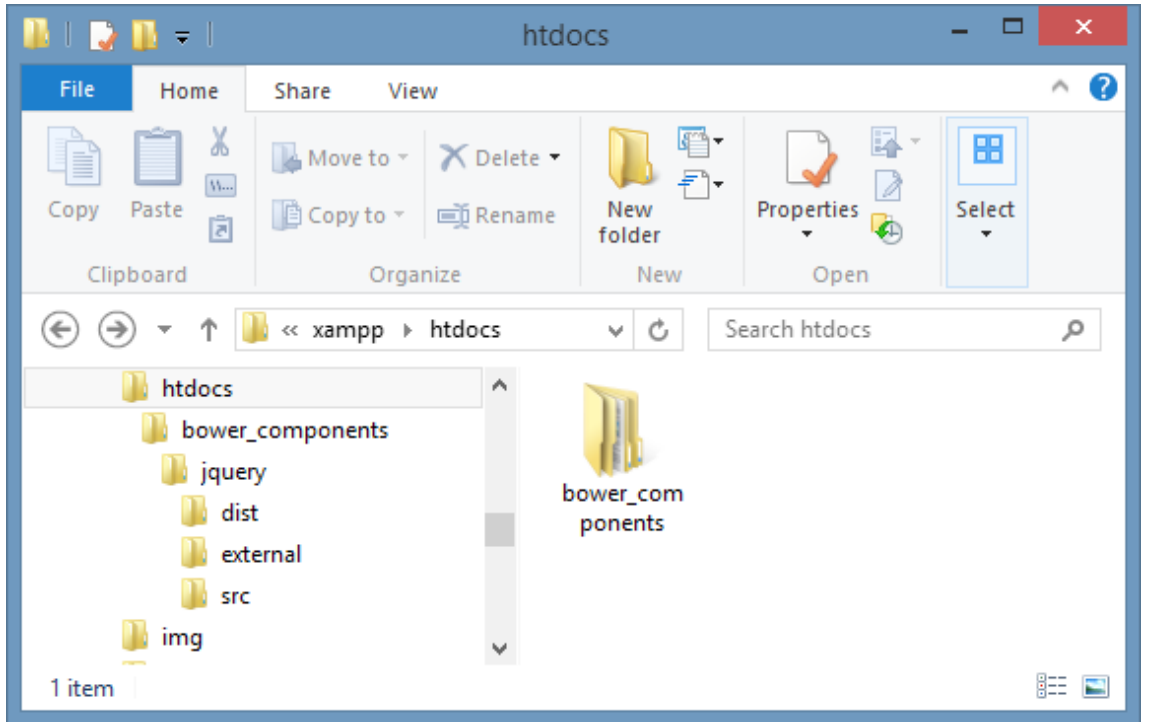


```
Command Prompt

C:\xampp\htdocs>bower install jquery
bower cached      https://github.com/jquery/jquery-dist.git#3.1.1
bower validate    3.1.1 against https://github.com/jquery/jquery-dist.git#*
bower install     jquery#3.1.1

jquery#3.1.1 bower_components\jquery
C:\xampp\htdocs>
```

Ekran alıntısında da görüldüğü gibi jQuery indirildi.



Default olarak: Bower indirdiği dosyaları “bower_components” dosyasına konumlandırır. Sol taraftaki listede de bunu görebilirsiniz.

Burada dikkat etmeniz gereken 2 konu var.

1. İndirilecek paketin versiyonu belirtilmezse: en son versiyon indirilir.
2. Bower, Github üzerindeki ilgili paketin tüm dosyalarını indirir.

Şayet; son versiyon yerine önceki sürümleri kullanmak istiyorsak bunu komut ekranında belirtebiliriz.

```
$ bower install jquery#2.0.1
```

Örneğin; yukarıdaki komutta, jQuery 2.0.1 sürümünün indirilmesini istedik. Burada: # ile sürüm numarasını belirtiyoruz.

Sadece paket ismiyle indirme yapmak yerine: doğrudan github hesabından, .git uzantılı adresleri vererek ve url kullanarak indirme işlemi yapabilirsiniz.

```
# github hesabi
$ bower install user_name/package_name

# .git uzantili
$ bower install git://github.com/user/package_name.git

# url
$ bower install http://example.com/script_name.js
```

-install konusunda göstermek istediğim son bir şey var. Birden fazla paket indirecekseniz; bu paketleri tek tek indirmenize gerek yok. Paketleri tek seferde indirebilirsiniz. Nasıl mı? Aşağıdaki gibi.

```
$ bower install package_name1 package_name2 package_name3 ...
```

Örneğin: jQuery, Bootstrap ve Ulkit'i indirelim.

```
$ bower install jquery#2.0.1 bootstrap#3 uikit
```

2.4 uninstall

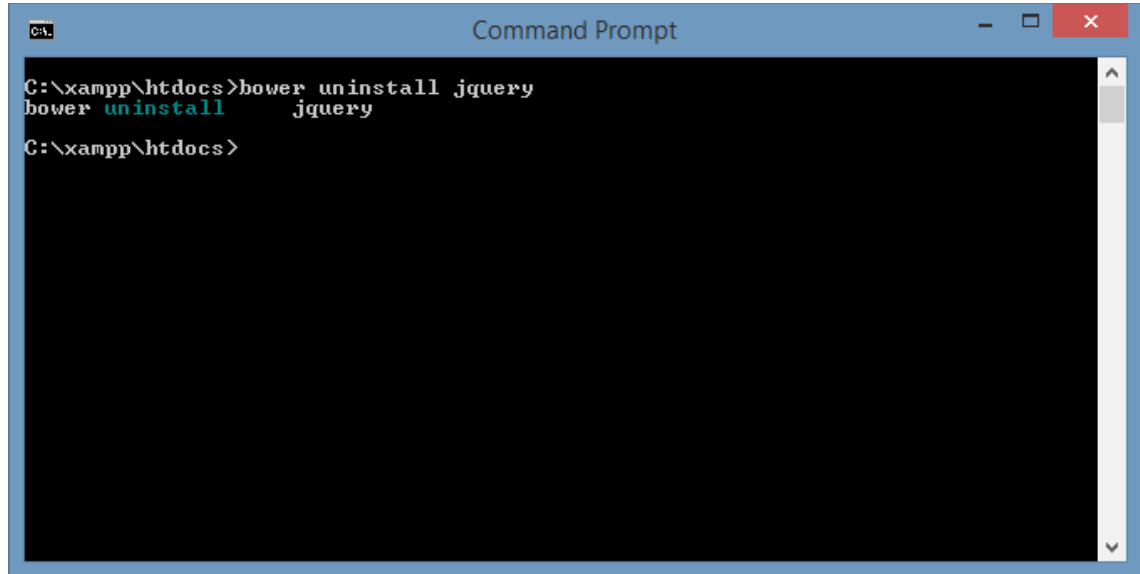
-uninstall komutu, indirmiş olduğunuz paketlerin silinmesini gerçekleştirir.

-uninstall komutunun kullanımı aşağıdaki gibidir.

```
$ bower uninstall package_name
```

Örneğin; daha önce jQuery'i indirmiştik. Şimdi bunu kaldırabiliriz.

```
$ bower uninstall jquery
```



Ekran alıntısında da görüldüğü üzere; jQuery’i library’sini sildik.

Ayrıca birden fazla paketi de silebilirsiniz. -install komutunda yaptığımız gibi paket isimlerini yan yana yazmamız yeterli. Aşağıda olduğu gibi.

```
$ bower uninstall package_name1 package_name2 package_name3 ...
```

2.5 update

Kullanmış olduğunuz paketleri yükseltmek isteyebilirsiniz. Son sürümleri kullanmak her zaman iyidir. Çünkü güvenlik güncellemeleri v.s geldiği zaman bu paketlerin güvenlik sebebiyle mutlaka güncellemeleri gerekir. Bower ortamında da -update işlemi oldukça basit.

-update komutunun kullanımı aşağıdaki gibidir.

```
$ bower update
```

Yukarıdaki komuttan sonra; indirmiş olduğunuz tüm paketler güncellenir. Tüm paketlerin güncellemesi yerine sadece belli paketlerin güncellemesini isteyebilirsiniz. Bunun içinde paket ismini yazmanız yeterlidir.

```
$ bower update package_name
```

Birden fazla paketiniz varsa güncellenecek; bu paketleri yan yana yazarak güncelleyebilirsiniz.

```
$ bower update package_name1 package_name2 package_name3 ...
```

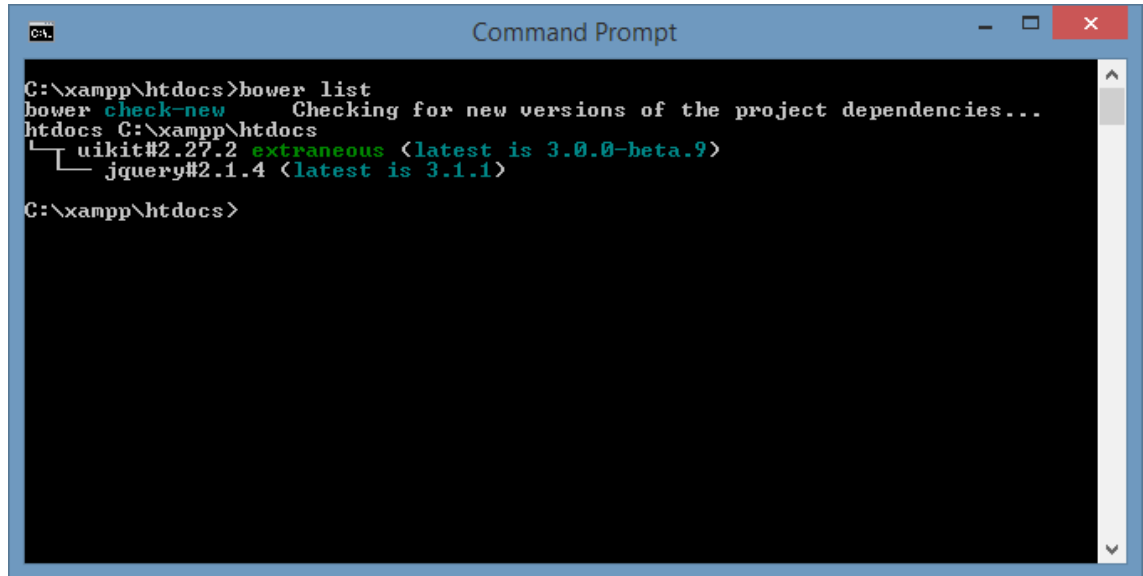
2.6 list

-list komutu, daha önce indirilmiş olan paketleri listeler.

-list komutunun kullanımı aşağıdaki gibidir.

```
$ bower list
```

Örneğin; benim daha önce indirmiş olduğum paketleri listeyeyim.



```
C:\xampp\htdocs>bower list
bower check-new    Checking for new versions of the project dependencies...
htdocs C:\xampp\htdocs
├─ ui-kit#2.27.2 extraneous <latest is 3.0.0-beta.9>
└─ jquery#2.1.4 <latest is 3.1.1>
C:\xampp\htdocs>
```

Yukarıdaki ekran alıntısında da görüldüğü gibi indirmiş olduğum paketler listeleniyor. Ayrıca; bu paketlerin yeni sürümleri var mı yok mu bunu da bana bildiriyor. Örneğin; Ulkit için; 3.0.0-beta.9 ve jQuery için; 3.1.1 sürümleri mevcut. Bir önceki başlıktaki; -update komutunu kullanarak bu paketlerimi update edebilirim.

BÖLÜM 3: .bowerrc

Önceki bölümde; genel olarak Bower hakkında detaylıca bahsetmeye çalıştık. Sadece bir önceki bölümde anlatılanları bilerseniz bile Bower'i rahatça kullanabilirsiniz. Lakin bu kadarı bize yetmez. Daha da derinlere ineceğiz ve A'dan Z'ye Bower'i anlamaya daha doğrusu anlayarak kullanmaya çalışacağız.

.bowerrc, bir dosyadır. Daha önce hiç Vim kullandınız mı bilmiyorum. Kullanmış iseniz; vim'in de .vimrc dosyası bulunur. İşte .bowerrc dosyası da .vimrc dosyası gibidir. Vim'den haberiniz yoksa; .browserrc, .htaccess dosyasına da benzer diyebiliriz. Yani kullanım amaçları aynı: ayarlar...

.bowerrc dosyası, bower ile ilgili ayarların saklandığı bir dosyadır. Kısaca; Bower'ın konfigürasyon dosyasıdır. Bu dosyanın uzantısı yok. İstedığınız daha doğrusu çalıştığınız dizine .bowerrc isminde bir dosya açmanız yeterli. İçeriği de bir JSON yapısında olmalı.

Windows ortamında; doğrudan başında nokta (.) olan bir dosya oluşturamazsınız. GNU / Linux dünyasında ise; başında nokta (.) olan bir dosya gizli bir dosya olur. O yüzden bu dosyayı kullandığınız editor ortamında oluşturmanız gerekecektir. Mesela biz bir tane oluşturalım. Peki bu dosyanın içerisine ne yazacağız?

Bower'in kendi sitesinde örnek bir .bowerrc dosyası oluşturulmuş. Bu dosyanın içeriği aşağıdaki gibidir.

```
{
  "cwd": "~/.my-project",
  "directory": "bower_components",
  "registry": "https://bower.herokuapp.com",
  "shorthand-resolver": "git://github.com//.git",
  "proxy": "http://proxy.local",
  "https-proxy": "http://proxy.local",
  "ca": "/var/certificate.pem",
  "color": true,
  "timeout": 60000,
  "save": true,
  "save-exact": true,
  "strict-ssl": true,
  "storage": {
    "packages" : "~/.bower/packages",
    "registry" : "~/.bower/registry",
    "links" : "~/.bower/links"
  },
  "interactive": true,
  "resolvers": [
    "mercurial-bower-resolver"
  ],
  "shallowCloneHosts": [
```

```
    "myGitHub.example.com"
  ],
  "scripts": {
    "preinstall": "",
    "postinstall": "",
    "preuninstall": ""
  },
  "ignoredDependencies": [
    "jquery"
  ]
}
```

Siz de buna benzer bir .bowerrc dosyası oluşturabilirsiniz. Ben de devamlı bir .bowerrc dosyası oluşturuyorum. Lakin yukarıdaki gibi karmakarışık bir dosya oluşturmak yerine daha sade çok sade bir dosya oluşturuyorum.

Benim genelde hazırladığım .bowerrc dosyamın içeriği aşağıdaki gibi olur.

```
{
  "directory": "assets"
}
```

Peki yukarıdaki dosyada ne belirtiyorum? Hatırlarsanız; Bower, indirdiği paketleri “bower_components” dizinine konumlandırıyor. Lakin “directory” anahtar kelimesi ile ben basit olarak; yeni paketlerin “assets” dizinine konumlandırmasını söylüyorum. Dilerseniz; farklı bir dizin ismi de yazabilirsiniz. Seçim sizlerin, dostlarım!

Gelelim Bower’in sitesinde oluşturduğu örnek .bowerrc dosyasına. Açıkcası ben bu dosyada yazılanların çoğu hakkında bir bilgim yok. Bu konuda tek cahil ben miyim diye küçük bir araştırma yaptım lakin bu konuda herhangi bir yazıyla karşılaşamadım. Baktığım sayfaların çoğu ya .bowerrc dosyasına hiç değinmemiş, değinenler de .bowerrc dosyalarını benim oluşturduğum gibi oluşturmuş. O yüzden bu konuda herhangi bir bilgim yok, dostlarım!

BÖLÜM 4: bower.json

Bir önceki bölümde; .bowerrc dosyasından bahsetmiştik. Bu bölümde ise; bower.json dosyası ile paketlerimizi daha doğrusu projemizi yönetmeyi anlamaya çalışacağız.

bower.json dosyası, projeniz hakkında çeşitli bilgileri ve özellikle de projenizin bağımlılıklarını belirlemenizi sağlar. Yani bu dosya içerisine; projeniz hakkındaki detayları ve hangi paketleri kullandığınızı yazabilirsiniz. Bu dosyayı oluşturmak zorunda değilsiniz lakin projelerinizi otomatize etmek isterseniz; oluşturmanızda yarar var. Çünkü her seferinde projenizdeki paketleri güncelleme v.s gibi işlerle uğraşmak yerine, bower.json dosyasında belirlediğiniz bağımlılıkları otomatik olarak; Bower sizin yerinize kontrol edebilir.

4.1 init

bower.json dosyasını kendiniz oluşturabileceğiniz gibi cmd (Command Prompt, Terminal veya Console) ekranından da oluşturabilirsiniz. Bunun için aşağıdaki komut vermelisiniz.

```
bower init
```

Yukarıdaki komuttan sonra; komut ekranını size bazı sorular sormaya başlayacaktır. Bu sorulara verdiğiniz yanıtlara göre bower.json dosyasının içeriğini belirleyecektir.

O halde yukarıdaki komutumuzu kullanarak bir bower.json dosyası oluşturalım ve bu dosya üzerinden konuşmaya devam edelim.

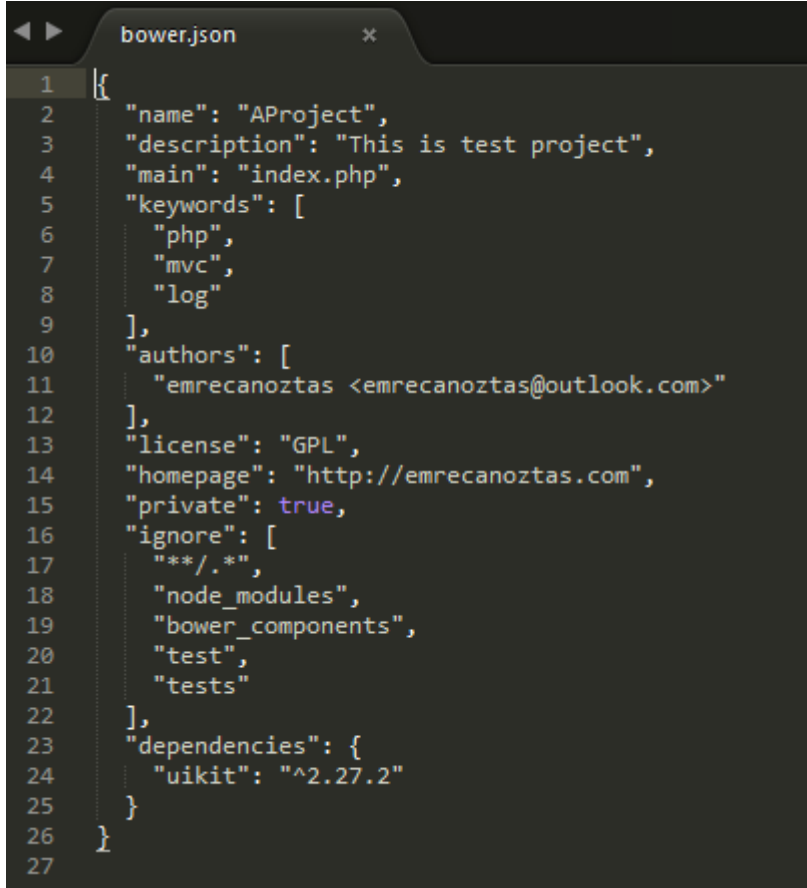
```
Command Prompt

C:\xampp\htdocs>bower init
? name AProject
? description This is test project
? main file index.php
? keywords php mvc log
? authors emrecaoztas <emrecaoztas@outlook.com>
? license GPL
? homepage http://emrecaoztas.com
? set currently installed components as dependencies? Yes
? add commonly ignored files to ignore list? Yes
? would you like to mark this package as private which prevents it from being ac
? would you like to mark this package as private which prevents it from being ac
? cidentally published to the registry? Yes

{
  name: 'AProject',
  description: 'This is test project',
  main: 'index.php',
  keywords: [
    'php',
    'mvc',
    'log'
  ],
  authors: [
    'emrecaoztas <emrecaoztas@outlook.com>'
  ],
  license: 'GPL',
  homepage: 'http://emrecaoztas.com',
  private: true,
  ignore: [
    '**/*.*',
    'node_modules',
    'bower_components',
    'test',
    'tests'
  ],
  dependencies: {
    uikit: '^2.27.2'
  }
}

? Looks good? Yes
C:\xampp\htdocs>
```

Yukarıdaki ekran alıntısında da görüldüğü üzere ./htdocs dizininde bu komutumuzu verdim. Daha öncede dediğim gibi komutumuzdan sonra bana çeşitli sorular sordu. Bu sorular ve bu sorulara verdiğim yanıtları ekran alıntısında da görebilirsiniz. Şimdi oluşturmuş olduğumuz bower.json dosyasına bir bakalım.



```
1 {
2   "name": "AProject",
3   "description": "This is test project",
4   "main": "index.php",
5   "keywords": [
6     "php",
7     "mvc",
8     "log"
9   ],
10  "authors": [
11    "emrecaoztas <emrecaoztas@outlook.com>"
12  ],
13  "license": "GPL",
14  "homepage": "http://emrecaoztas.com",
15  "private": true,
16  "ignore": [
17    "**/*.*",
18    "node_modules",
19    "bower_components",
20    "test",
21    "tests"
22  ],
23  "dependencies": {
24    "uikit": "^2.27.2"
25  }
26 }
27
```

Gördüğünüz gibi dosyamız oluşturulmuş. Ben herhangi bir müdahale de bulunmadım. Olduğu gibi paylaştım.

Burada dikkatinizi çekmek istediğim bir konu var. “dependencies” başlığı altında daha önce indirdiğim Ulkit’i eklendi. Şayet daha önce indirmiş olduğunuz paketler varsa; bower.json dosyasını oluştururken o dizini kontrol eder ve indirilen paketler varsa dependencies’e ekler. Yoksa; o zaman dependencies başlığı boş kalır. Siz burayı daha sonra herhangi bir text editor ile doldurabilirsiniz veya paketleri indirirken bower.json dosyasına eklenmesini sağlayabilirsiniz.

Bower ile paketler indirilirken; bower.json dosyasına paketlerin nasıl eklendiğine değineceğiz. Lakin önce bu sayfadaki başlıklara bir bakalım.

name: paket veya projenin ismi. -name ismini oluştururken; 50 karakterin üzerine çıkmamanız önerilir. Ayrıca -name, içerisinde karakter (küçük karakterler), sayılar ve sadece alt çizgi işareti kullanılabilir. Lakin proje ismi alt çizgi ile başlayamaz veya bitemez. Örneğin; biz projemizde büyük harflerde kullandık. Aslında bu yanlış. Bu yanlış daha sonra karşımıza çıkacak. Yani bu konuya değineceğiz.

description: proje veya paket hakkında açıklama yazmak için kullanılan başlıktır. Kullanımı zorunlu değildir. Lakin kullanırsanız; sizden sonrakilere yol gösterici olur. Maksimum 140 karakterlik açıklamalar yazabilirsiniz.

main: Projenin veya paketin main dosyasını belirtmek için kullanılır.

keywords: Proje veya paketin etiketlerinin açıklanmasının yapıldığı alandır. Bildiğiniz gibi daha önce -search komutu ile sadece paket ismine göre değil, etiket ismine de göre arama işlemi yapabiliyorduk. İşte oradaki etiket buradaki etiketlerdir. Etiket isimlerini de her kelimenin arasına birer boşluk bırakarak, cmd ekranında oluşturmuştuk. -name alanını oluştururken dediklerimiz -keywords için de geçerlidir.

authors: Proje veya paketin sahip veya sahiplerinin olduğu alandır. Burada da birden fazla -authors varsa o zaman; her isimden sonra virgül koymamız gerekir.

license: Proje veya paketin kullanım ve dağıtım lisansıdır.

homepage: Proje veya paketin yayınlandığı web sayfasıdır. Birden fazla lisans belirleyebilirsiniz. Lisanlar hakkında herhangi bir bilginiz yoksa aşağıdaki sayfayı ziyaret edebilirsiniz.

<https://spdx.org/licenses/>

private: Proje veya paketin serbest veya dağıtımına kapalı olup olmadığını belirtir. Buradaki değer boolean bir değer olmalıdır.

ignore: Proje veya paket yüklenirken görmemezlikten gelineceklerin listesidir.

dependencies: Bu konudan daha önce bahsetmiştik lakin yine bahsedelim. Proje veya paketteki bağımlılıklardır.

Bakın! Yukarıdakiler dışında diğer alanlar da var. Bu alanları; kendi paketinizi yayınlarken de kullanabilirsiniz. Yani oluşturmuş olduğumuz bower.json dosyası aynı zamanda; sizin oluşturacağınız ve Bower üzerinden yayınlayacağınız paketlerde de bulunmalıdır. Daha da kısa açıklaması; Bower üzerinden kendi paket veya projelerinizi de yayınlatabilirsiniz. Bu konuda aşağıdaki sayfayı inceleyebilirsiniz.

<https://bower.io/docs/creating-packages/>

4.2 save

Oluşturulan bower.json dosyasına, “dependencies” yani “bağımlılık”lar iki farklı şekilde eklenir veya düzenlenir. Bunlar:

1. Herhangi bir text editor ile doğrudan ekleme/silme.
2. Paketlerin indirilmesi sırasında, Bower tarafından bu bağımlılıkların eklenme/silme işlemi.

İki farklı seçeneğiniz var. Yani hangisini kullanmak size kolay gelirse! Ben genelde ikinci yolu seçiyorum. Çünkü bower.json dosyasıyla uğraşmak yerine; paketlerin indirilmesi sırasında Bower tarafından bower.json dosyasının kontrol edilmesini istiyorum.

Bu başlık altında ikinci madde hakkında konuşacağız.

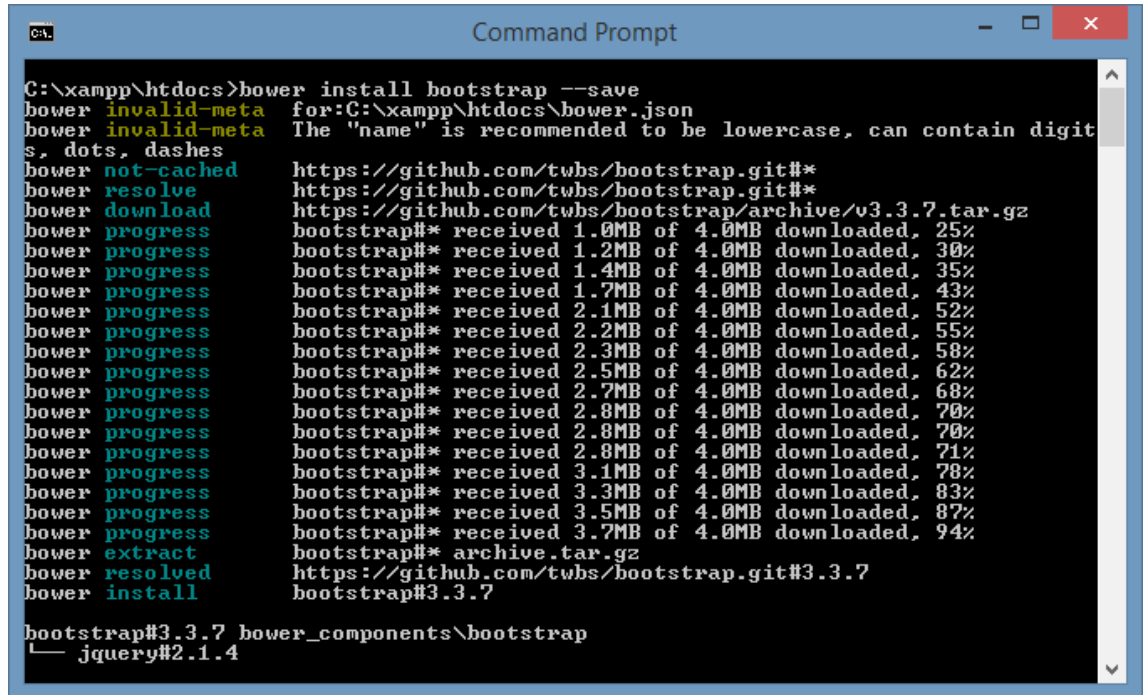
Bir paket indirilirken bu paketin doğrudan bower.json dosyasına yazılabileceğinden bahsettik. Bu işlem de -save komutu ile gerçekleştirilir. -save komutunun kullanımı aşağıdaki gibidir.

```
bower install package_name --save
```

Yukarıdaki komutumuzu anlamak için basit bir örnek yapalım. Mesela daha önce oluşturduğumuz bower.json dosyasına bir de Bootstrap'i ekleyelim.

```
bower install bootstrap --save
```

Şimdi yukarıdaki komutu ekleyelim.



```
C:\xampp\htdocs>bower install bootstrap --save
bower invalid-meta for:C:\xampp\htdocs\bower.json
bower invalid-meta The "name" is recommended to be lowercase, can contain digit
s, dots, dashes
bower not-cached https://github.com/twbs/bootstrap.git#*
bower resolve https://github.com/twbs/bootstrap.git#*
bower download https://github.com/twbs/bootstrap/archive/v3.3.7.tar.gz
bower progress bootstrap#* received 1.0MB of 4.0MB downloaded, 25%
bower progress bootstrap#* received 1.2MB of 4.0MB downloaded, 30%
bower progress bootstrap#* received 1.4MB of 4.0MB downloaded, 35%
bower progress bootstrap#* received 1.7MB of 4.0MB downloaded, 43%
bower progress bootstrap#* received 2.1MB of 4.0MB downloaded, 52%
bower progress bootstrap#* received 2.2MB of 4.0MB downloaded, 55%
bower progress bootstrap#* received 2.3MB of 4.0MB downloaded, 58%
bower progress bootstrap#* received 2.5MB of 4.0MB downloaded, 62%
bower progress bootstrap#* received 2.7MB of 4.0MB downloaded, 68%
bower progress bootstrap#* received 2.8MB of 4.0MB downloaded, 70%
bower progress bootstrap#* received 2.8MB of 4.0MB downloaded, 70%
bower progress bootstrap#* received 2.8MB of 4.0MB downloaded, 71%
bower progress bootstrap#* received 3.1MB of 4.0MB downloaded, 78%
bower progress bootstrap#* received 3.3MB of 4.0MB downloaded, 83%
bower progress bootstrap#* received 3.5MB of 4.0MB downloaded, 87%
bower progress bootstrap#* received 3.7MB of 4.0MB downloaded, 94%
bower extract bootstrap#* archive.tar.gz
bower resolved https://github.com/twbs/bootstrap.git#3.3.7
bower install bootstrap#3.3.7
bootstrap#3.3.7 bower_components\bootstrap
└─ jquery#2.1.4
```

Görüldüğü gibi projemize Bootstrap'i de ekledik. Bakın burada -name alanında bize uyarı verdi. Bu konudan daha önce bahsetmiştik. -name alanı küçük harflerle yazılmalıdır. Biz büyük harfler de kullanmıştık. Lakin herhangi büyük bir hata meydana gelmedi. Çünkü biz projemizi Bower üzerinden yayınlamayacağız. O halde sorun yok!

Şimdi bower.json dosyasına bir bakalım. Son hali nasıl olmuş?



```
1 {
2   "name": "AProject",
3   "description": "This is test project",
4   "main": "index.php",
5   "keywords": [
6     "php",
7     "mvc",
8     "log"
9   ],
10  "authors": [
11    "emrecaoztas <emrecaoztas@outlook.com>"
12  ],
13  "license": "GPL",
14  "homepage": "http://emrecaoztas.com",
15  "private": true,
16  "ignore": [
17    "**/*.*",
18    "node_modules",
19    "bower_components",
20    "test",
21    "tests"
22  ],
23  "dependencies": {
24    "uikit": "^2.27.2",
25    "bootstrap": "^3.3.7"
26  }
27 }
```

Yukarıdaki ekran alıntımızdan da görüldüğü üzere; Bootstrap, bower.json dosyasına eklendi. İşte her şey bu kadar basit, dostlarım!

-save komutu yerine daha kısa hali olan -S komutunu da kullanabilirsiniz. Yani:

```
bower install bootstrap -S
```

şeklinde de kullanabilirsiniz.

4.3 Paketleri Yükleme

Paketlerin nasıl yüklendiğini biliyoruz. Bu işlemi -install komutuyla yapıyoruz. Yani:

```
bower install package_name
```

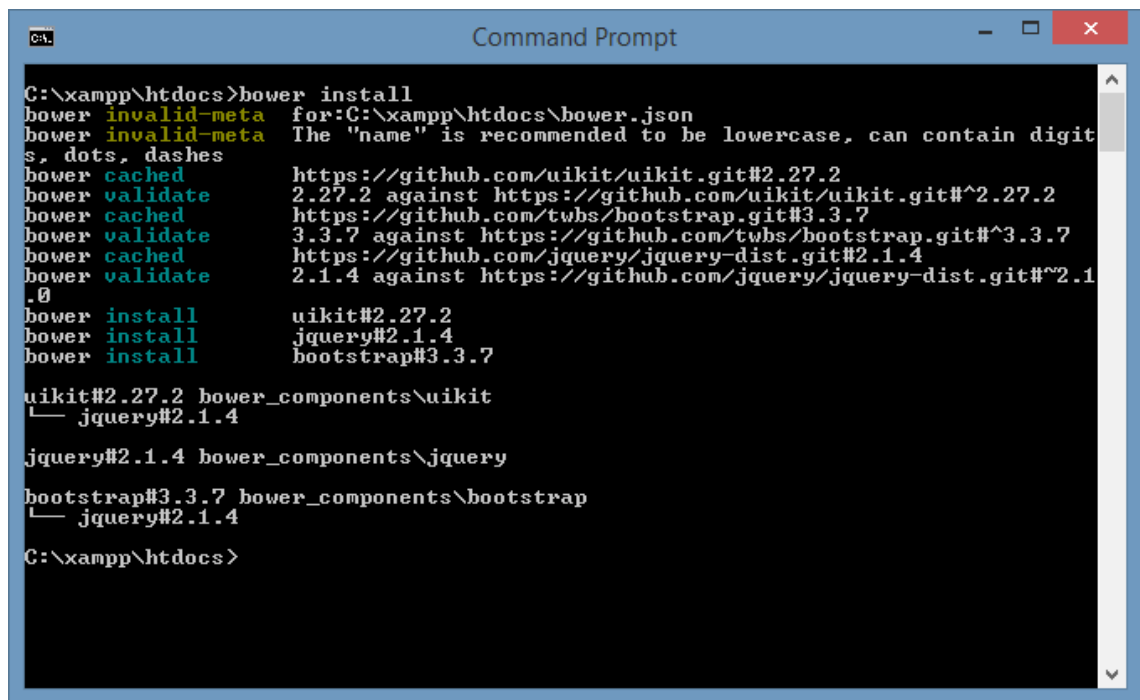
Şeklinde, istediğimiz herhangi bir paket veya paketleri yükleyebiliyoruz. Bunları biliyoruz lakin benim değinmek istediğim bu değil. Öyle ya bower.json dosyasını

boşuna oluşturmadık! Elimizde bower.json dosyası varsa; sadece -install komutunu vermemiz yeterli.

```
bower install
```

Yukarıda verdiğimiz komut ile bower.json dosyasındaki paketler yüklenir. Yani bower.json dosyasını elle oluşturmuş iseniz; bu yolla istediğiniz paketleri kolayca yükleyebilirsiniz.

Örneğin; daha önce üzerinde çalıştığımız “bower_components” dizinini silelim ve elimizde yalnızca bower.json dosyası kalsın. Şimdi komutumuzu yazalım ve Bower bizim için gidip gereken paketleri yüklesin.



```
C:\xampp\htdocs>bower install
bower invalid-meta for:C:\xampp\htdocs\bower.json
bower invalid-meta The "name" is recommended to be lowercase, can contain digit
s, dots, dashes
bower cached      https://github.com/uikit/uikit.git#2.27.2
bower validate    2.27.2 against https://github.com/uikit/uikit.git#^2.27.2
bower cached      https://github.com/twbs/bootstrap.git#3.3.7
bower validate    3.3.7 against https://github.com/twbs/bootstrap.git#^3.3.7
bower cached      https://github.com/jquery/jquery-dist.git#2.1.4
bower validate    2.1.4 against https://github.com/jquery/jquery-dist.git#~2.1
.0
bower install     uikit#2.27.2
bower install     jquery#2.1.4
bower install     bootstrap#3.3.7

uikit#2.27.2 bower_components\uikit
└─ jquery#2.1.4

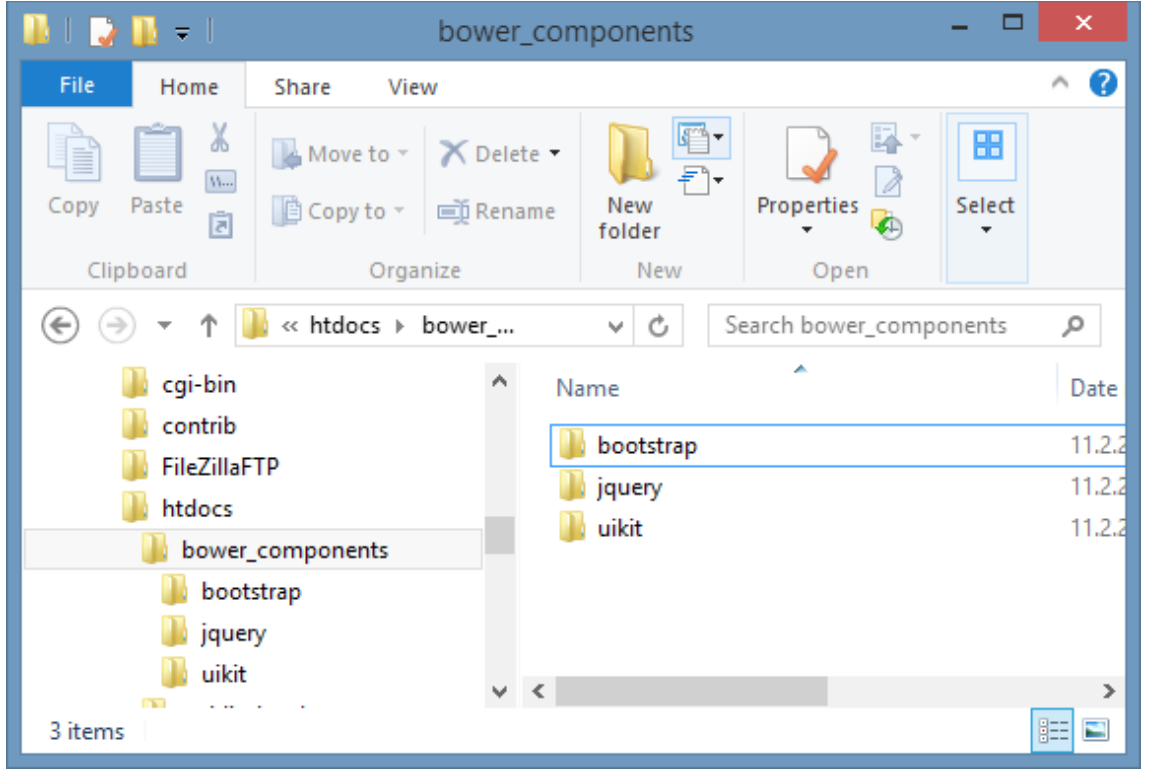
jquery#2.1.4 bower_components\jquery

bootstrap#3.3.7 bower_components\bootstrap
└─ jquery#2.1.4

C:\xampp\htdocs>
```

Görüldüğü gibi; bower.json dosyası içerisindeki “dependencies” yani bağımlılıklar kolayca yüklendi. Bakın! Yine bana -name alanında uyarı veriyor. -name alanında küçük harfler kullanmanız önemli. Yoksa siz de benim gibi her seferinde Bower tarafından uyarılırsınız!

Son olarak bir de dizinimize bakalım.



Evet, gördüğünüz gibi istediğimiz her şey yüklenmiş durumda. Bize sadece kullanmak kalıyor!

BÖLÜM 5: Proje Yapısı Oluşturma

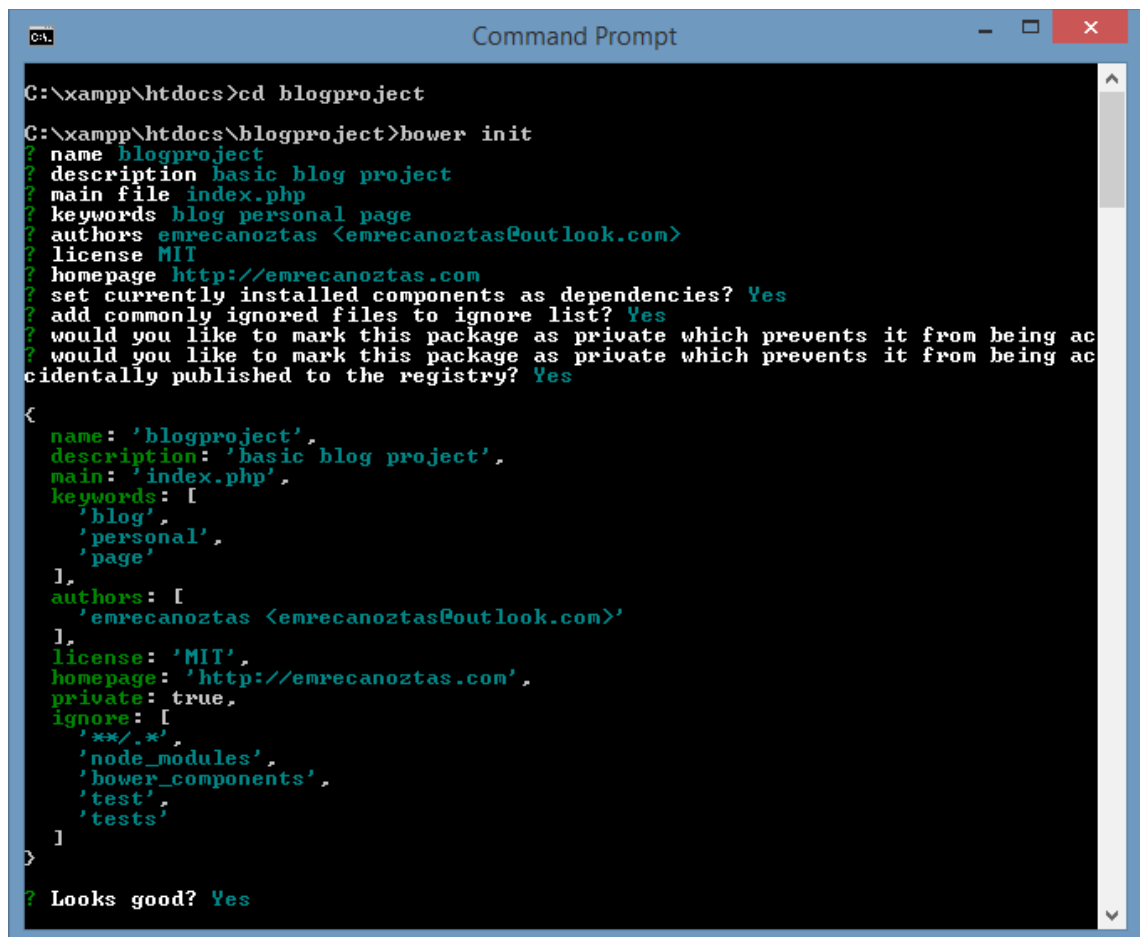
Artık Bower kullanımı hakkında bilgimiz var lakin bu bölümde bir projeye nasıl başlanır, Bower bu işin neresinde bunlara değinelim. Anlatacaklarımız aslında bildiğiniz klasik şeyler lakin bunları pratiğe dökelim.

Öncelikle kendimize bir proje dosyası oluşturalım. Örneğin; ben ./htdocs dizininde bir dosya açacağım ve bu dosyaya “blogproject” diyeceğim. Şimdi bu dosyamızın içerisine bir .bowerrc dosyası ekleyelim. Bu dosyanın içeriğini de aşağıdaki gibi belirleyelim.

```
{  
  "directory": "assets"  
}
```

Yani; Bower, paketleri “assets” dizinine konumlandırılacak.

Projemize bir de bower.json dosyası ekleyelim.



```
C:\>cd \xampp\htdocs>cd blogproject  
C:\xampp\htdocs\blogproject>bower init  
? name blogproject  
? description basic blog project  
? main file index.php  
? keywords blog personal page  
? authors emrecaoztas <emrecaoztas@outlook.com>  
? license MIT  
? homepage http://emrecaoztas.com  
? set currently installed components as dependencies? Yes  
? add commonly ignored files to ignore list? Yes  
? would you like to mark this package as private which prevents it from being ac  
? would you like to mark this package as private which prevents it from being ac  
? cidentially published to the registry? Yes  
<  
  name: 'blogproject',  
  description: 'basic blog project',  
  main: 'index.php',  
  keywords: [  
    'blog',  
    'personal',  
    'page'  
  ],  
  authors: [  
    'emrecaoztas <emrecaoztas@outlook.com>'  
  ],  
  license: 'MIT',  
  homepage: 'http://emrecaoztas.com',  
  private: true,  
  ignore: [  
    '**/*.js',  
    'node_modules',  
    'bower_components',  
    'test',  
    'tests'  
  ],  
  >  
? Looks good? Yes
```

Her şey tamam. Şimdi sıra geldi projemize ekleyeceğimiz paketlere. Projemizde; basit olması için jQuery ve Bootstrap paketlerini kullanalım. Siz isterseniz başka paketleri de kullanabilirsiniz.

O halde paketlerimizi de indirelim ve bu paketleri bower.json dosyasına da ekleyelim.

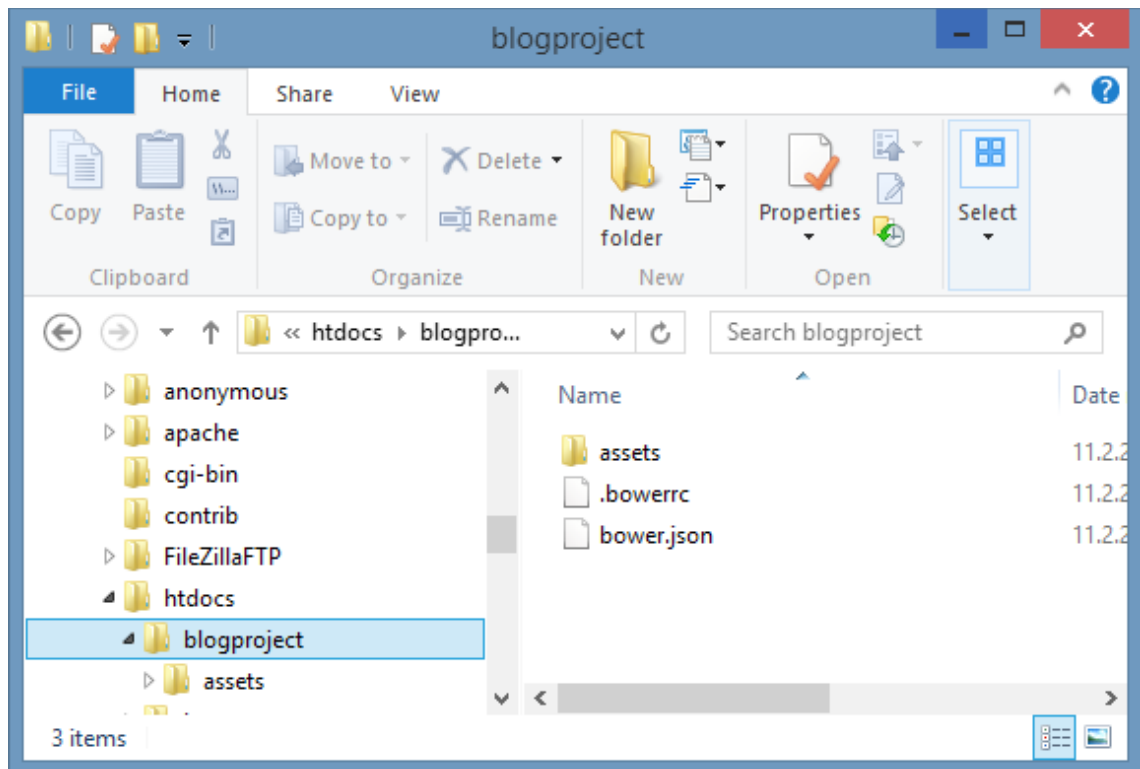


```
C:\xampp\htdocs\blogproject>bower install jquery bootstrap -S
bower cached      https://github.com/jquery/jquery-dist.git#3.1.1
bower validate    3.1.1 against https://github.com/jquery/jquery-dist.git#*
bower cached      https://github.com/twbs/bootstrap.git#3.3.7
bower validate    3.3.7 against https://github.com/twbs/bootstrap.git#*
bower install     jquery#3.1.1
bower install     bootstrap#3.3.7

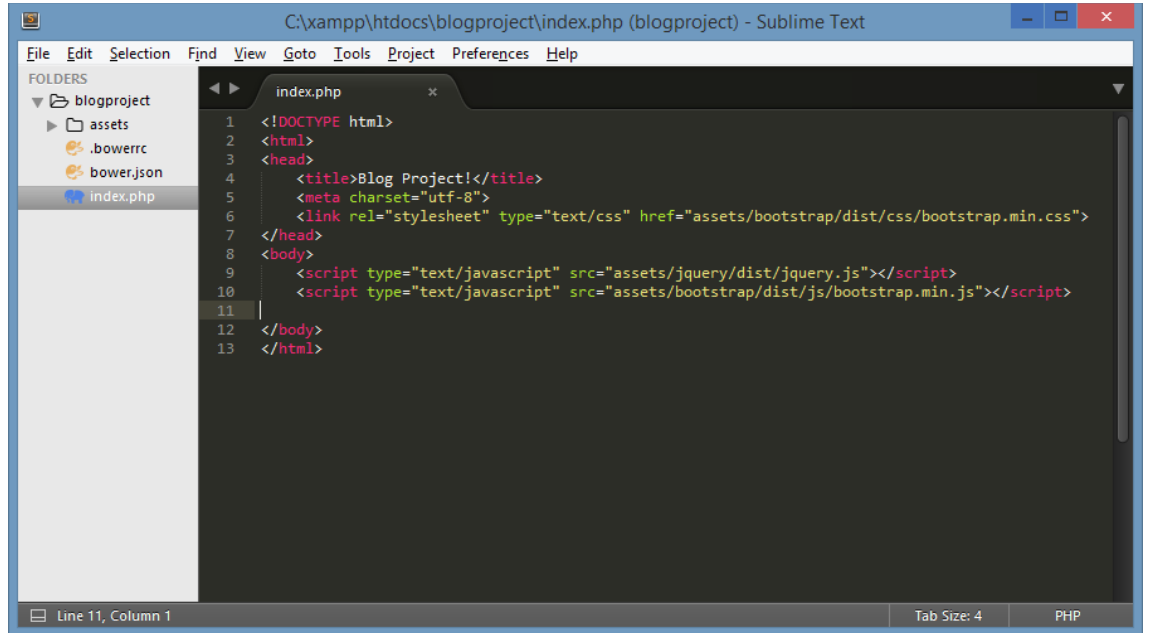
jquery#3.1.1 assets\jquery
bootstrap#3.3.7 assets\bootstrap
└─ jquery#3.1.1

C:\xampp\htdocs\blogproject>
```

Paketlerimizi de indirdik ve inen paketleri bower.json dosyasına ekledik. Projemizin son haline bir bakalım.



Görüldüğü gibi proje bağımlılıkları ve dizinimiz hazır. Artık projeye başlayabiliriz. Son olarak bir de projemize index.php dosyası açmak istiyorum. Daha sonra da paketlerimizi index.php dosyasına ekleyelim.



The screenshot shows the Sublime Text editor with the file 'index.php' open. The file content is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Blog Project!</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" type="text/css" href="assets/bootstrap/dist/css/bootstrap.min.css">
7 </head>
8 <body>
9   <script type="text/javascript" src="assets/jquery/dist/jquery.js"></script>
10  <script type="text/javascript" src="assets/bootstrap/dist/js/bootstrap.min.js"></script>
11
12 </body>
13 </html>
```

The left sidebar shows the project structure with folders 'assets', '.bowerrc', 'bower.json', and 'index.php'.

Görüldüğü gibi artık projemizi kodlamaya başlayabiliriz.



BÖLÜM 6: Paket Yayınlama

Bu bölümde; Bower üzerinden kendi paketlerimizi yayınlamayı göreceğiz. Örneğin; oldukça kullanışlı ve yararlı paketler oluşturmuş olabilirsiniz. Dilerseniz; bu paketinizi Bower'a ekleyim diğer kullanıcılarında kullanımına açabilirsiniz.

Bower üzerinden kendi paketinizi yayınlamak için ilk yapmanız gereken: paketiniz hakkında detaylı bir bower.json dosyası oluşturmanızdır. Bunu da -init komutuyla yapıyorduk hatırlarsanız.

```
$ bower init
```

Detaylı bir bower.json dosyası oluşturmanız önemli. Nasıl oluşturacağınızı ve içeriği hakkında daha önce detaylı olarak konuşmuştuk. Yine de yardım almak isterseniz; aşağıdaki sayfaya bakmanız yeterli olacaktır.

<https://github.com/bower/spec/blob/master/json.md>

Oluşturmuş olduğunuz paketinizde; bazı bağımlılıklar kullanmış olabilirsiniz. Daha bower.json dosyasına indirilen paketlerin nasıl eklendiğinden konuşmuştuk. Bu işlemi de aşağıdaki şekilde yapıyorduk

```
$ bower install package_name --save
```

Veya aynı şekilde aşağıdaki gibi de yapabilirsiniz.

```
$ bower install package_name --save-dev
```

Buraya kadar herhangi bir sıkıntı yoksa; şimdi paketimizi yayınlamaya geçebiliriz.

6.1 Register

Paketlerinizi Bower'a kaydetmeniz oldukça kolay. Bu işlemi -register komutu ile gerçekleştireceğiz. Şimdi aşağıdaki örnek kullanımımıza bakalım.

```
$ bower register package_name <git-endpoint>  
# örneğin:  
$ bower register example git://github.com/user/example.git
```

Yukarıdaki kullanımda; paket ismini ve Github adresini yazmamız yeterli. Paketimizi Bower'a eklenir. Daha sonra isteyen kullanıcılar, paketimizi indirip kullanabilir.

Register işleminde yapmanız veya uymanız gereken belli prosedürler var. Şimdi bu prosedürlere sırasıyla değinelim.

1. bower.json dosyası, düzgün ve kaliteli bir şekilde olmalıdır. Daha önce bahsettiğimiz kurallara uymalıdır.
2. bower.json dosyası, kök dizinde yerelmalıdır.
3. Paketiniz Github üzerinde korumalı olmamalıdır. Herkese açık yani public halde olmalıdır.

Yukarıdaki kurallara uyduğumuz takdirde herhangi bir sorun çıkmadan paketimiz yayınlanır. Böyle herkes kullanabilir.

6.2 Unregister

Bower üzerinden dosyasınızı kaldırmak isteyebilirsiniz. Bu işlemde oldukça basittir.

Bower üzerinden dosyanızı kaldırmak istiyorsanız; ilk olarak -login olmalısınız.

```
$ bower login
```

Yukarıdaki komuttan sonra aşağıdaki işlemler gerçekleşir.

```
? username:  
? password:
```

Buradaki “username” ve “password” alanları, Github hesanıza aittir. Yani Bower Github hesabı üzerinden login olmanızı sağlar.

Paketinizi Bower üzerinden kaldırmak için de aşağıdaki komutu vermeniz yeterlidir.

```
$ bower unregister package_name
```

Yukarıdaki komuttan sonra; paketinizi artık Bower görmemeye başlayacaktır.

Görüldüğü gibi Bower’in hem kullanımı hem de paket yayınlama işlemi oldukça kolaydır.

BÖLÜM 7: Referanslar

Bu bölümde daha önce değinmediğimiz, çeşitli komutları ve komut seçeneklerini anlatmaya çalışacağız. Bu komut veya komut seçenekleri yardımıyla Bower'i daha efektif olarak kullanabilirsiniz.

7.1 Diğer Komutlar

Bu zamana kadar konuşmadığımız komutlar hakkında bu başlık altında konuşalım.

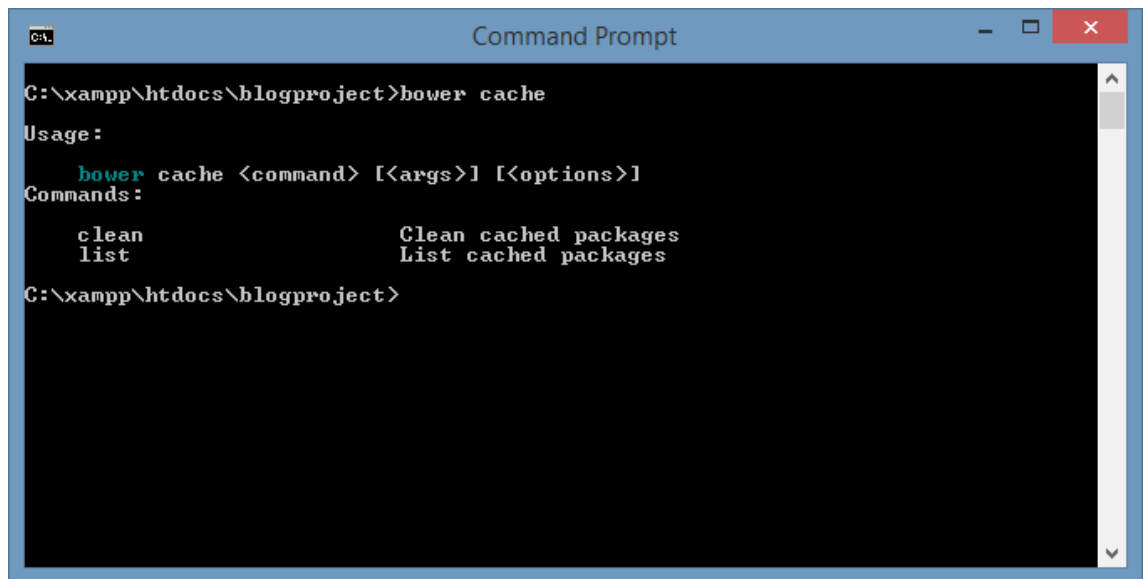
7.1.1 cache

Bower daha önce indirilmiş ya da aranmış olan paketleri cache'ler. Yani önbellege alır. Böylelikle paketler istendiği zaman cache'den daha çabuk okur.

- cache komutunun kullanımı aşağıdaki gibidir.

```
$ bower cache
```

Örneğin; yukarıdaki komutu kullanayım. Bakalım ne sonuç verecek?



```
Command Prompt

C:\xampp\htdocs\blogproject>bower cache

Usage:
  bower cache <command> [<args>] [<options>]

Commands:
  clean          Clean cached packages
  list           List cached packages

C:\xampp\htdocs\blogproject>
```

Yukarıdaki ekran alıntısında da görüldüğü üzere; bana -cache ile kullanılabilecek 2 farklı komut olduğunu söylüyor. Yani yalnız başına kullanılamaz!

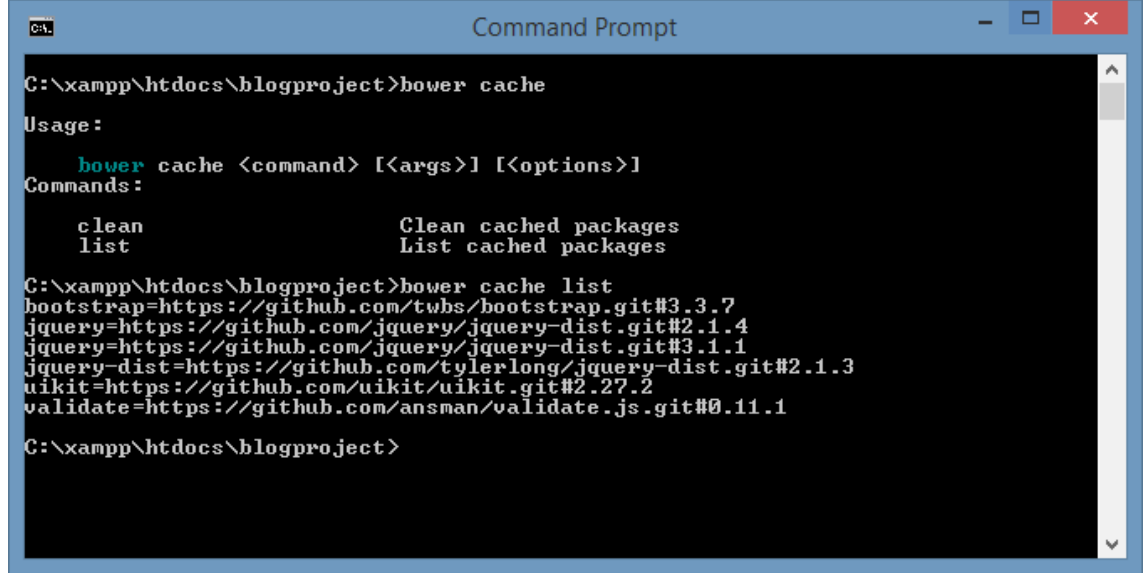
-cache clean komutu, cache'i siler.

-cache list komutu, daha önce indirilmiş paketleri listeler.

Önce list'i kullanalım. Yani:

```
$ bower cache list
```

Yukarıdaki komutumuzun çıktısı aşağıdaki gibi olacaktır. En azından bana göre!



```
C:\xampp\htdocs\blogproject>bower cache list

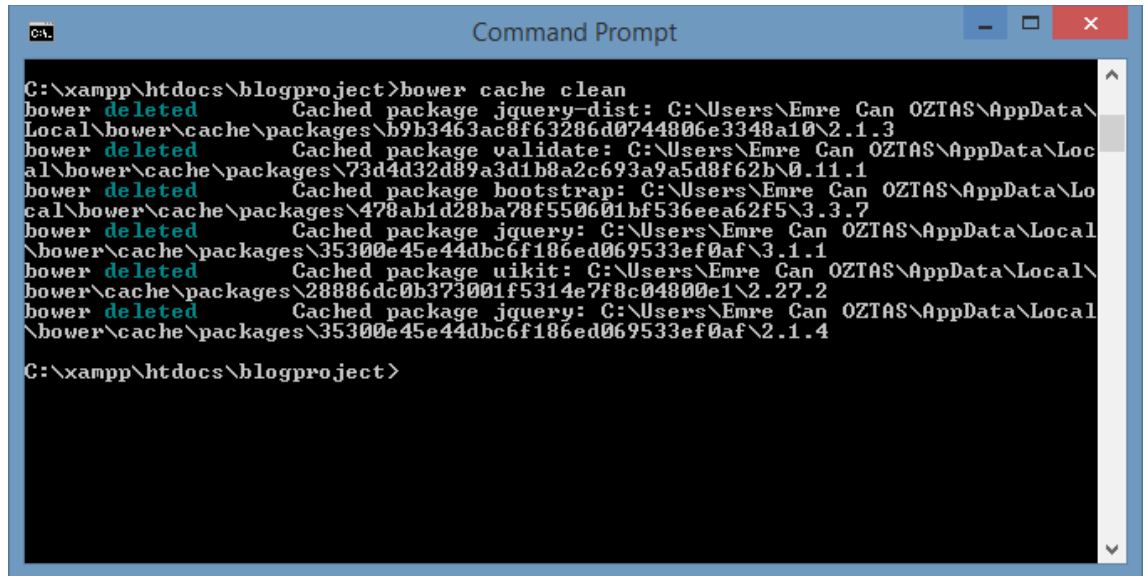
Usage:
  bower cache <command> [<args>] [<options>]
Commands:
  clean      Clean cached packages
  list       List cached packages

C:\xampp\htdocs\blogproject>bower cache list
bootstrap=https://github.com/twbs/bootstrap.git#3.3.7
jquery=https://github.com/jquery/jquery-dist.git#2.1.4
jquery=https://github.com/jquery/jquery-dist.git#3.1.1
jquery-dist=https://github.com/tylerlong/jquery-dist.git#2.1.3
uikit=https://github.com/uikit/uikit.git#2.27.2
validate=https://github.com/ansman/validate.js.git#0.11.1

C:\xampp\htdocs\blogproject>
```

Görüldüğü gibi daha önce araştırdığım veya indirdiğim paketleri göstermekte.

Şimdi de clean komutu ile cache'i temizleyelim.



```
C:\xampp\htdocs\blogproject>bower cache clean
bower deleted    Cached package jquery-dist: C:\Users\Emre Can OZTAS\AppData\Local\
Local\bower\cache\packages\b9b3463ac8f63286d0744806e3348a10\2.1.3
bower deleted    Cached package validate: C:\Users\Emre Can OZTAS\AppData\Local\
al\bower\cache\packages\73d4d32d89a3d1b8a2c693a9a5d8f62b\0.11.1
bower deleted    Cached package bootstrap: C:\Users\Emre Can OZTAS\AppData\Local\
cal\bower\cache\packages\478ab1d28ba78f550601bf536eea62f5\3.3.7
bower deleted    Cached package jquery: C:\Users\Emre Can OZTAS\AppData\Local\
\bower\cache\packages\35300e45e44dbc6f186ed069533ef0af\3.1.1
bower deleted    Cached package uikit: C:\Users\Emre Can OZTAS\AppData\Local\
bower\cache\packages\28886dc0b373001f5314e7f8c04800e1\2.27.2
bower deleted    Cached package jquery: C:\Users\Emre Can OZTAS\AppData\Local\
\bower\cache\packages\35300e45e44dbc6f186ed069533ef0af\2.1.4

C:\xampp\htdocs\blogproject>
```

cache'deki herşeyi silindi. Tekrar list komutunu kullanırsak herhangi bir şey çıkmayacaktır.

cache'deki herhangi bir paketi de silebilirsiniz. Yani hepsini silmek zorunda değilsiniz!

```
$ bower cache clean package_name
```

7.1.2 home

-home komutu, favori tarayıcınızda bower.json dosyasındaki “homepage” alanında yazan web adresini açar.

-home komutunun kullanımı aşağıdaki gibidir.

```
$ bower home
```

Örneğin; bizim bir önceki bölüm oluşturduğumuz “blogproject” dizini içerisinde bu komutu verirsem, aşağıdaki sayfa açılacaktır.



Açılan sayfa: <http://emrecaoztas.com> adresidir.

Bower, bower.json dosyasındaki adres yerine istediğiniz herhangi bir paketin de sayfasını açabilirsiniz.

```
$ bower home package_name  
# veya  
$ bower home package#version
```

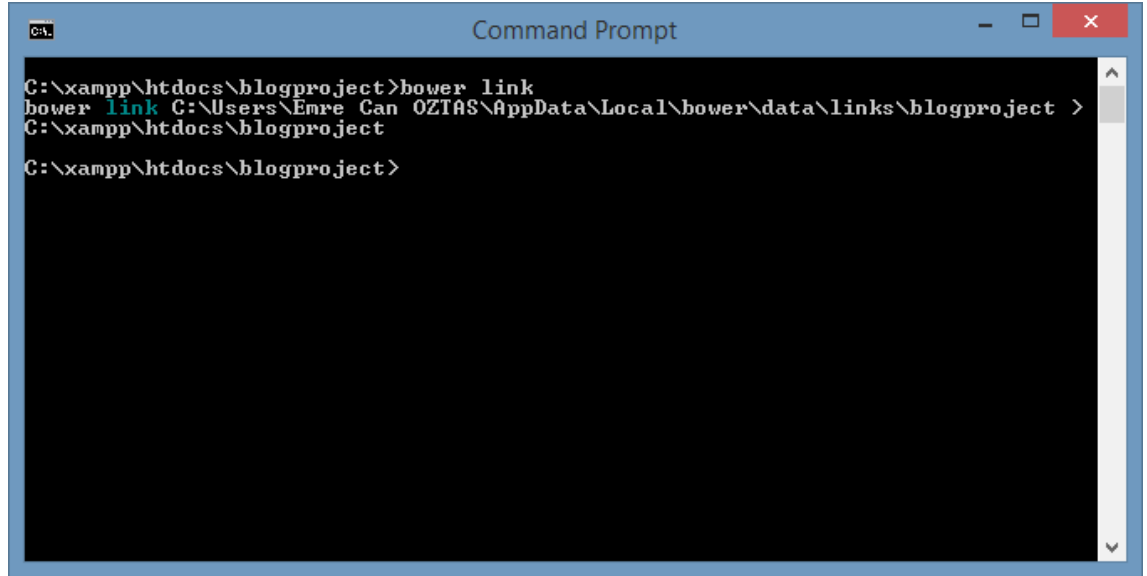
7.1.3 link

-link komutu, kullanıcıların paketlerini test etmelerini sağlar.

-link komutunun kullanımı aşağıdaki gibidir.

```
$ bower link
```

Örneğin; komutumu “blogproject” içerisinde kullanalım. Bakalım ne sonuç verecek?



```
Command Prompt
C:\xampp\htdocs\blogproject>bower link
bower link C:\Users\Emre Can OZTAS\AppData\Local\bowe\data\links\blogproject >
C:\xampp\htdocs\blogproject
C:\xampp\htdocs\blogproject>
```

Yukarıdaki ekran alıntısında da görüldüğü gibi “blogproject” hakkında bize istenilen bilgileri ilettiler.

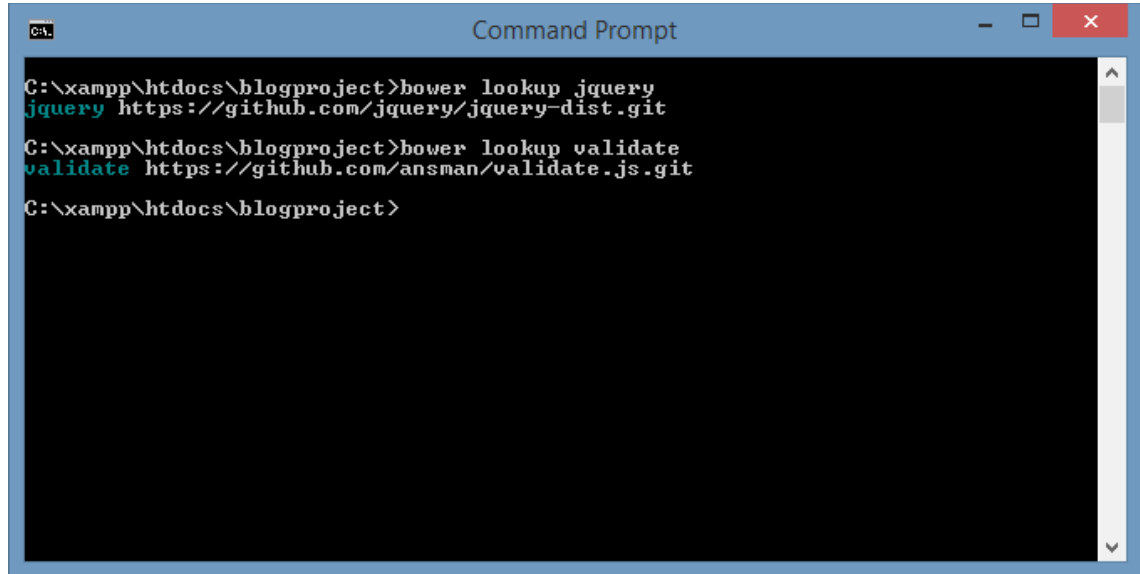
7.1.4 lookup

-lookup komutu, belirtilen paketin URL’ini verir.

-lookup komutunun kullanımı aşağıdaki gibidir.

```
$ bower lookup package_name
```

Komutumuzun kullanımı.



```
C:\xampp\htdocs\blogproject>bower lookup jquery
jquery https://github.com/jquery/jquery-dist.git
C:\xampp\htdocs\blogproject>bower lookup validate
validate https://github.com/ansman/validate.js.git
C:\xampp\htdocs\blogproject>
```

Görüldüğü gibi belirtilen paketlerin adreslerini bize iletti.

7.1.5 prune

-prune, yerel olmayan paketleri kaldırır.

-prune komutunun kullanımı aşağıdaki gibidir.

```
$ bower prune
```

7.2 Seçenekler

Komutlarla birlikte kullanılabilecek çeşitli komut seçeneklerine bir gözatalım.

7.2.1 force

-force kullanılan komutu belli işlemlere zorlar.

-force seçeneğinin kullanımı aşağıdaki gibidir.

```
-f, --force
```

Örneğin; aşağıdaki gibi -install komutu ile kullandığımızda, paketlerin yeniden yüklenmesini sağlar.

```
$ bower install -f
```

7.2.2 json

Çıktıların JSON formatında olmasını sağlar.

-json seçeneğinin kullanımı aşağıdaki gibidir.

```
-j, --json
```

7.2.3 loglevel

log seviyesini gösterir. Örneğin bu seviye: error, conflict, warn, action, info ve debug olabilir.

-loglevel kullanımı aşağıdaki gibidir.

```
-l, --loglevel
```

7.2.4 offline

İnternet bağlantısı kullanmadan işlemleri yerine getirilmesini sağlar.

-loglevel seçeneğinin kullanımı aşağıdaki gibidir.

```
-o, --offline
```

7.2.5 quit

Sadece önemli bilgileri gösterir.

-quit seçeneğinin kullanımı aşağıdaki gibidir.

```
-q, --quit
```

7.2.6 silent

Hata mesajları dışında herhangi bir çıktı verilmemesini sağlar.

-silent seçeneğinin kullanımı aşağıdaki gibidir.

```
-s, --silent
```

7.2.7 verbose

Çıktıların daha ayrıntılı olmasını sağlar.

-verbose seçeneğinin kullanımı aşağıdaki gibidir.

```
-v, --verbose
```

7.2.8 allow-root

root olarakta komutların işletilmesini sağlar. Normalde Bower root izni olmadan da çalışabilir.

-allow-root seçeneğinin kullanımı aşağıdaki gibidir.


```
--allow-root
```

KAYNAKLAR

- [1]. <https://bower.io/>
- [2]. <http://sixrevisions.com/tutorials/bower/>
- [3]. <http://blog.teamtreehouse.com/getting-started-bower>
- [4]. <https://www.codementor.io/bower/tutorial/beginner-tutorial-getting-started-bower-package-manager>
- [5]. <http://andy-carter.com/blog/a-beginners-guide-to-package-manager-bower-and-using-gulp-to-manage-components>
- [6]. <https://www.digitalocean.com/community/tutorials/how-to-manage-front-end-javascript-and-css-dependencies-with-bower-on-ubuntu-14-04>