

Video 1.6

Setting Up the Environment





In this Video, we are going to take a look at...

- Creating an Express.js application
- Generating Twitter API keys
- Installing MongoDB



1x



Browse Q&A

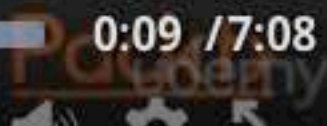


Add Bookmark

Continue >



0:09 / 7:08



Express.js

- Express.js is a web framework written in Node.js

Express.js

- Express.js is a web framework written in Node.js
- It helps us to:
 - Accept HTTP requests
 - Read the data in HTTP requests
 - Create HTTP responses
 - Send HTTP responses

Creating an Express.js Application

- Create a Node.js application
- Install the Express.js module
- Code an Express.js server

Salehs-Macintosh:Desktop saleh\$ mkdir twitter-notes

Salehs-Macintosh:Desktop saleh\$ cd twitter-notes/

Salehs-Macintosh:twitter-notes saleh\$ npm init

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.

name: (twitter-notes) █

Go to Dashboard

1:11 / 7:08



1x



Browse Q&A



Add Bookmark

Continue >




```
[name: (twitter-notes)
[version: (1.0.0)
[description: An application for leaving notes on Twitter friends and followers.
[entry point: (index.js)
[test command:
[git repository:
[keywords:
[author: Saleh Hamadeh
[license: (ISC)
About to write to /Users/saleh/Desktop/twitter-notes/package.json:
```

```
{
  "name": "twitter-notes",
  "version": "1.0.0",
  "description": "An application for leaving notes on Twitter friends and followers.",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Saleh Hamadeh",
  "license": "ISC"
}
```

Is this ok? (yes) ☐

description: An application for leaving notes on Twitter friends and followers.

entry point: (index.js)

test command:

git repository:

keywords:

author: Saleh Hamadeh

license: (ISC)

About to write to /Users/saleh/Desktop/twitter-notes/package.json:

```
{
  "name": "twitter-notes",
  "version": "1.0.0",
  "description": "An application for leaving notes on Twitter friends and followers.",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Saleh Hamadeh",
  "license": "ISC"
}
```

Is this ok? (yes)

Saleh's Macintosh:twitter-notes saleh\$ ls


```
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "Saleh Hamadeh",
"license": "ISC"
}
```

[Is this ok? (yes)

[Salehs-Macintosh:twitter-notes saleh\$ ls

package.json

[Salehs-Macintosh:twitter-notes saleh\$ cat package.json

```
{
  "name": "twitter-notes",
  "version": "1.0.0",
  "description": "An application for leaving notes on Twitter friends and followers.",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Saleh Hamadeh",
  "license": "ISC"
}
```

Salehs-Macintosh:twitter-notes saleh\$

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1"  
},  
"author": "Saleh Hamadeh",  
"license": "ISC"  
}
```

[Is this ok? (yes)]

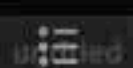
[Salehs-Macintosh:twitter-notes saleh\$ ls]
package.json

[Salehs-Macintosh:twitter-notes saleh\$ cat package.json]

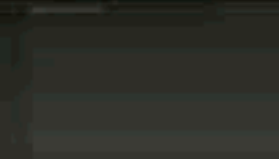
```
{  
  "name": "twitter-notes",  
  "version": "1.0.0",  
  "description": "An application for leaving notes on Twitter friends and followers.",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "Saleh Hamadeh",  
  "license": "ISC"  
}
```

[Salehs-Macintosh:twitter-notes saleh\$ touch index.js]

[Salehs-Macintosh:twitter-notes saleh\$ sublime index.js]



```
1 console.log("Hello world");
```



```
"author": "Saleh Hamadeh",  
"license": "ISC"
```

```
}
```

```
Is this ok? (yes)
```

```
Salehs-Macintosh:twitter-notes saleh$ ls
```

```
package.json
```

```
Salehs-Macintosh:twitter-notes saleh$ cat package.json
```

```
{
```

```
  "name": "twitter-notes",
```

```
  "version": "1.0.0",
```

```
  "description": "An application for leaving notes on Twitter friends and followers.",
```

```
  "main": "index.js",
```

```
  "scripts": {
```

```
    "test": "echo \"Error: no test specified\" && exit 1"
```

```
  },
```

```
  "author": "Saleh Hamadeh",
```

```
  "license": "ISC"
```

```
}
```

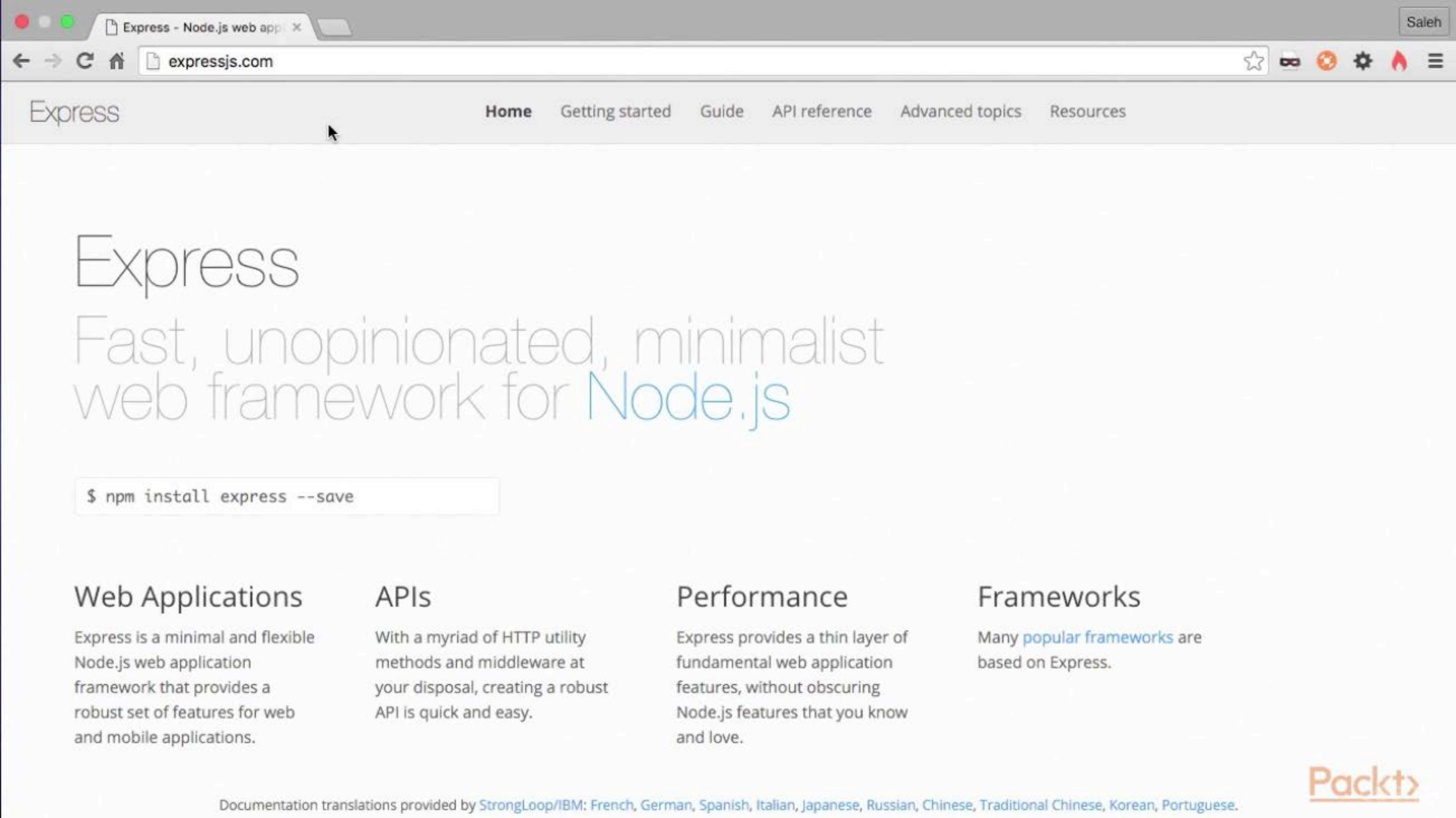
```
Salehs-Macintosh:twitter-notes saleh$ touch index.js
```

```
Salehs-Macintosh:twitter-notes saleh$ sublime index.js
```

```
Salehs-Macintosh:twitter-notes saleh$ node index.js
```

```
Hello world
```

```
Salehs-Macintosh:twitter-notes saleh$
```

Express

Fast, unopinionated, minimalist web framework for Node.js

```
$ npm install express --save
```

Web Applications

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

APIs

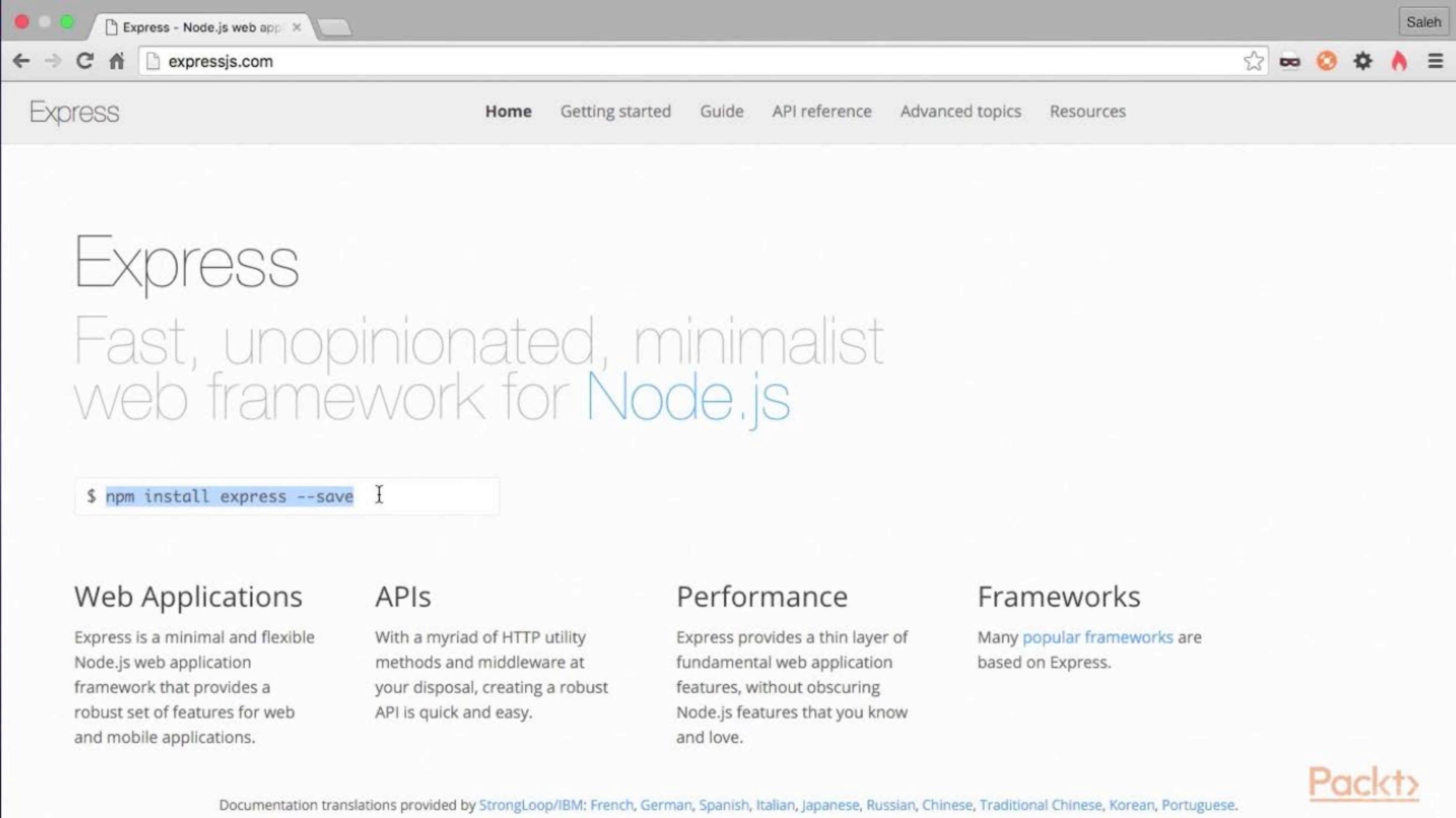
With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Frameworks

Many popular frameworks are based on Express.



Express

Fast, unopinionated, minimalist web framework for Node.js

```
$ npm install express --save
```

Web Applications

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

APIs

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Frameworks

Many [popular frameworks](#) are based on Express.

```
Salehs-Macintosh:twitter-notes saleh$ npm install express --save
```

```
}
```

Setting Up the Environment

var@1.3.1
Section 1, Lecture 6

methods@1.1.2

content-disposition@0.5.1

cookie@0.1.5

etag@1.7.0

fresh@0.3.0

serve-static@1.10.2

range-parser@1.0.3

merge-descriptors@1.0.1

parseurl@1.3.1

path-to-regexp@0.1.7

depd@1.1.0

qs@4.0.0

on-finished@2.3.0 (ee-first@1.1.1)

finalhandler@0.4.1 (unpipe@1.0.0)

debug@2.2.0 (ms@0.7.1)

proxy-addr@1.0.10 (forwarded@0.1.0, ipaddr.js@1.0.5)

accepts@1.2.13 (negotiator@0.5.3, mime-types@2.1.10)

type-is@1.6.12 (media-typer@0.3.0, mime-types@2.1.10)

send@0.13.1 (ms@0.7.1, statuses@1.2.1, destroy@1.0.4, mime@1.3.4, http-errors@1.3.1)

Salehs-Macintosh:twitter-notes saleh\$ ls

index.js node_modules package.json

Salehs-Macintosh:twitter-notes saleh\$ ls node_modules/

express

Salehs-Macintosh:twitter-notes saleh\$


```
└─ debug@2.2.0 (ms@0.7.1)
└─ proxy-addr@1.0.10 (forwarded@0.1.0, ipaddr.js@1.0.5)
└─ accepts@1.2.13 (negotiator@0.5.3, mime-types@2.1.10)
└─ type-is@1.6.12 (media-typer@0.3.0, mime-types@2.1.10)
└─ send@0.13.1 (ms@0.7.1, statuses@1.2.1, destroy@1.0.4, mime@1.3.4, http-errors@1.3.1)
```

```
[Salehs-Macintosh:twitter-notes saleh$ ls
```

```
index.js      node_modules  package.json
```

```
[Salehs-Macintosh:twitter-notes saleh$ ls node_modules/
express
```

```
[Salehs-Macintosh:twitter-notes saleh$ cat package.json
```

```
{
  "name": "twitter-notes",
  "version": "1.0.0",
  "description": "An application for leaving notes on Twitter friends and followers.",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Saleh Hamadeh",
  "license": "ISC",
  "dependencies": {
    "express": "^4.13.4"
  }
}
```

```
[Salehs-Macintosh:twitter-notes saleh$
```

```
1 var express = require('express');
2 var app = express();
3
4 app.get("/", function(req, res) {
5   res.send("ho ho ho");
6 });
7
8 app.listen(8080, function() {
9   console.log("Express server is running on port 8080");
10 });
```


[Salehs-Macintosh:twitter-notes saleh\$ ls

index.js node_modules package.json

[Salehs-Macintosh:twitter-notes saleh\$ ls node_modules/

express

[Salehs-Macintosh:twitter-notes saleh\$ cat package.json

{

 "name": "twitter-notes",

 "version": "1.0.0",

 "description": "An application for leaving notes on Twitter friends and followers.",

 "main": "index.js",

 "scripts": {

 "test": "echo \"Error: no test specified\" && exit 1"

 },

 "author": "Saleh Hamadeh",

 "license": "ISC",

 "dependencies": {

 "express": "^4.13.4"

 }

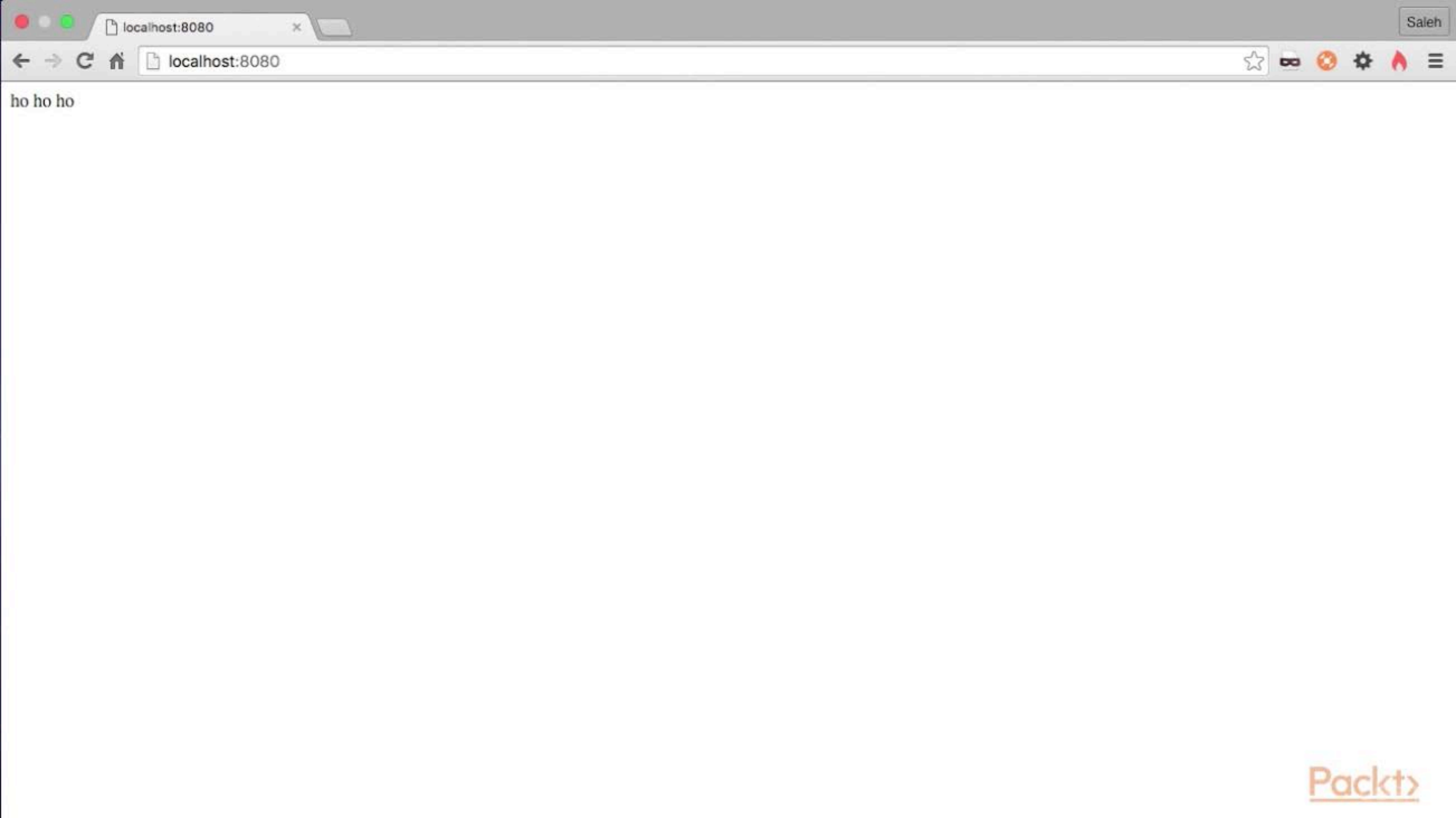
}

[Salehs-Macintosh:twitter-notes saleh\$ node index.js

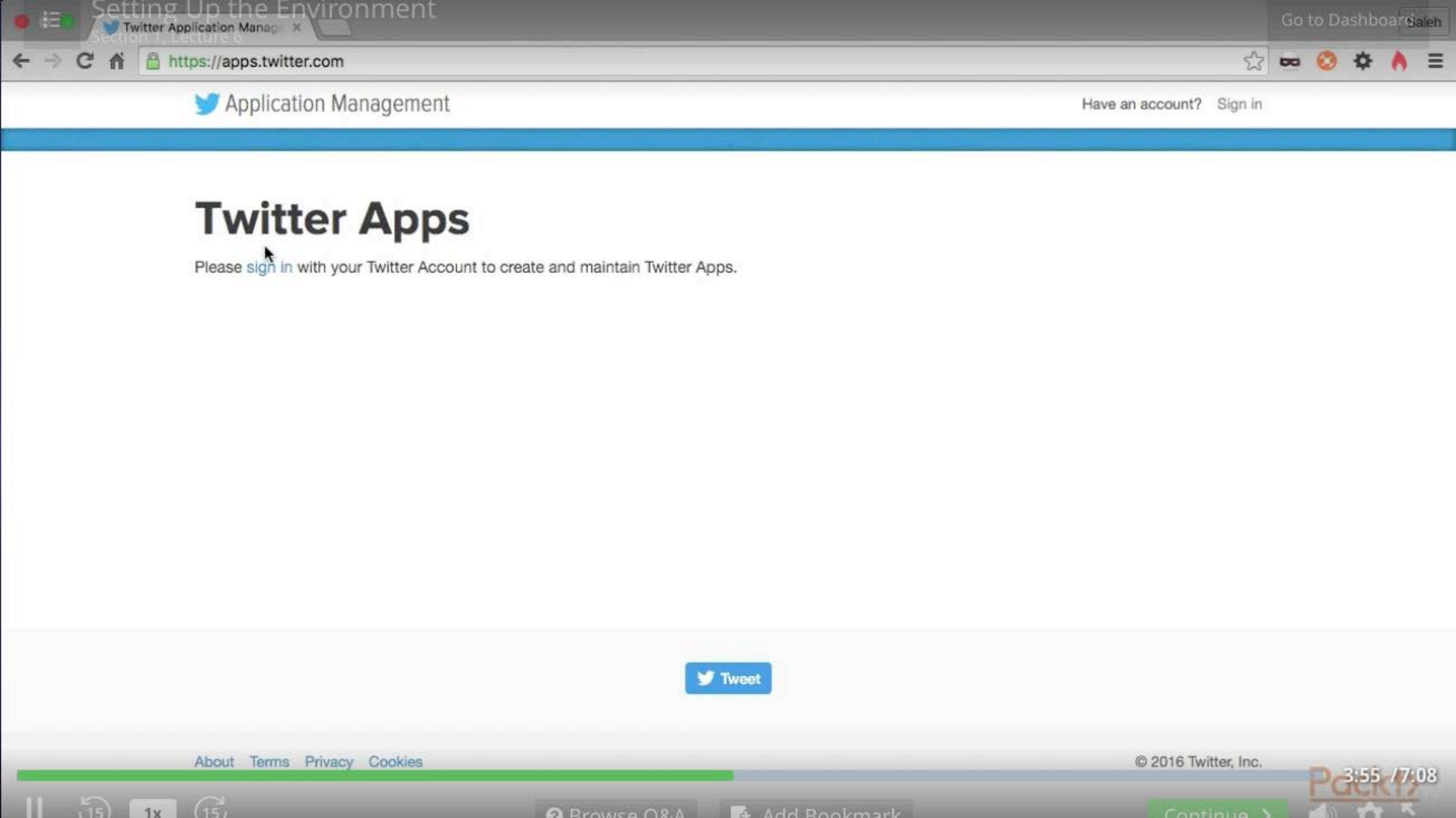
Hello world

[Salehs-Macintosh:twitter-notes saleh\$ node index.js

Express server is running on port 8080



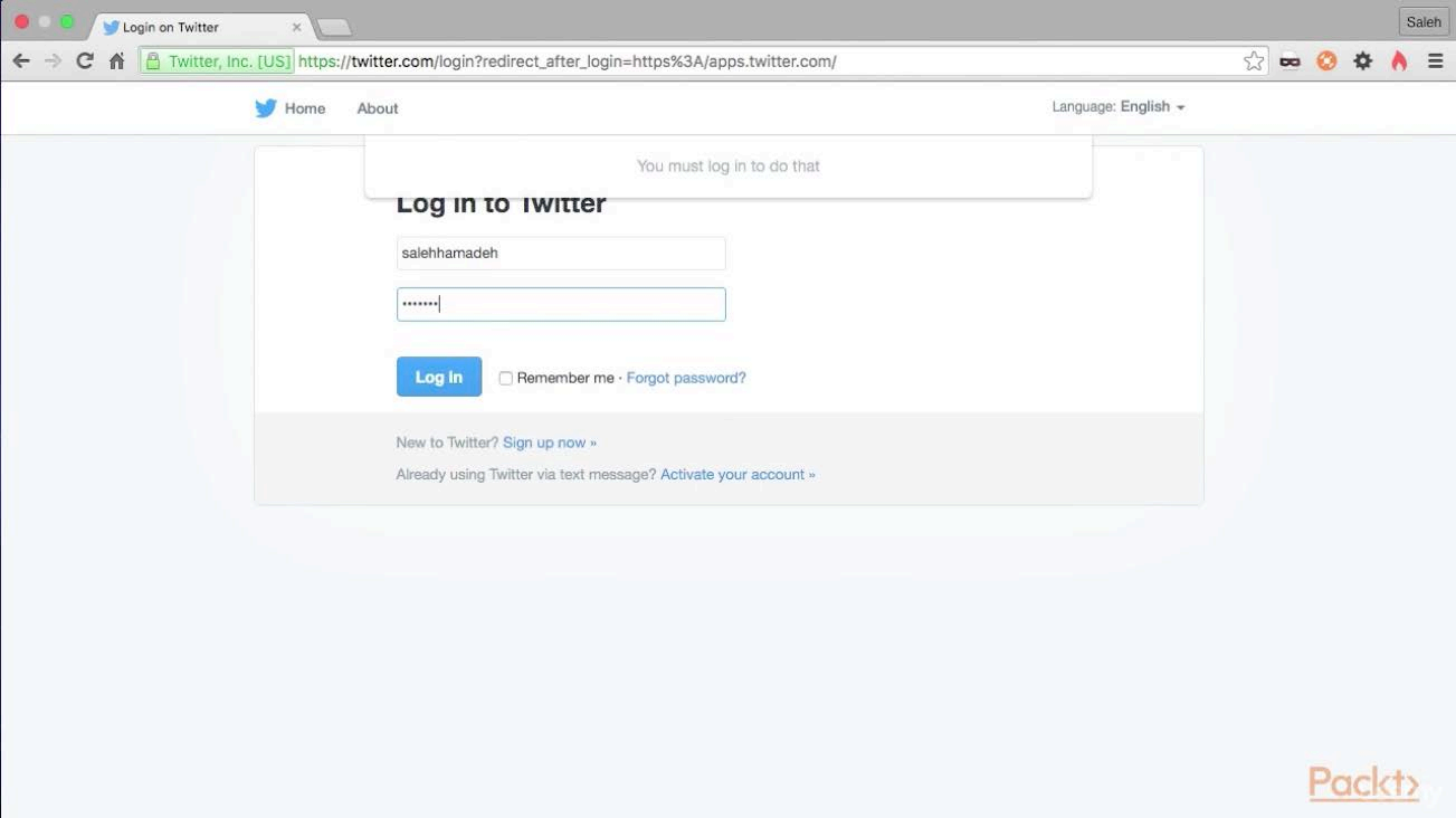
ho ho ho



Twitter Apps

Please [sign in](#) with your Twitter Account to create and maintain Twitter Apps.

 Tweet



Log in to Twitter

salehhamadeh

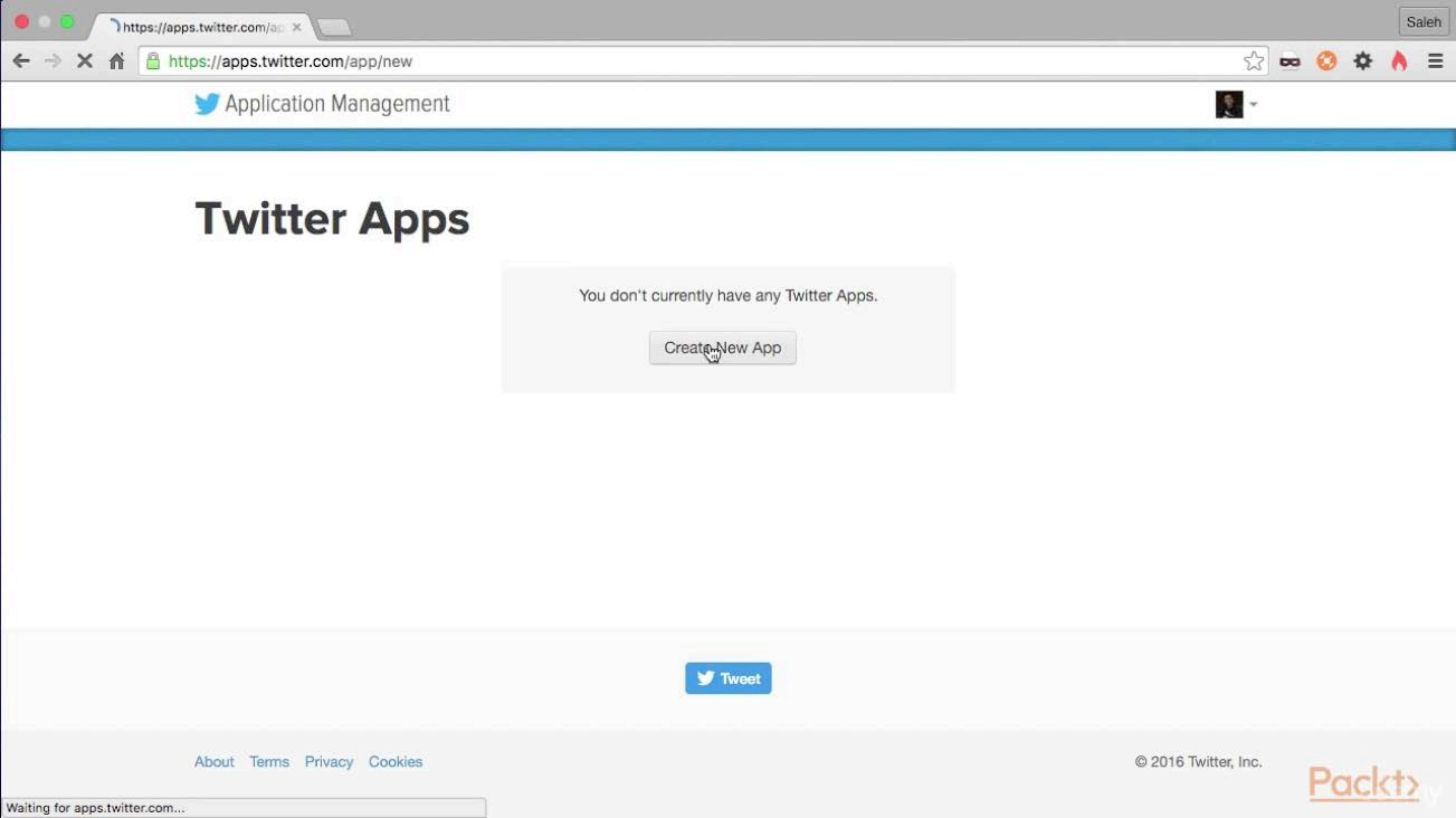
.....

Log in

☐ Remember me · [Forgot password?](#)

New to Twitter? [Sign up now »](#)

Already using Twitter via text message? [Activate your account »](#)

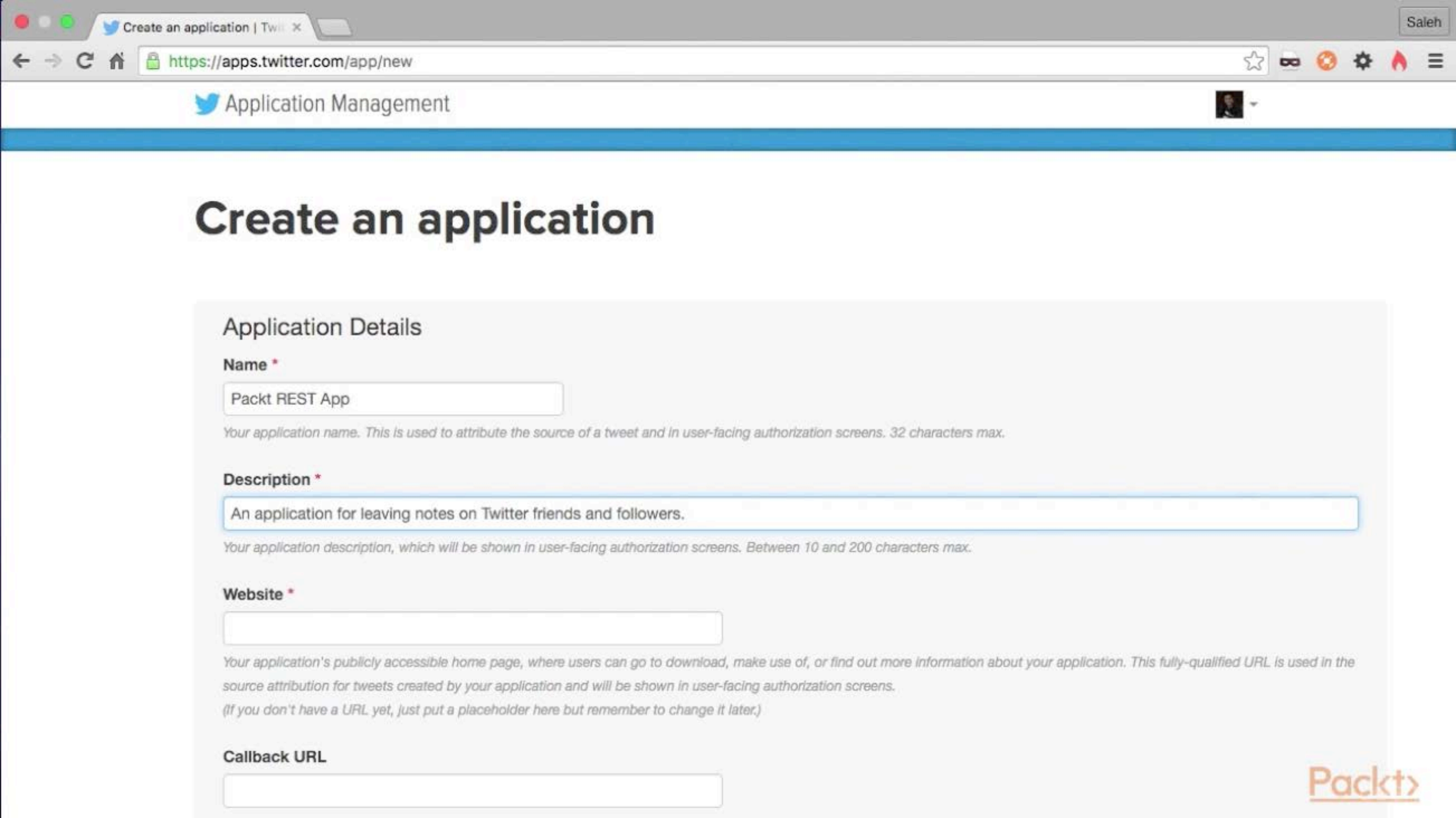


Twitter Apps

You don't currently have any Twitter Apps.

Create New App

Tweet



Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

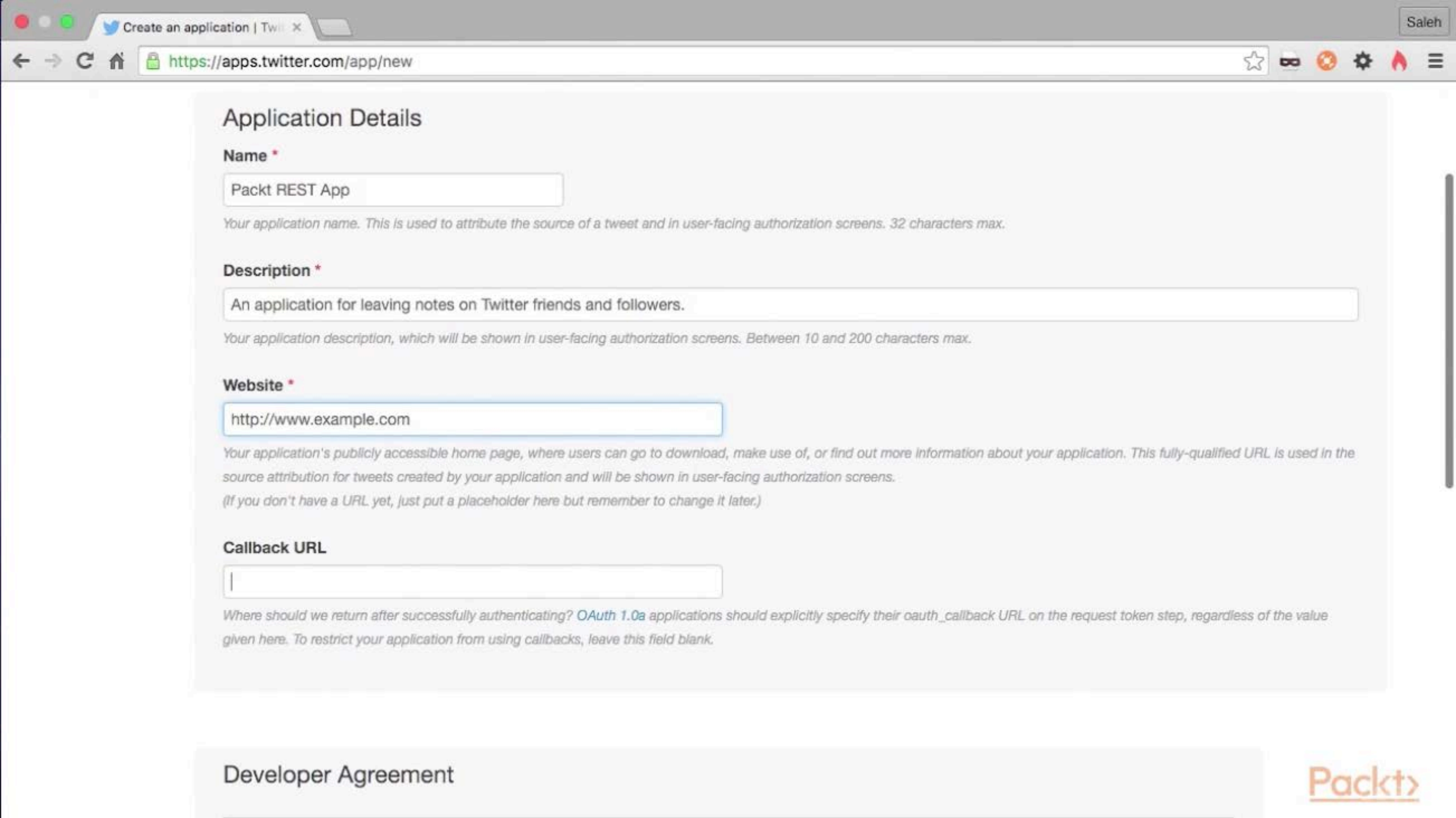
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL



Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

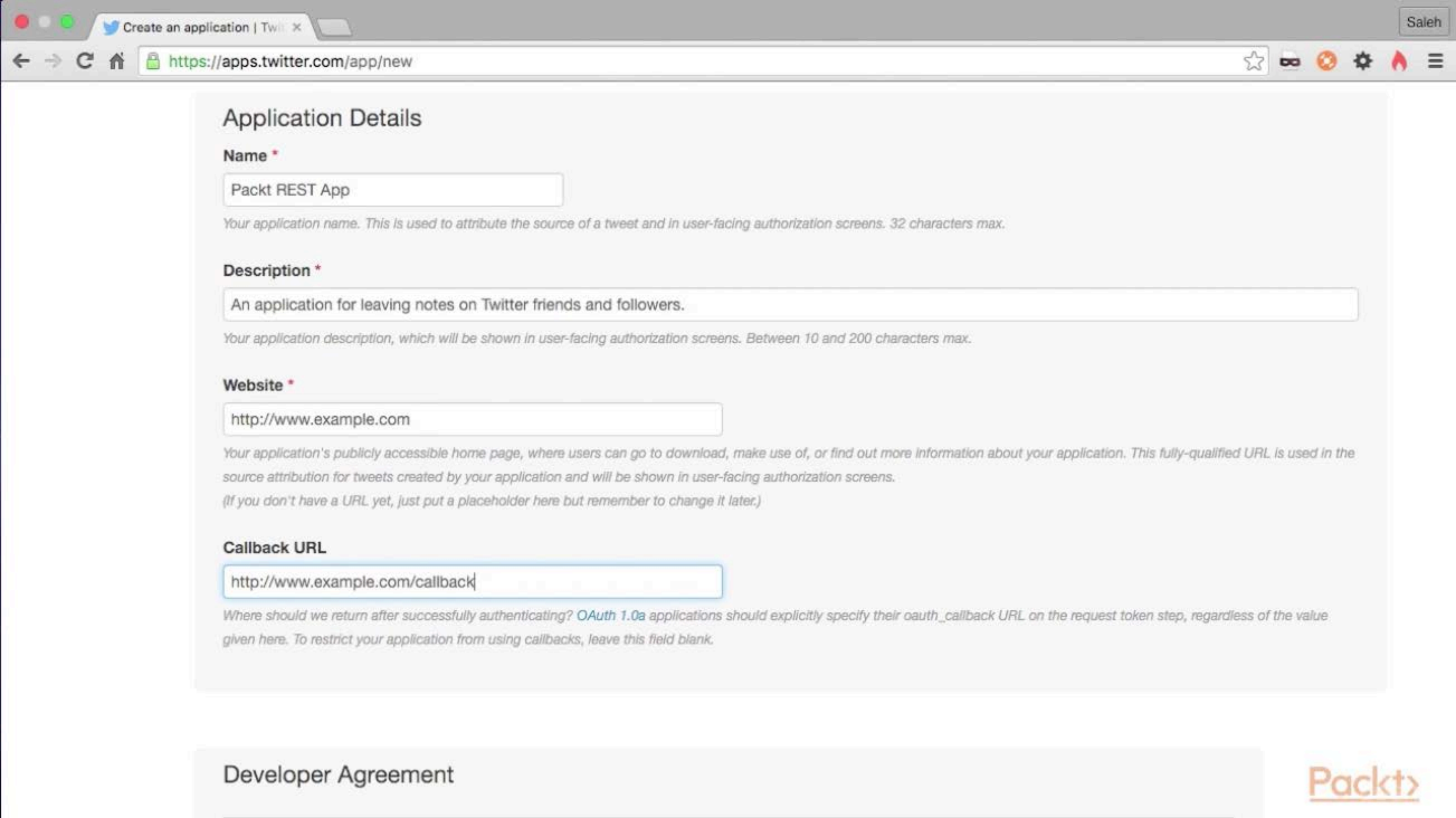
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

Packt



Application Details

Name *

Packt REST App

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

An application for leaving notes on Twitter friends and followers.

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://www.example.com

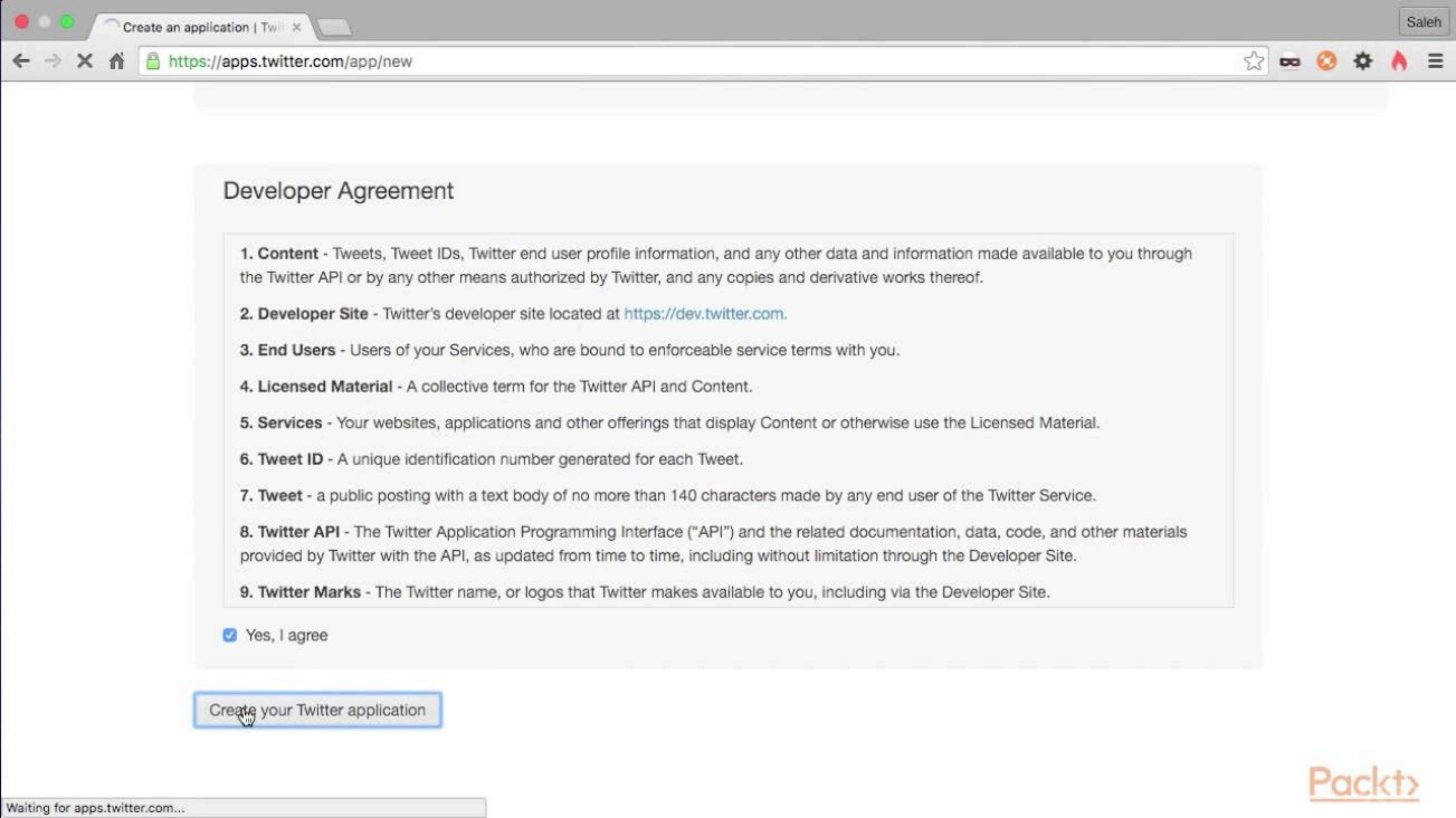
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

http://www.example.com/callback

Where should we return after successfully authenticating? [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

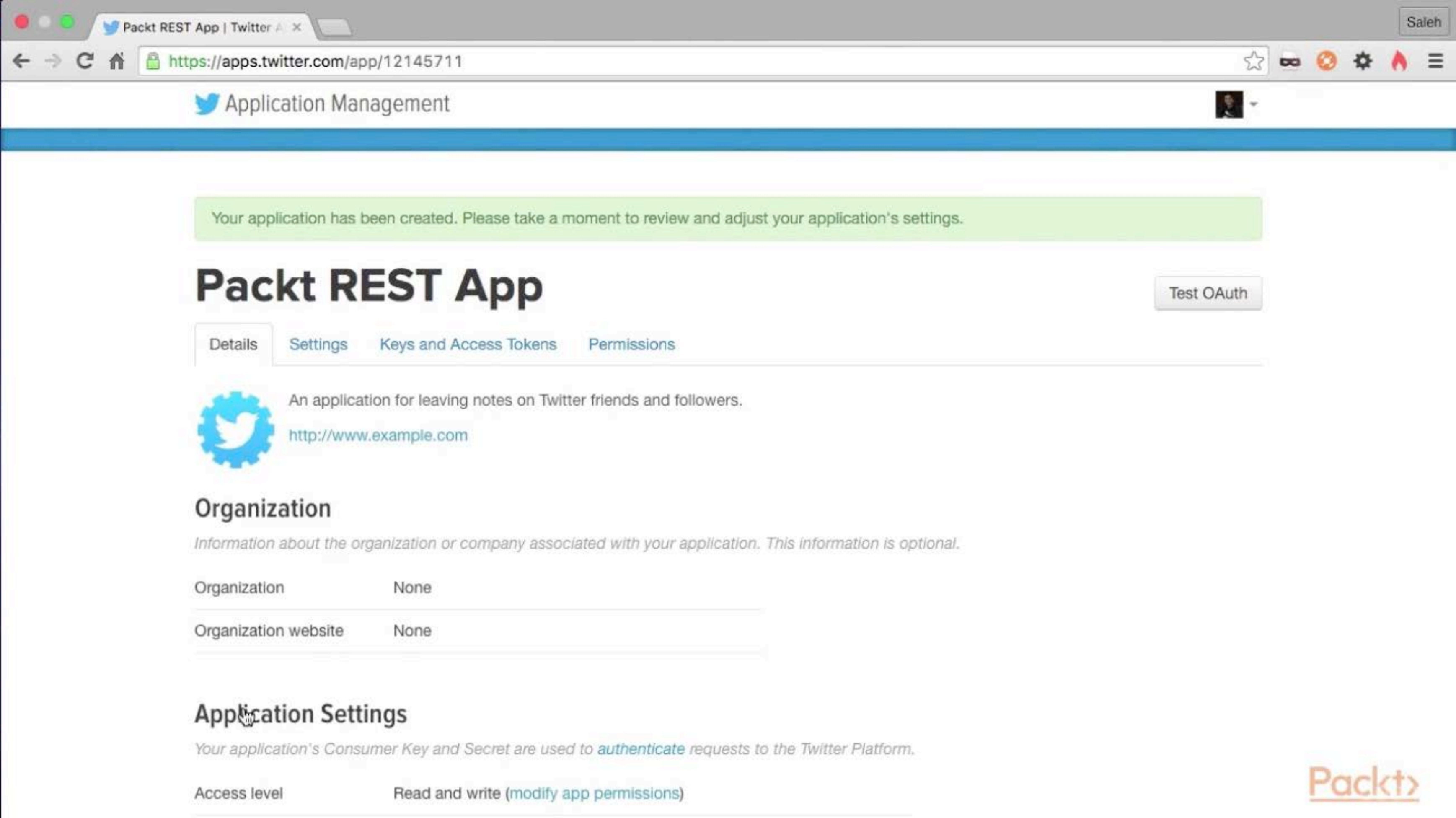


Developer Agreement

- 1. Content** - Tweets, Tweet IDs, Twitter end user profile information, and any other data and information made available to you through the Twitter API or by any other means authorized by Twitter, and any copies and derivative works thereof.
- 2. Developer Site** - Twitter's developer site located at <https://dev.twitter.com>.
- 3. End Users** - Users of your Services, who are bound to enforceable service terms with you.
- 4. Licensed Material** - A collective term for the Twitter API and Content.
- 5. Services** - Your websites, applications and other offerings that display Content or otherwise use the Licensed Material.
- 6. Tweet ID** - A unique identification number generated for each Tweet.
- 7. Tweet** - a public posting with a text body of no more than 140 characters made by any end user of the Twitter Service.
- 8. Twitter API** - The Twitter Application Programming Interface ("API") and the related documentation, data, code, and other materials provided by Twitter with the API, as updated from time to time, including without limitation through the Developer Site.
- 9. Twitter Marks** - The Twitter name, or logos that Twitter makes available to you, including via the Developer Site.

☒ Yes, I agree

Create your Twitter application



Your application has been created. Please take a moment to review and adjust your application's settings.

Packt REST App

Test OAuth

- Details
- Settings
- Keys and Access Tokens
- Permissions



An application for leaving notes on Twitter friends and followers.

<http://www.example.com>

Organization

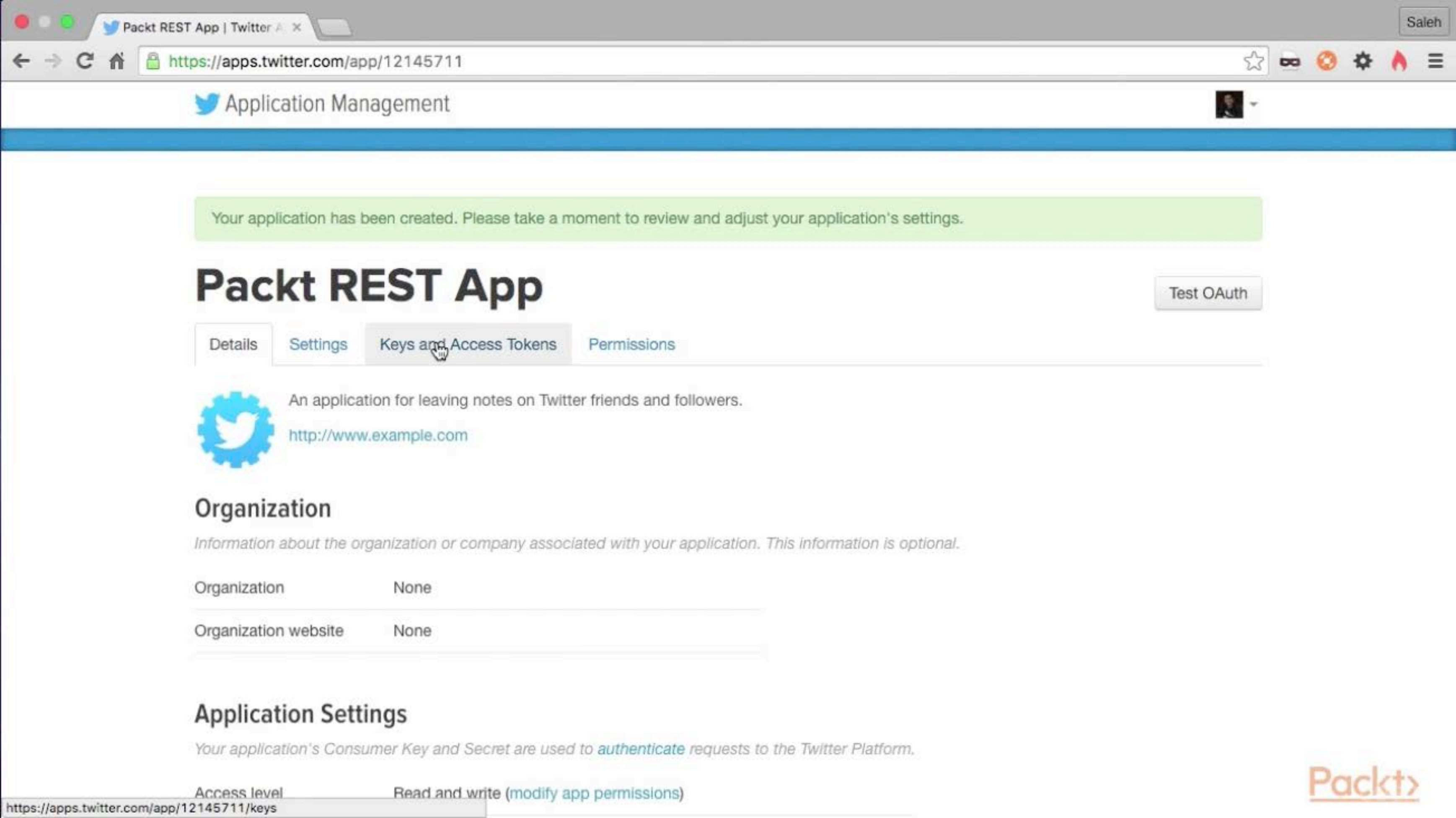
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application Settings

Your application's Consumer Key and Secret are used to [authenticate](#) requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
--------------	-----------------------------------------------------------



Your application has been created. Please take a moment to review and adjust your application's settings.

Packt REST App

Test OAuth

- Details
- Settings
- Keys and Access Tokens
- Permissions



An application for leaving notes on Twitter friends and followers.

http://www.example.com

Organization

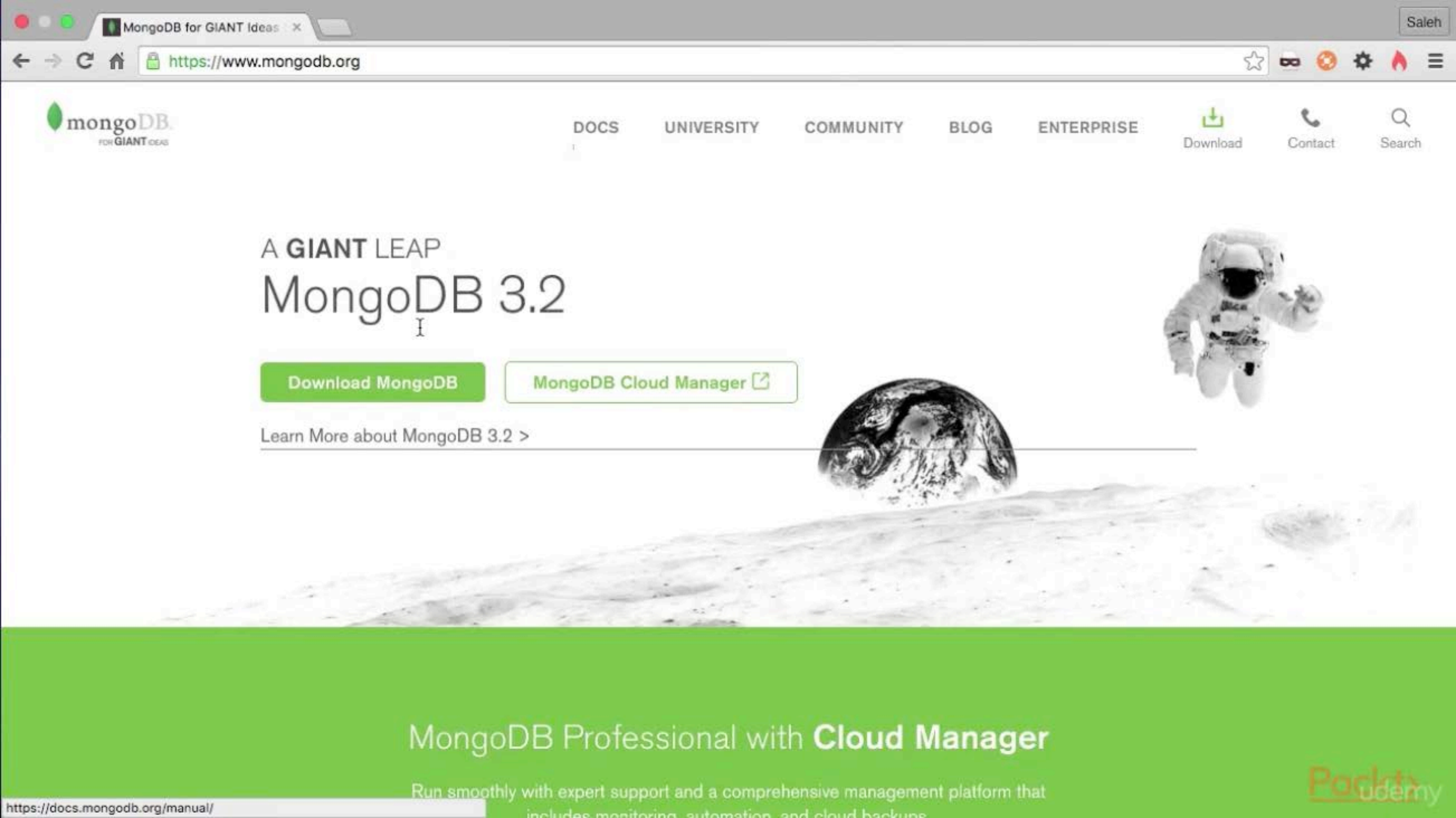
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
--------------	-----------------------------------------



A GIANT LEAP MongoDB 3.2

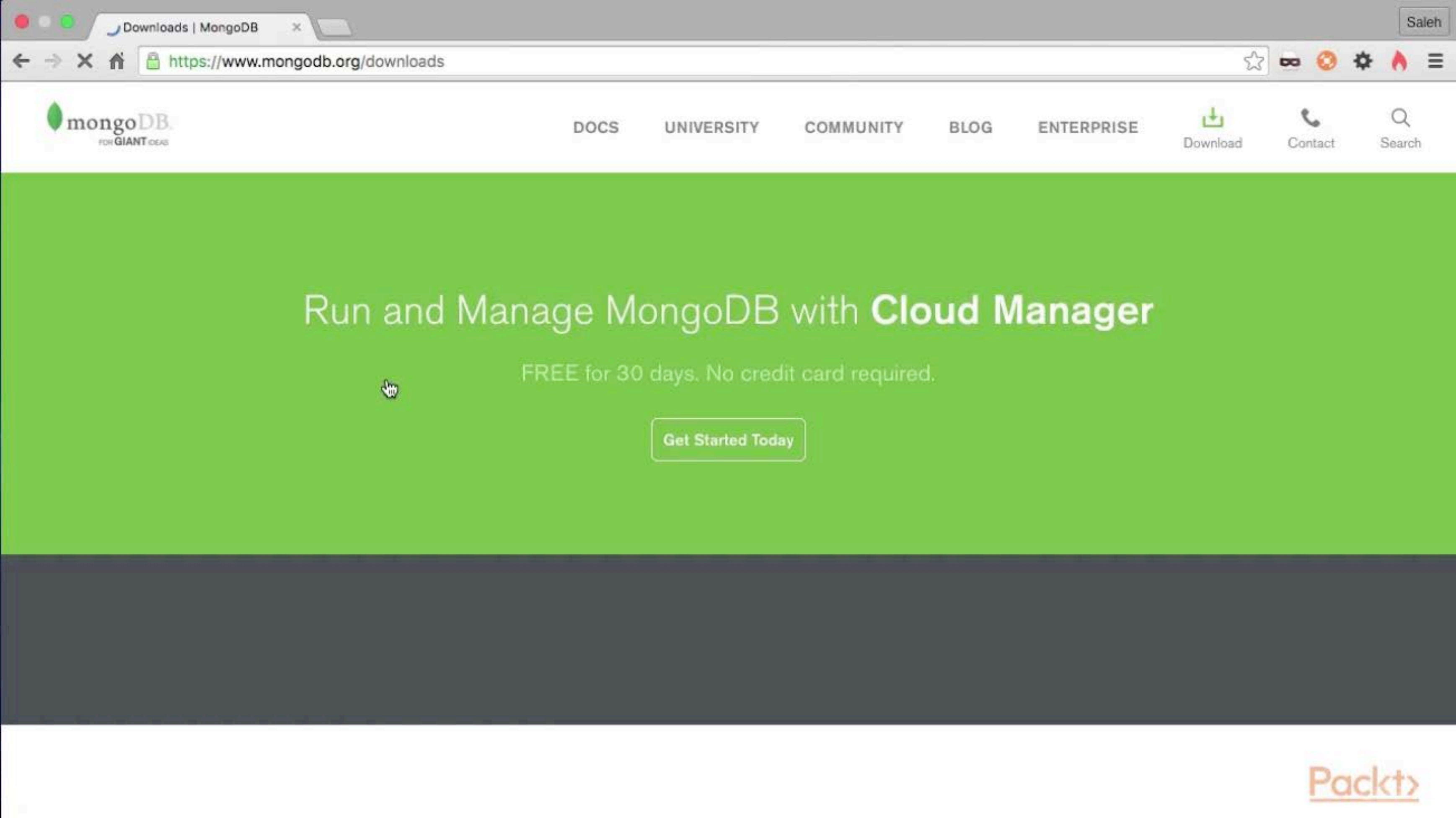
Download MongoDB MongoDB Cloud Manager

Learn More about MongoDB 3.2 >



MongoDB Professional with Cloud Manager

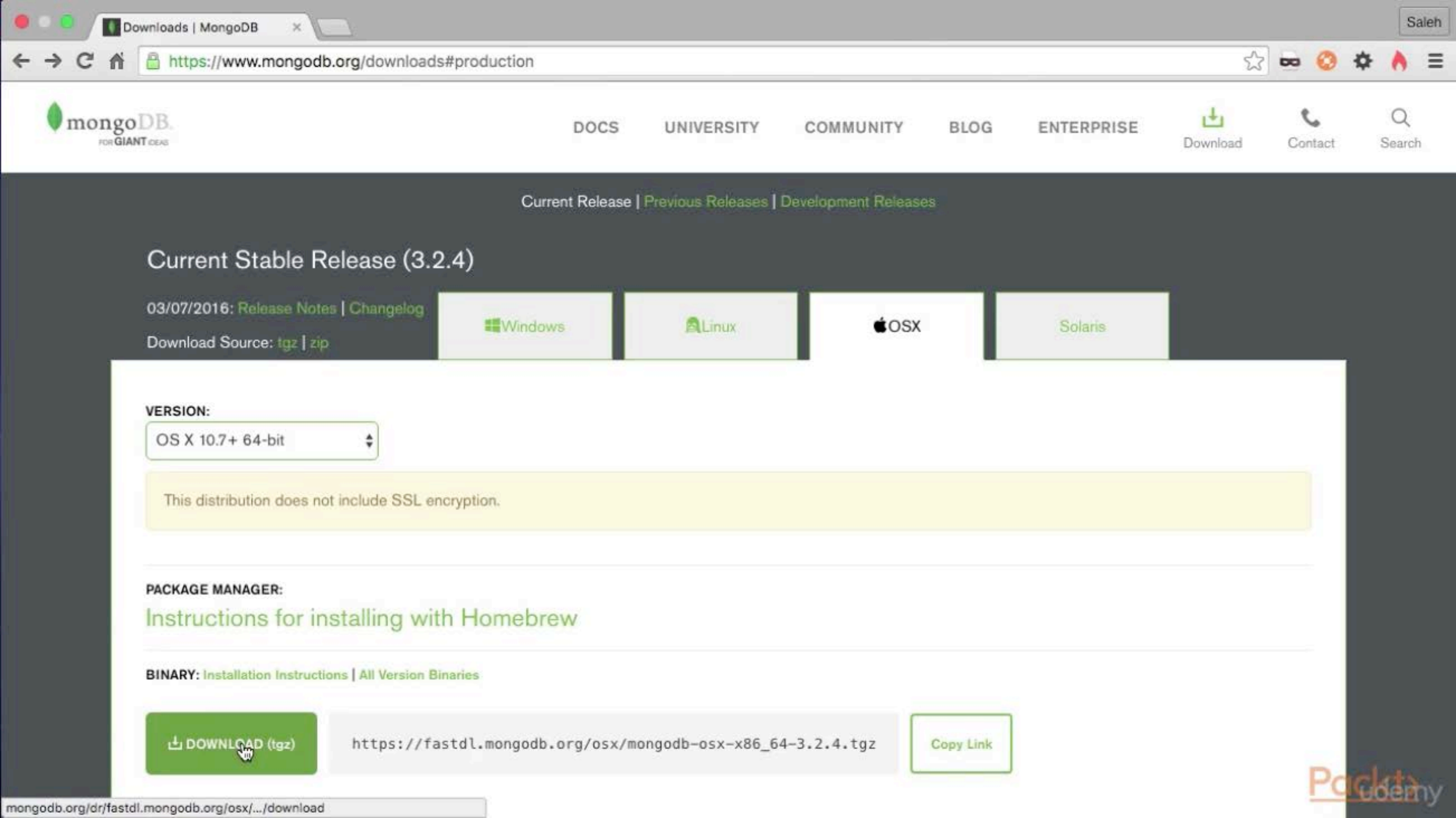
Run smoothly with expert support and a comprehensive management platform that includes monitoring, automation, and cloud backups.



Run and Manage MongoDB with **Cloud Manager**

FREE for 30 days. No credit card required.

Get Started Today



Current Release | [Previous Releases](#) | [Development Releases](#)

Current Stable Release (3.2.4)

03/07/2016: [Release Notes](#) | [Changelog](#)

Download Source: [tgz](#) | [zip](#)

Windows

Linux

OSX

Solaris

VERSION:

OS X 10.7+ 64-bit

This distribution does not include SSL encryption.

PACKAGE MANAGER:

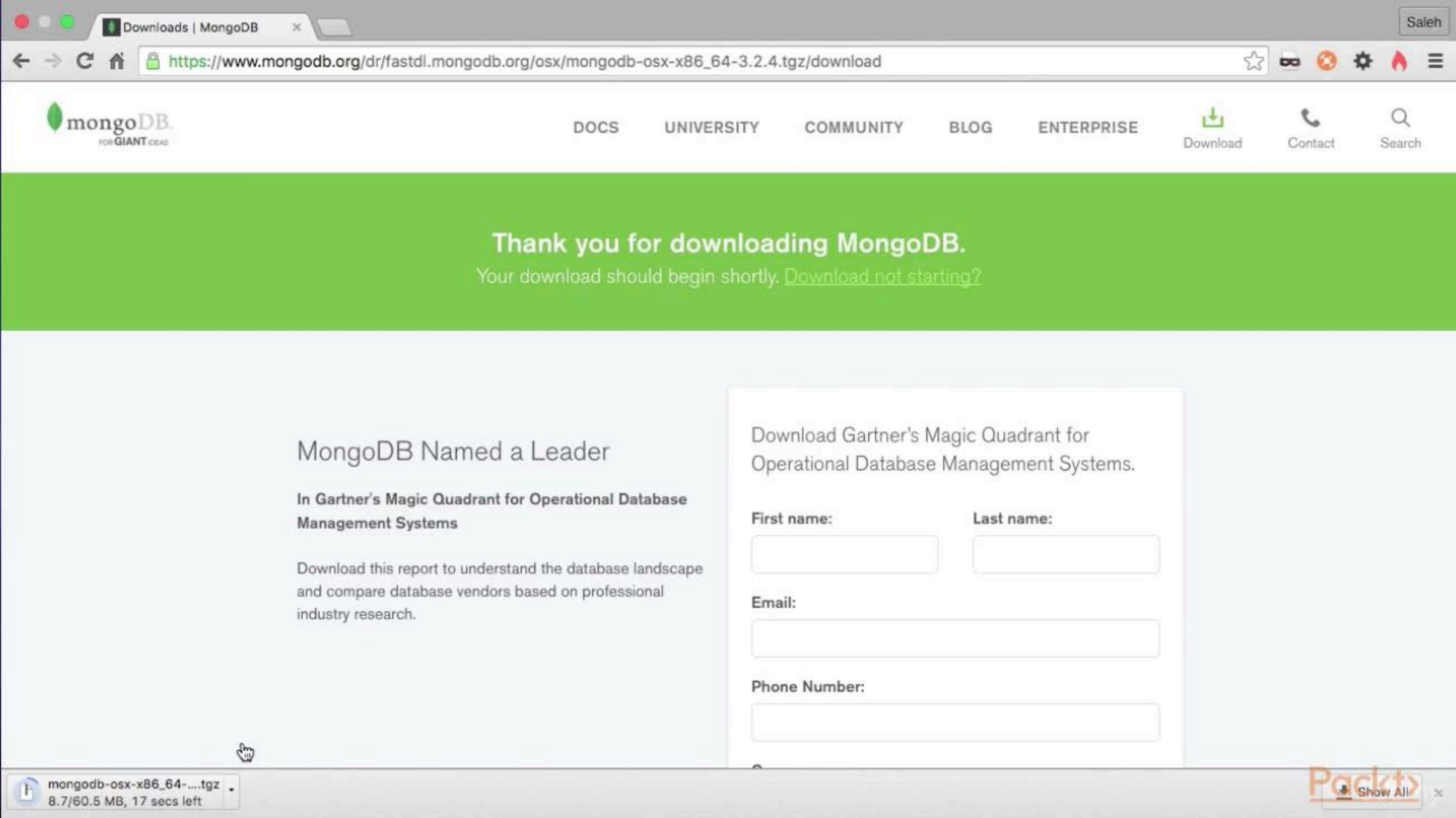
[Instructions for installing with Homebrew](#)

BINARY: [Installation Instructions](#) | [All Version Binaries](#)

DOWNLOAD (tgz)

https://fastdl.mongodb.org/osx/mongodb-osx-x86_64-3.2.4.tgz

Copy Link



Thank you for downloading MongoDB.
Your download should begin shortly. [Download not starting?](#)

MongoDB Named a Leader

In Gartner's Magic Quadrant for Operational Database Management Systems

Download this report to understand the database landscape and compare database vendors based on professional industry research.

Download Gartner's Magic Quadrant for Operational Database Management Systems.

First name:

Last name:

Email:

Phone Number:

```
Salehs-Macintosh:Desktop saleh$ mkdir database
Salehs-Macintosh:Desktop saleh$ mv ~/Downloads/mongodb-osx-x86_64-3.2.4.tgz database/
Salehs-Macintosh:Desktop saleh$ cd database/
Salehs-Macintosh:database saleh$ tar xzvf mongodb-osx-x86_64-3.2.4.tgz
x mongodb-osx-x86_64-3.2.4/README
x mongodb-osx-x86_64-3.2.4/THIRD-PARTY-NOTICES
x mongodb-osx-x86_64-3.2.4/MPL-2
x mongodb-osx-x86_64-3.2.4/GNU-AGPL-3.0
x mongodb-osx-x86_64-3.2.4/bin/mongodump
x mongodb-osx-x86_64-3.2.4/bin/mongorestore
x mongodb-osx-x86_64-3.2.4/bin/mongoexport
x mongodb-osx-x86_64-3.2.4/bin/mongoimport
x mongodb-osx-x86_64-3.2.4/bin/mongostat
x mongodb-osx-x86_64-3.2.4/bin/mongotop
x mongodb-osx-x86_64-3.2.4/bin/bsondump
x mongodb-osx-x86_64-3.2.4/bin/mongofiles
x mongodb-osx-x86_64-3.2.4/bin/mongooplog
x mongodb-osx-x86_64-3.2.4/bin/mongoperf
x mongodb-osx-x86_64-3.2.4/bin/mongosniff
x mongodb-osx-x86_64-3.2.4/bin/mongod
x mongodb-osx-x86_64-3.2.4/bin/mongos
x mongodb-osx-x86_64-3.2.4/bin/mongo
Salehs-Macintosh:database saleh$
```



```
[Salehs-Macintosh:database saleh$ tar xzvf mongodb-osx-x86_64-3.2.4.tgz
```

```
x mongodb-osx-x86_64-3.2.4/README
x mongodb-osx-x86_64-3.2.4/THIRD-PARTY-NOTICES
x mongodb-osx-x86_64-3.2.4/MPL-2
x mongodb-osx-x86_64-3.2.4/GNU-AGPL-3.0
x mongodb-osx-x86_64-3.2.4/bin/mongodump
x mongodb-osx-x86_64-3.2.4/bin/mongorestore
x mongodb-osx-x86_64-3.2.4/bin/mongoexport
x mongodb-osx-x86_64-3.2.4/bin/mongoimport
x mongodb-osx-x86_64-3.2.4/bin/mongostat
x mongodb-osx-x86_64-3.2.4/bin/mongotop
x mongodb-osx-x86_64-3.2.4/bin/bsondump
x mongodb-osx-x86_64-3.2.4/bin/mongofiles
x mongodb-osx-x86_64-3.2.4/bin/mongooplog
x mongodb-osx-x86_64-3.2.4/bin/mongoperf
x mongodb-osx-x86_64-3.2.4/bin/mongosniff
x mongodb-osx-x86_64-3.2.4/bin/mongod
x mongodb-osx-x86_64-3.2.4/bin/mongos
x mongodb-osx-x86_64-3.2.4/bin/mongo
```

```
[Salehs-Macintosh:database saleh$ mkdir db
```

```
[Salehs-Macintosh:database saleh$ cd db/
```

```
[Salehs-Macintosh:db saleh$ pwd
```

```
/Users/saleh/Desktop/database/db
```

```
[Salehs-Macintosh:db saleh$ cd ../mongodb-osx-x86_64-3.2.4/bin/
```

```
Salehs-Macintosh:bin saleh$
```



```
x mongodb-osx-x86_64-3.2.4/bin/mongos
x mongodb-osx-x86_64-3.2.4/bin/mongo
Salehs-Macintosh:database saleh$ mkdir db
Salehs-Macintosh:database saleh$ cd db/
Salehs-Macintosh:db saleh$ pwd
/Users/saleh/Desktop/database/db
Salehs-Macintosh:db saleh$ cd ../mongodb-osx-x86_64-3.2.4/bin/
Salehs-Macintosh:bin saleh$ ./mongod --dbpath /Users/saleh/Desktop/database/db
2016-03-27T15:48:25.621-0400 I CONTROL [initandlisten] MongoDB starting : pid=1189 port=27017 db
path=/Users/saleh/Desktop/database/db 64-bit host=Salehs-Macintosh.local
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] db version v3.2.4
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] git version: e2ee9ffcf9f5a94fad76802e28cc
978718bb7a30
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] allocator: system
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] modules: none
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] build environment:
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] distarch: x86_64
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] target_arch: x86_64
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] options: { storage: { dbPath: "/Users/sal
eh/Desktop/database/db" } }
2016-03-27T15:48:25.648-0400 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size
=1G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=t
rue,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint
=(wait=60,log_size=2GB),statistics_log=(wait=0),
```



```
2016-03-27T15:48:25.621-0400 I CONTROL [initandlisten] MongoDB starting : pid=1189 port=27017 db
path=/Users/saleh/Desktop/database/db 64-bit host=Salehs-Macintosh.local
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] db version v3.2.4
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] git version: e2ee9ffcf9f5a94fad76802e28cc
978718bb7a30
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] allocator: system
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] modules: none
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] build environment:
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten]     distarch: x86_64
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten]     target_arch: x86_64
2016-03-27T15:48:25.622-0400 I CONTROL [initandlisten] options: { storage: { dbPath: "/Users/sal
eh/Desktop/database/db" } }
2016-03-27T15:48:25.648-0400 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size
=1G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=t
rue,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint
=(wait=60,log_size=2GB),statistics_log=(wait=0),
2016-03-27T15:48:30.352-0400 I CONTROL [initandlisten]
2016-03-27T15:48:30.352-0400 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number
of files is 256, should be at least 1000
2016-03-27T15:48:30.756-0400 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canoni
calization worker
2016-03-27T15:48:30.756-0400 I FTDC [initandlisten] Initializing full-time diagnostic data ca
pture with directory '/Users/saleh/Desktop/database/db/diagnostic.data'
2016-03-27T15:48:32.735-0400 I NETWORK [initandlisten] waiting for connections on port 27017
```

Summary

- Discussed what is an API
- Communication via HTTP
- REST resources and properties
- Set up the environment

Next Section

Consuming a RESTful API