Source signup, set up, download, activation, usage

Unregistered user learns about Source on Apigee.com, signs up, sets up source downloads JS library, supplies credentials, signs in to sample app, views sample app.

1. A developer visits Source product page on Apigee.com
2. Signs up
3. Sees welcome. Gets an overview.
4. Sets up a source application via online command line interface.
5. selects a JS library to download
6. Downloads and extracts JS code library
7. Opens in IDE, pastes application endpoint and app ID into file
8. Runs Source application
9. Creates an end-user account in local demo app
10. Signs in with Twitter, makes the demo app a trusted Twitter application
11. Views demo app
12. Returns to Apigee Source to manage the demo app via online command line interface
13. Explores Twitter API with Source-enhanced Console

View

Apigee Free

Apigee labs feature

Apigee Source (for Twitter)

A simple, primitive Twitter prototyping tool for the brave.

Sign up and give Source a try

What is Apigee Source?

Apigee source is our first, really early attempt to develop tools that help developers get to building something functional faster. Source is currently only good for quickly roughing out single-user Twitter applications, but we're hoping there's enough you'll try it out and let us know what you think.

Apigee Source is Free to use. get started »

How to get started with Source

1. Make a valid authenticated request

Make an Authenticated Twitter API request in the Console.

2. Get a valid response, view source code

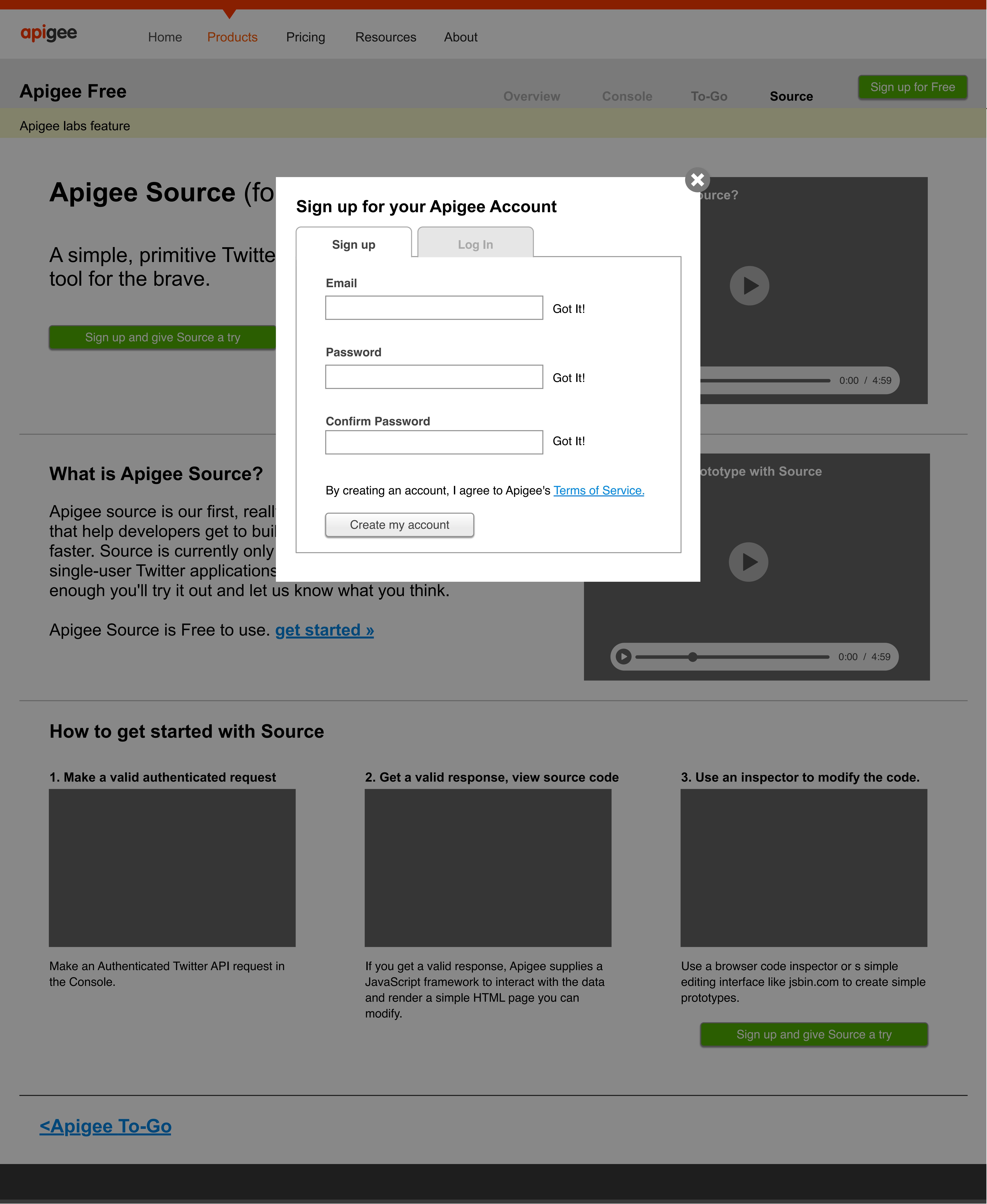
If you get a valid response, Apigee supplies a JavaScript framework to interact with the data and render a simple HTML page you can modify.

3. Use an inspector to modify the code.

Use a browser code inspector or s simple editing interface like jsbin.com to create simple prototypes.

Sign up and give Source a try

<Apigee To-Go



Apigee Free

Apigee labs feature

Apigee Source (fo

A simple, primitive Twitte
tool for the brave.

Sign up for your Apigee Account

Email

Got It!

Password

Got It!

Confirm Password

Got It!

By creating an account, I agree to Apigee's Terms of Service.

What is Apigee Source?

Apigee source is our first, reall
that help developers get to buil
faster. Source is currently only
single-user Twitter applications
enough you'll try it out and let us know what you think.

Apigee Source is Free to use. get started »

How to get started with Source

1. Make a valid authenticated request

Make an Authenticated Twitter API request in the Console.

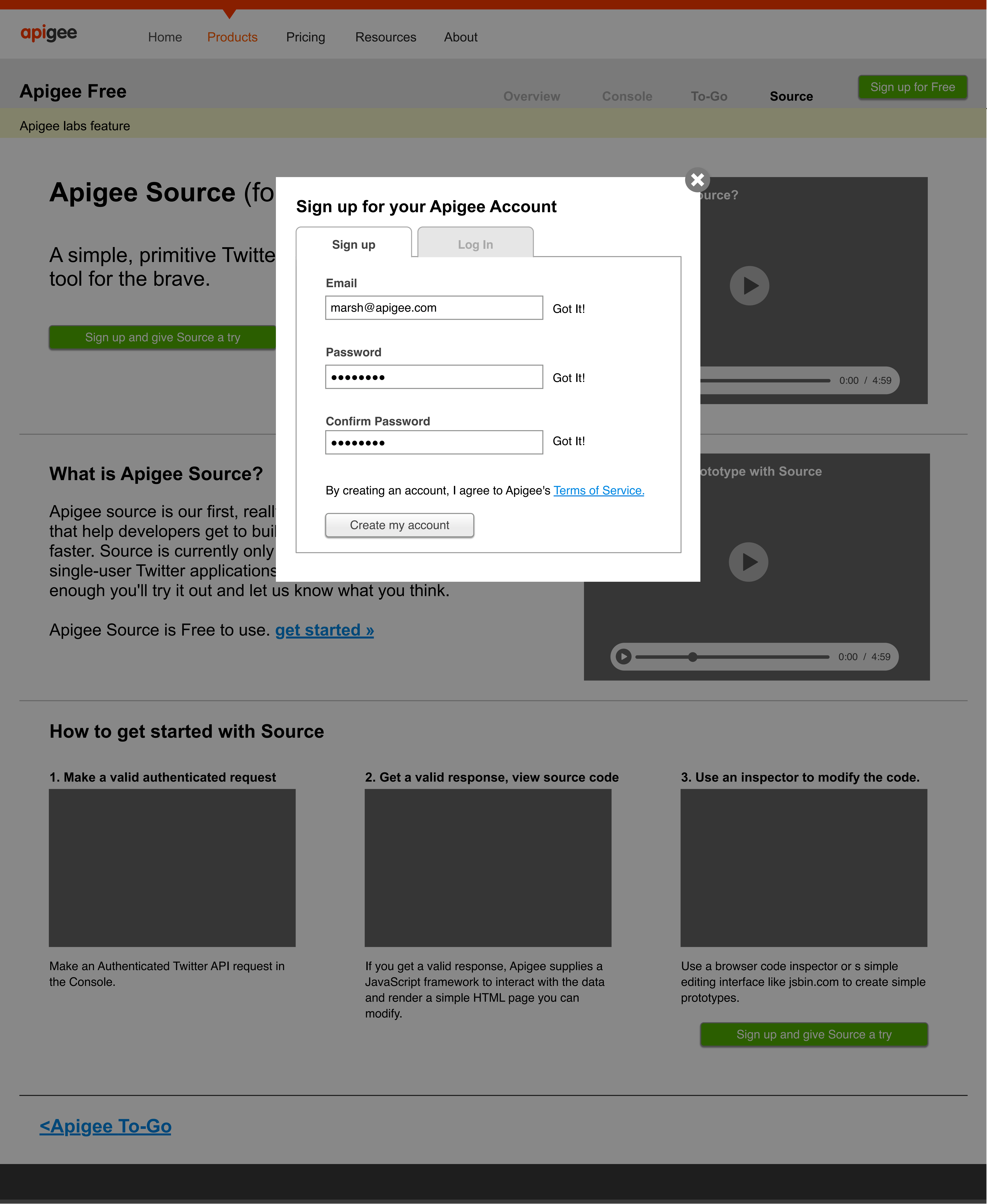
2. Get a valid response, view source code

If you get a valid response, Apigee supplies a JavaScript framework to interact with the data and render a simple HTML page you can modify.

3. Use an inspector to modify the code.

Use a browser code inspector or s simple editing interface like jsbin.com to create simple prototypes.

<Apigee To-Go



Apigee Free

Apigee labs feature

Apigee Source (fo

A simple, primitive Twitte
tool for the brave.

Sign up for your Apigee Account

Email
marsh@apigee.com Got It!

Password
•••••••• Got It!

Confirm Password
•••••••• Got It!

By creating an account, I agree to Apigee's Terms of Service.

What is Apigee Source?

Apigee source is our first, reall
that help developers get to buil
faster. Source is currently only
single-user Twitter applications
enough you'll try it out and let us know what you think.

Apigee Source is Free to use. get started »

How to get started with Source

1. Make a valid authenticated request

Make an Authenticated Twitter API request in the Console.

2. Get a valid response, view source code

If you get a valid response, Apigee supplies a JavaScript framework to interact with the data and render a simple HTML page you can modify.

3. Use an inspector to modify the code.

Use a browser code inspector or s simple editing interface like jsbin.com to create simple prototypes.

<Apigee To-Go

apigee → **Source**

marsh@apigee.com ▼

✓ Welcome! You are signed in to Apigee and Source has been activated.

Welcome to Source for Twitter

Apigee Source offers a way to prototype web applications faster. We handle the OAuth mess so you can get to building faster.

Set up	Use Apigee's command line interface for setting up source application >>
Manage	Use Apigee's command line interface for managing your source applications and credentials >>
Libraries	Use these libraries to get started quickly in your language of choice. >>
Console	Make Twitter API requests and get the code snippet to make the same request in your Source application. >>

Updates

6/16/11
Twitter API has been added

6/14/11
JQuery plug in has been revised to include simplified setup. Download

6/13/11
Ruby Library added. Download

6/11/11
OAuth 1.0a support added to Source

apigee → **Source**

marsh@apigee.com ▼

Welcome to Source for Twitter

Apigee Source offers a way to prototype web applications faster. We handle the OAuth mess so you can get to building faster.

Set up	Use Apigee's command line interface for setting up source application >>
Manage	Use Apigee's command line interface for managing your source applications and credentials >>
Libraries	Use these libraries to get started quickly in your language of choice. >>
Console	Make Twitter API requests and get the code snippet to make the same request in your Source application. >>

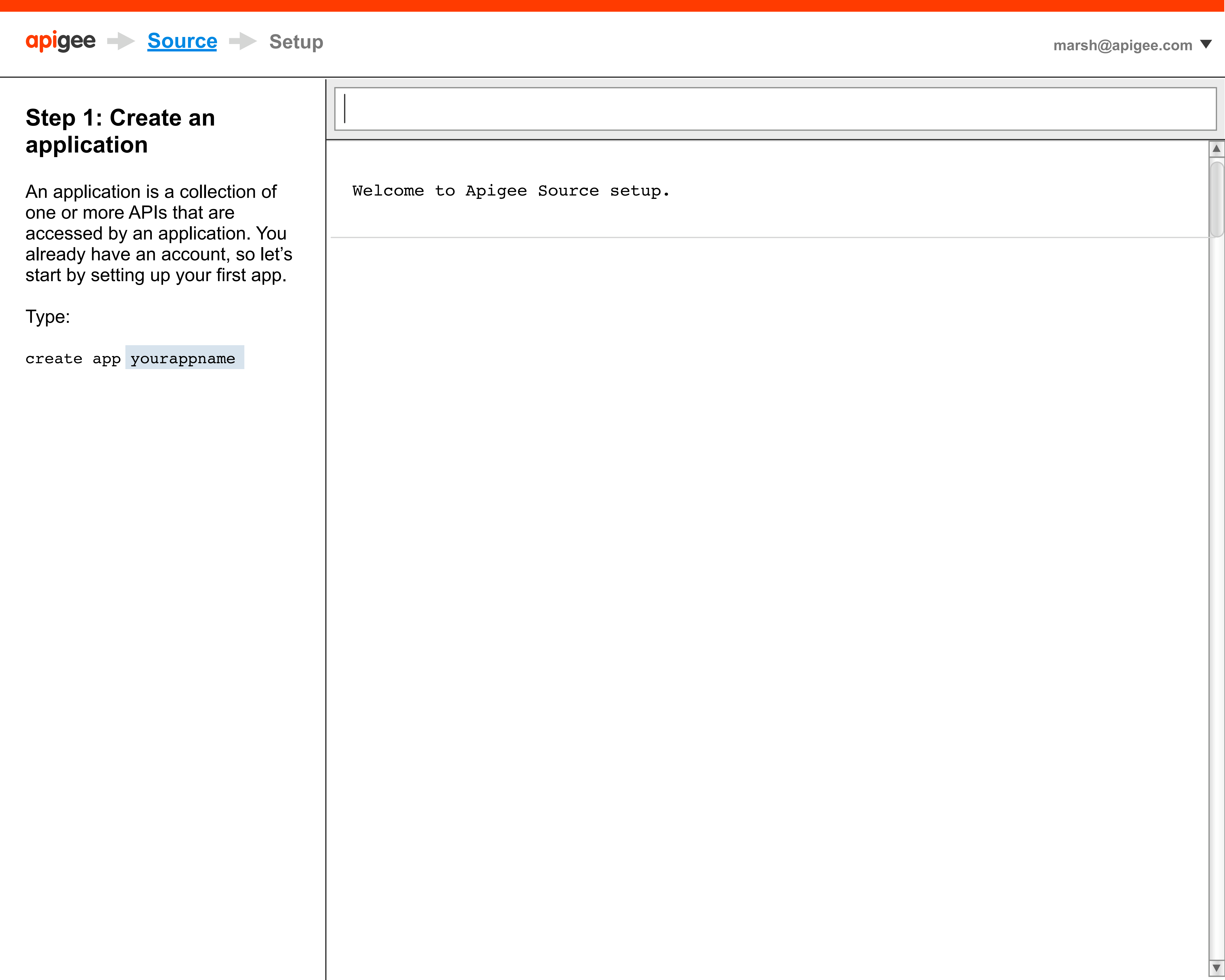
Updates

6/16/11
Twitter API has been added

6/14/11
JQuery plug in has been revised to include simplified setup. Download

6/13/11
Ruby Library added. Download

6/11/11
OAuth 1.0a support added to Source



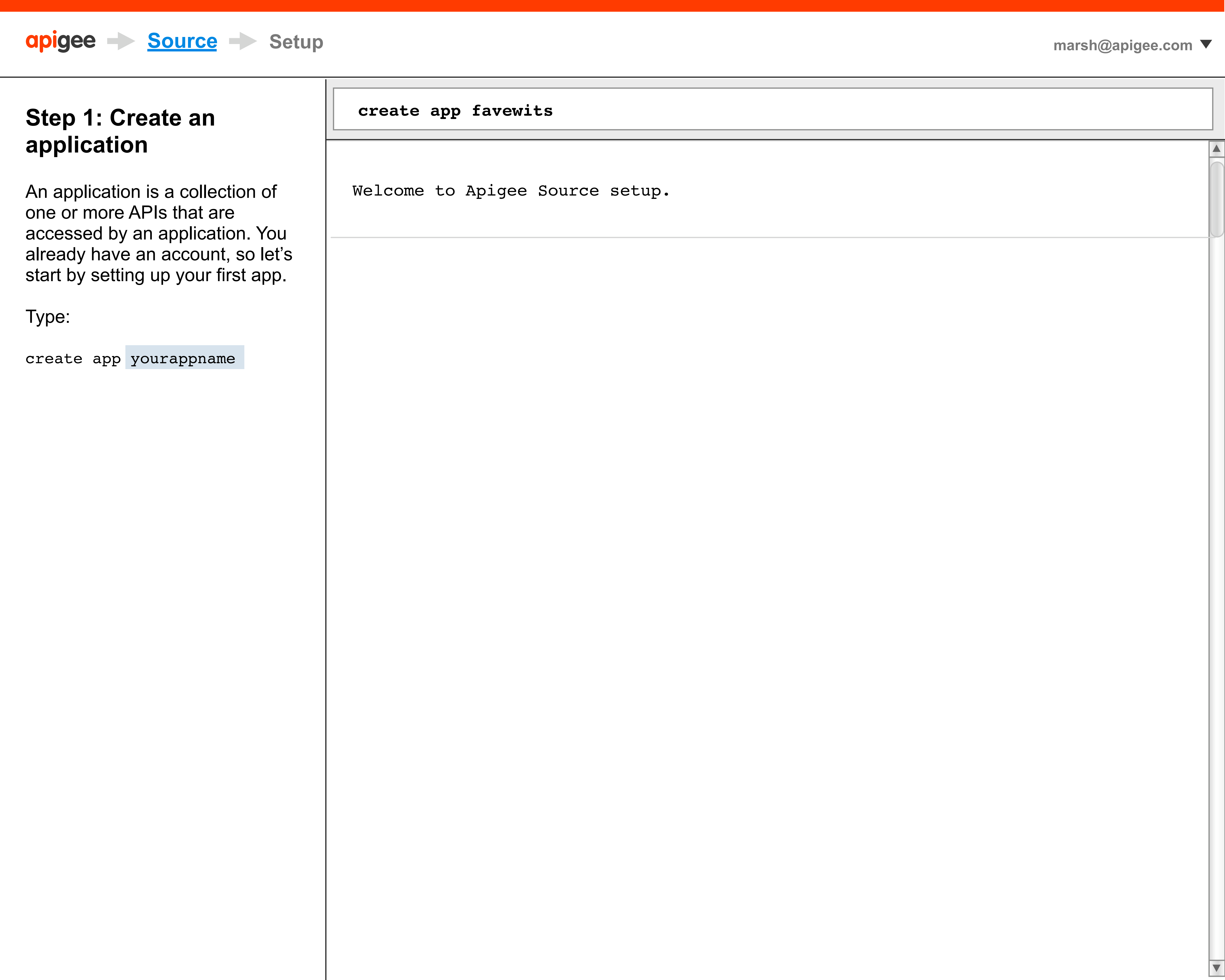
Step 1: Create an application

An application is a collection of one or more APIs that are accessed by an application. You already have an account, so let's start by setting up your first app.

Type:

```
create app yourappname
```

```
Welcome to Apigee Source setup.
```



Step 1: Create an application

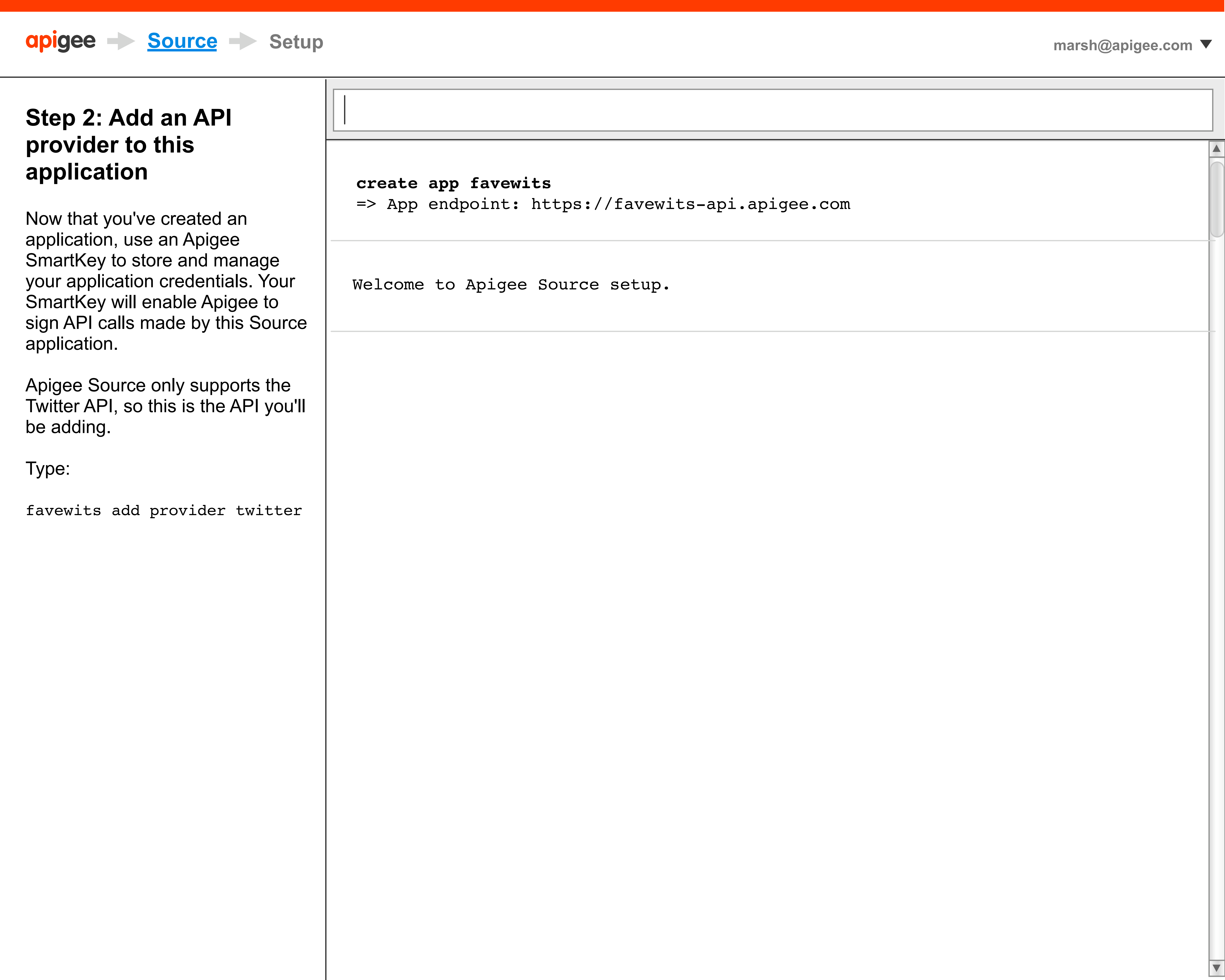
An application is a collection of one or more APIs that are accessed by an application. You already have an account, so let's start by setting up your first app.

Type:

```
create app yourappname
```

```
create app favewits
```

```
Welcome to Apigee Source setup.
```



Step 2: Add an API provider to this application

Now that you've created an application, use an Apigee SmartKey to store and manage your application credentials. Your SmartKey will enable Apigee to sign API calls made by this Source application.

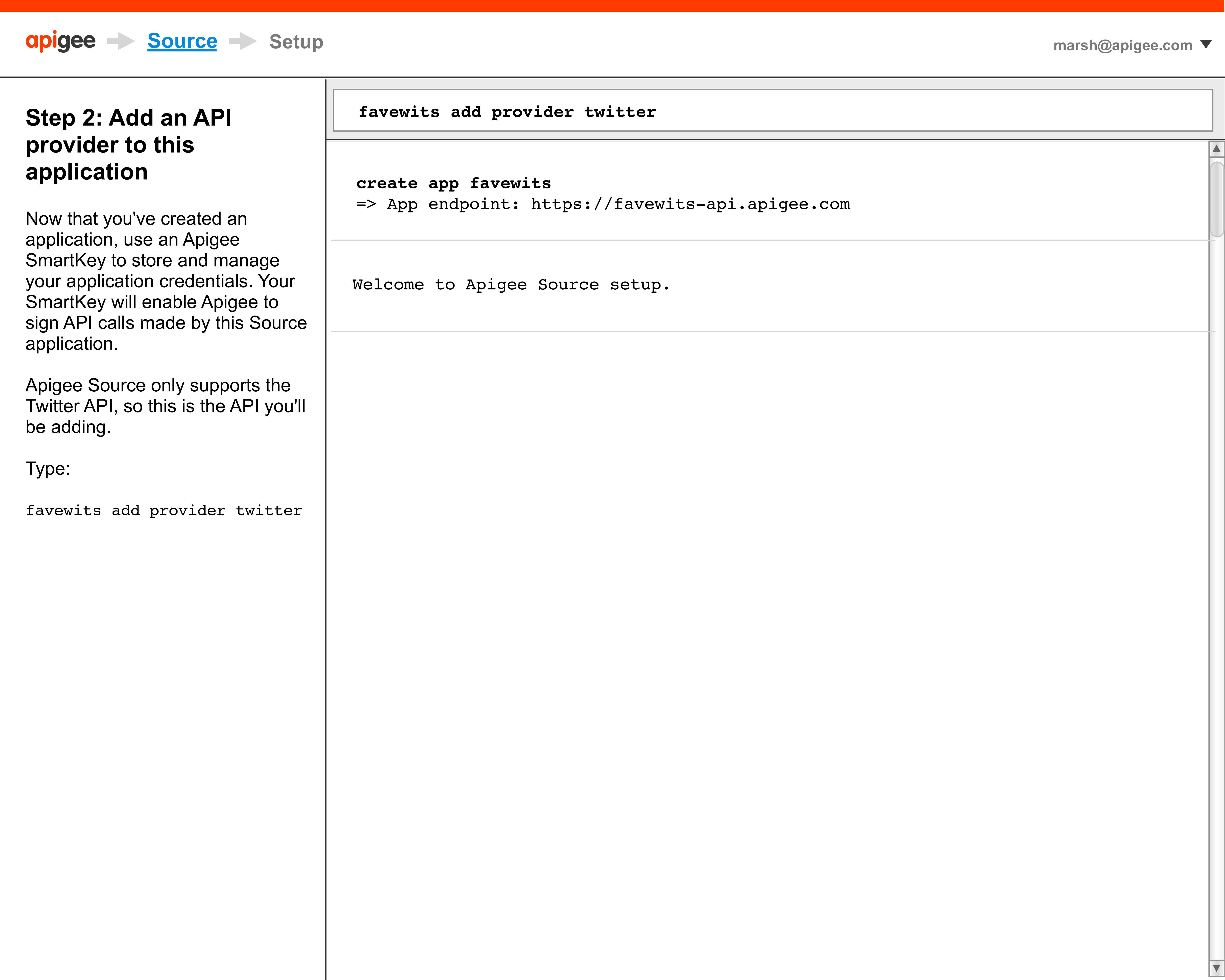
Apigee Source only supports the Twitter API, so this is the API you'll be adding.

Type:

```
favewits add provider twitter
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 2: Add an API provider to this application

Now that you've created an application, use an Apigee SmartKey to store and manage your application credentials. Your SmartKey will enable Apigee to sign API calls made by this Source application.

Apigee Source only supports the Twitter API, so this is the API you'll be adding.

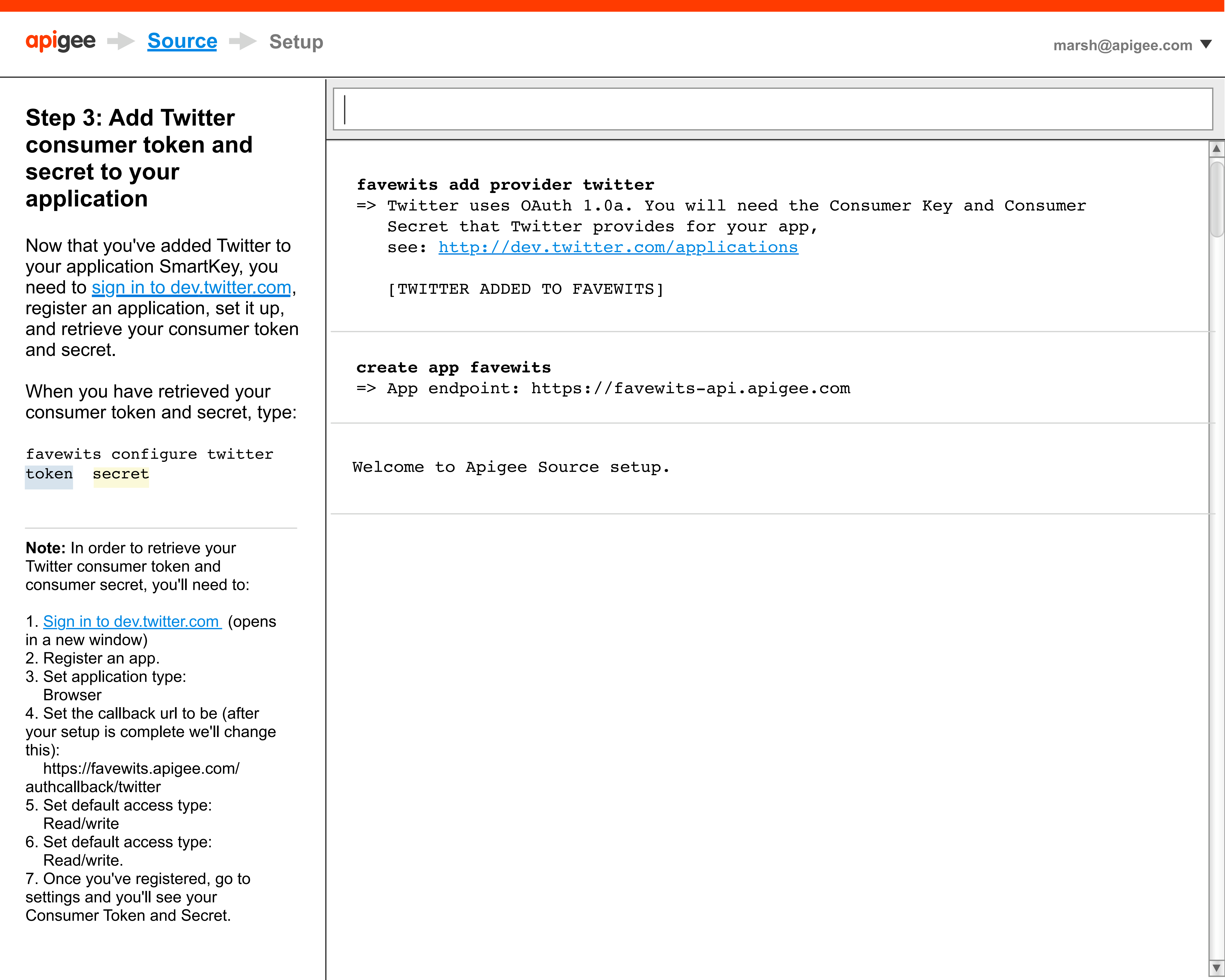
Type:

```
favewits add provider twitter
```

```
favewits add provider twitter
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 3: Add Twitter consumer token and secret to your application

Now that you've added Twitter to your application SmartKey, you need to sign in to dev.twitter.com, register an application, set it up, and retrieve your consumer token and secret.

When you have retrieved your consumer token and secret, type:

```
favewits configure twitter
token  secret
```

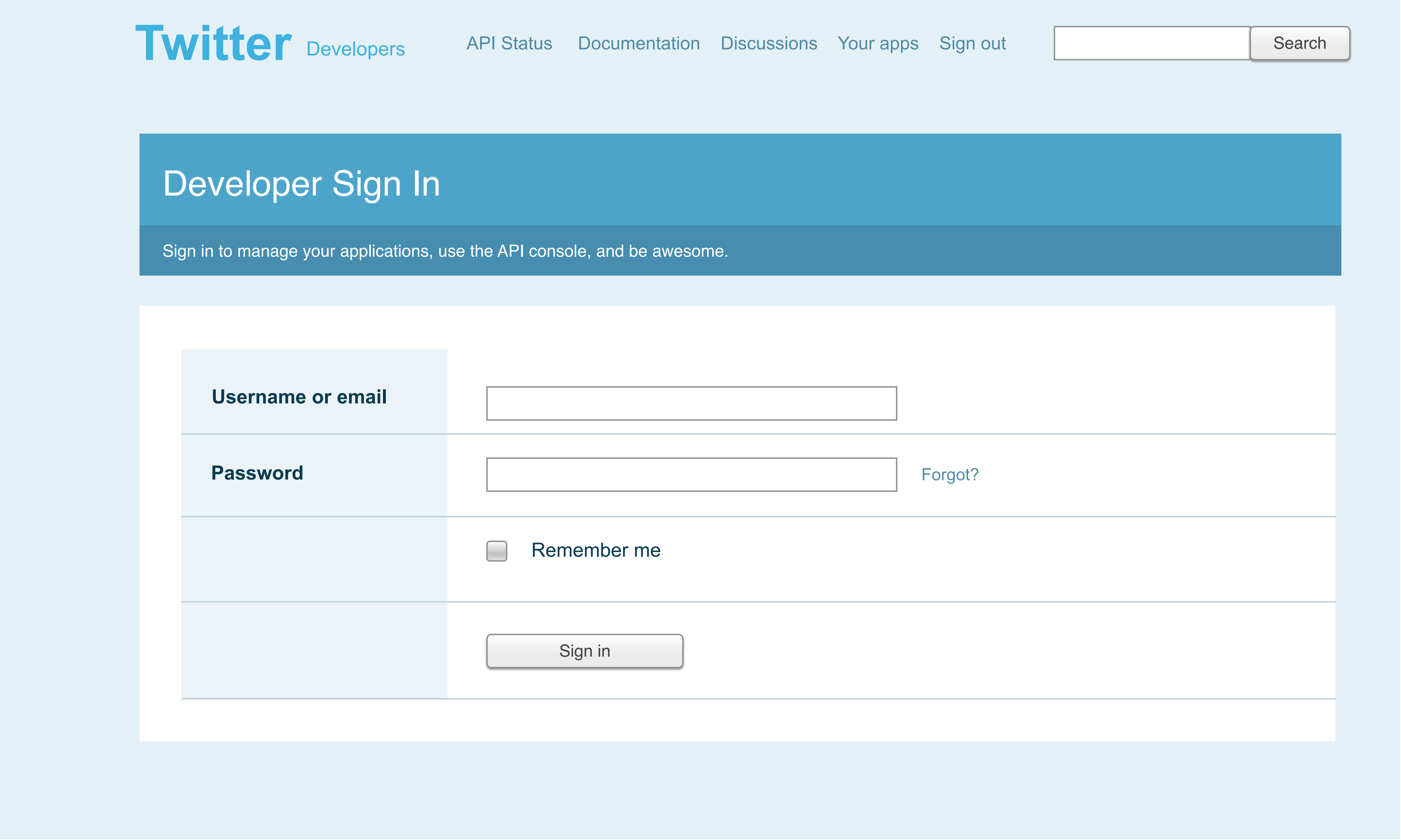
Note: In order to retrieve your Twitter consumer token and consumer secret, you'll need to:

1. Sign in to dev.twitter.com (opens in a new window)
2. Register an app.
3. Set application type:
   Browser
4. Set the callback url to be (after your setup is complete we'll change this):
   https://favewits.apigee.com/authcallback/twitter
5. Set default access type:
   Read/write
6. Set default access type:
   Read/write.
7. Once you've registered, go to settings and you'll see your Consumer Token and Secret.

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

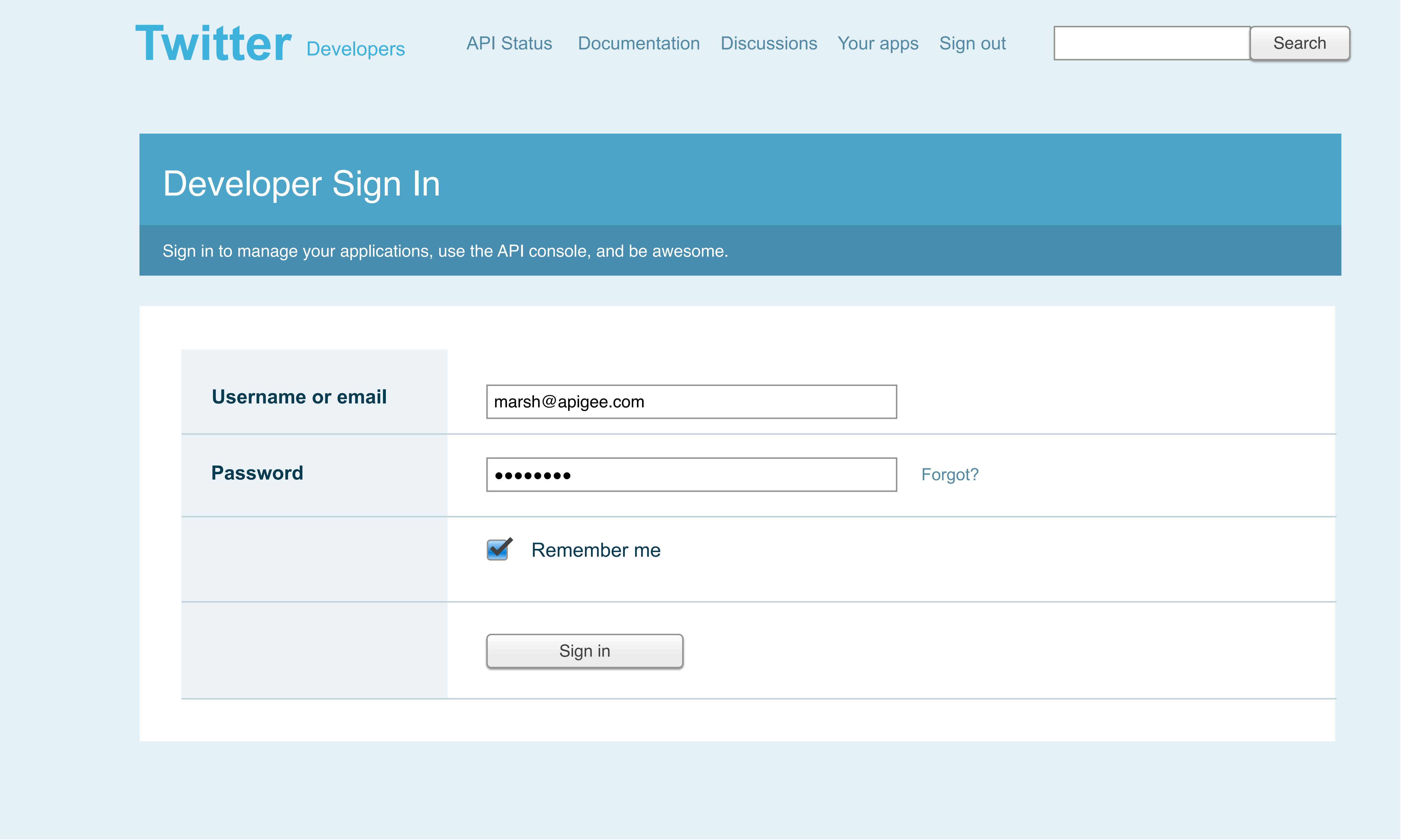
```
Welcome to Apigee Source setup.
```



Developer Sign In

Sign in to manage your applications, use the API console, and be awesome.

Username or email	
Password	Forgot?
	□ Remember me
	Sign in



Developer Sign In

Sign in to manage your applications, use the API console, and be awesome.

Username or email marsh@apigee.com

Password ●●●●●●●●●

☑ Remember me

Twitter Developers

API Status Documentation Discussions Your apps Sign out

Search

Create cool applications that integrate with Twitter

1. Get Started

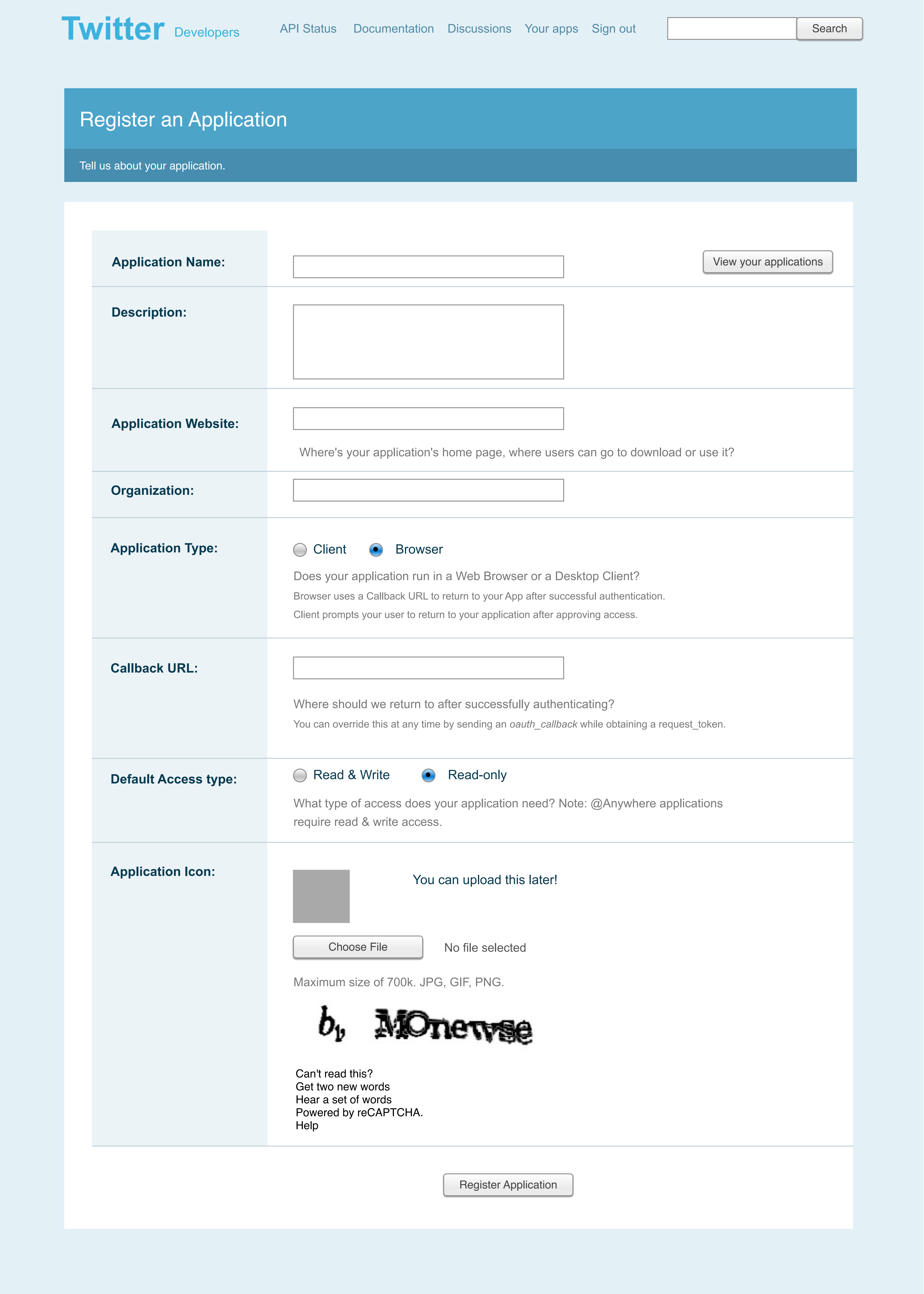
Explore all of Twitter from your timeline and mentions

2. Register an App

Register an application to start using the twitter API. It's easy

3. Discuss

Get in touch with the Twitter API Team and the community of API developers



Register an Application

Tell us about your application.

Application Name:

Description:

Application Website:

Where's your application's home page, where users can go to download or use it?

Organization:

Application Type: ○ Client ● Browser

Does your application run in a Web Browser or a Desktop Client?

Browser uses a Callback URL to return to your App after successful authentication.

Client prompts your user to return to your application after approving access.

Callback URL:

Where should we return to after successfully authenticating?

You can override this at any time by sending an *oauth_callback* while obtaining a request_token.

Default Access type: ○ Read & Write ● Read-only

What type of access does your application need? Note: @Anywhere applications require read & write access.

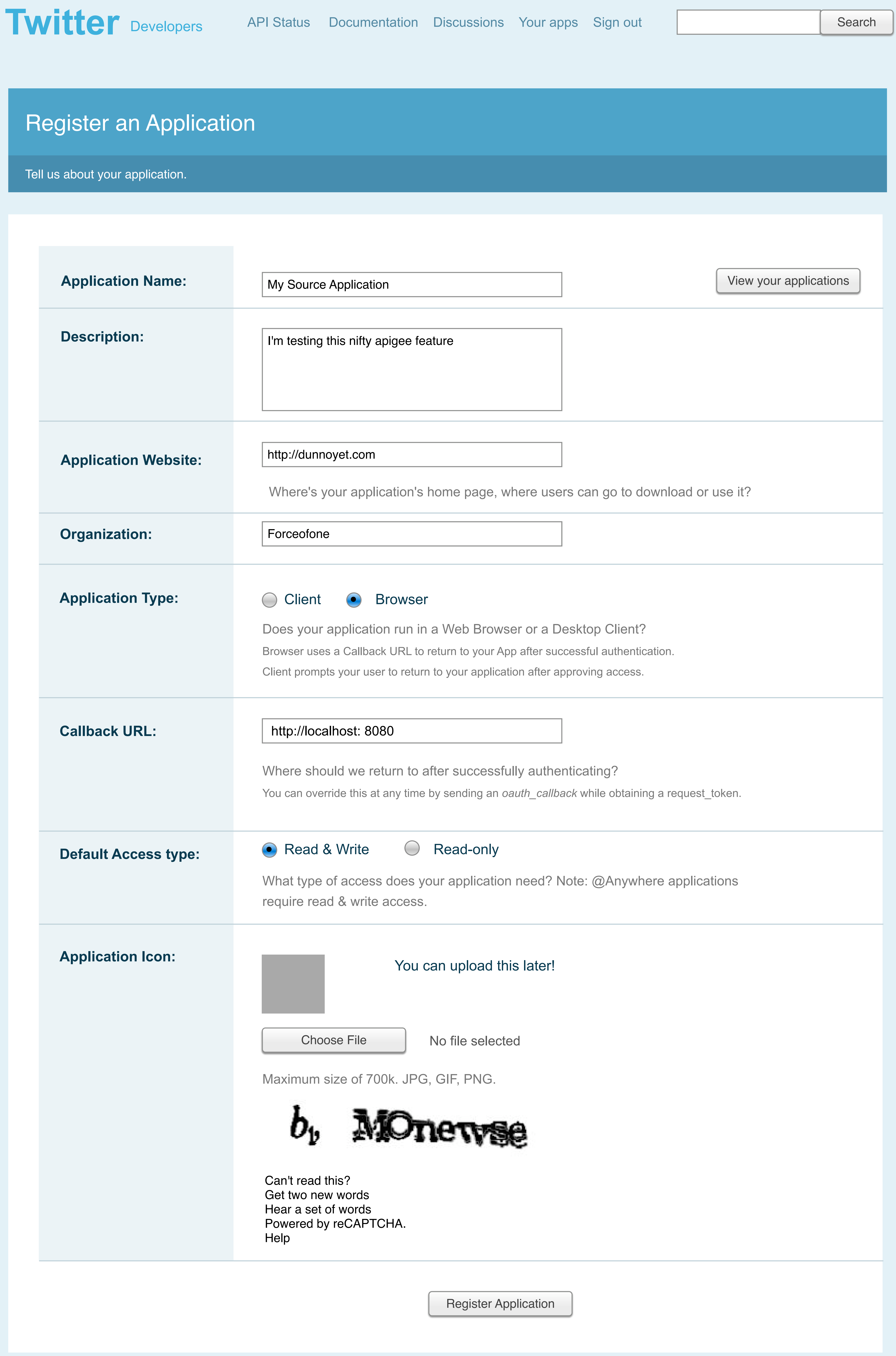
Application Icon:

You can upload this later!

No file selected

Maximum size of 700k. JPG, GIF, PNG.

Can't read this?
Get two new words
Hear a set of words
Powered by reCAPTCHA.
Help



Register an Application

Tell us about your application.

Application Name: My Source Application

Description: I'm testing this nifty apigee feature

Application Website: http://dunnoyet.com

Where's your application's home page, where users can go to download or use it?

Organization: Forceofone

Application Type: ○ Client ● Browser

Does your application run in a Web Browser or a Desktop Client?

Browser uses a Callback URL to return to your App after successful authentication.

Client prompts your user to return to your application after approving access.

Callback URL: http://localhost: 8080

Where should we return to after successfully authenticating?

You can override this at any time by sending an *oauth_callback* while obtaining a request_token.

Default Access type: ● Read & Write ○ Read-only

What type of access does your application need? Note: @Anywhere applications require read & write access.

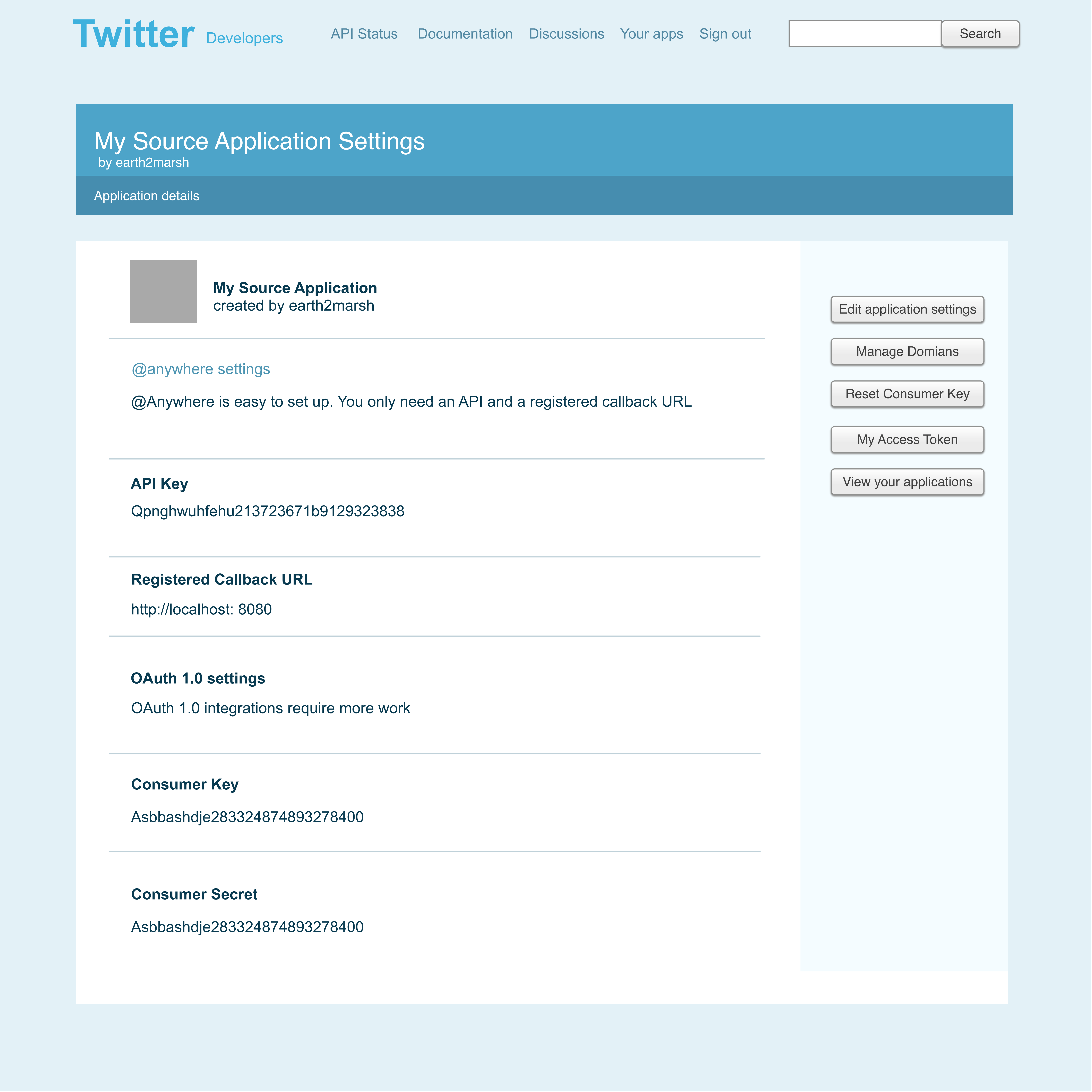
Application Icon:

You can upload this later!

Choose File No file selected

Maximum size of 700k. JPG, GIF, PNG.

Can't read this?
Get two new words
Hear a set of words
Powered by reCAPTCHA.
Help



My Source Application Settings

by earth2marsh

Application details

My Source Application
created by earth2marsh

@anywhere settings

@Anywhere is easy to set up. You only need an API and a registered callback URL

API Key

Qpnghwuhfehu213723671b9129323838

Registered Callback URL

http://localhost: 8080

OAuth 1.0 settings

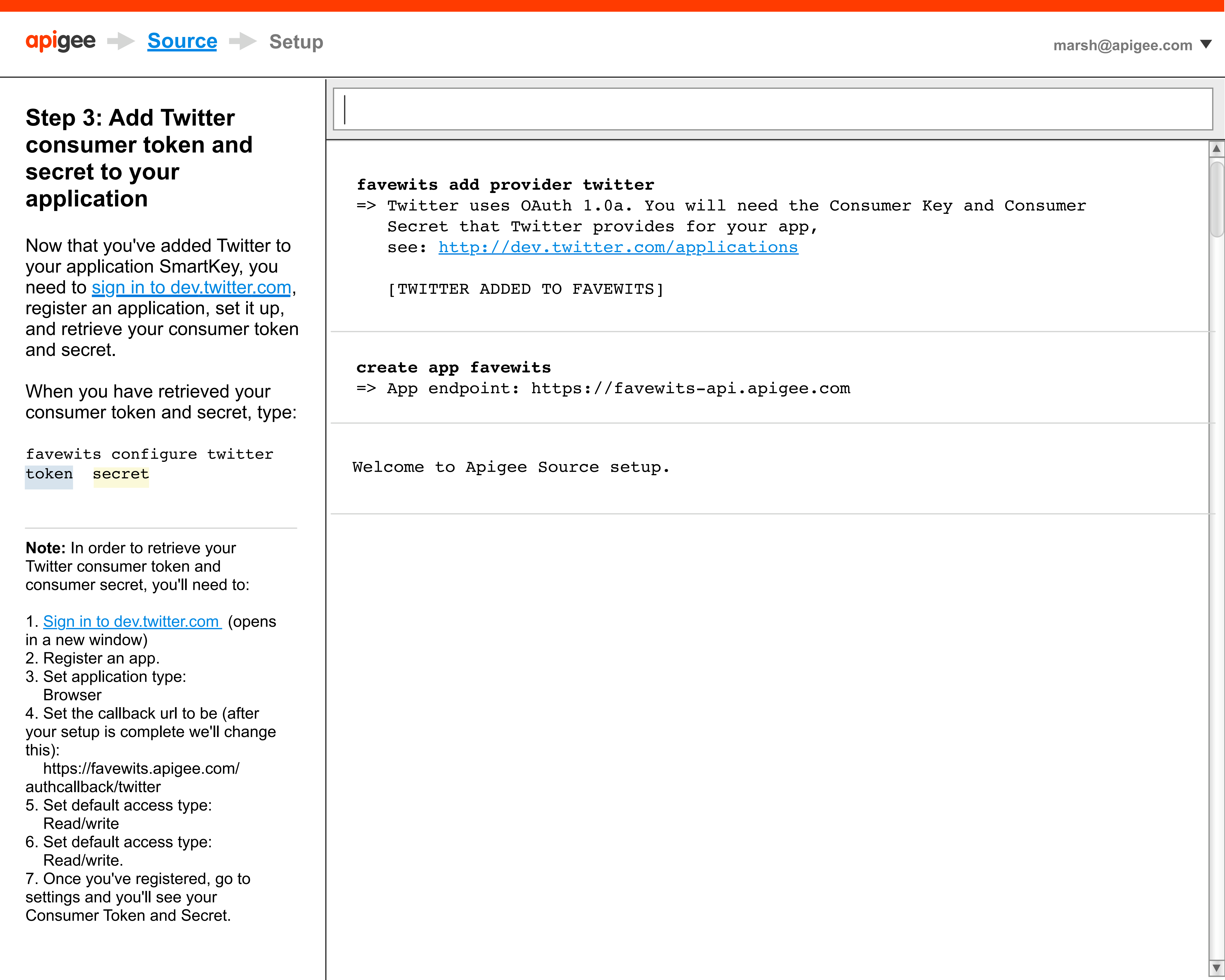
OAuth 1.0 integrations require more work

Consumer Key

Asbbashdje283324874893278400

Consumer Secret

Asbbashdje283324874893278400



Step 3: Add Twitter consumer token and secret to your application

Now that you've added Twitter to your application SmartKey, you need to sign in to dev.twitter.com, register an application, set it up, and retrieve your consumer token and secret.

When you have retrieved your consumer token and secret, type:

```
favewits configure twitter
token  secret
```

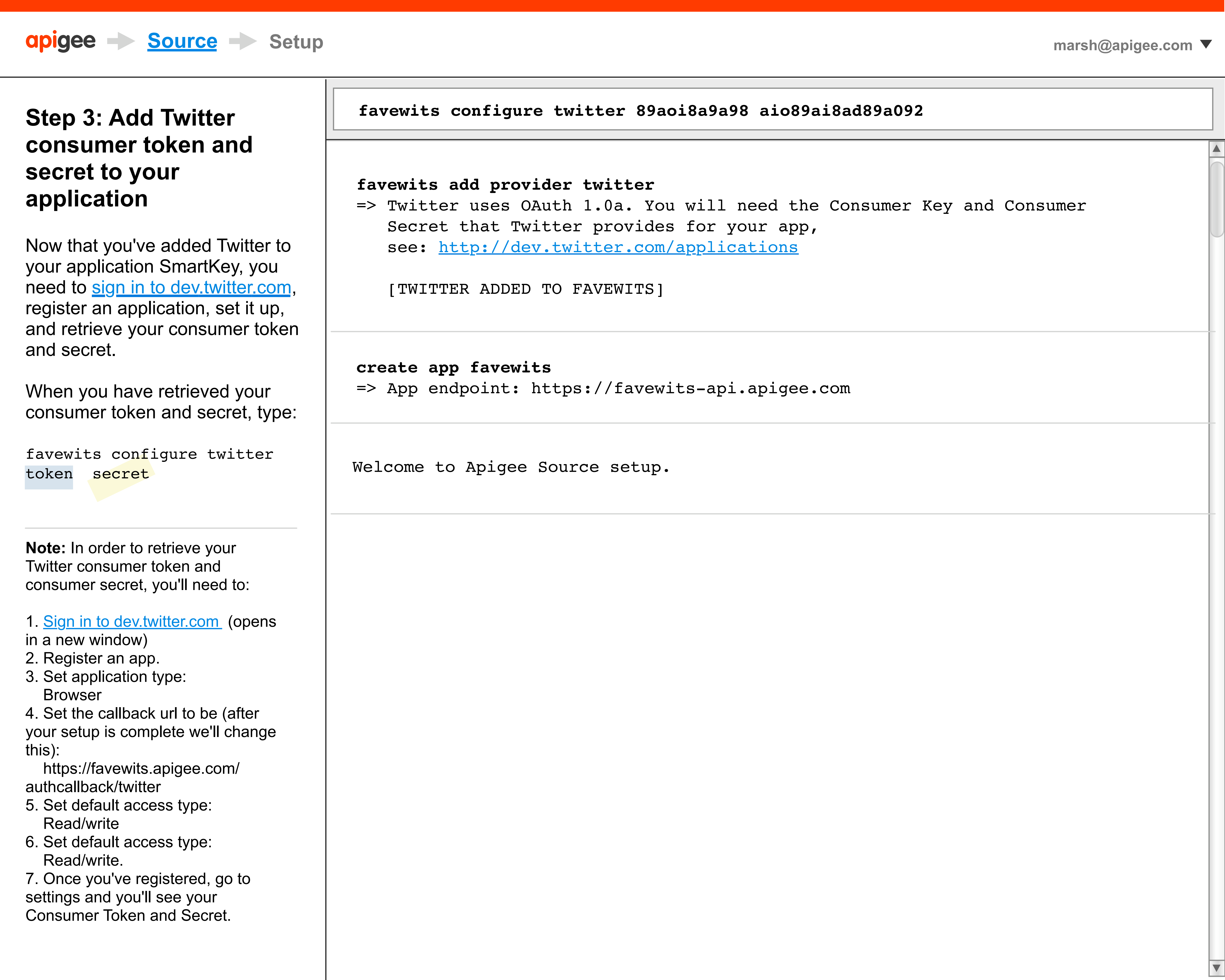
Note: In order to retrieve your Twitter consumer token and consumer secret, you'll need to:

1. Sign in to dev.twitter.com (opens in a new window)
2. Register an app.
3. Set application type:
   Browser
4. Set the callback url to be (after your setup is complete we'll change this):
   https://favewits.apigee.com/authcallback/twitter
5. Set default access type:
   Read/write
6. Set default access type:
   Read/write.
7. Once you've registered, go to settings and you'll see your Consumer Token and Secret.

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 3: Add Twitter consumer token and secret to your application

Now that you've added Twitter to your application SmartKey, you need to sign in to dev.twitter.com, register an application, set it up, and retrieve your consumer token and secret.

When you have retrieved your consumer token and secret, type:

```
favewits configure twitter
token  secret
```

Note: In order to retrieve your Twitter consumer token and consumer secret, you'll need to:

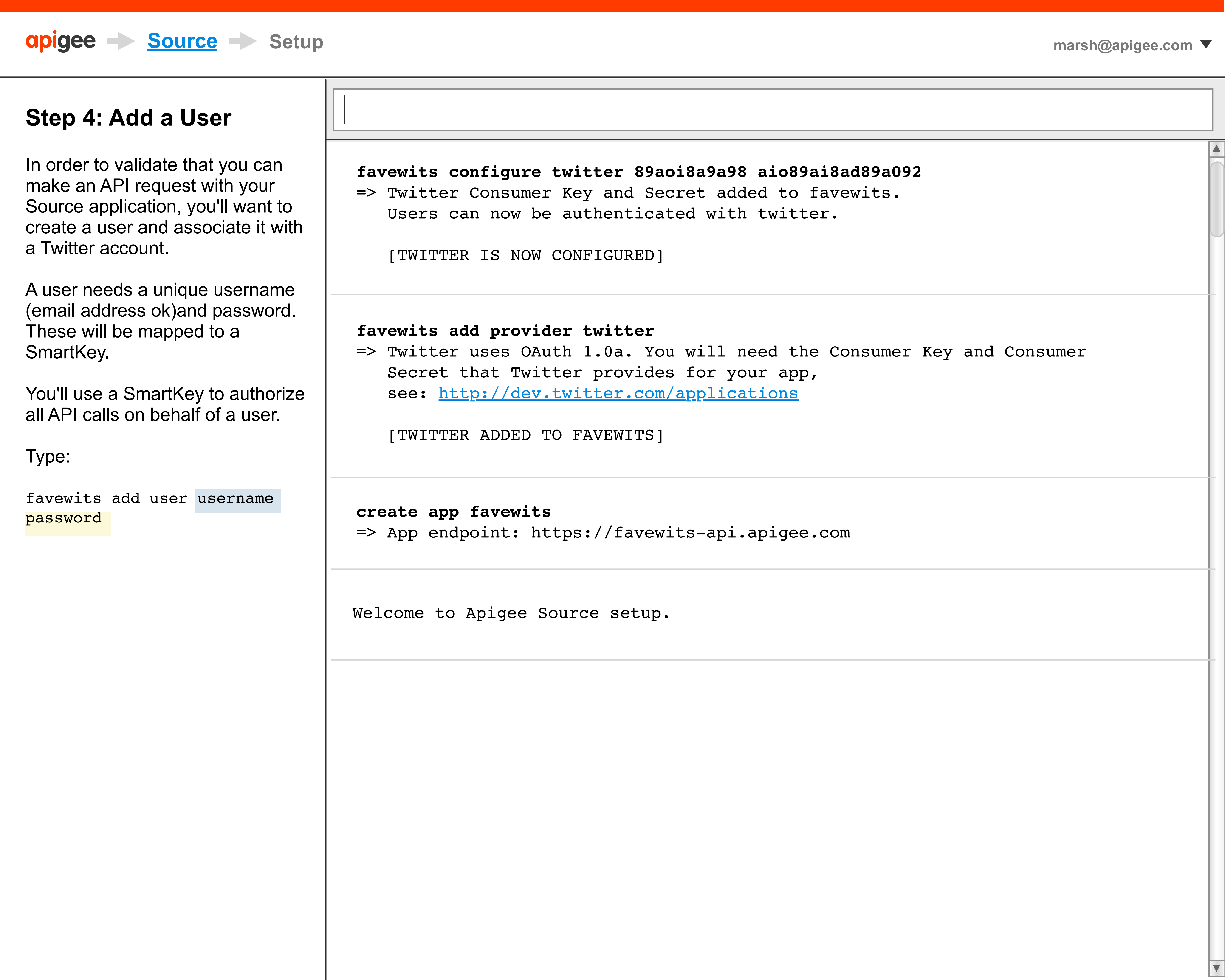
1. Sign in to dev.twitter.com (opens in a new window)
2. Register an app.
3. Set application type:
   Browser
4. Set the callback url to be (after your setup is complete we'll change this):
   https://favewits.apigee.com/authcallback/twitter
5. Set default access type:
   Read/write
6. Set default access type:
   Read/write.
7. Once you've registered, go to settings and you'll see your Consumer Token and Secret.

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 4: Add a User

In order to validate that you can make an API request with your Source application, you'll want to create a user and associate it with a Twitter account.

A user needs a unique username (email address ok)and password. These will be mapped to a SmartKey.

You'll use a SmartKey to authorize all API calls on behalf of a user.

Type:

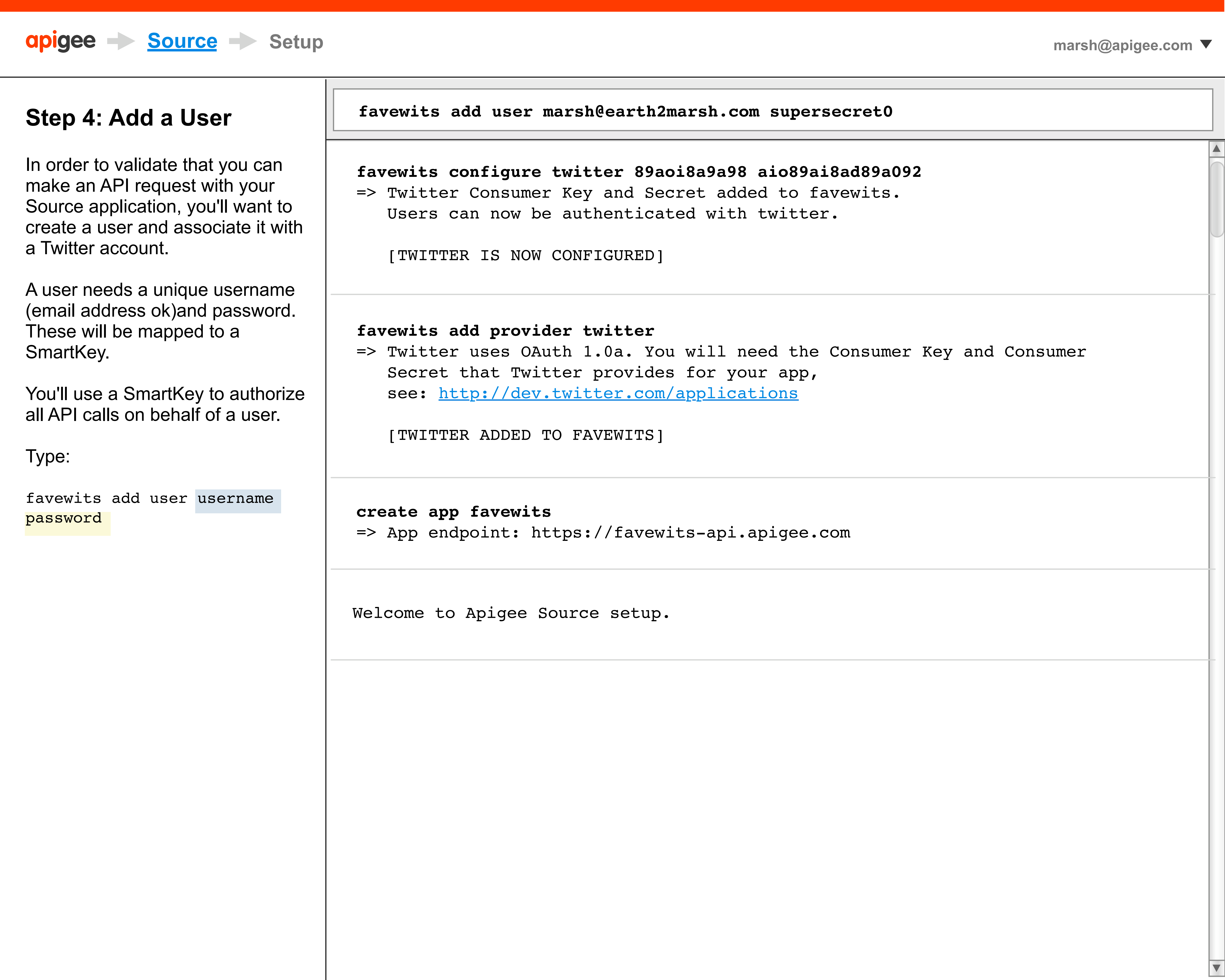
```
favewits add user username
password
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 4: Add a User

In order to validate that you can make an API request with your Source application, you'll want to create a user and associate it with a Twitter account.

A user needs a unique username (email address ok)and password. These will be mapped to a SmartKey.

You'll use a SmartKey to authorize all API calls on behalf of a user.

Type:

```
favewits add user username
password
```

```
favewits add user marsh@earth2marsh.com supersecret0
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```

apigee → Source → Setup   marsh@apigee.com ▼

Step 5: Generate an Authorization URL

In order to validate that you can make an API request with your SmartKey, you'll want to sign in with Twitter and authorize favewits as a trusted application.

Type:

```
favewits get twitter
authorization for smartkey
220961tty9089
```

and get a URL to authenticate with Twitter as an end user.

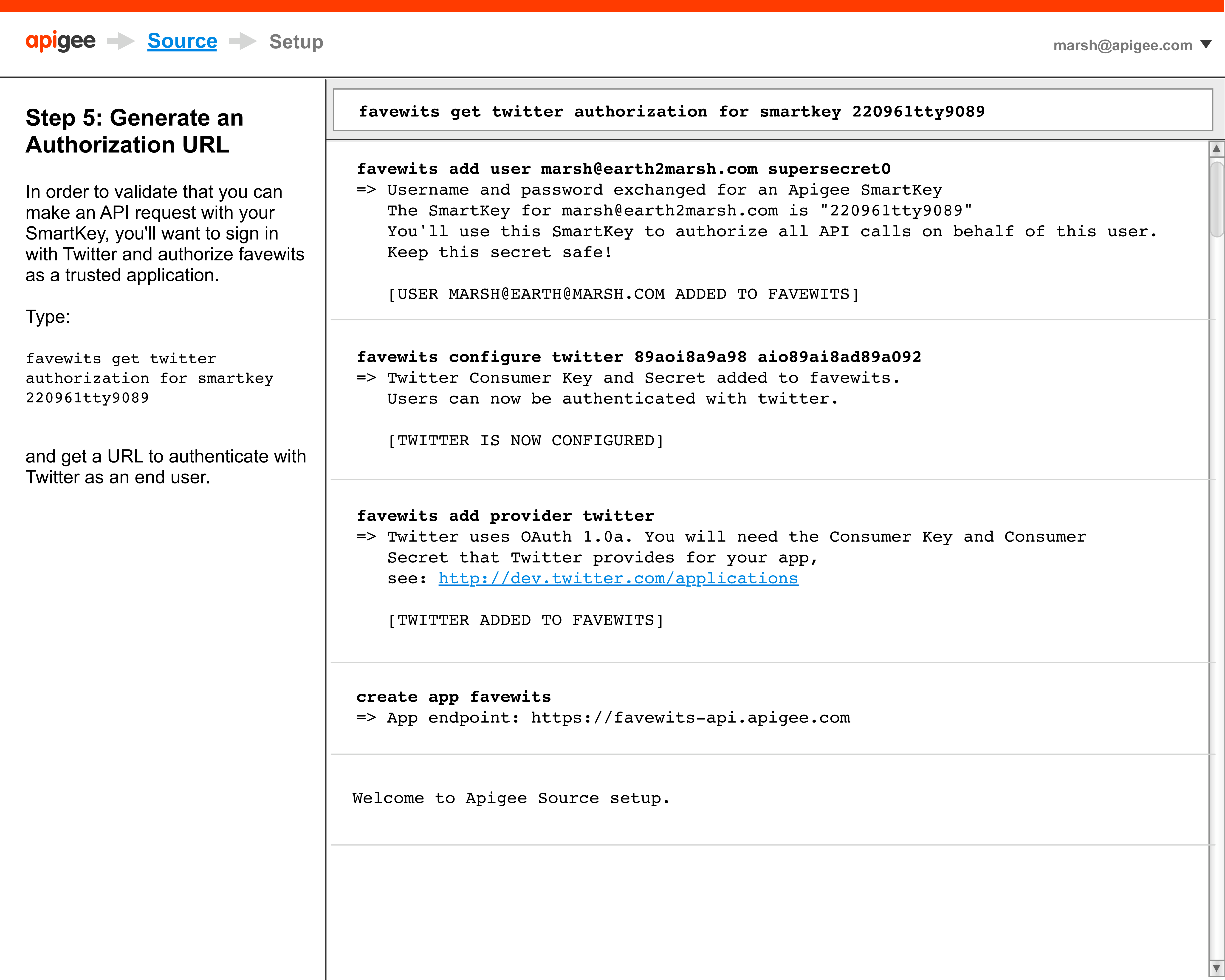
```
favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



apigee → Source → Setup

marsh@apigee.com ▼

Step 5: Generate an Authorization URL

In order to validate that you can make an API request with your SmartKey, you'll want to sign in with Twitter and authorize favewits as a trusted application.

Type:

```
favewits get twitter
authorization for smartkey
220961tty9089
```

and get a URL to authenticate with Twitter as an end user.

```
favewits get twitter authorization for smartkey 220961tty9089
```

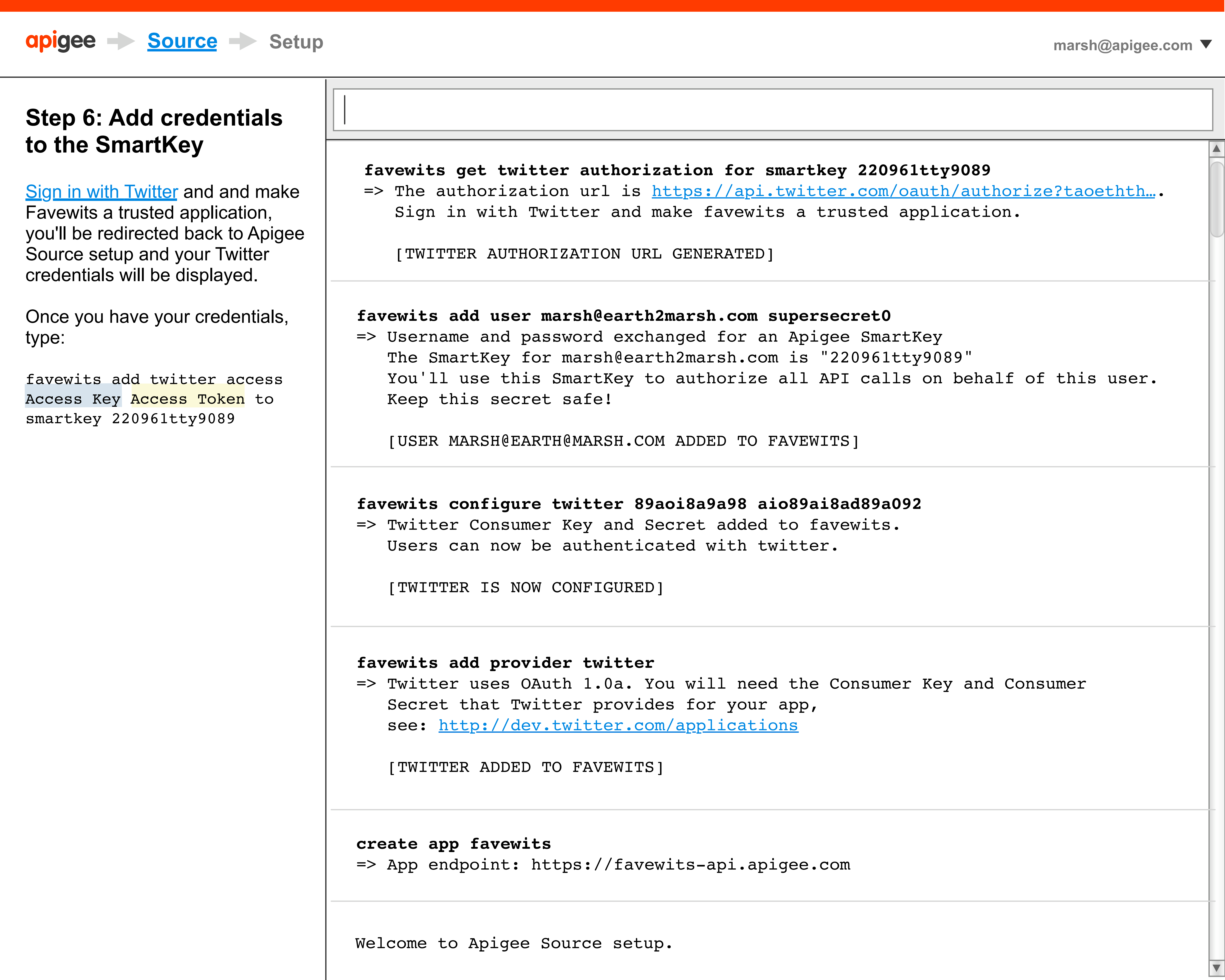
```
favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



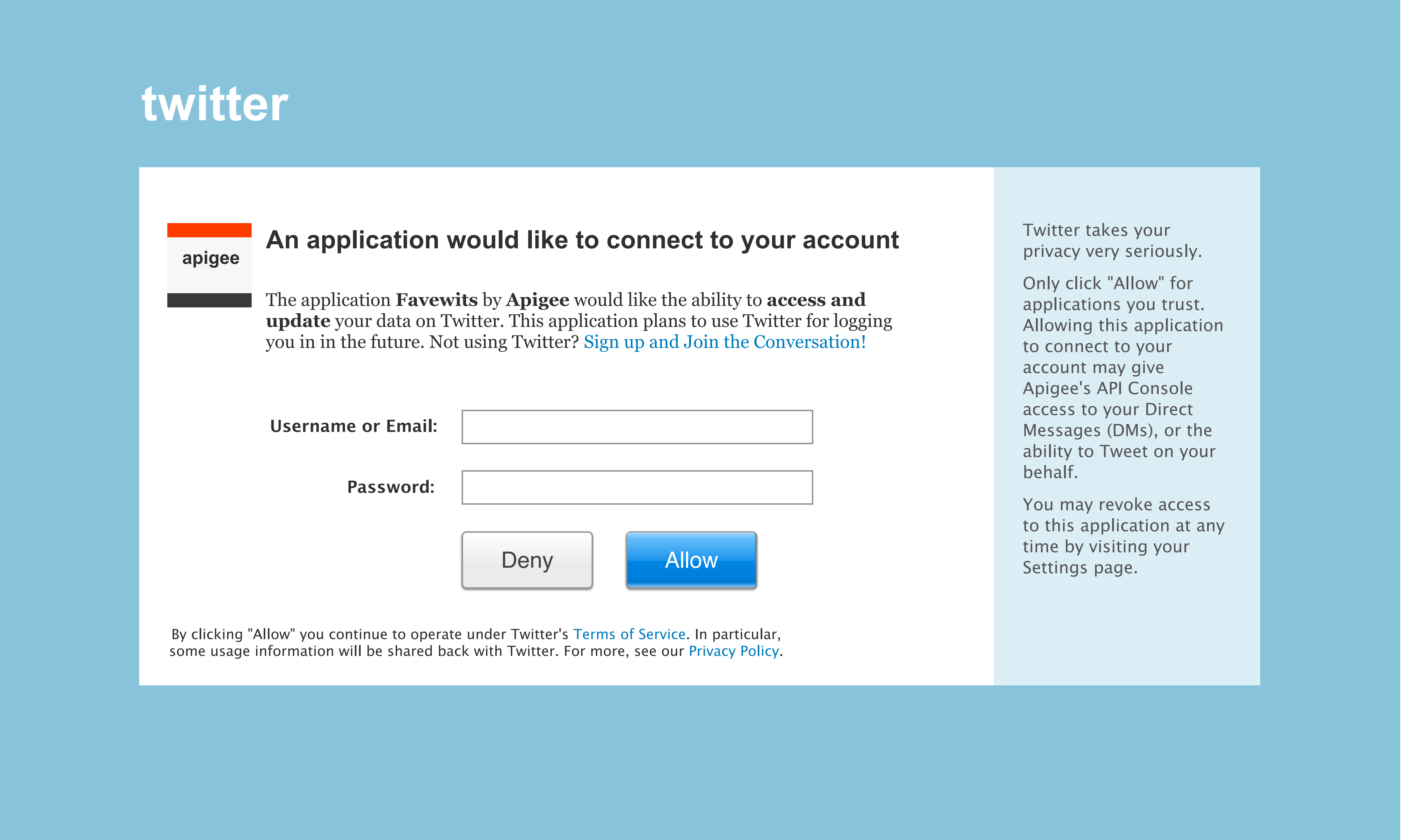
Step 6: Add credentials to the SmartKey

Sign in with Twitter and and make Favewits a trusted application, you'll be redirected back to Apigee Source setup and your Twitter credentials will be displayed.

Once you have your credentials, type:

```
favewits add twitter access
Access Key Access Token to
smartkey 220961tty9089
```

```
favewits get twitter authorization for smartkey 220961tty9089
=> The authorization url is https://api.twitter.com/oauth/authorize?taoeth th….
   Sign in with Twitter and make favewits a trusted application.

   [TWITTER AUTHORIZATION URL GENERATED]

favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]

favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]

favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]

create app favewits
=> App endpoint: https://favewits-api.apigee.com

Welcome to Apigee Source setup.
```



An application would like to connect to your account

The application **Favewits** by **Apigee** would like the ability to **access and update** your data on Twitter. This application plans to use Twitter for logging you in in the future. Not using Twitter? Sign up and Join the Conversation!

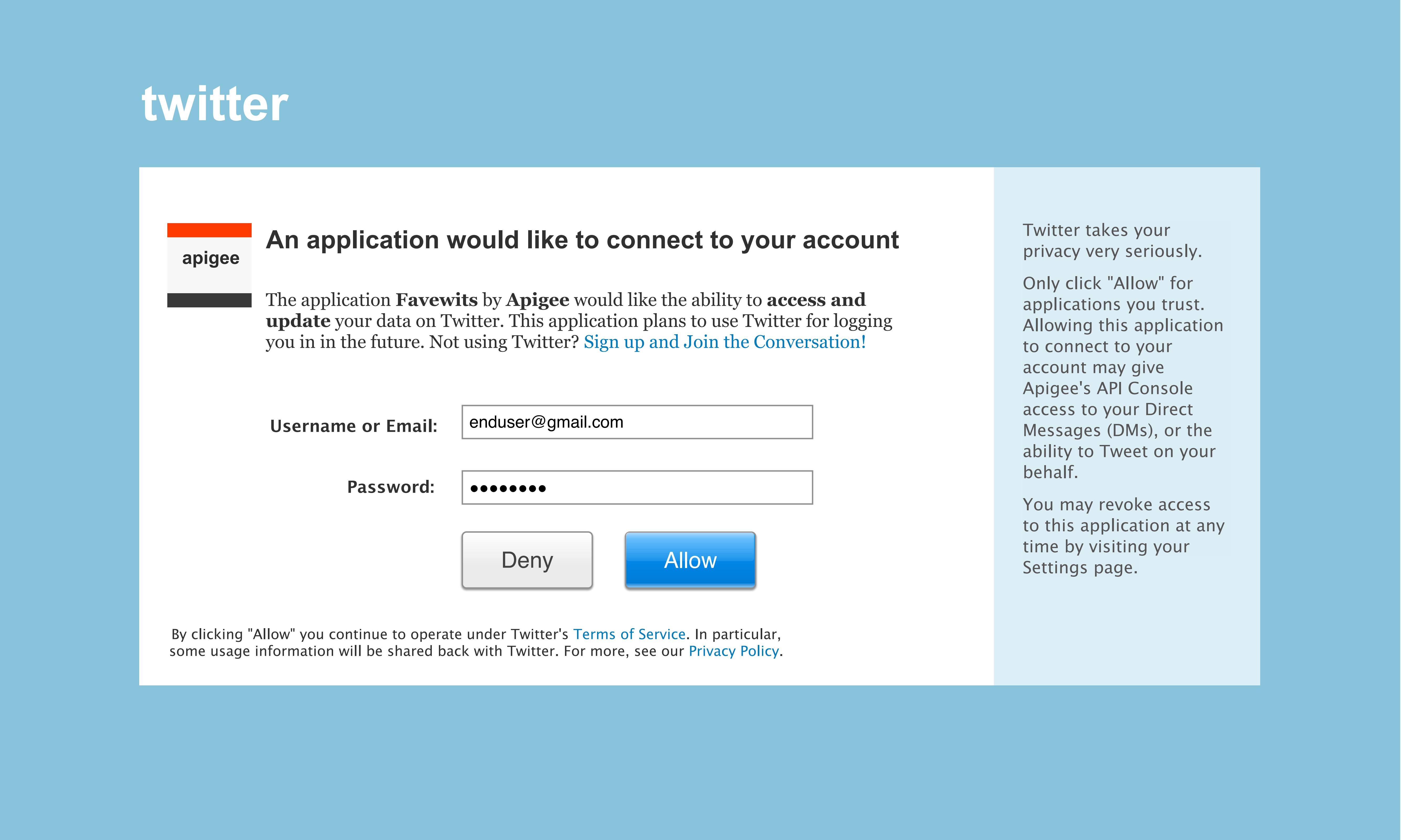
Username or Email:	
Password:	

By clicking "Allow" you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Twitter takes your privacy very seriously.

Only click "Allow" for applications you trust. Allowing this application to connect to your account may give Apigee's API Console access to your Direct Messages (DMs), or the ability to Tweet on your behalf.

You may revoke access to this application at any time by visiting your Settings page.



twitter

apigee

An application would like to connect to your account

The application **Favewits** by **Apigee** would like the ability to **access and update** your data on Twitter. This application plans to use Twitter for logging you in in the future. Not using Twitter? Sign up and Join the Conversation!

Username or Email: enduser@gmail.com

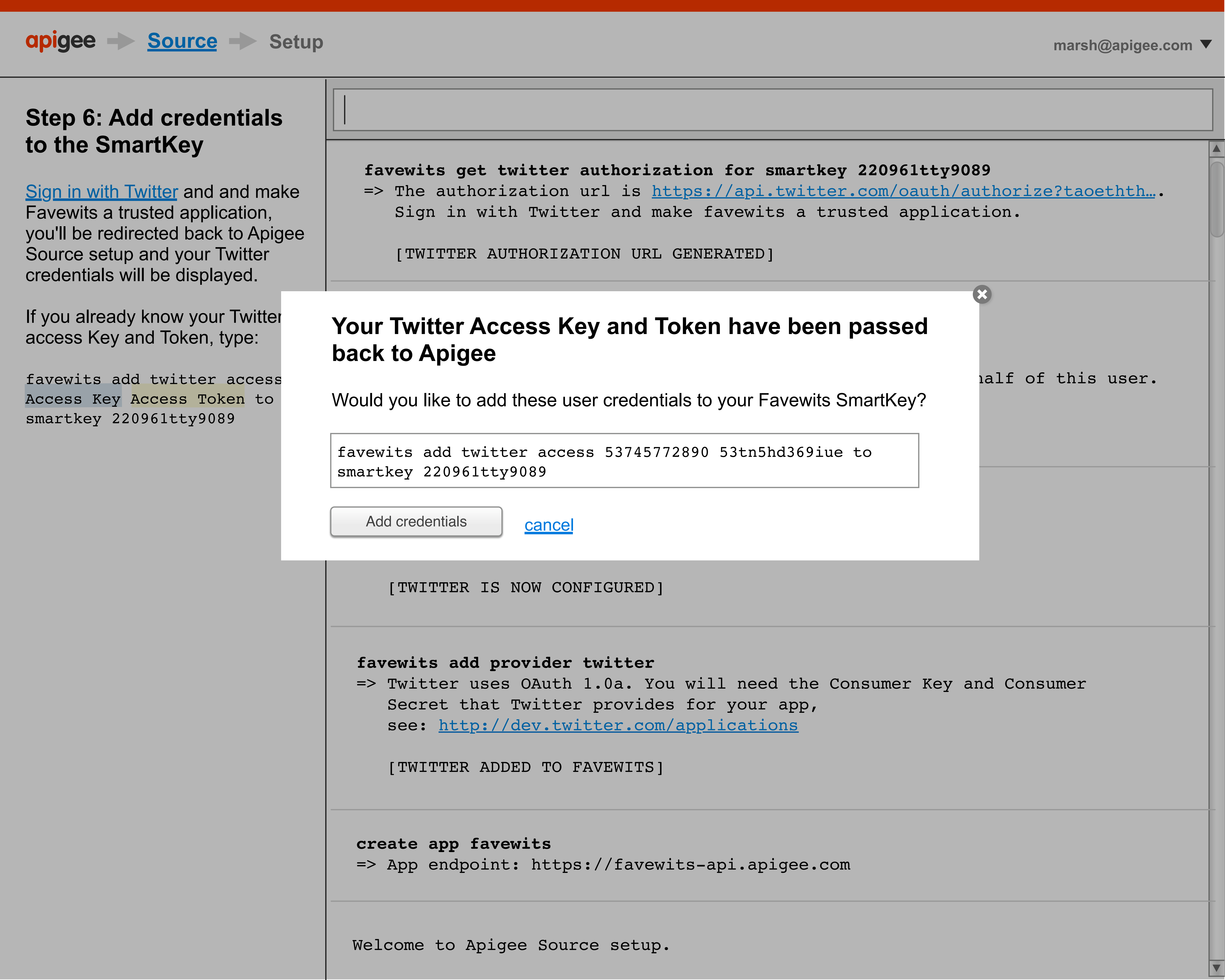
Password: ●●●●●●●●●

By clicking "Allow" you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Twitter takes your privacy very seriously.

Only click "Allow" for applications you trust. Allowing this application to connect to your account may give Apigee's API Console access to your Direct Messages (DMs), or the ability to Tweet on your behalf.

You may revoke access to this application at any time by visiting your Settings page.



Step 6: Add credentials to the SmartKey

Sign in with Twitter and and make Favewits a trusted application, you'll be redirected back to Apigee Source setup and your Twitter credentials will be displayed.

If you already know your Twitter access Key and Token, type:

```
favewits add twitter access
Access Key Access Token to
smartkey 220961tty9089
```

```
favewits get twitter authorization for smartkey 220961tty9089
=> The authorization url is https://api.twitter.com/oauth/authorize?taoethth….
   Sign in with Twitter and make favewits a trusted application.

   [TWITTER AUTHORIZATION URL GENERATED]
```

```
alf of this user.
```

Your Twitter Access Key and Token have been passed back to Apigee

Would you like to add these user credentials to your Favewits SmartKey?

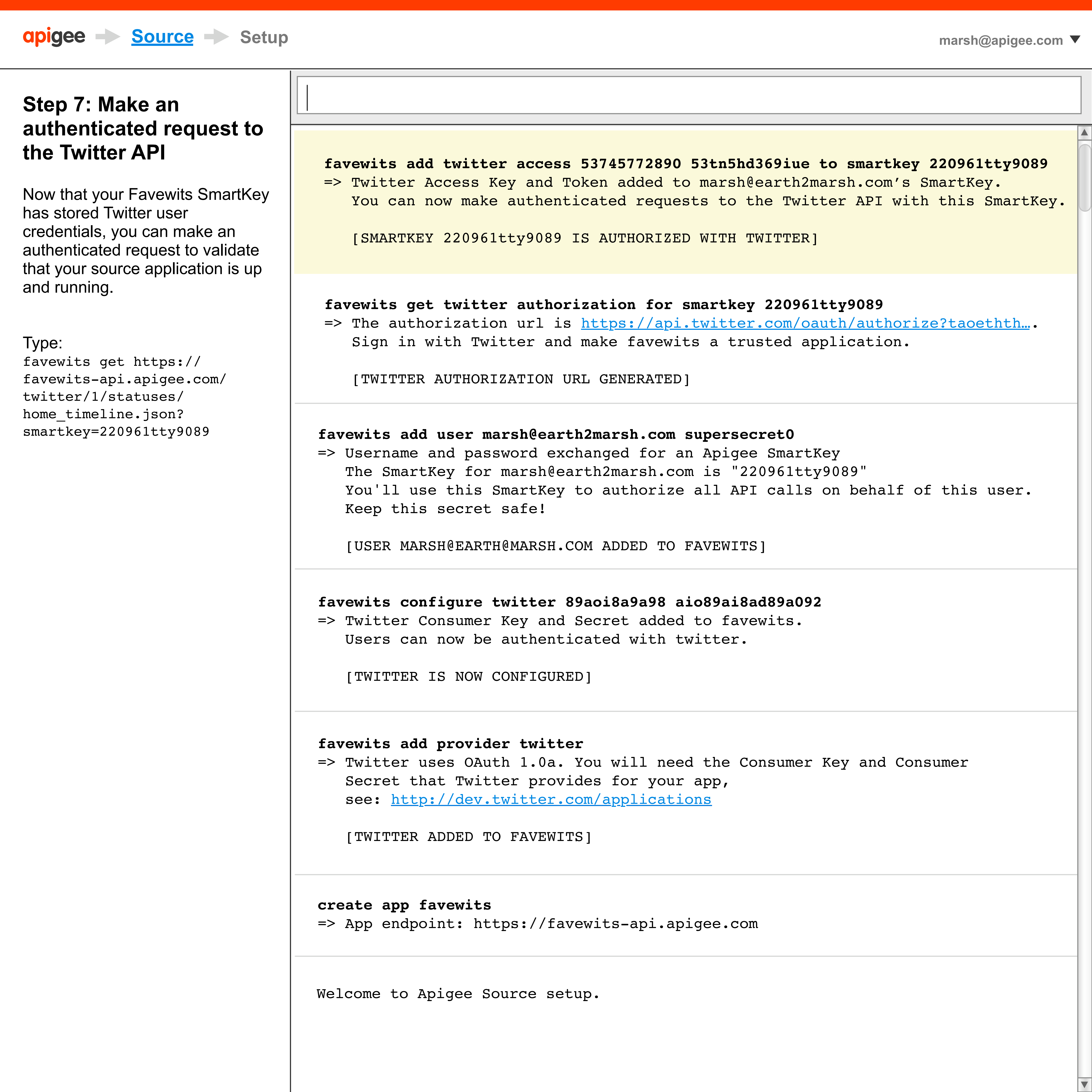
```
favewits add twitter access 53745772890 53tn5hd369iue to
smartkey 220961tty9089
```

```
   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 7: Make an authenticated request to the Twitter API

Now that your Favewits SmartKey has stored Twitter user credentials, you can make an authenticated request to validate that your source application is up and running.

Type:

```
favewits get https://
favewits-api.apigee.com/
twitter/1/statuses/
home_timeline.json?
smartkey=220961tty9089
```

```
favewits add twitter access 53745772890 53tn5hd369iue to smartkey 220961tty9089
=> Twitter Access Key and Token added to marsh@earth2marsh.com's SmartKey.
   You can now make authenticated requests to the Twitter API with this SmartKey.

   [SMARTKEY 220961tty9089 IS AUTHORIZED WITH TWITTER]
```

```
favewits get twitter authorization for smartkey 220961tty9089
=> The authorization url is https://api.twitter.com/oauth/authorize?taoeththth….
   Sign in with Twitter and make favewits a trusted application.

   [TWITTER AUTHORIZATION URL GENERATED]
```

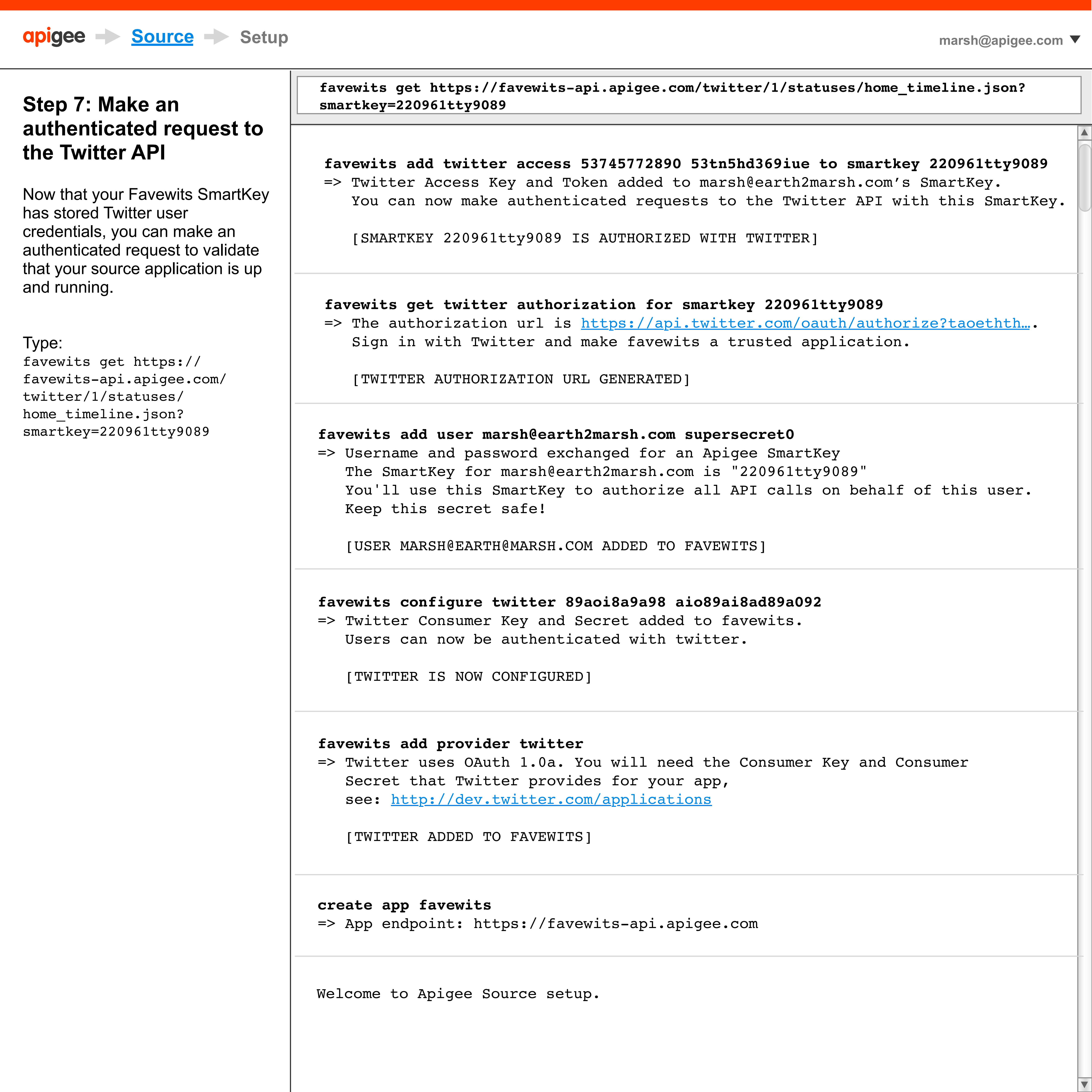
```
favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Step 7: Make an authenticated request to the Twitter API

Now that your Favewits SmartKey has stored Twitter user credentials, you can make an authenticated request to validate that your source application is up and running.

Type:

```
favewits get https://
favewits-api.apigee.com/
twitter/1/statuses/
home_timeline.json?
smartkey=220961tty9089
```

```
favewits get https://favewits-api.apigee.com/twitter/1/statuses/home_timeline.json?
smartkey=220961tty9089
```

```
favewits add twitter access 53745772890 53tn5hd369iue to smartkey 220961tty9089
=> Twitter Access Key and Token added to marsh@earth2marsh.com’s SmartKey.
   You can now make authenticated requests to the Twitter API with this SmartKey.

   [SMARTKEY 220961tty9089 IS AUTHORIZED WITH TWITTER]
```

```
favewits get twitter authorization for smartkey 220961tty9089
=> The authorization url is https://api.twitter.com/oauth/authorize?taoeththth….
   Sign in with Twitter and make favewits a trusted application.

   [TWITTER AUTHORIZATION URL GENERATED]
```

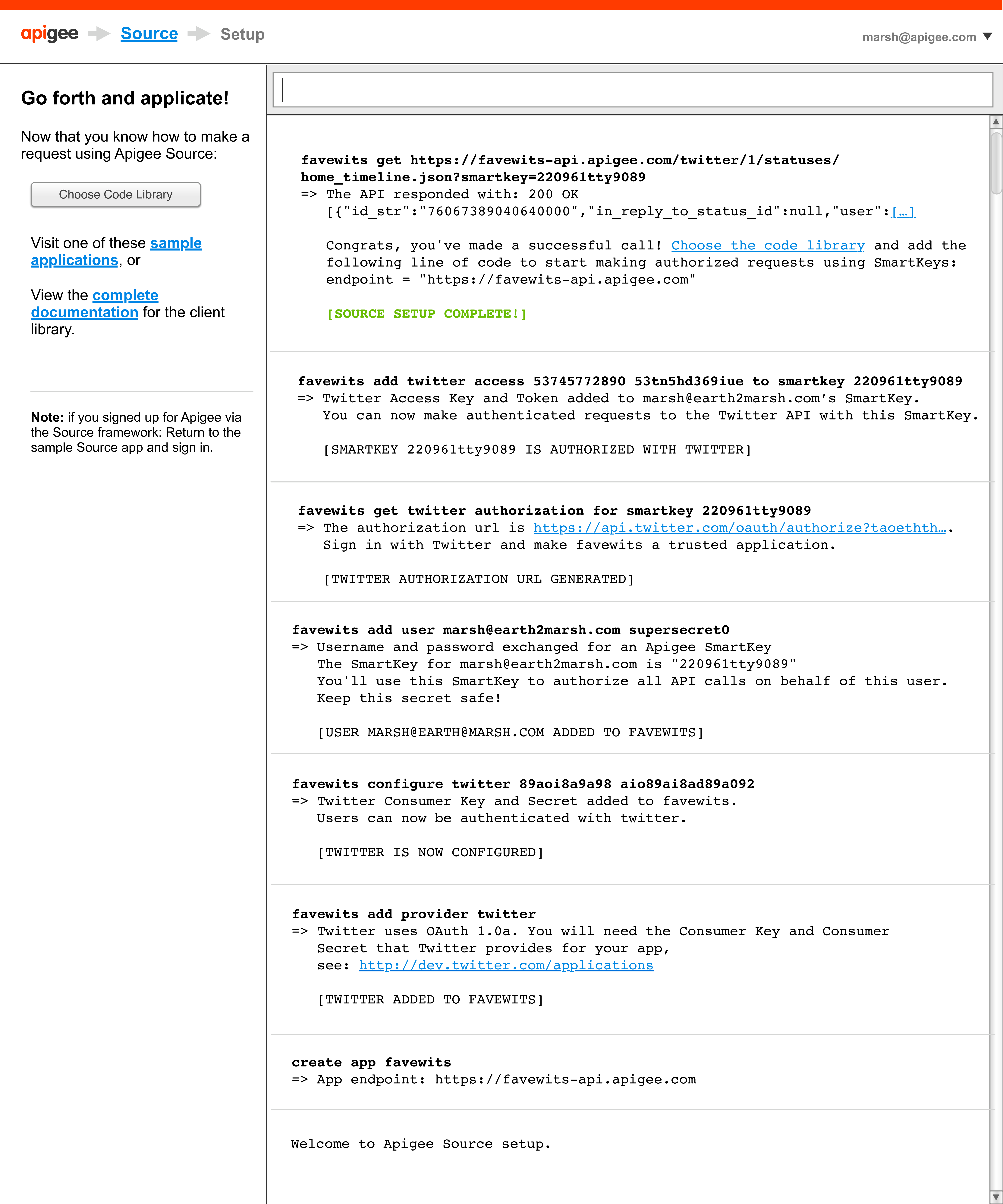
```
favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



Go forth and applicate!

Now that you know how to make a request using Apigee Source:

Visit one of these sample applications, or

View the complete documentation for the client library.

Note: if you signed up for Apigee via the Source framework: Return to the sample Source app and sign in.

```
favewits get https://favewits-api.apigee.com/twitter/1/statuses/
home_timeline.json?smartkey=220961tty9089
=> The API responded with: 200 OK
   [{"id_str":"76067389040640000","in_reply_to_status_id":null,"user":[…]

   Congrats, you've made a successful call! Choose the code library and add the
   following line of code to start making authorized requests using SmartKeys:
   endpoint = "https://favewits-api.apigee.com"

   [SOURCE SETUP COMPLETE!]
```

```
favewits add twitter access 53745772890 53tn5hd369iue to smartkey 220961tty9089
=> Twitter Access Key and Token added to marsh@earth2marsh.com’s SmartKey.
   You can now make authenticated requests to the Twitter API with this SmartKey.

   [SMARTKEY 220961tty9089 IS AUTHORIZED WITH TWITTER]
```

```
favewits get twitter authorization for smartkey 220961tty9089
=> The authorization url is https://api.twitter.com/oauth/authorize?taoeththt….
   Sign in with Twitter and make favewits a trusted application.

   [TWITTER AUTHORIZATION URL GENERATED]
```

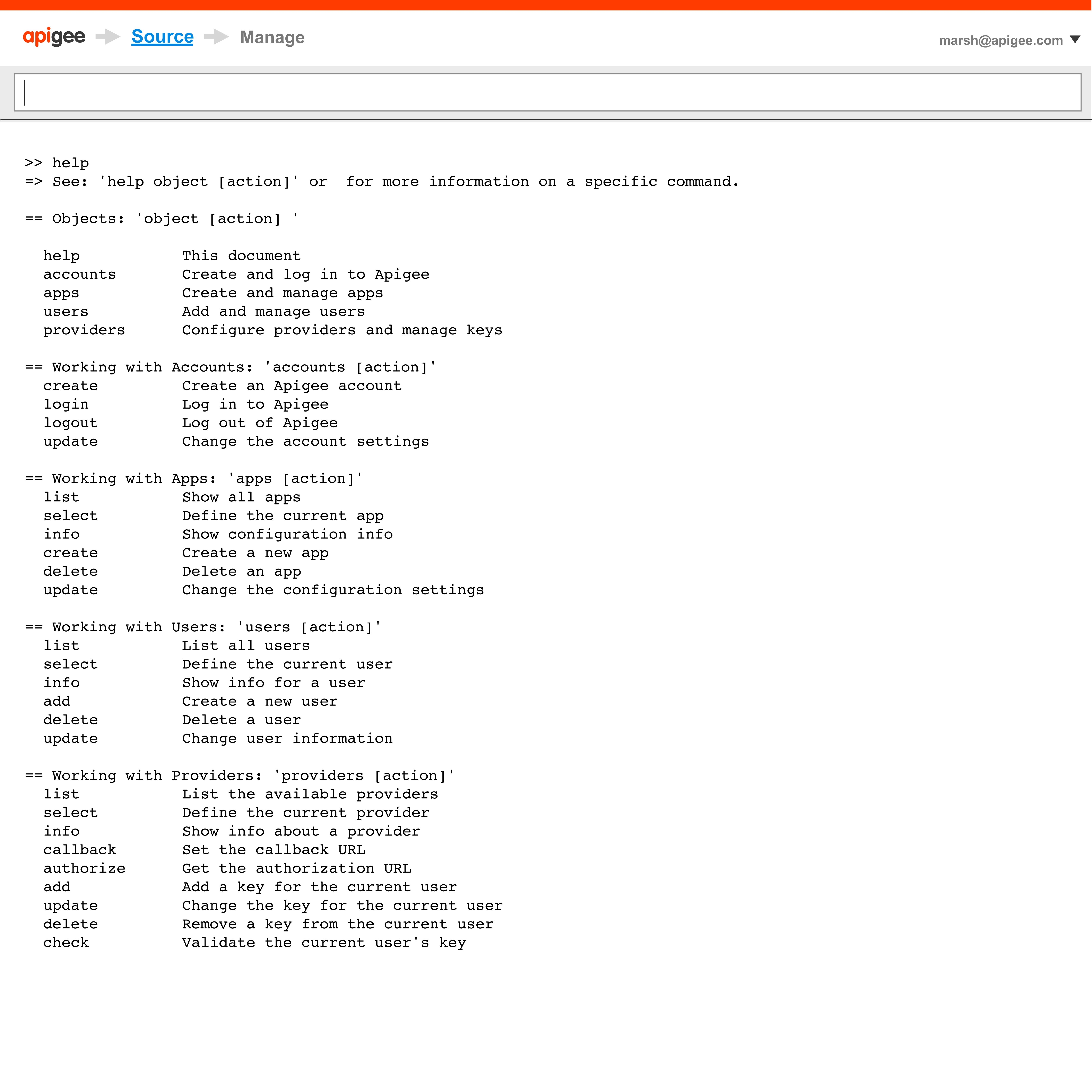
```
favewits add user marsh@earth2marsh.com supersecret0
=> Username and password exchanged for an Apigee SmartKey
   The SmartKey for marsh@earth2marsh.com is "220961tty9089"
   You'll use this SmartKey to authorize all API calls on behalf of this user.
   Keep this secret safe!

   [USER MARSH@EARTH@MARSH.COM ADDED TO FAVEWITS]
```

```
favewits configure twitter 89aoi8a9a98 aio89ai8ad89a092
=> Twitter Consumer Key and Secret added to favewits.
   Users can now be authenticated with twitter.

   [TWITTER IS NOW CONFIGURED]
```

```
favewits add provider twitter
=> Twitter uses OAuth 1.0a. You will need the Consumer Key and Consumer
   Secret that Twitter provides for your app,
   see: http://dev.twitter.com/applications

   [TWITTER ADDED TO FAVEWITS]
```

```
create app favewits
=> App endpoint: https://favewits-api.apigee.com
```

```
Welcome to Apigee Source setup.
```



```
>> help
=> See: 'help object [action]' or  for more information on a specific command.

== Objects: 'object [action] '

  help            This document
  accounts        Create and log in to Apigee
  apps            Create and manage apps
  users           Add and manage users
  providers       Configure providers and manage keys

== Working with Accounts: 'accounts [action]'
  create          Create an Apigee account
  login           Log in to Apigee
  logout          Log out of Apigee
  update          Change the account settings

== Working with Apps: 'apps [action]'
  list            Show all apps
  select          Define the current app
  info            Show configuation info
  create          Create a new app
  delete          Delete an app
  update          Change the configuration settings

== Working with Users: 'users [action]'
  list            List all users
  select          Define the current user
  info            Show info for a user
  add             Create a new user
  delete          Delete a user
  update          Change user information

== Working with Providers: 'providers [action]'
  list            List the available providers
  select          Define the current provider
  info            Show info about a provider
  callback        Set the callback URL
  authorize       Get the authorization URL
  add             Add a key for the current user
  update          Change the key for the current user
  delete          Remove a key from the current user
  check           Validate the current user's key
```

apigee → Source → Code Libraries

marsh@apigee.com

Code Libraries

Use these libraries to get quickly started quickly in your language of choice.

Javascript

The official JQuery client library for Apigee Source, available as a plug-in through github.

Download

View on Github

Coming soon

Objective-C SDK

Apigee's Source Objective-C SDK is a code library that you can drop into any iOS app to get working with APIs faster. It consists of two related pieces: a UI SDK, which provides the interface elements for displaying API data within your app, and a lower-level Data SDK, which helps your app communicate with the Apigee API and on which the UI SDK depends. The Data SDK can be used by itself in Mac OS apps.

Ruby

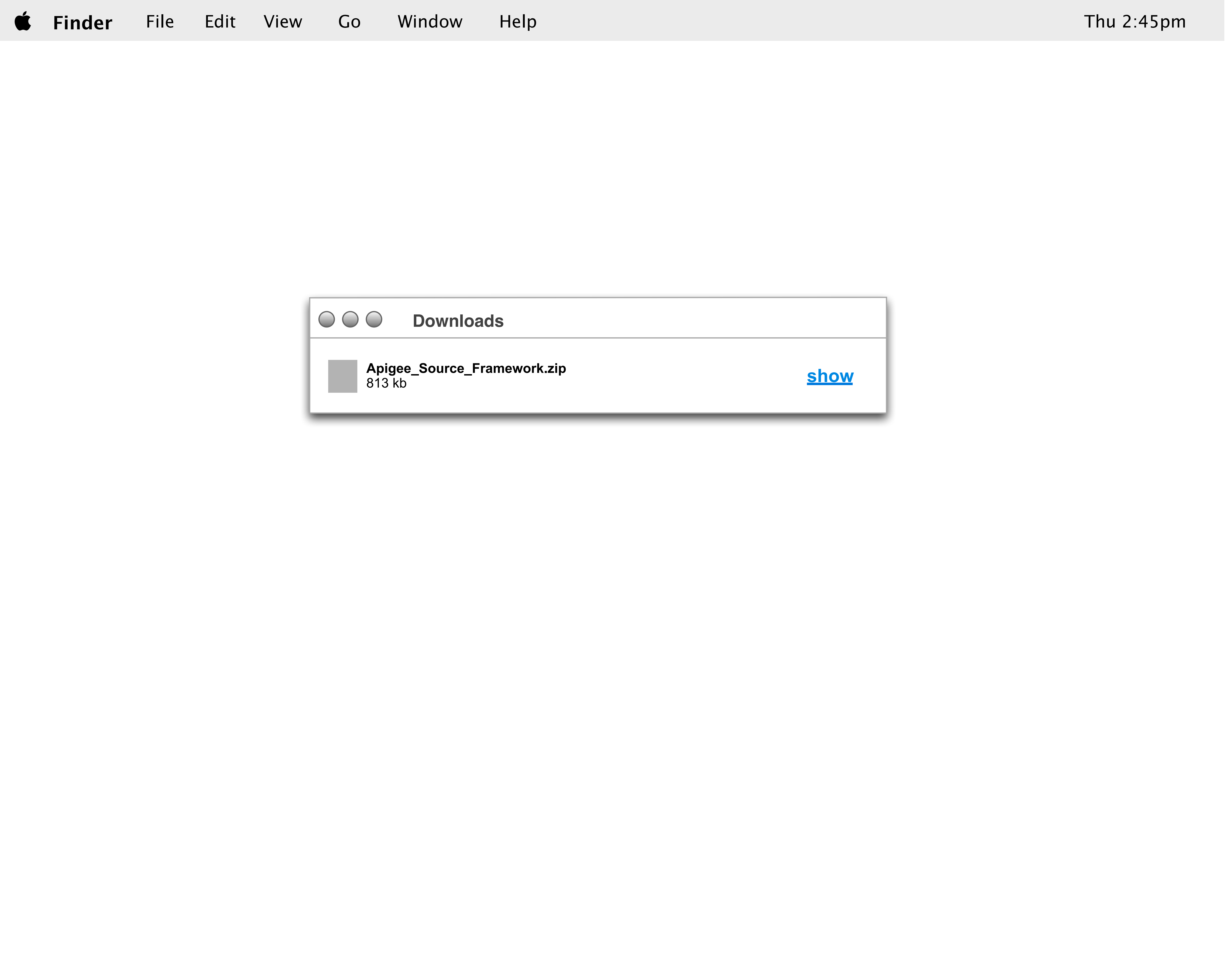
The official Ruby gem for Apigee Source, available on rubygems and github. To install, gem install apigeesource

Team up with us

We're looking to develop as many code libraries for Apigee Source as we can. Are you a guru in a language like **PHP**, **Ruby, or Python**?

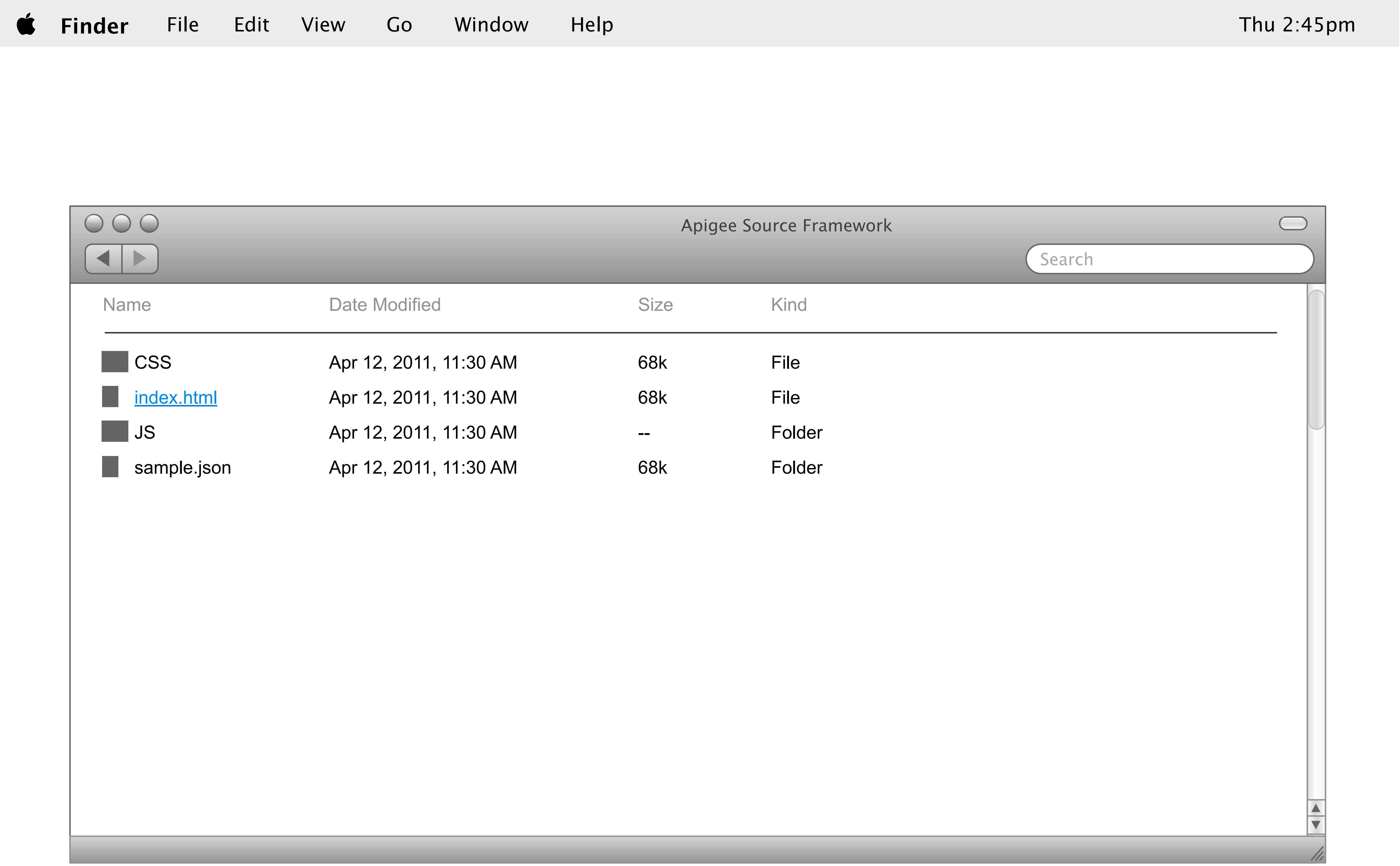
Do you have experience creating code frameworks? We'd love to hear from you (we'll pay handsomely).

Drop us a line!

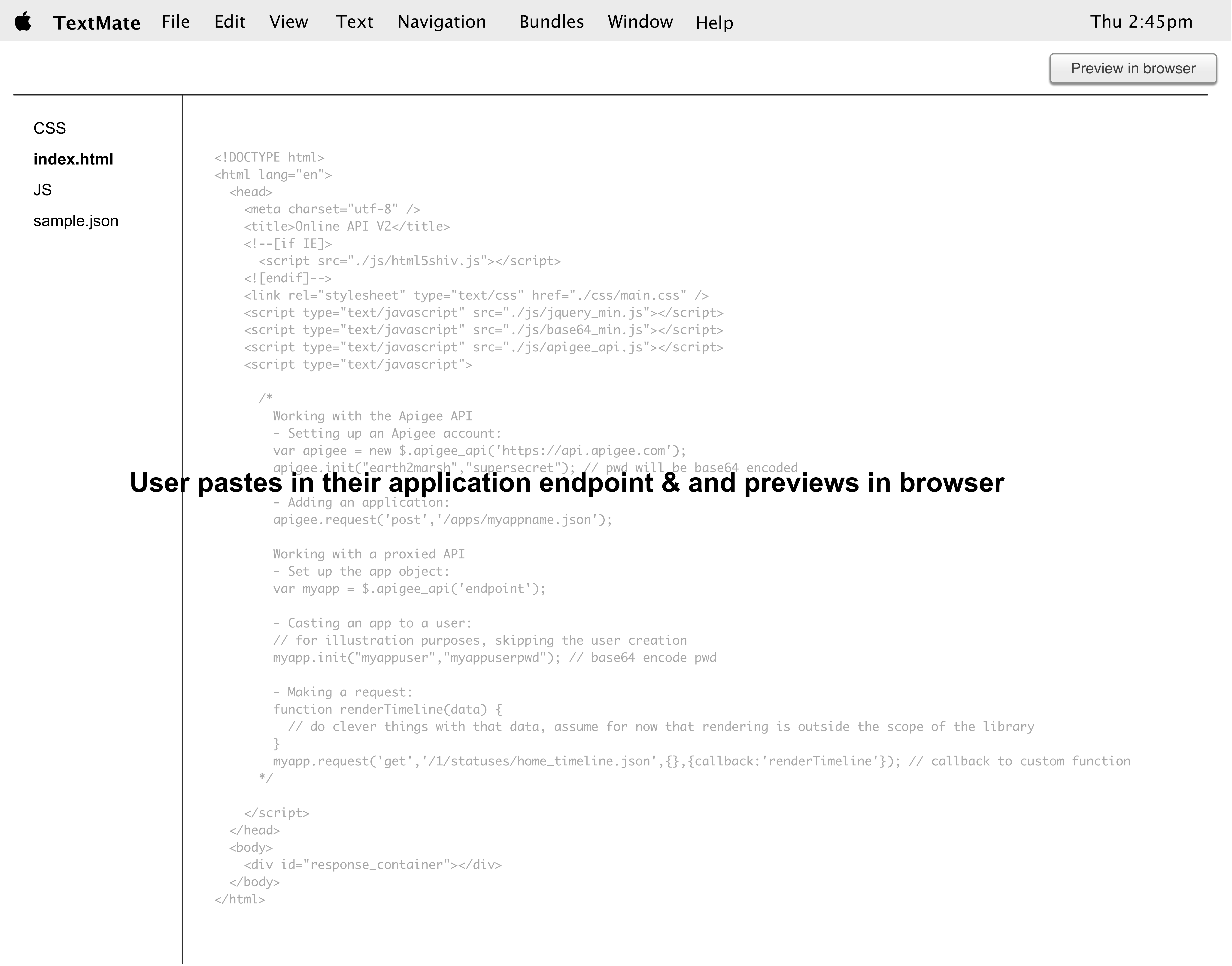


Downloads

Apigee_Source_Framework.zip
813 kb

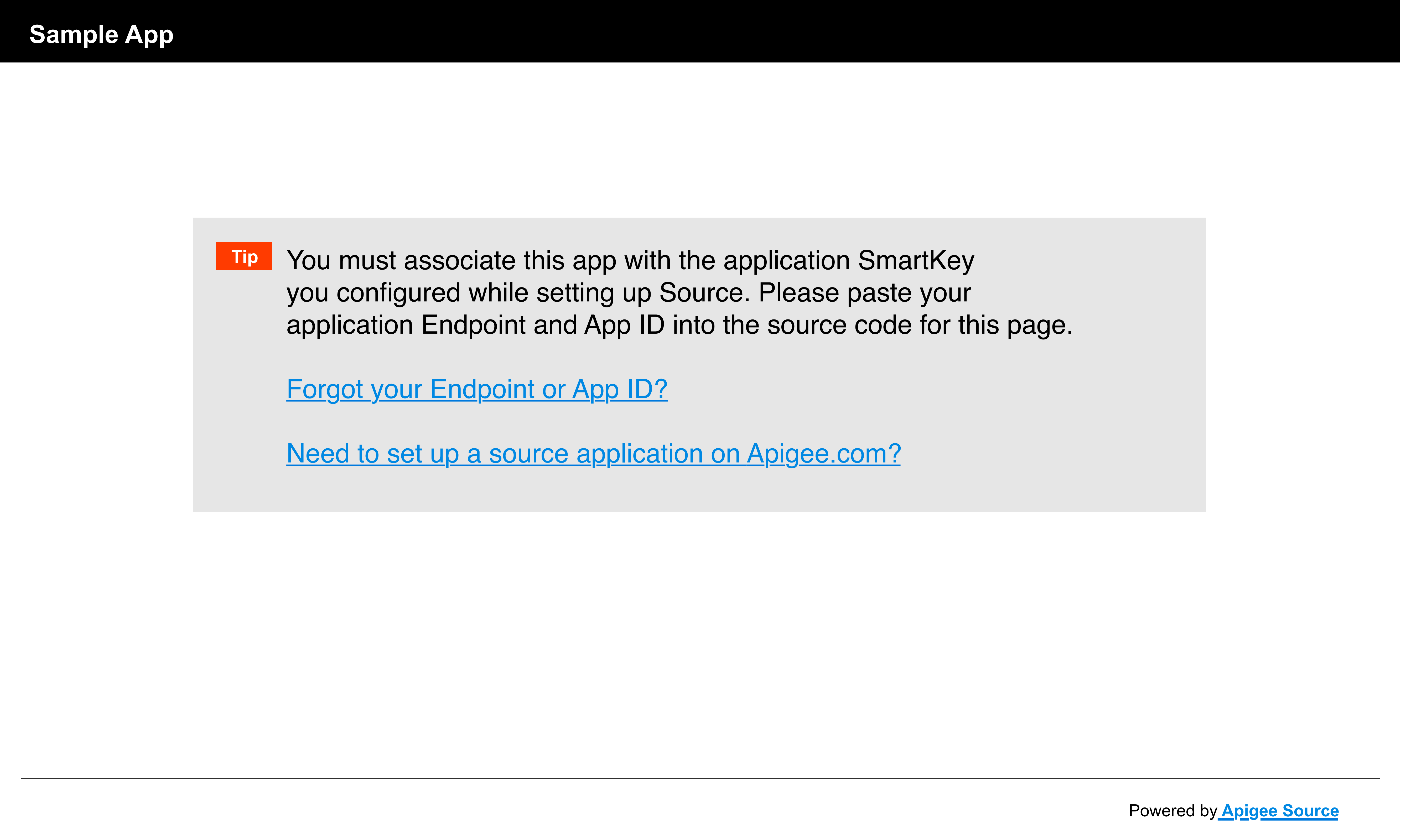


Name	Date Modified	Size	Kind
■ CSS	Apr 12, 2011, 11:30 AM	68k	File
■ index.html	Apr 12, 2011, 11:30 AM	68k	File
■ JS	Apr 12, 2011, 11:30 AM	--	Folder
■ sample.json	Apr 12, 2011, 11:30 AM	68k	Folder



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Online API V2</title>
    <!--[if IE]>
      <script src="./js/html5shiv.js"></script>
    <![endif]-->
    <link rel="stylesheet" type="text/css" href="./css/main.css" />
    <script type="text/javascript" src="./js/jquery_min.js"></script>
    <script type="text/javascript" src="./js/base64_min.js"></script>
    <script type="text/javascript" src="./js/apigee_api.js"></script>
    <script type="text/javascript">

      /*
        Working with the Apigee API
        - Setting up an Apigee account:
        var apigee = new $.apigee_api('https://api.apigee.com');
        apigee.init("earth2marsh","supersecret"); // pwd will be base64 encoded

        - Adding an application:
        apigee.request('post','/apps/myappname.json');

        Working with a proxied API
        - Set up the app object:
        var myapp = $.apigee_api('endpoint');

        - Casting an app to a user:
        // for illustration purposes, skipping the user creation
        myapp.init("myappuser","myappuserpwd"); // base64 encode pwd

        - Making a request:
        function renderTimeline(data) {
          // do clever things with that data, assume for now that rendering is outside the scope of the library
        }
        myapp.request('get','/1/statuses/home_timeline.json',{},{callback:'renderTimeline'}); // callback to custom function
      */

    </script>
  </head>
  <body>
    <div id="response_container"></div>
  </body>
</html>
```

User pastes in their application endpoint & and previews in browser



Sample App

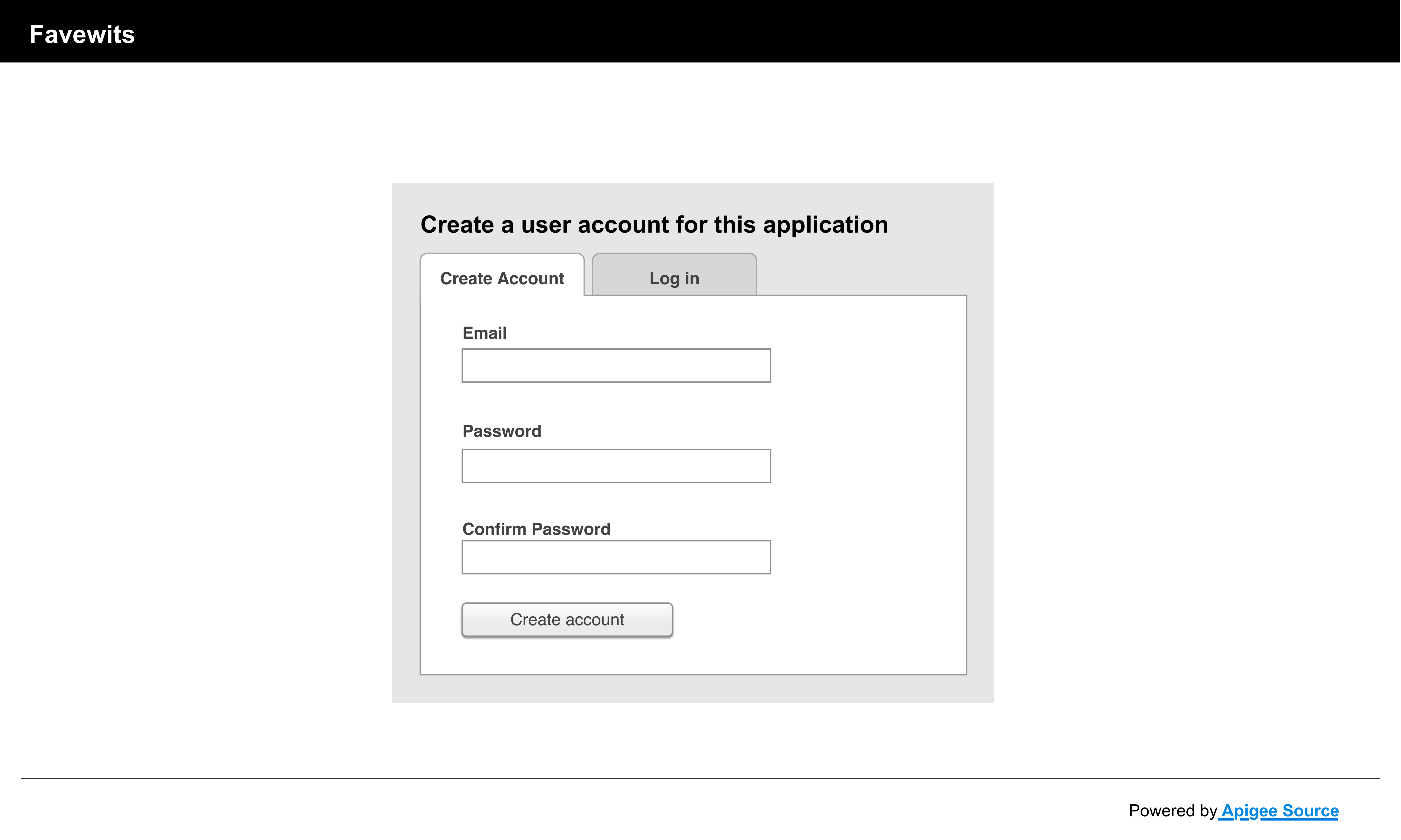
Tip

You must associate this app with the application SmartKey you configured while setting up Source. Please paste your application Endpoint and App ID into the source code for this page.

Forgot your Endpoint or App ID?

Need to set up a source application on Apigee.com?

Powered by Apigee Source



Favewits

Create a user account for this application

Create Account | Log in

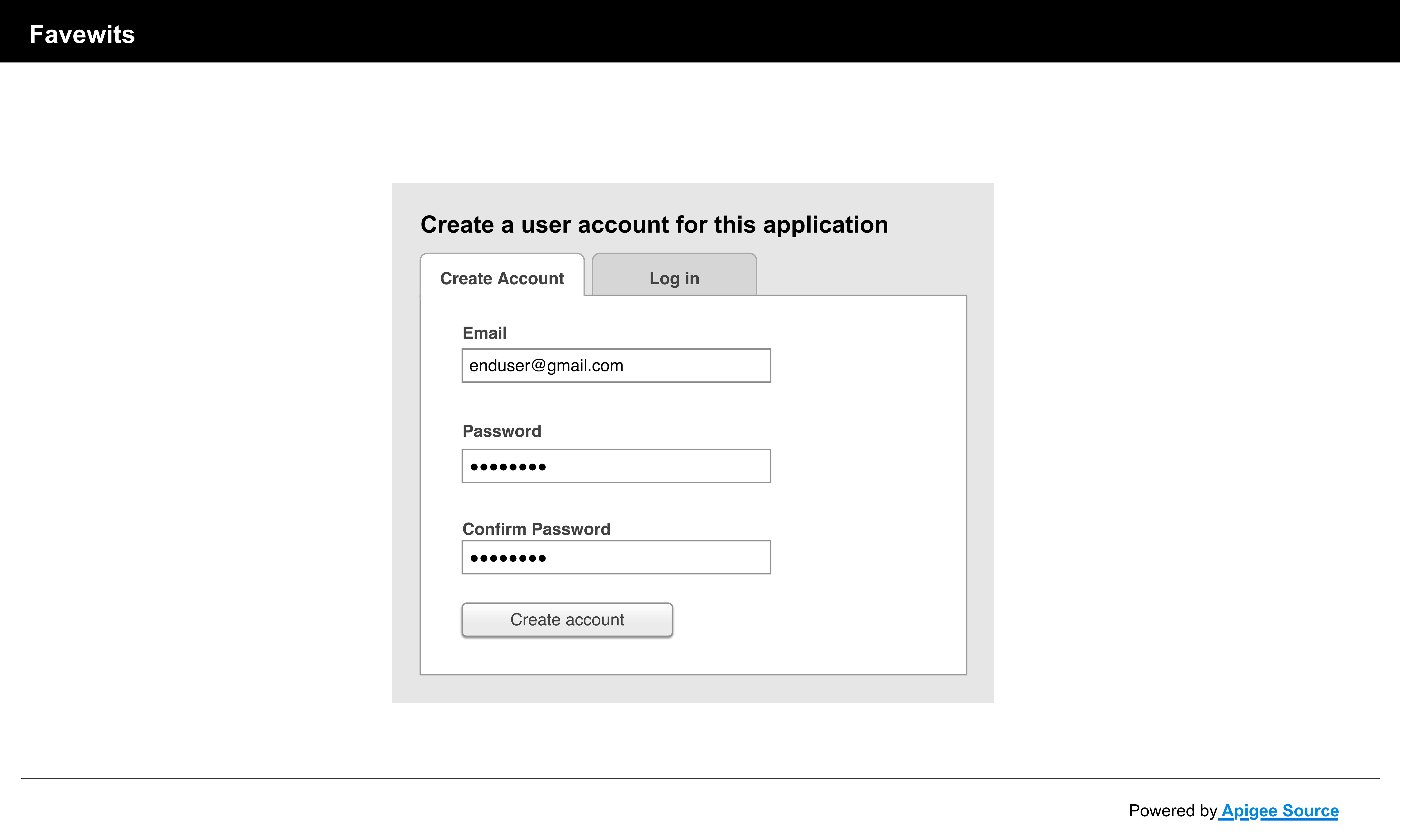
Email

Password

Confirm Password

Create account

Powered by Apigee Source



Favewits

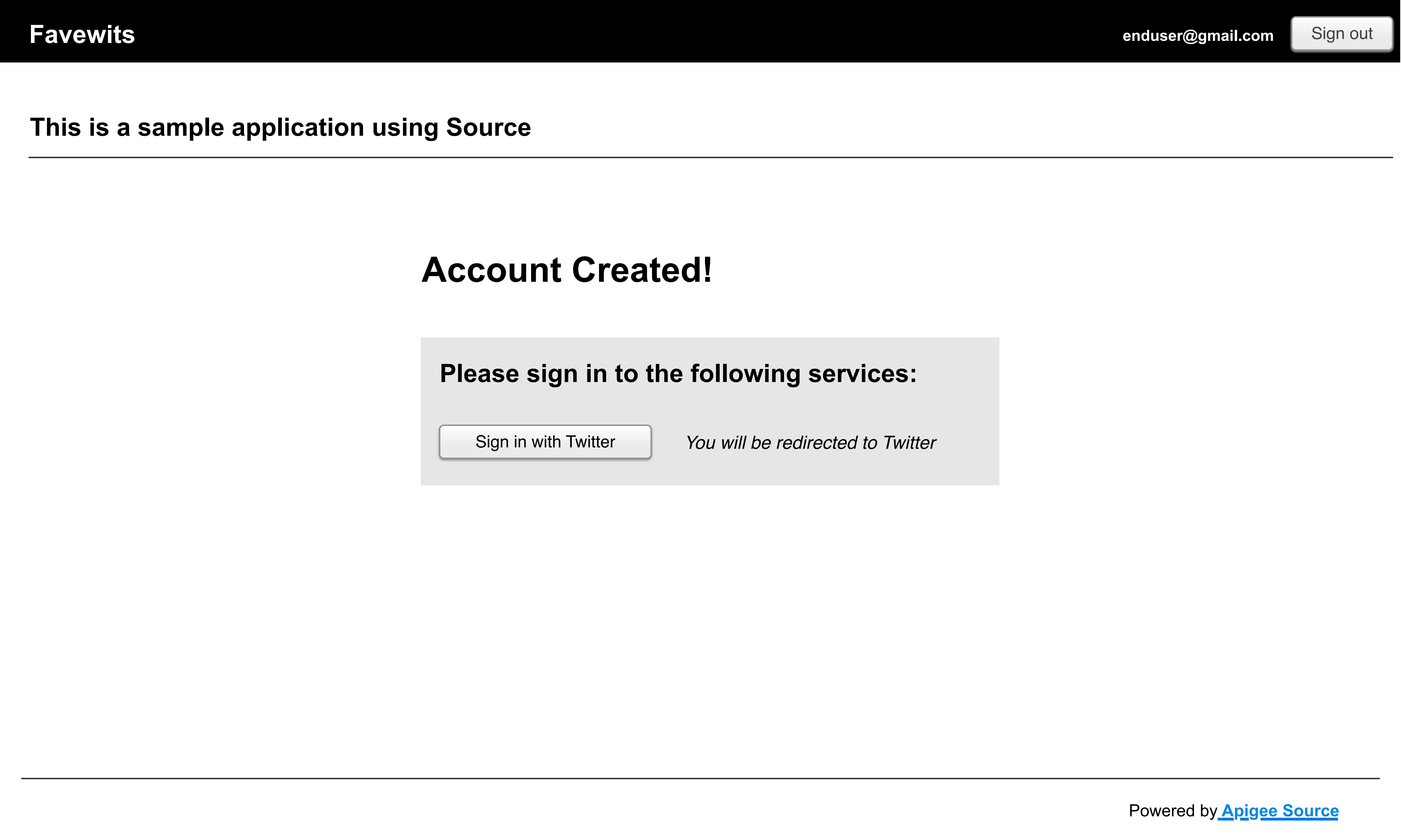
Create a user account for this application

Email: enduser@gmail.com

Password: •••••••••

Confirm Password: •••••••••

Powered by Apigee Source



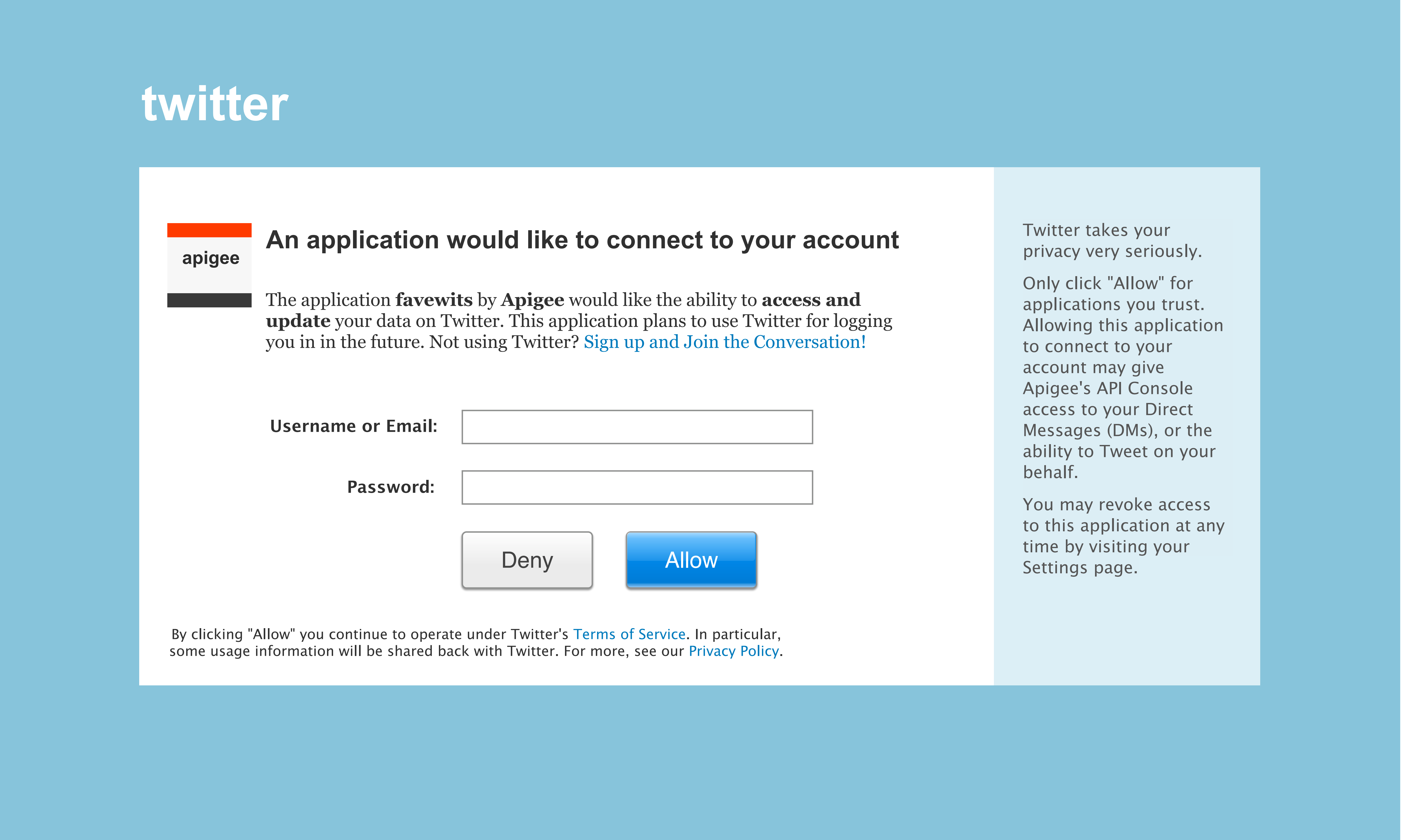
This is a sample application using Source

Account Created!

Please sign in to the following services:

You will be redirected to Twitter

Powered by [Apigee Source]



An application would like to connect to your account

The application **favewits** by **Apigee** would like the ability to **access and update** your data on Twitter. This application plans to use Twitter for logging you in in the future. Not using Twitter? Sign up and Join the Conversation!

Username or Email:

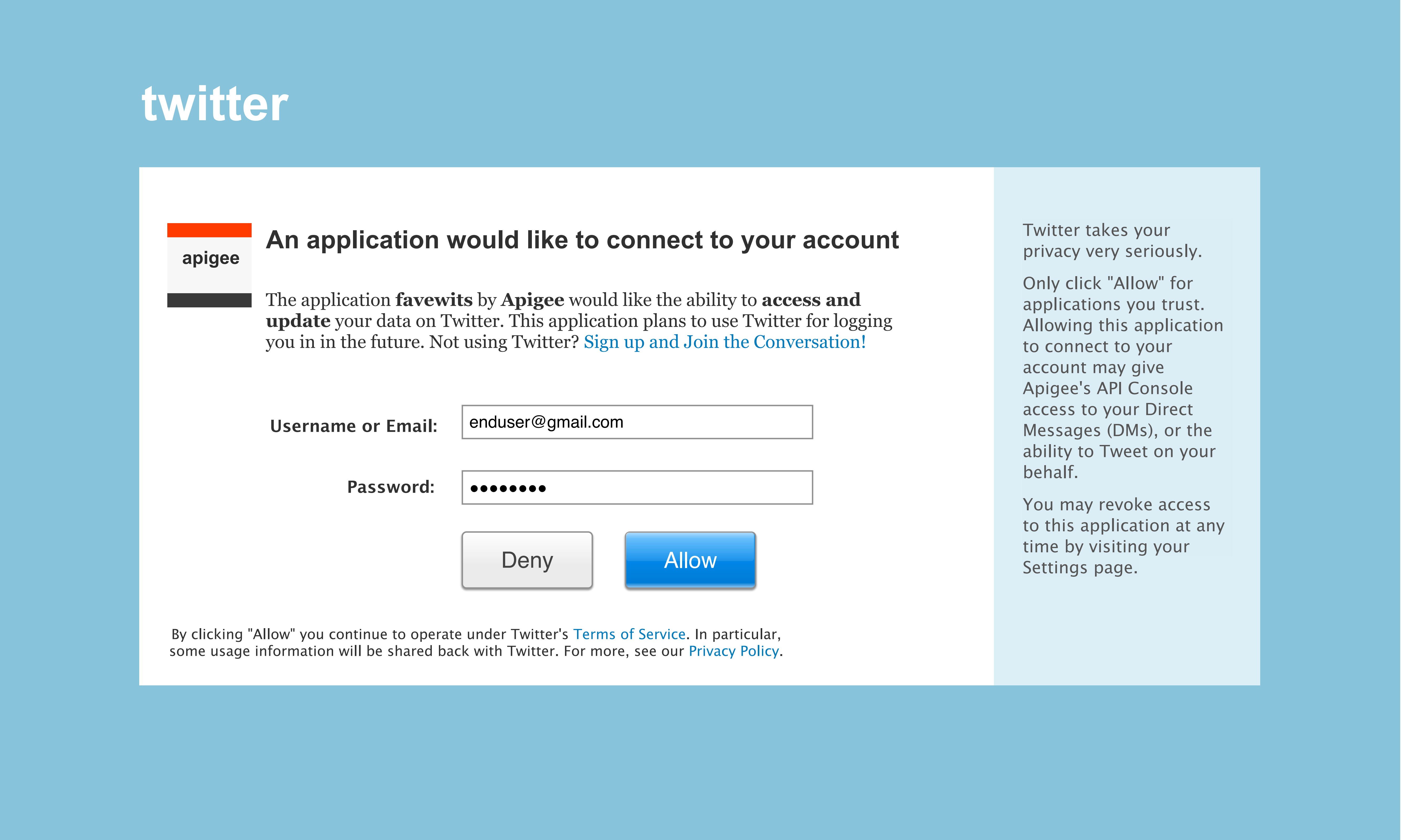
Password:

By clicking "Allow" you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Twitter takes your privacy very seriously.

Only click "Allow" for applications you trust. Allowing this application to connect to your account may give Apigee's API Console access to your Direct Messages (DMs), or the ability to Tweet on your behalf.

You may revoke access to this application at any time by visiting your Settings page.



An application would like to connect to your account

The application **favewits** by **Apigee** would like the ability to **access and update** your data on Twitter. This application plans to use Twitter for logging you in in the future. Not using Twitter? Sign up and Join the Conversation!

Username or Email: enduser@gmail.com

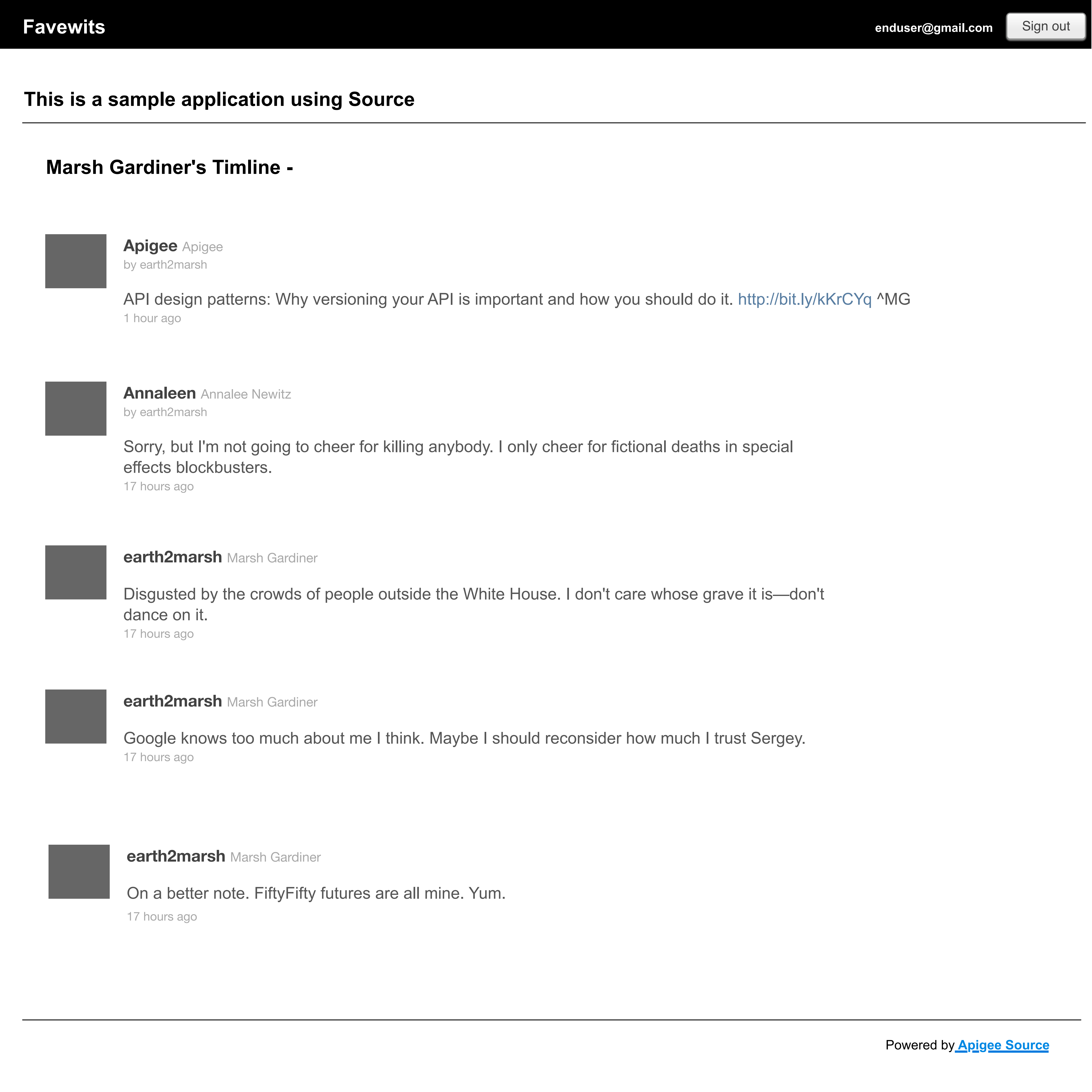
Password: ●●●●●●●●●

By clicking "Allow" you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Twitter takes your privacy very seriously.

Only click "Allow" for applications you trust. Allowing this application to connect to your account may give Apigee's API Console access to your Direct Messages (DMs), or the ability to Tweet on your behalf.

You may revoke access to this application at any time by visiting your Settings page.



This is a sample application using Source

Marsh Gardiner's Timline -

Apigee Apigee
by earth2marsh

API design patterns: Why versioning your API is important and how you should do it. http://bit.ly/kKrCYq ^MG
1 hour ago

Annaleen Annalee Newitz
by earth2marsh

Sorry, but I'm not going to cheer for killing anybody. I only cheer for fictional deaths in special effects blockbusters.
17 hours ago

earth2marsh Marsh Gardiner

Disgusted by the crowds of people outside the White House. I don't care whose grave it is—don't dance on it.
17 hours ago

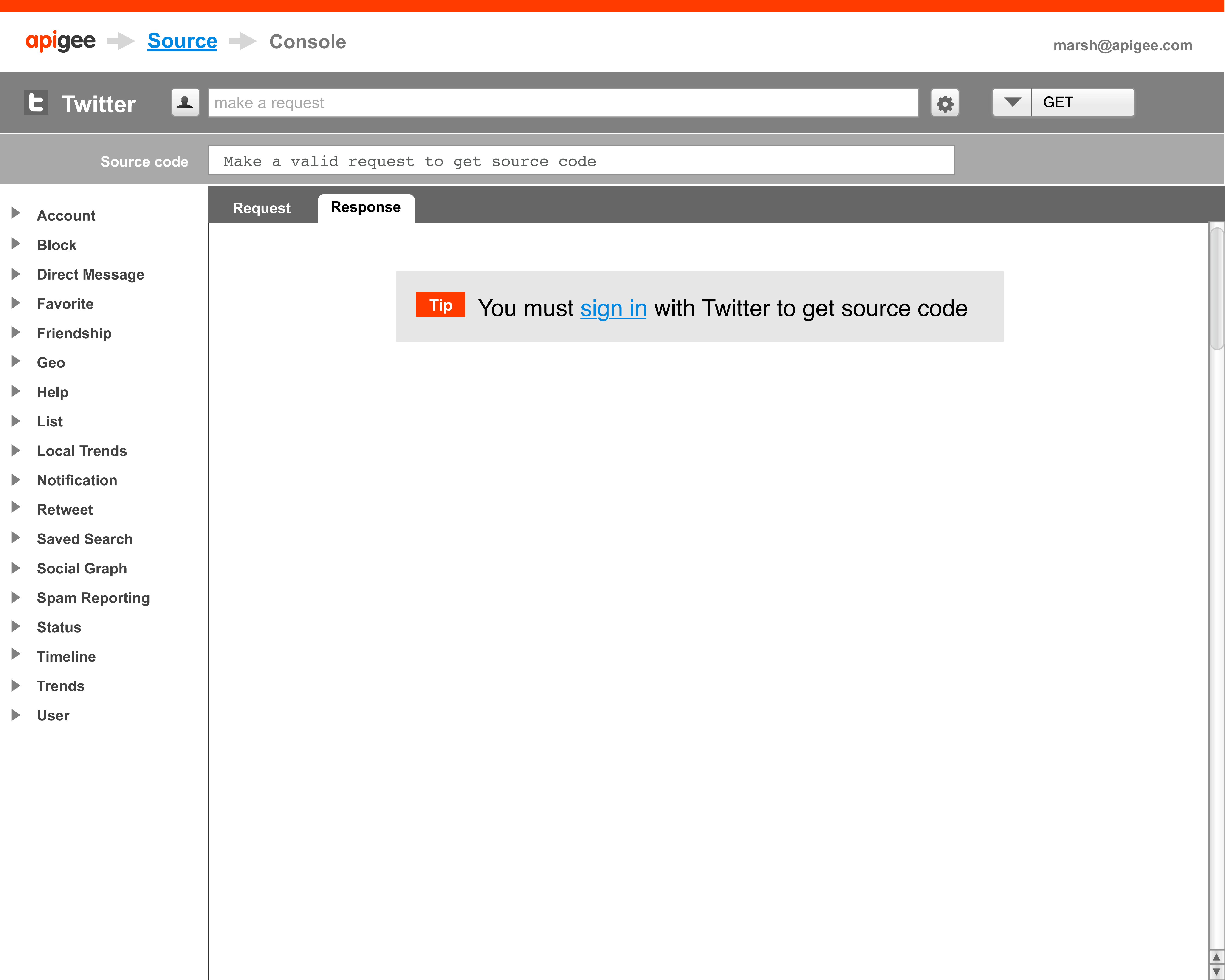
earth2marsh Marsh Gardiner

Google knows too much about me I think. Maybe I should reconsider how much I trust Sergey.
17 hours ago

earth2marsh Marsh Gardiner

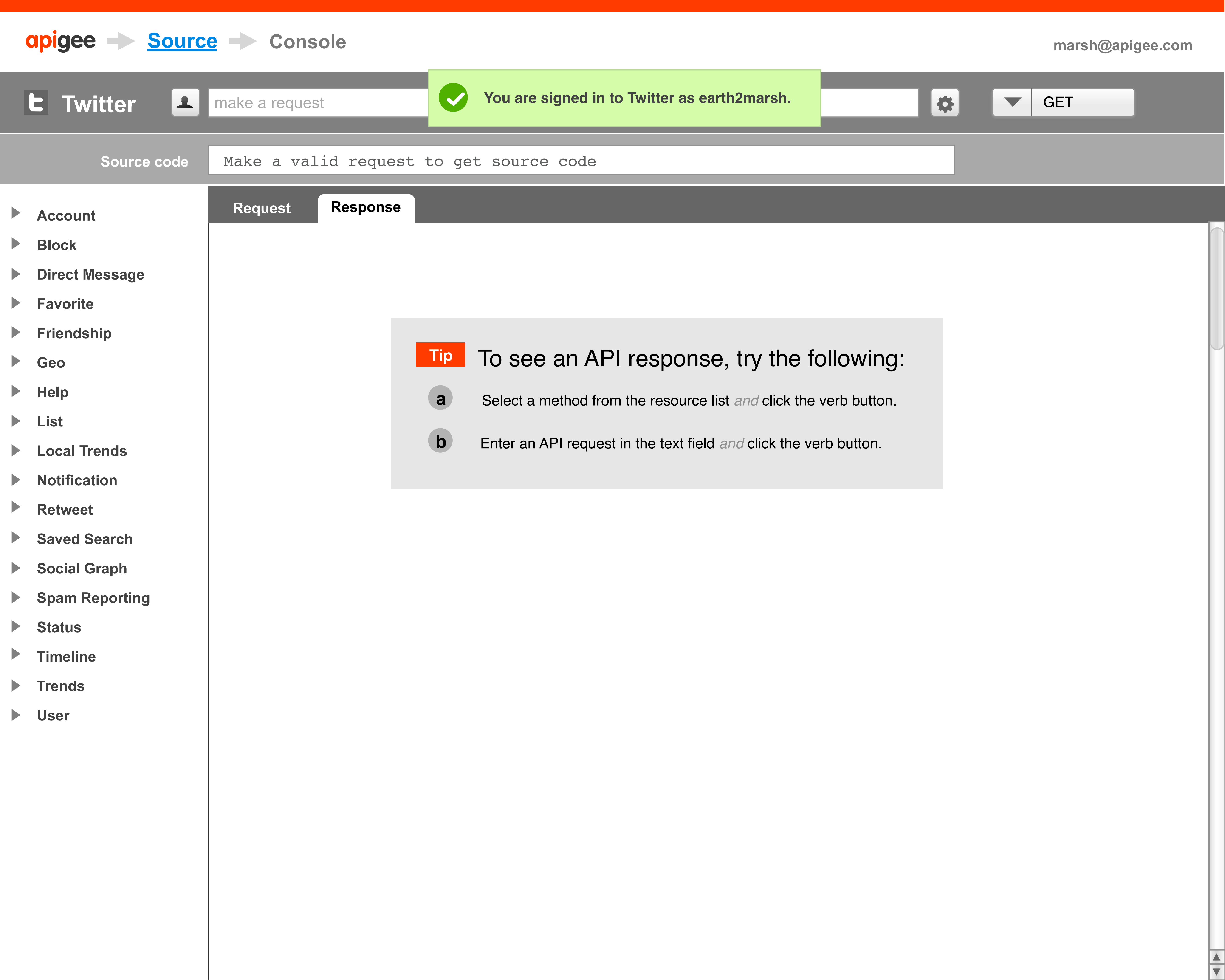
On a better note. FiftyFifty futures are all mine. Yum.
17 hours ago

Powered by [Apigee Source]



```
Make a valid request to get source code
```

Tip You must sign in with Twitter to get source code



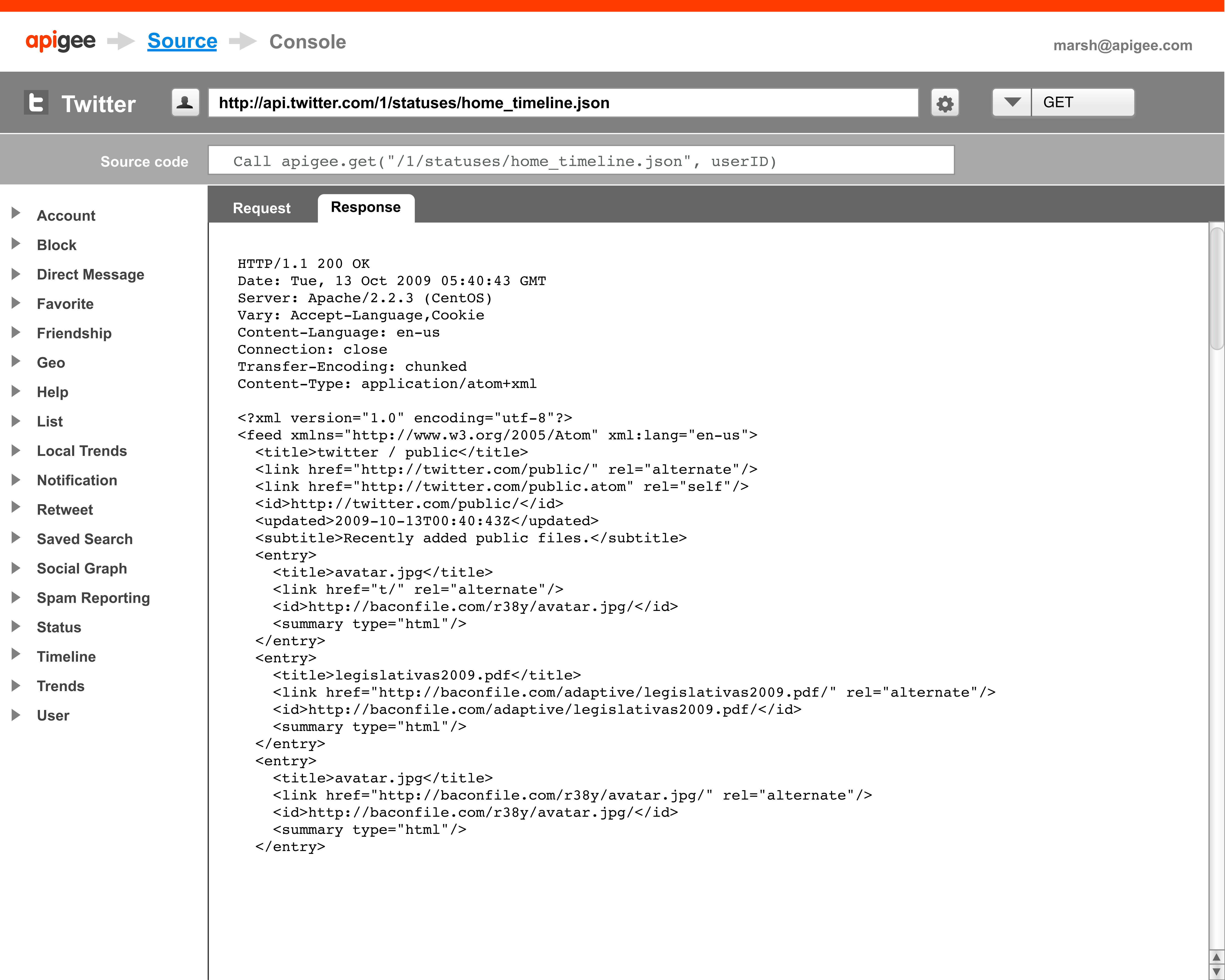
You are signed in to Twitter as earth2marsh.

```
Make a valid request to get source code
```

Tip To see an API response, try the following:

a Select a method from the resource list *and* click the verb button.

b Enter an API request in the text field *and* click the verb button.



http://api.twitter.com/1/statuses/home_timeline.json

Source code

```
Call apigee.get("/1/statuses/home_timeline.json", userID)
```

```
HTTP/1.1 200 OK
Date: Tue, 13 Oct 2009 05:40:43 GMT
Server: Apache/2.2.3 (CentOS)
Vary: Accept-Language,Cookie
Content-Language: en-us
Connection: close
Transfer-Encoding: chunked
Content-Type: application/atom+xml

<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="en-us">
  <title>twitter / public</title>
  <link href="http://twitter.com/public/" rel="alternate"/>
  <link href="http://twitter.com/public.atom" rel="self"/>
  <id>http://twitter.com/public/</id>
  <updated>2009-10-13T00:40:43Z</updated>
  <subtitle>Recently added public files.</subtitle>
  <entry>
    <title>avatar.jpg</title>
    <link href="t/" rel="alternate"/>
    <id>http://baconfile.com/r38y/avatar.jpg/</id>
    <summary type="html"/>
  </entry>
  <entry>
    <title>legislativas2009.pdf</title>
    <link href="http://baconfile.com/adaptive/legislativas2009.pdf/" rel="alternate"/>
    <id>http://baconfile.com/adaptive/legislativas2009.pdf/</id>
    <summary type="html"/>
  </entry>
  <entry>
    <title>avatar.jpg</title>
    <link href="http://baconfile.com/r38y/avatar.jpg/" rel="alternate"/>
    <id>http://baconfile.com/r38y/avatar.jpg/</id>
    <summary type="html"/>
  </entry>
```

github

Pricing and Signup Explore GitHub Features Blog Login

apigee / **Apigee-Source-for-JavaScript**

Watch Fork

Source Commits Network Pull requests (0) Issues (0) Graphs Branch: gh-pages

Switch Branches (1) Switch Tags Branch List

apigee.github.com/Apigee-Source-for-JavaScript

Downloads

- **HTTP Git Read-Only** https://github.com/apigee/Apigee-Source-for-JavaScript.git This URL has **Read-Only** access

fixed link to source of example app

earth2marsh (author)
1 day ago

```
commit  6e0df084a783daaba644
tree    63e37f3d95b343cf93b9
parent  912e3a611f9ca6b09808
```

Apigee-Source-for-JavaScript /

name	age	message	history
examples/	1 day ago	fixed link to source of example app [earth2marsh]	
javascripts/	1 day ago	updated to production version of apigee.js [earth2marsh]	
jsdoc/	1 day ago	separating js into its own repo [earth2marsh]	
README.md	1 day ago	added collaborator text [earth2marsh]	
favicon.ico	1 day ago	separating js into its own repo [earth2marsh]	
index.html	1 day ago	removed API queries from landing page [earth2marsh]	

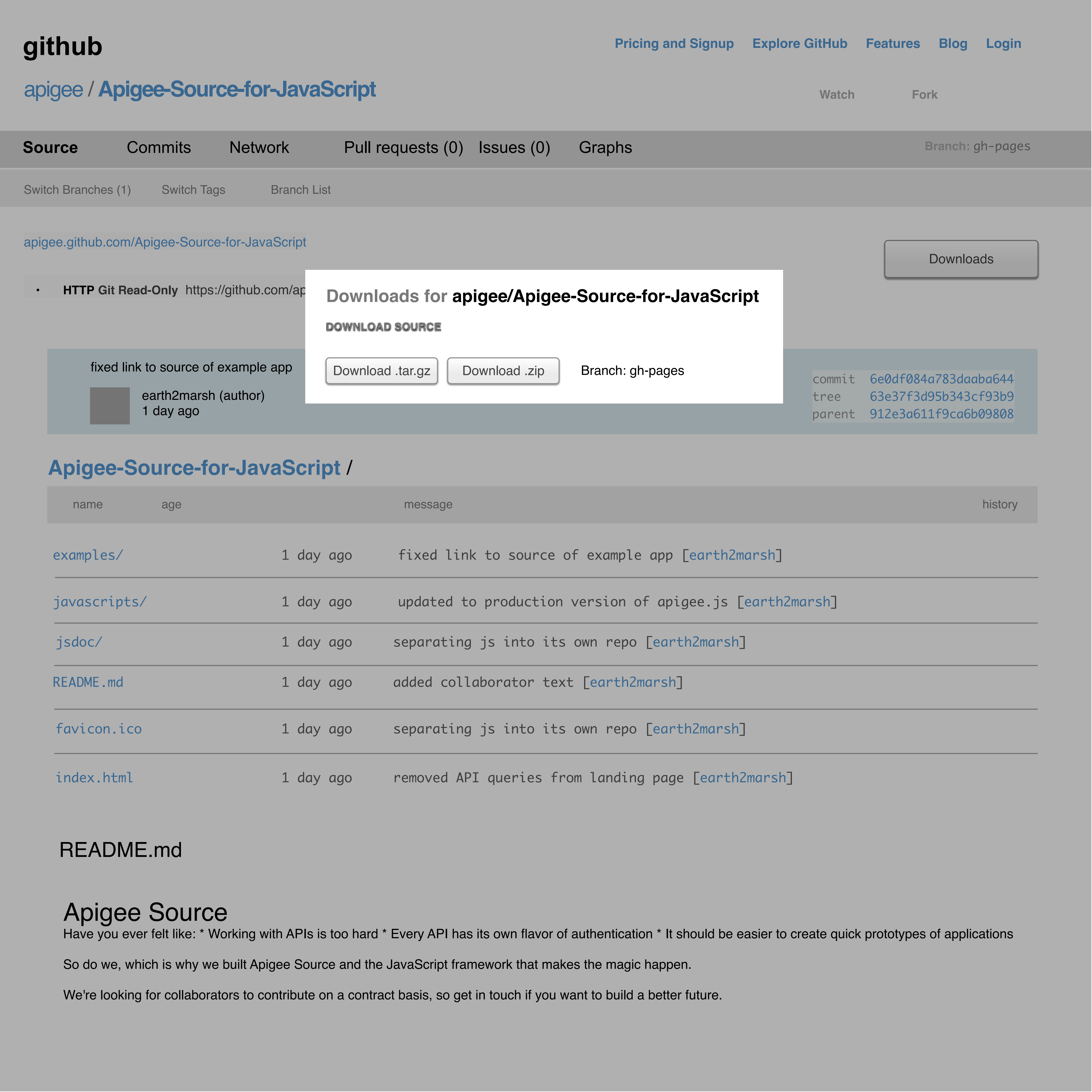
README.md

Apigee Source

Have you ever felt like: * Working with APIs is too hard * Every API has its own flavor of authentication * It should be easier to create quick prototypes of applications

So do we, which is why we built Apigee Source and the JavaScript framework that makes the magic happen.

We're looking for collaborators to contribute on a contract basis, so get in touch if you want to build a better future.



apigee / Apigee-Source-for-JavaScript

apigee.github.com/Apigee-Source-for-JavaScript

- **HTTP Git Read-Only** https://github.com/ap

Downloads for apigee/Apigee-Source-for-JavaScript

DOWNLOAD SOURCE

Download .tar.gz | Download .zip | Branch: gh-pages

fixed link to source of example app

earth2marsh (author)
1 day ago

```
commit  6e0df084a783daaba644
tree    63e37f3d95b343cf93b9
parent  912e3a611f9ca6b09808
```

Apigee-Source-for-JavaScript /

name	age	message	history
examples/	1 day ago	fixed link to source of example app [earth2marsh]	
javascripts/	1 day ago	updated to production version of apigee.js [earth2marsh]	
jsdoc/	1 day ago	separating js into its own repo [earth2marsh]	
README.md	1 day ago	added collaborator text [earth2marsh]	
favicon.ico	1 day ago	separating js into its own repo [earth2marsh]	
index.html	1 day ago	removed API queries from landing page [earth2marsh]	

README.md

Apigee Source

Have you ever felt like: * Working with APIs is too hard * Every API has its own flavor of authentication * It should be easier to create quick prototypes of applications

So do we, which is why we built Apigee Source and the JavaScript framework that makes the magic happen.

We're looking for collaborators to contribute on a contract basis, so get in touch if you want to build a better future.