Hey there,

This is a quick tutorial how to present my application.
It will help you adding the Watson User Modeling Service Live.

0. Prepare
1. Upload an alone standing application to Bluemix
2. Show how to use the alone standing application
3. Add the User Modeling Service to it
4. Activate the User Modeling Service
5. Upload the application
6. Show how to use the new application

If you do not know what my application does, check out my other slideshare presentation

0. Prepare

Instructions

There are some steps you have to do before you can present the application.

- Go to github.com/JDihlmann/moodlocator and follow the steps in the section presentation
- If you are presenting moodlocator you can start your presentation at any point you want, but you have to do all steps behind the scene if you wouldn't show it.

Read the GitHub instructions until point 5

At this point I'm assuming you read the GitHub instructions and set up everything

- Before we upload our alone standing application we have to make some changes to it. The default of the moodlocator application is set to show the Watson User Modeling Service.
- To change this default value we have to open the code

0. Prepare

Open style.css

Open the style.css file with a text editor you will find it in

/ node / public / stylesheets / style.css

Scroll to the bottom of the page

You will find something that looks like that, delete the text in line 449 and 452. I marked it red.

Before

```
448   /* For Presentation activate oder deactivate the Section below */
449   /*
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452   */
```

After

```
448   /* For Presentation activate oder deactivate the Section below */
449
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452
```

Save your changes. Perfect to understand what it does, it enables or disables the white Watson User Modeling bar at the bottom of the page.

0. Prepare

Open app.js

Open the app.js file with a text editor you will find it in

/ node / app.js

Scroll to line 189

You will find something that looks like that, add this bracket „ /* “ in line 190 and this bracket „ */ “ in line 211.
I marked the position red where to add them.

0. Prepare

Before

```
189     //For presentation start commenting here
190     
191     //Create a profile request with the text and the https options and call it
192     create_profile_request(profile_options,textToWatson)(function(error,profile_string) {
193         if (error) console.log(error.message);
194         else {
195             //Parse the Profile and format it
196             var profile_json = JSON.parse(profile_string);
197             var flat_traits = flatten.flat(profile_json.tree);
198 
199             //Sent Watson Data to Client
200             var watsonObj = {
201                 id: msg.id,
202                 watsonJSON: flat_traits
203                 }
204 
205 
206             //Send Watson Data to Client
207             socket.emit('watson message', watsonObj);
208             //console.log(flat_traits);
209         }
210     }); //For presentation stop commenting here
211     
212 }
```

After

```
189     //For presentation start commenting here
190     /*
191     //Create a profile request with the text and the https options and call it
192     create_profile_request(profile_options,textToWatson)(function(error,profile_string) {
193         if (error) console.log(error.message);
194         else {
195             //Parse the Profile and format it
196             var profile_json = JSON.parse(profile_string);
197             var flat_traits = flatten.flat(profile_json.tree);
198 
199             //Sent Watson Data to Client
200             var watsonObj = {
201                 id: msg.id,
202                 watsonJSON: flat_traits
203                 }
204 
205 
206             //Send Watson Data to Client
207             socket.emit('watson message', watsonObj);
208             //console.log(flat_traits);
209         }
210     }); //For presentation stop commenting here
211     */
212 }
```

0. Prepare

Stay in app.js

Scroll to line 238

You will find something that looks like that, start in line 238 and delete everything what comes after it, till the end of the document.

Save your changes

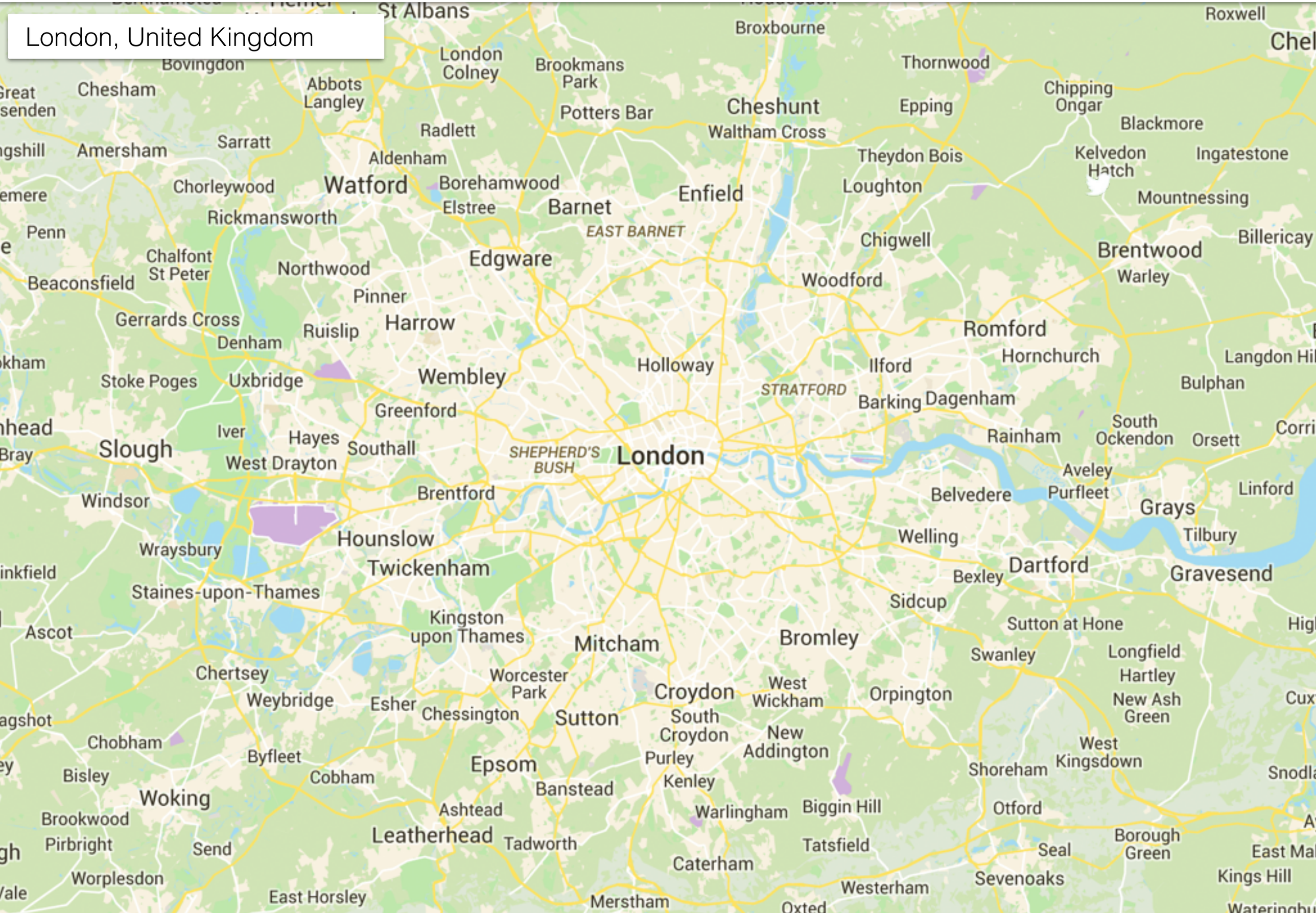
Okay we prepared our application now lets start presenting

```
238  //For presentation start delete this section
239
240  //Function you will find in the User Modeling documentation
241  var create_profile_request = function(options,content) {
242    return function (/*function*/ callback) {
243      // create the post data to send to the User Modeling service
244      var post_data = {
245        'contentItems' : [{
246          'userid' : 'dummy',
247          'id' : 'dummyUuid',
248          'sourceid' : 'freetext',
249          'contenttype' : 'text/plain',
250          'language' : 'en',
251          'content': content
252        }]
253      };
254      // Create a request to POST to the User Modeling service
255      var profile_req = https.request(options, function(result) {
256        result.setEncoding('utf-8');
257        var response_string = '';
258
259        result.on('data', function(chunk) {
260          response_string += chunk;
261        });
262
263        result.on('end', function() {
264
265          if (result.statusCode != 200) {
266            var error = JSON.parse(response_string);
267            callback({'message': error.user_message}, null);
268          } else
269            callback(null,response_string);
270        });
271      });
272
273      profile_req.on('error', function(e) {
274        callback(e,null);
275      });
276
277      profile_req.write(JSON.stringify(post_data));
278      profile_req.end();
279    }
280  };
281
282  //For presentation stop deleting this section
```

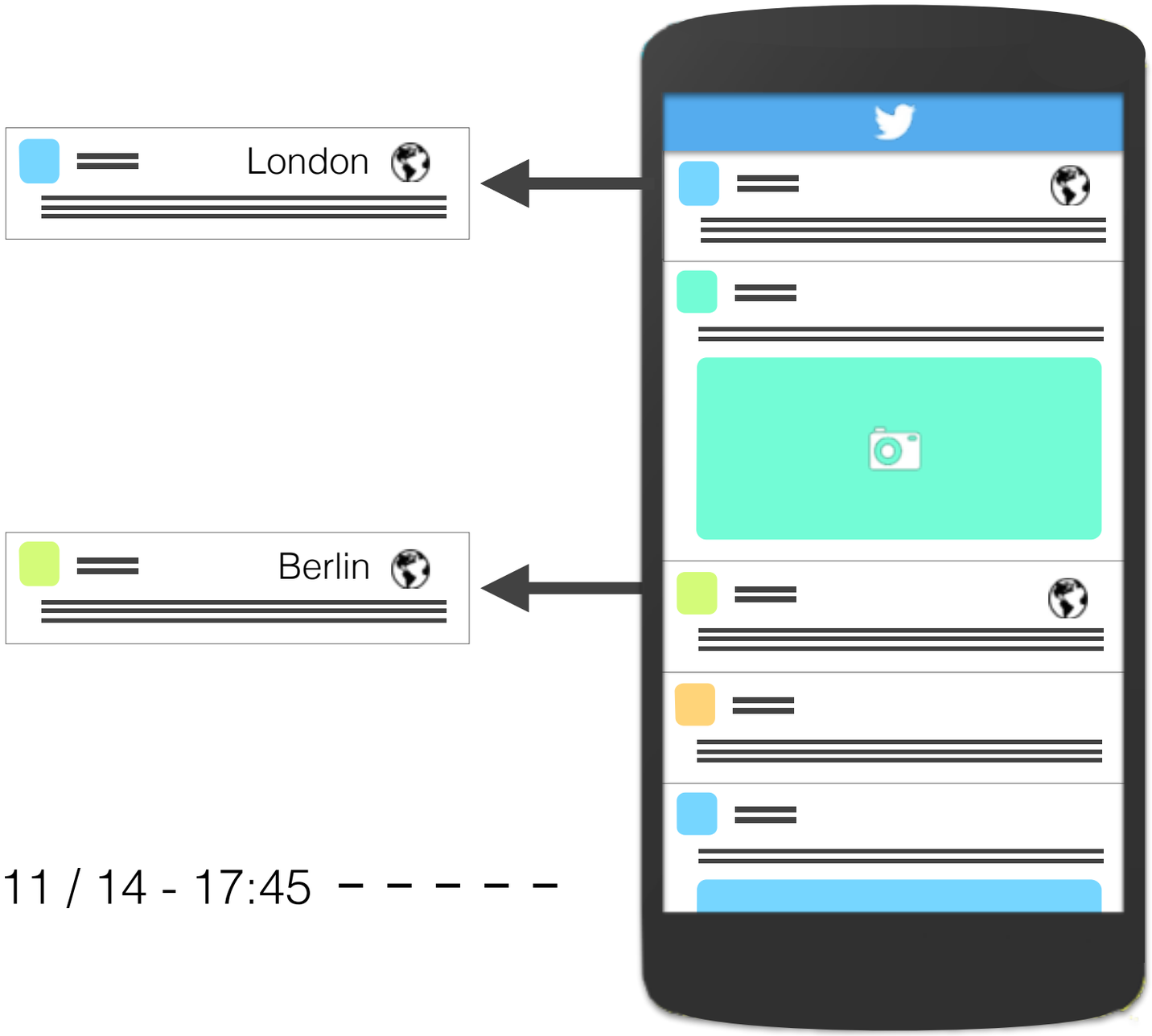
Bluemix Demo

MoodLocator

Google Maps API

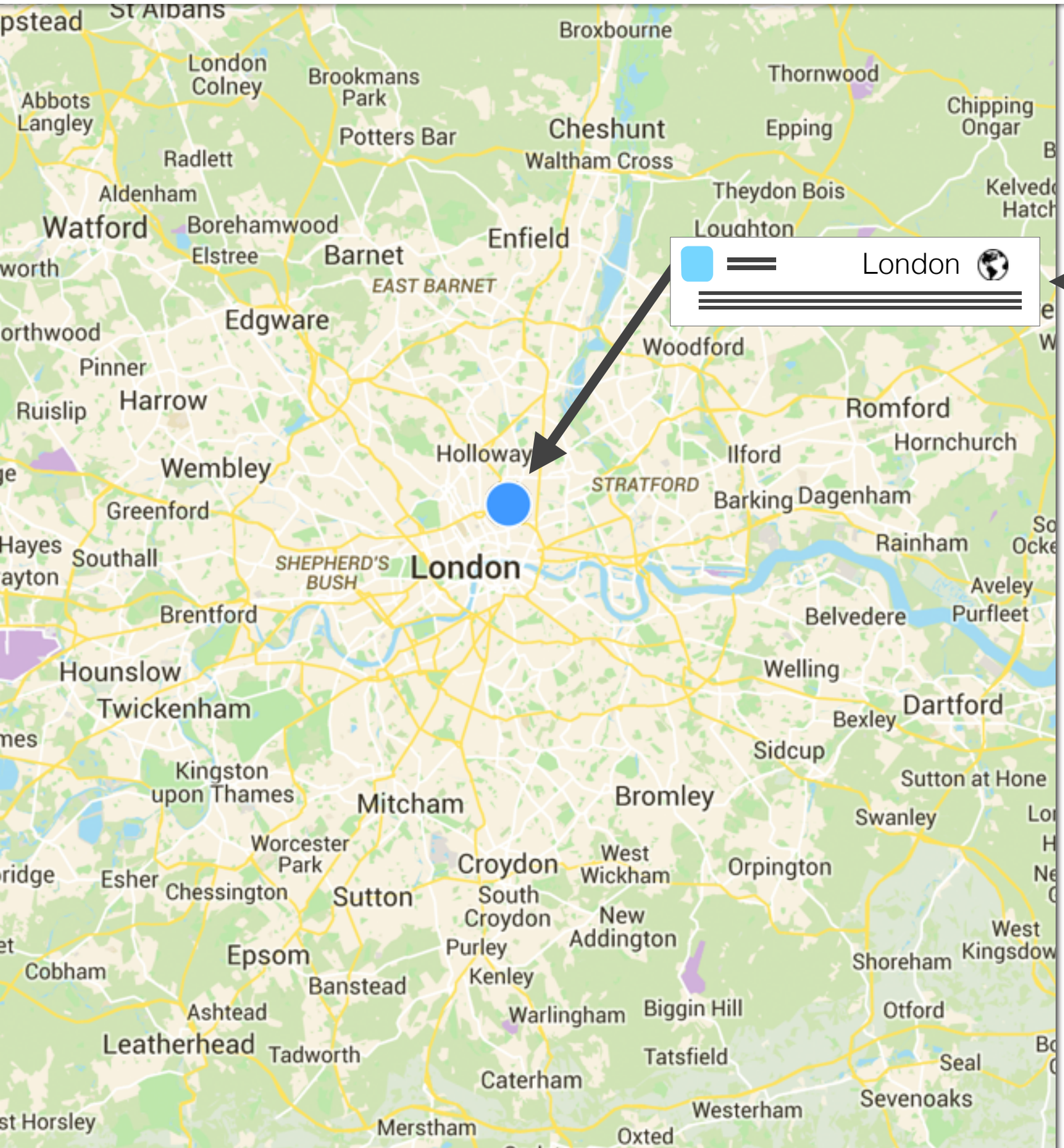


Twitter Stream API



– – – – – – User visited website 29 / 11 / 14 - 17:45 – – – – –

Google & Twitter

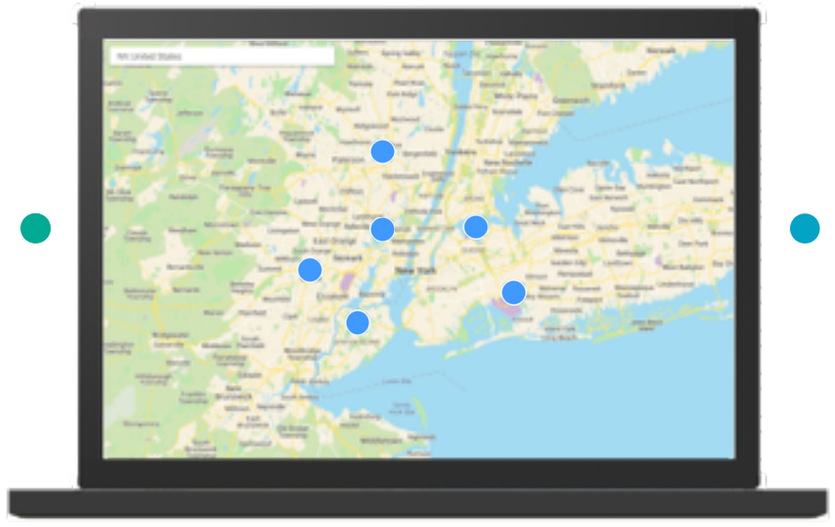


Structure

Maps API

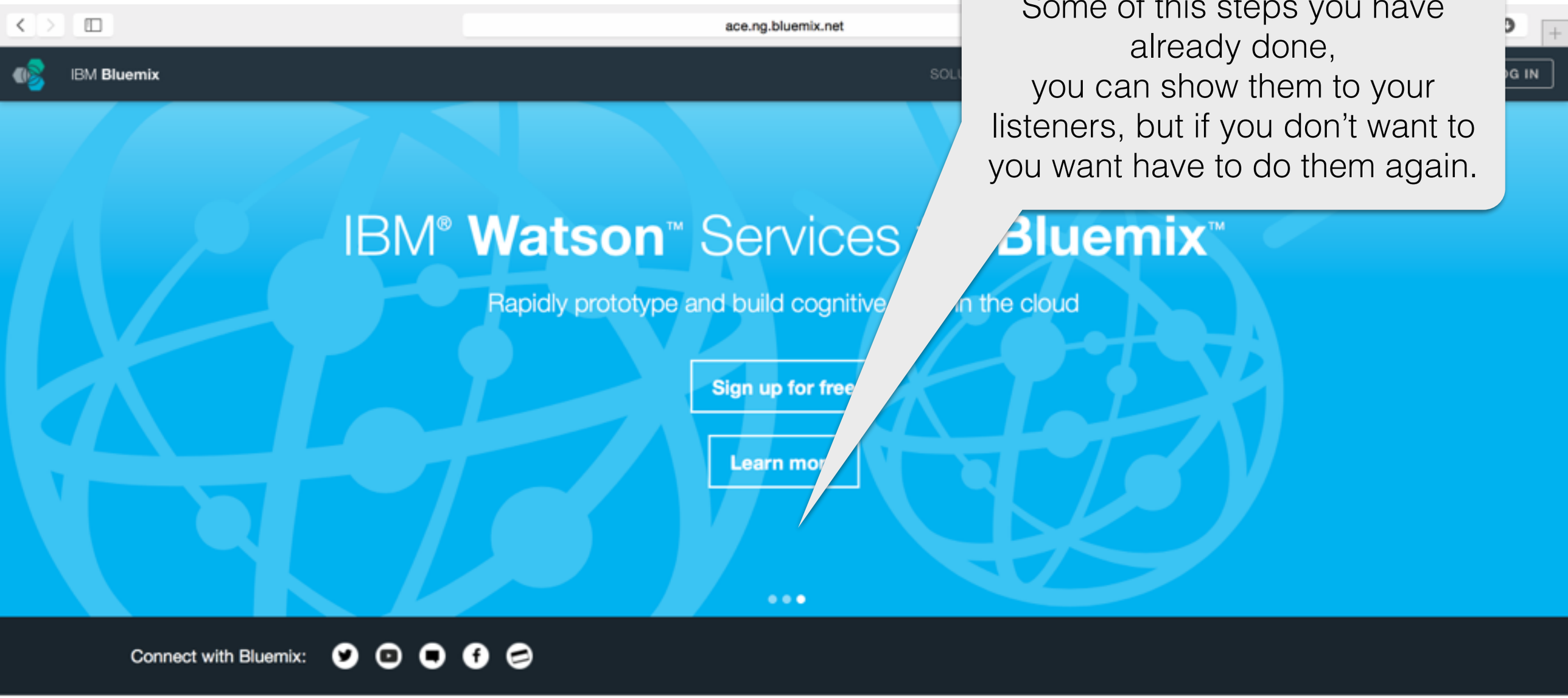
Stream API

Lokal

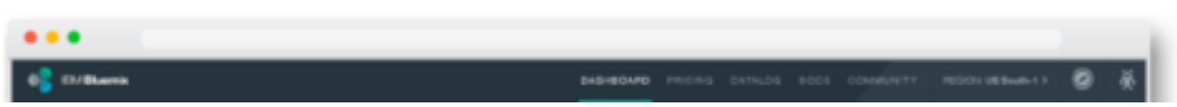


Live Demo

Bluemix Homepage



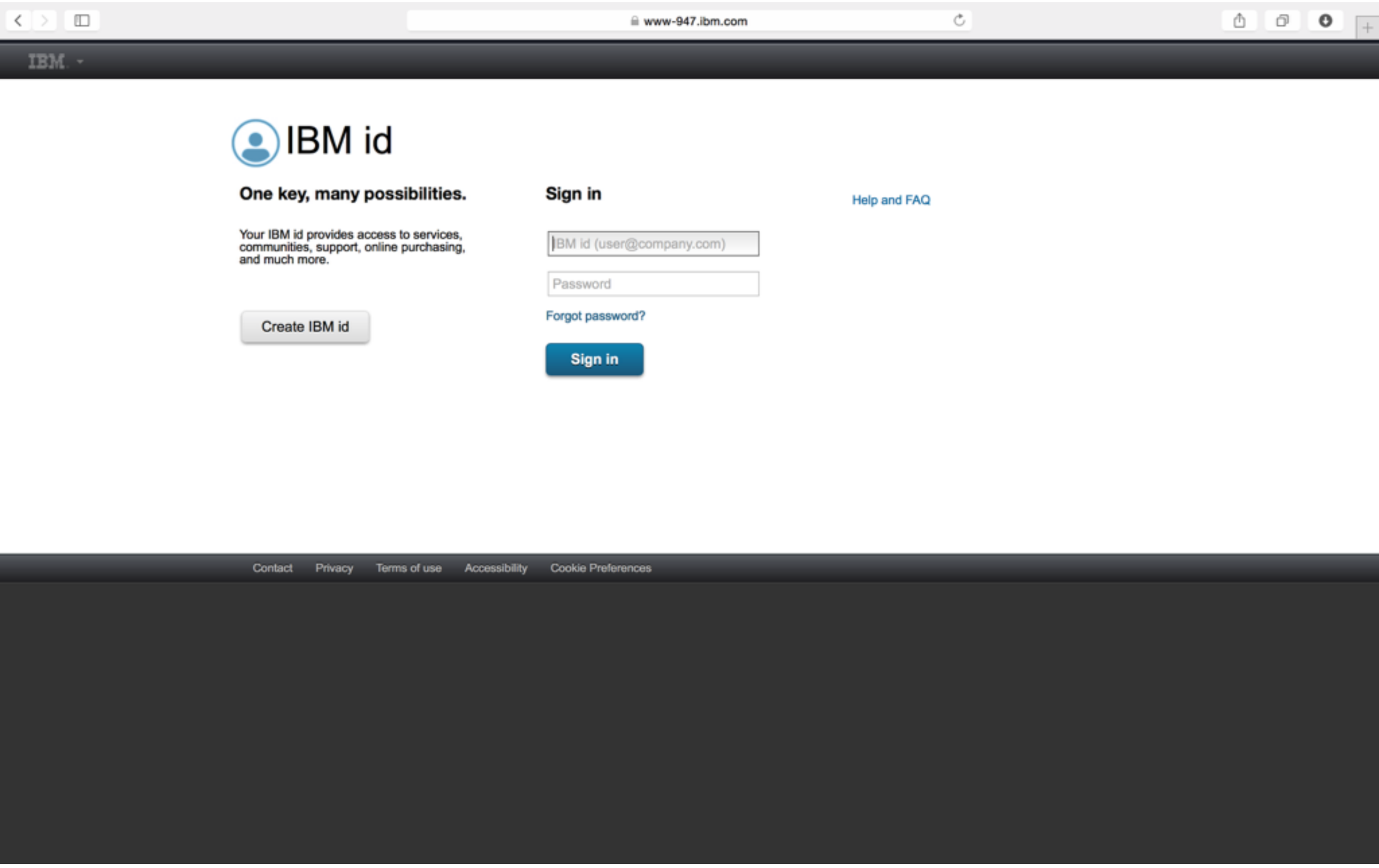
Some of this steps you have already done,
you can show them to your listeners, but if you don't want to you want have to do them again.



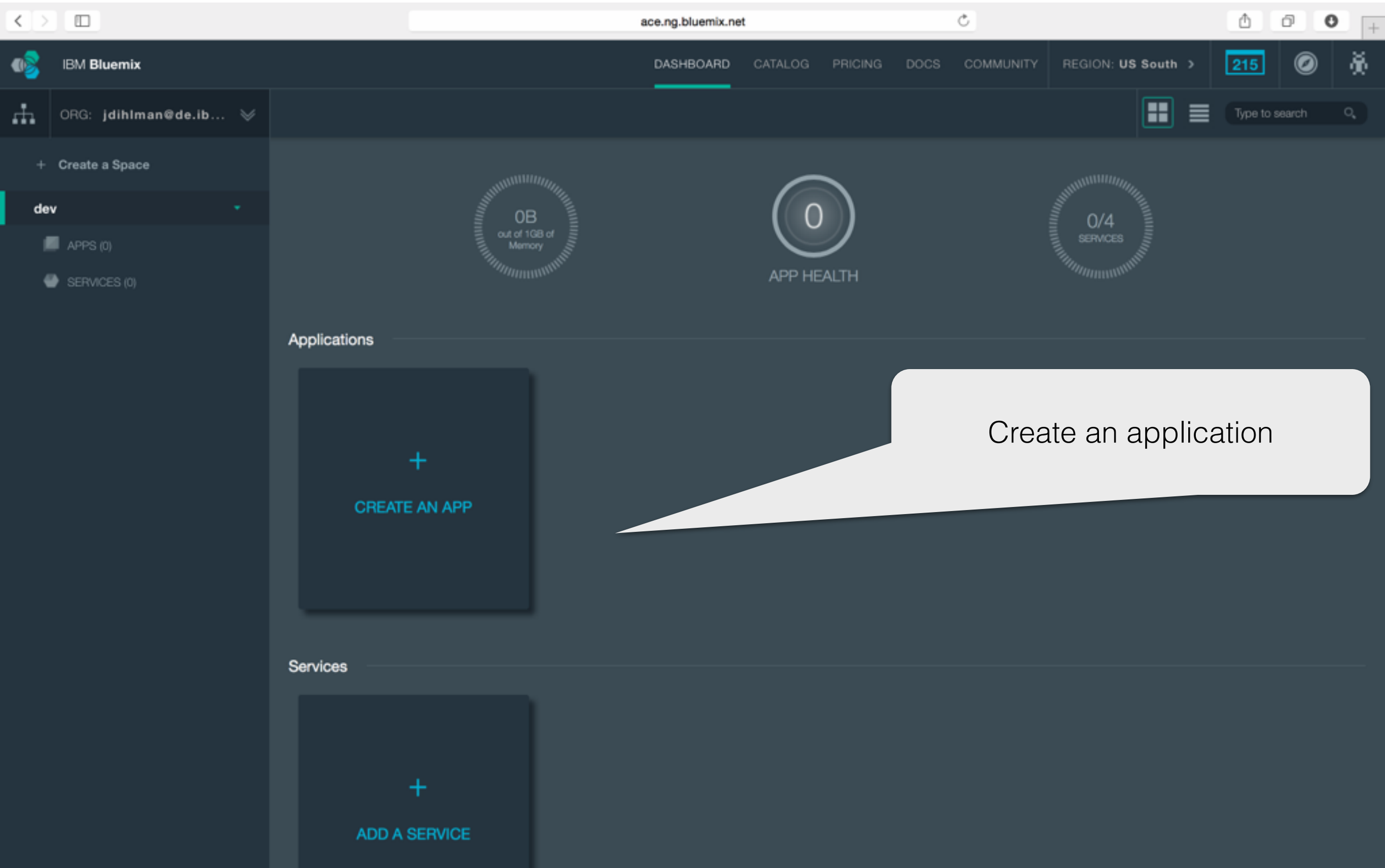
The future is yours.

If you're a developer, you now have a hand in creating the future. Bluemix offers you all the instant services, runtimes, and infrastructure you need to push your ideas into the present.

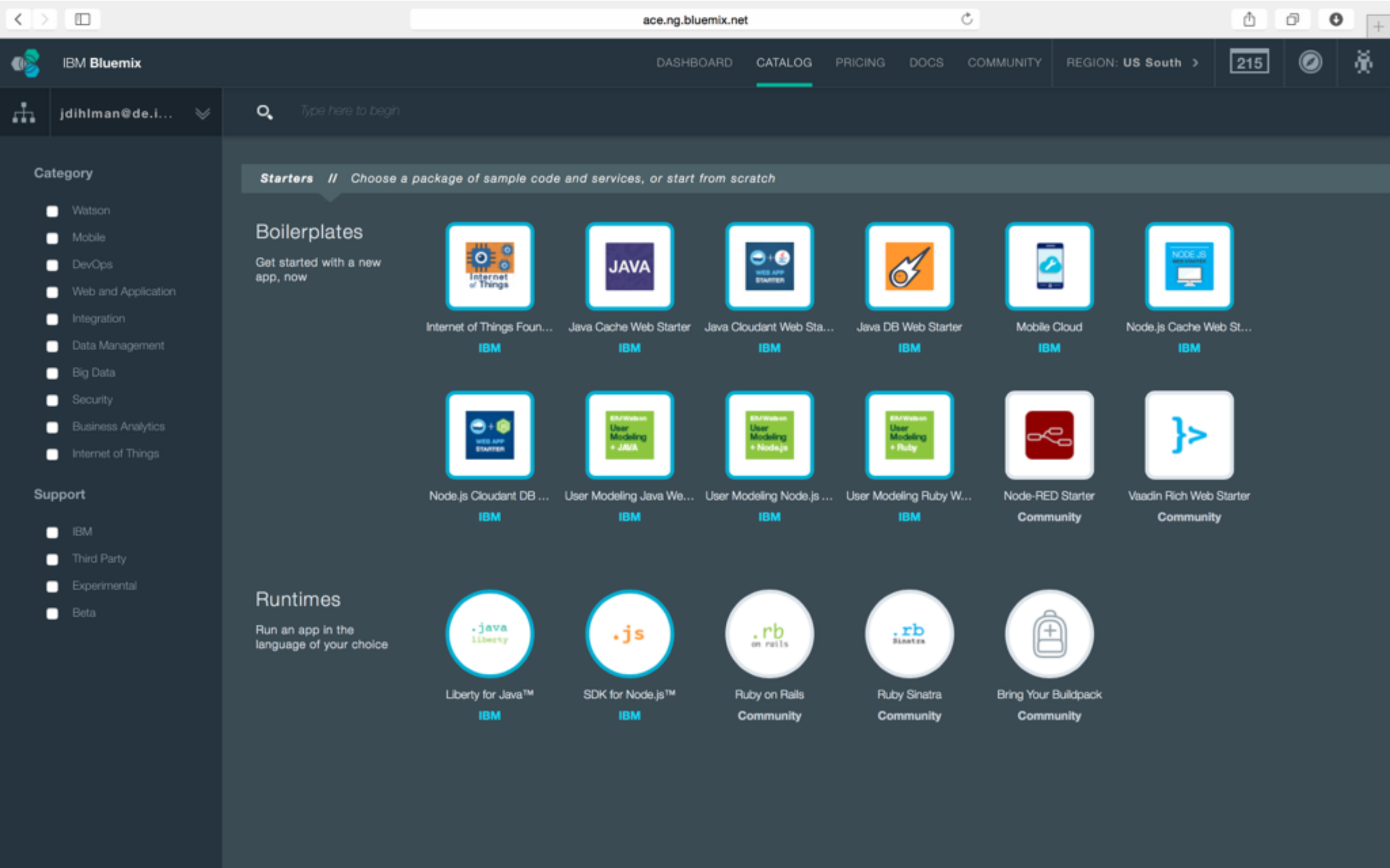
Login


www-947.ibm.com
IBM
IBM id
One key, many possibilities.
Your IBM id provides access to services, communities, support, online purchasing, and much more.
Create IBM id
Sign in
Help and FAQ
IBM id (user@company.com)
Password
Forgot password?
Sign in
Contact
Privacy
Terms of use
Accessibility
Cookie Preferences

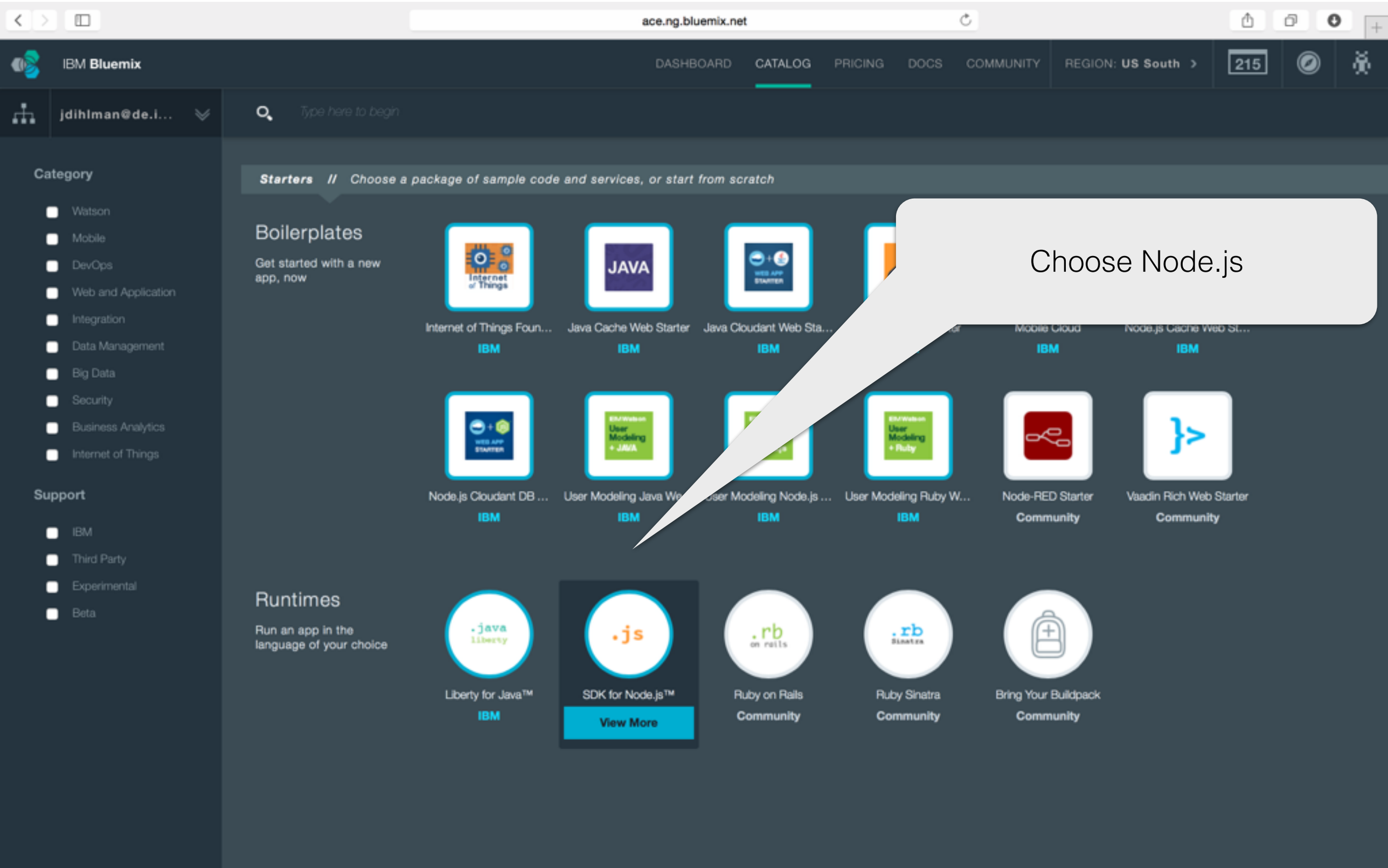
Dashboard


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
ORG: jdihlman@de.ib...
Type to search
Create a Space
dev
APPS (0)
SERVICES (0)
0B
out of 1GB of Memory
0
APP HEALTH
0/4
SERVICES
Applications
CREATE AN APP
Create an application
Services
ADD A SERVICE

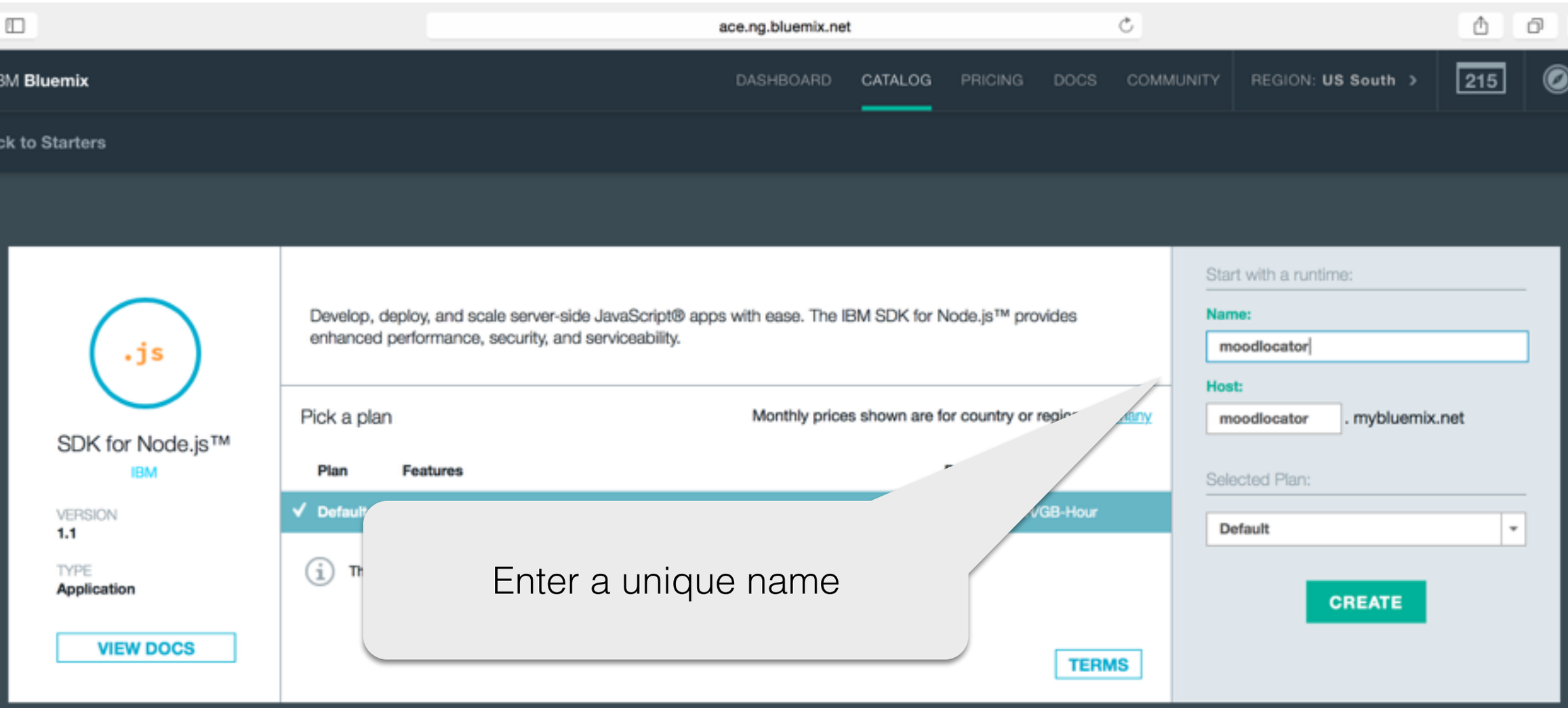
Catalog


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
jdihlman@de.i...
Type here to begin
Category
Watson
Mobile
DevOps
Web and Application
Integration
Data Management
Big Data
Security
Business Analytics
Internet of Things
Support
IBM
Third Party
Experimental
Beta
Starters // Choose a package of sample code and services, or start from scratch
Boilerplates
Get started with a new app, now
Internet of Things Foun...
IBM
Java Cache Web Starter
IBM
Java Cloudant Web Sta...
IBM
Java DB Web Starter
IBM
Mobile Cloud
IBM
Node.js Cache Web St...
IBM
Node.js Cloudant DB ...
IBM
User Modeling Java We...
IBM
User Modeling Node.js ...
IBM
User Modeling Ruby W...
IBM
Node-RED Starter
Community
Vaadin Rich Web Starter
Community
Runtimes
Run an app in the language of your choice
Liberty for Java™
IBM
SDK for Node.js™
IBM
Ruby on Rails
Community
Ruby Sinatra
Community
Bring Your Buildpack
Community

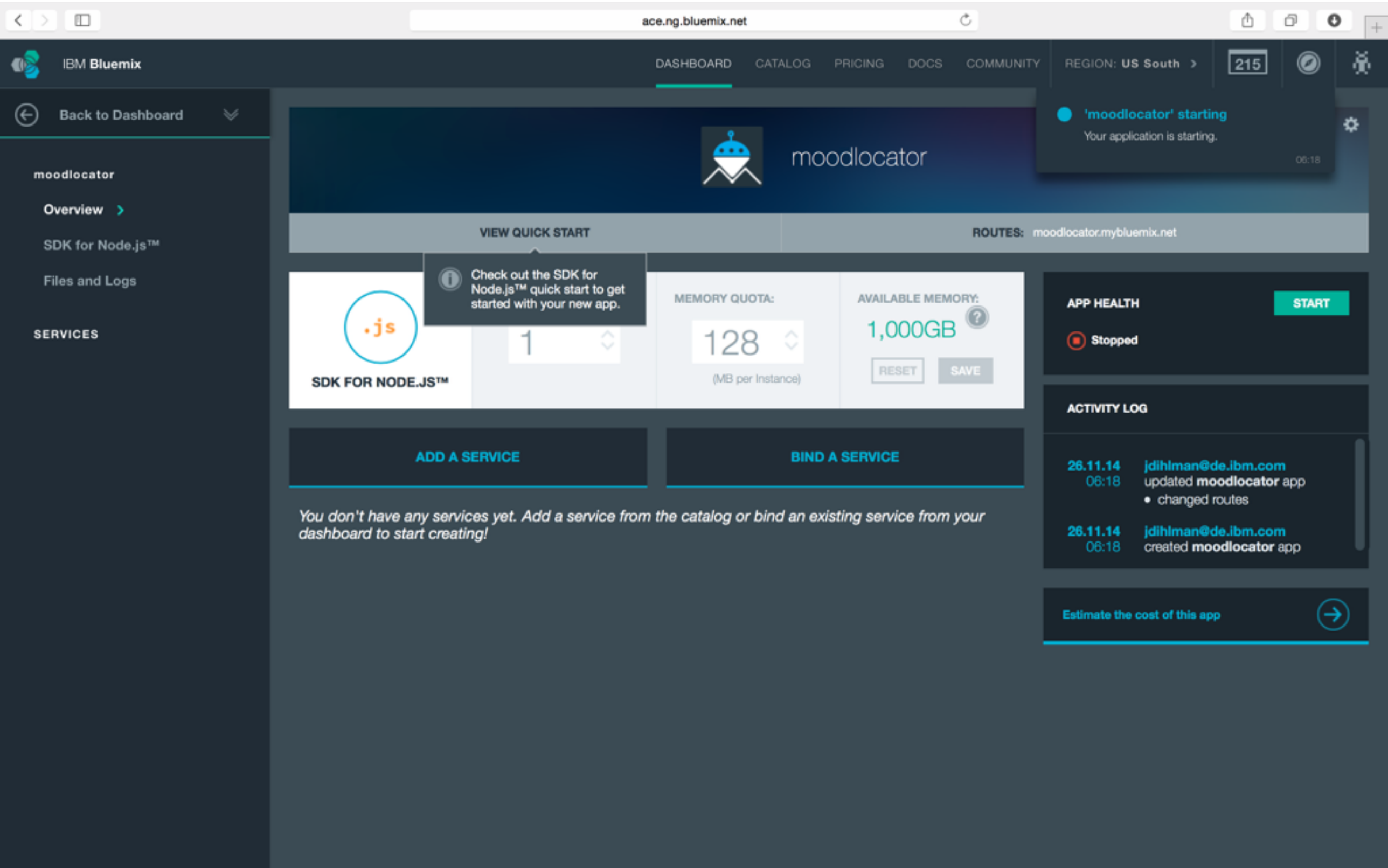
Node.js


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
jdihlman@de.i...
Type here to begin
Category
Watson
Mobile
DevOps
Web and Application
Integration
Data Management
Big Data
Security
Business Analytics
Internet of Things
Support
IBM
Third Party
Experimental
Beta
Starters // Choose a package of sample code and services, or start from scratch
Boilerplates
Get started with a new app, now
Internet of Things Foun...
IBM
Java Cache Web Starter
IBM
Java Cloudant Web Sta...
IBM
Mobile Cloud
IBM
IBM
Node.js Cloudant DB ...
IBM
User Modeling Java We...
IBM
IBM
User Modeling Ruby W...
IBM
Node-RED Starter
Community
Vaadin Rich Web Starter
Community
Runtimes
Run an app in the language of your choice
Liberty for Java™
IBM
SDK for Node.js™
View More
Ruby on Rails
Community
Ruby Sinatra
Community
Bring Your Buildpack
Community
Choose Node.js

Create App


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
Back to Starters
.js
SDK for Node.js™
IBM
VERSION
1.1
TYPE
Application
VIEW DOCS
Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.
Pick a plan
Monthly prices shown are for country or regi
Plan
Features
Default
vGB-Hour
TERMS
Start with a runtime:
Name:
moodlocator
Host:
moodlocator
. mybluemix.net
Selected Plan:
Default
CREATE
Enter a unique name

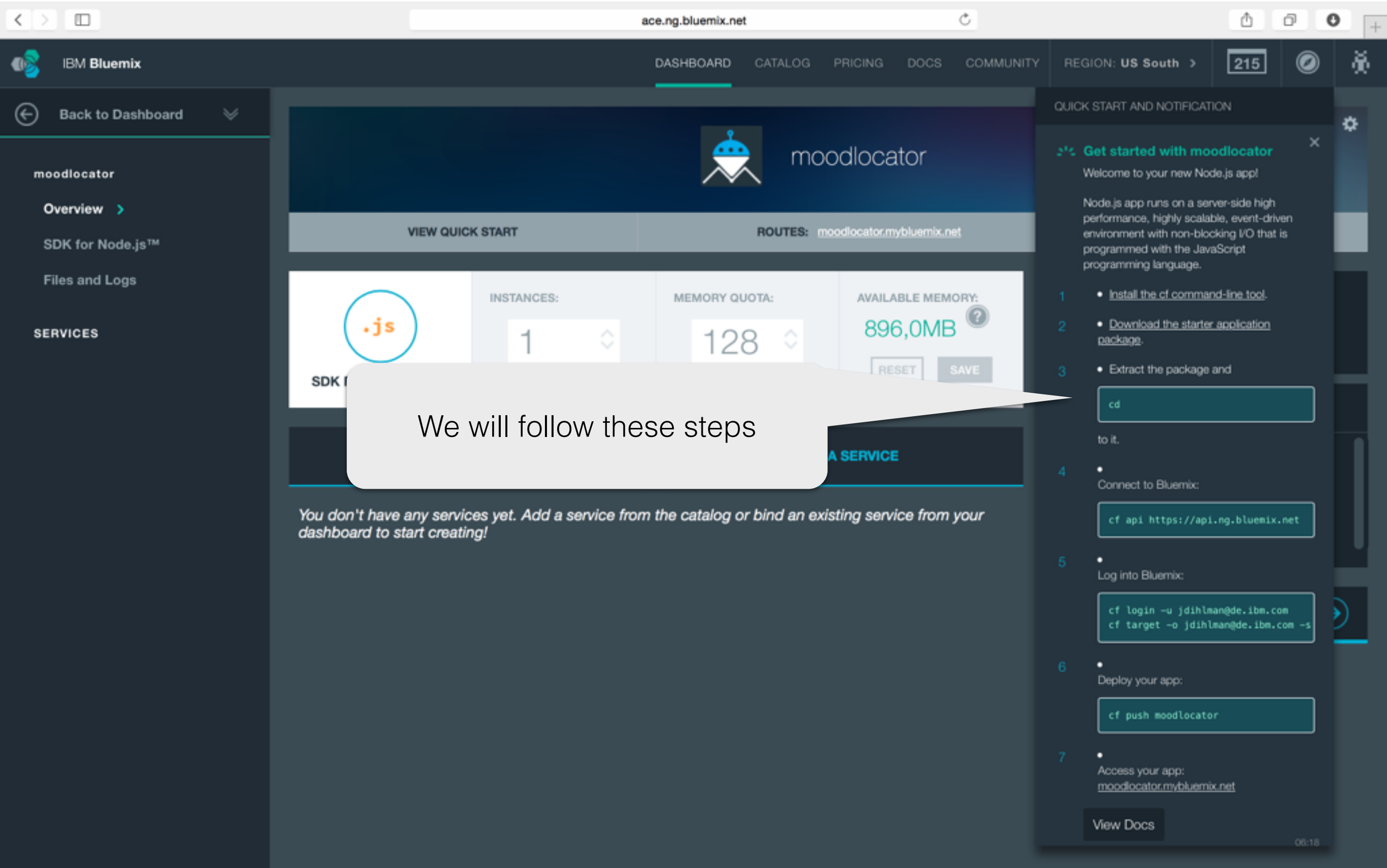
App Info


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
'moodlocator' starting
Your application is starting.
06:18
Back to Dashboard
moodlocator
Overview
SDK for Node.js™
Files and Logs
SERVICES
moodlocator
VIEW QUICK START
ROUTES: moodlocator.mybluemix.net
Check out the SDK for Node.js™ quick start to get started with your new app.
.js
SDK FOR NODE.JS™
1
MEMORY QUOTA:
128
(MB per Instance)
AVAILABLE MEMORY:
1,000GB
RESET
SAVE
APP HEALTH
START
Stopped
ACTIVITY LOG
26.11.14 06:18 jdihlman@de.ibm.com updated moodlocator app
• changed routes
26.11.14 06:18 jdihlman@de.ibm.com created moodlocator app
ADD A SERVICE
BIND A SERVICE
You don't have any services yet. Add a service from the catalog or bind an existing service from your dashboard to start creating!
Estimate the cost of this app

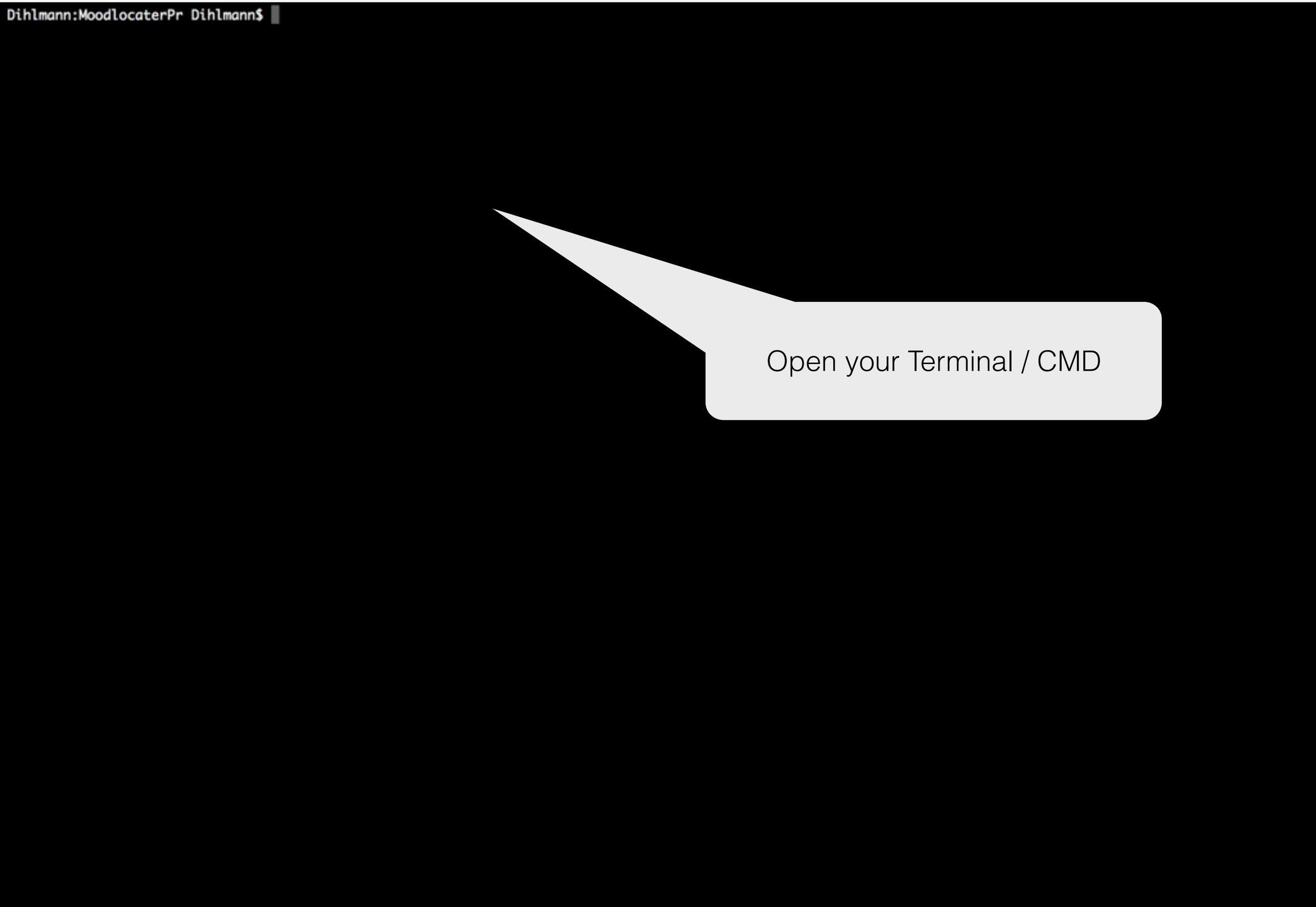
1. Upload an alone standing application to Bluemix

Go back to GitHub and take a look at section 6

Quick Start


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
Back to Dashboard
moodlocator
Overview
SDK for Node.js™
Files and Logs
SERVICES
moodlocator
VIEW QUICK START
ROUTES: moodlocator.mybluemix.net
INSTANCES:
1
MEMORY QUOTA:
128
AVAILABLE MEMORY:
896,0MB
RESET
SAVE
SDK
A SERVICE
We will follow these steps
You don't have any services yet. Add a service from the catalog or bind an existing service from your dashboard to start creating!
QUICK START AND NOTIFICATION
Get started with moodlocator
Welcome to your new Node.js app!
Node.js app runs on a server-side high performance, highly scalable, event-driven environment with non-blocking I/O that is programmed with the JavaScript programming language.
1 • Install the cf command-line tool.
2 • Download the starter application package.
3 • Extract the package and
cd
to it.
4 • Connect to Bluemix:
cf api https://api.ng.bluemix.net
5 • Log into Bluemix:
cf login -u jdihlman@de.ibm.com
cf target -o jdihlman@de.ibm.com -s
6 • Deploy your app:
cf push moodlocator
7 • Access your app:
moodlocator.mybluemix.net
View Docs
06:18

Terminal



cd MoodLocator

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$
```

Instead of „MoodlocaterPr“ write „node“

Connect to Bluemix

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$
```

Copy & Paste

Login

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf login -u jdihlman@de.ibm.com
API endpoint: https://api.ng.bluemix.net

Password>
Authenticating...
OK

Targeted org jdihlman@de.ibm.com

Targeted space dev


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$
```

Enter the password

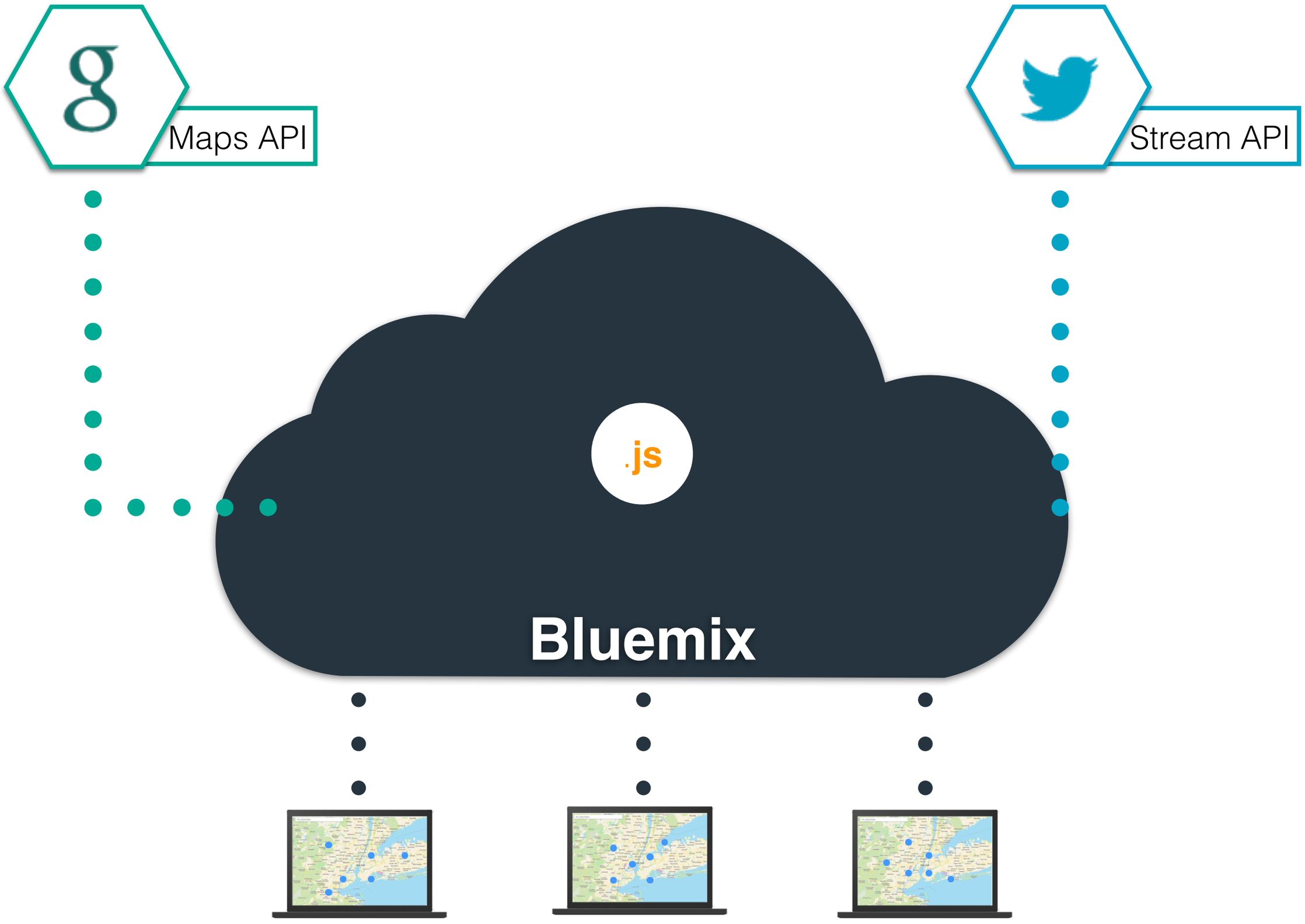
cf push

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf login -u jdihlman@de.ibm.com
API endpoint: https://api.ng.bluemix.net

Password>
Authenticating...
OK

Targeted org jdihlman@de.ibm.com

Targeted space dev


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf push moodlocator
Using manifest file /Users/Dihlmann/Desktop/MoodlocaterPr/manifest.yml

Updating app moodlocator in org jdihlman@de.ibm.com / space dev as jdihlman@de.ibm.com...
OK

```

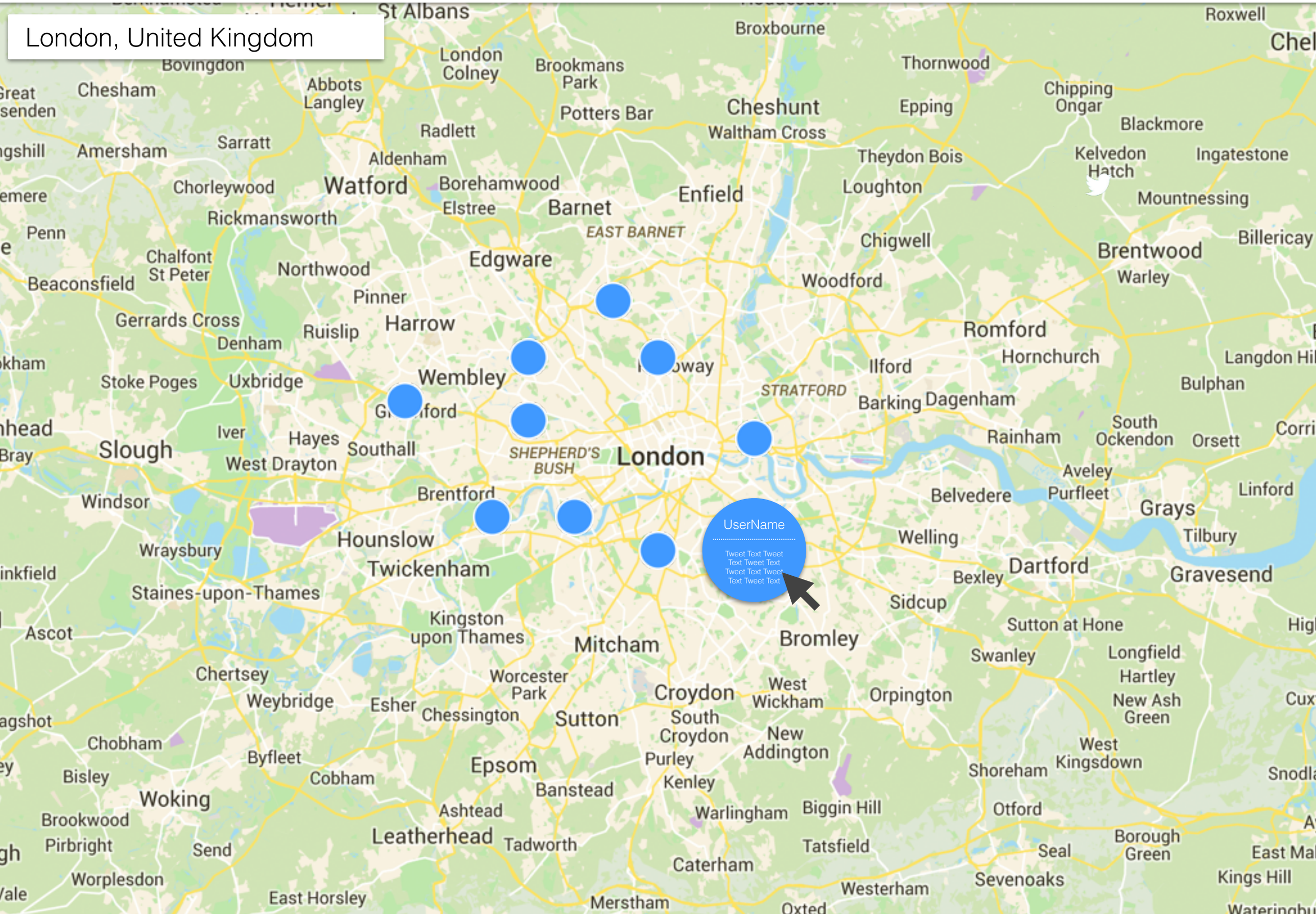
Instead of „moodlocator“ enter your application name

Your stand alone application is uploading. It may take 1minute

Structure

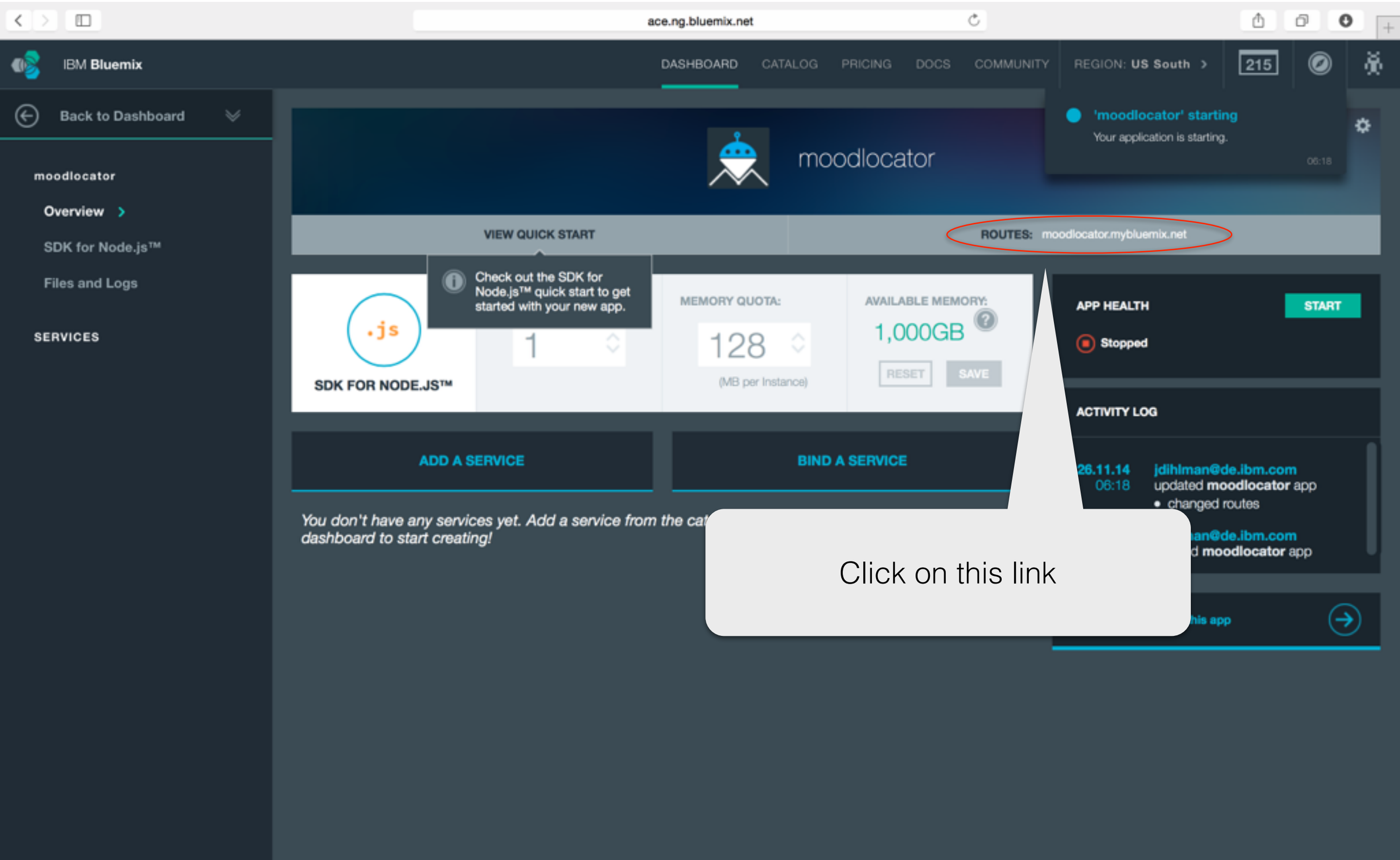


Twitter Locator



2. Show how to use the alone standing application

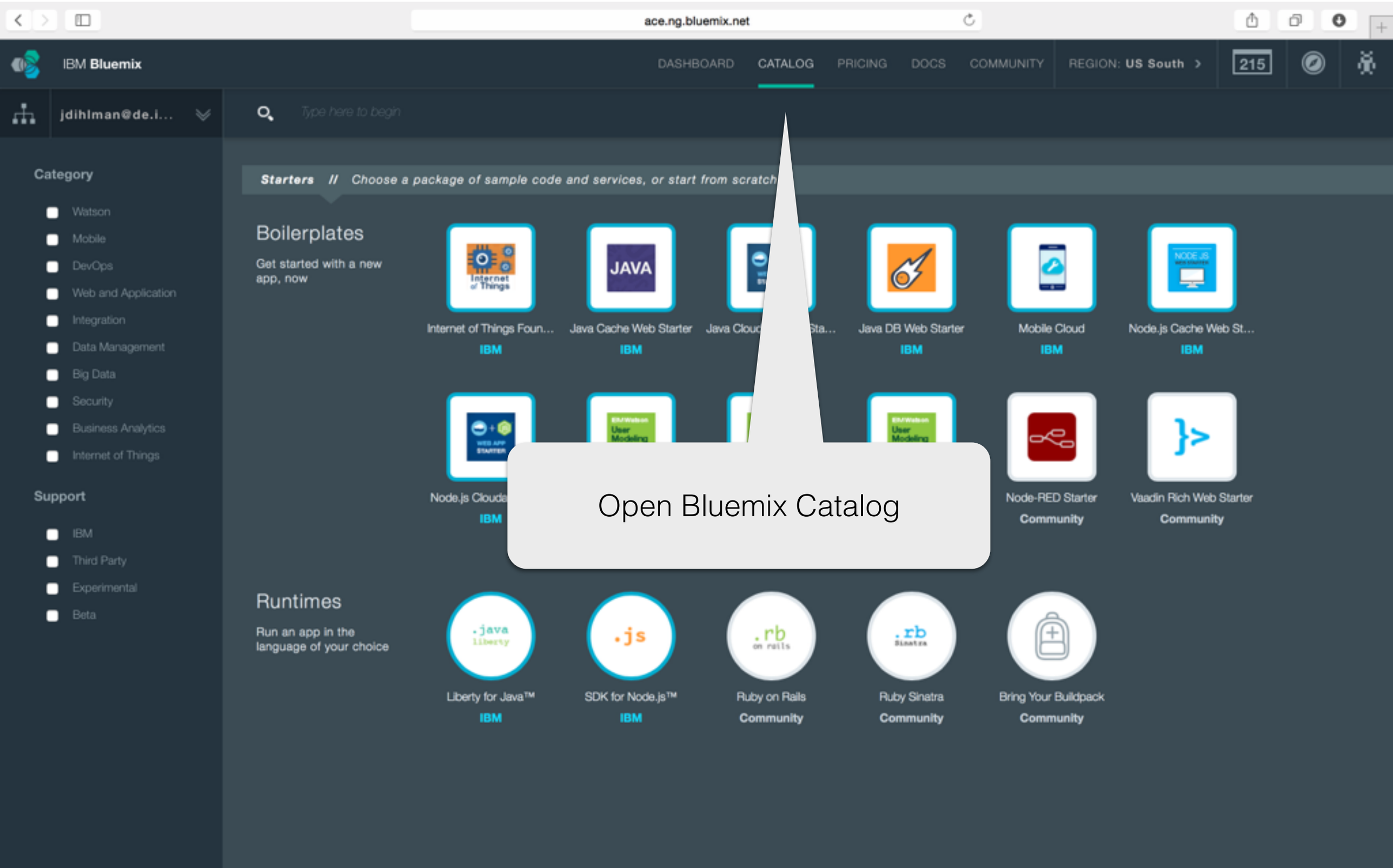
App Info


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
Back to Dashboard
moodlocator
Overview
SDK for Node.js™
Files and Logs
SERVICES
'moodlocator' starting
Your application is starting.
06:18
moodlocator
VIEW QUICK START
ROUTES: moodlocator.mybluemix.net
Check out the SDK for Node.js™ quick start to get started with your new app.
.js
SDK FOR NODE.JS™
MEMORY QUOTA:
1
128
(MB per Instance)
AVAILABLE MEMORY:
1,000GB
RESET
SAVE
APP HEALTH
START
Stopped
ACTIVITY LOG
26.11.14
06:18
jdihlman@de.ibm.com
updated moodlocator app
• changed routes
ADD A SERVICE
BIND A SERVICE
You don't have any services yet. Add a service from the cat
dashboard to start creating!
Click on this link

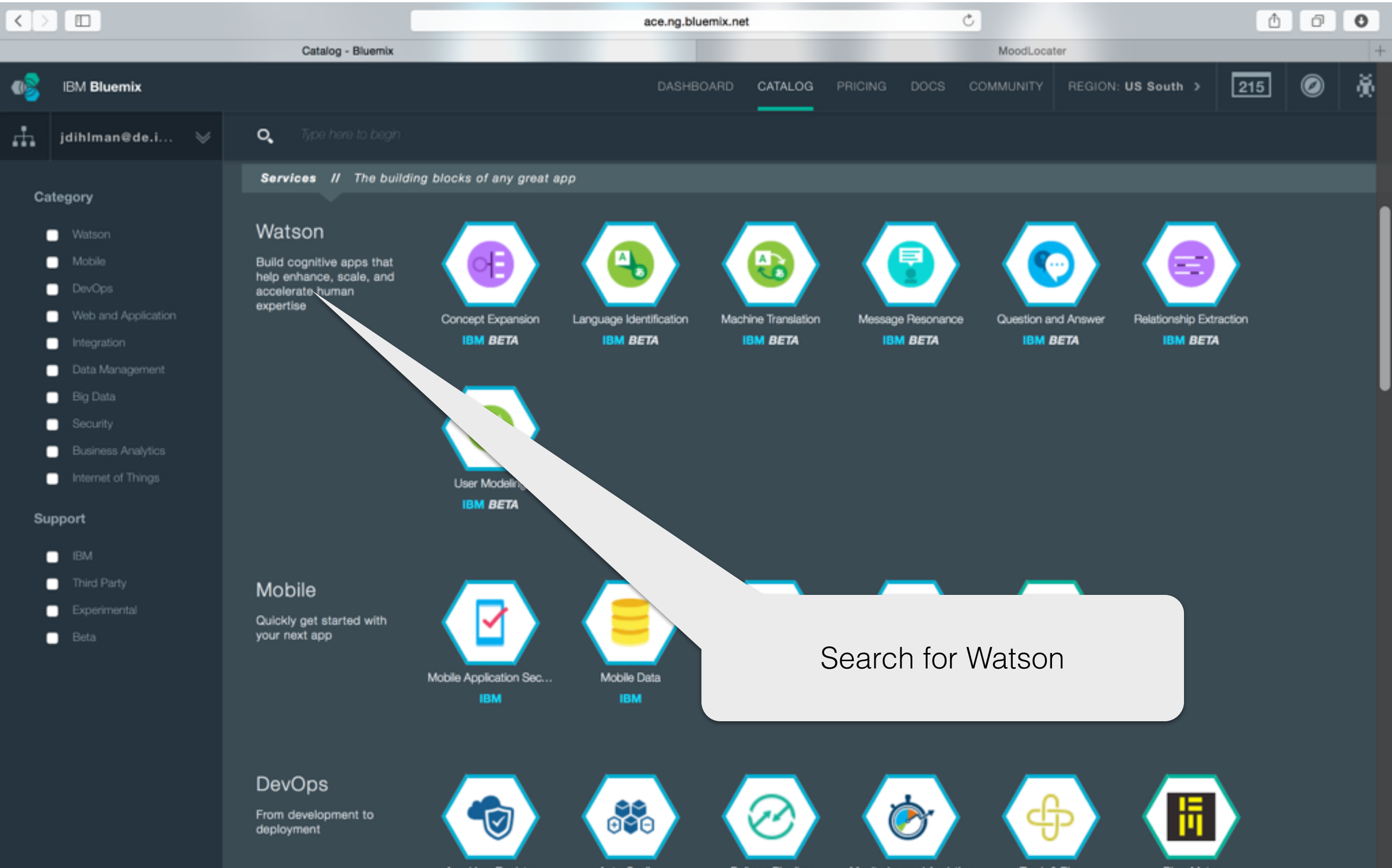
3. Add the User Modeling Service to it

Go back to GitHub and take a look at section 5

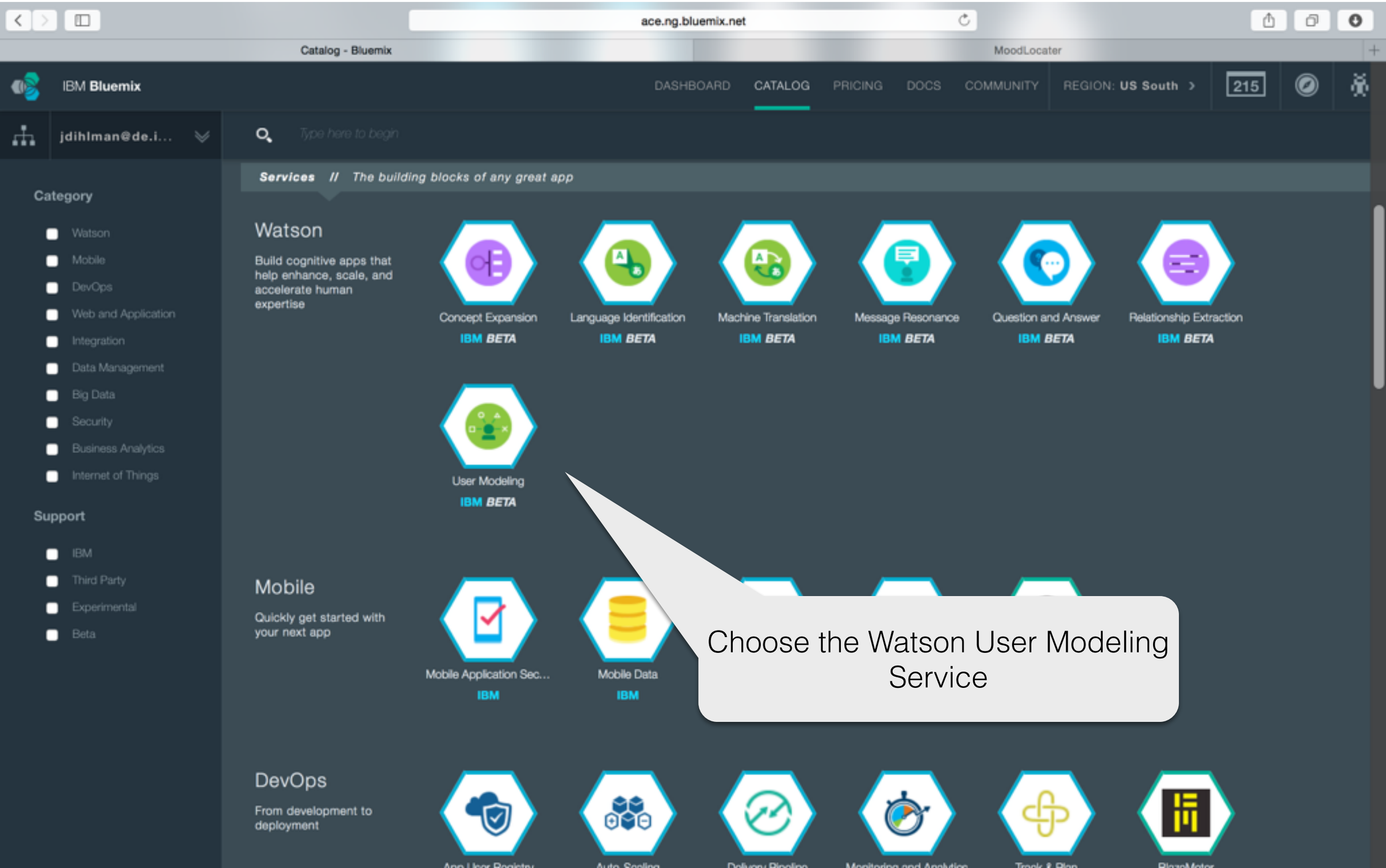
Catalog


ace.ng.bluemix.net
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
jdihlman@de.i...
Type here to begin
Category
Watson
Mobile
DevOps
Web and Application
Integration
Data Management
Big Data
Security
Business Analytics
Internet of Things
Support
IBM
Third Party
Experimental
Beta
Starters // Choose a package of sample code and services, or start from scratch
Boilerplates
Get started with a new app, now
Internet of Things Foun...
IBM
Java Cache Web Starter
IBM
Java DB Web Starter
IBM
Mobile Cloud
IBM
Node.js Cache Web St...
IBM
Node-RED Starter
Community
Vaadin Rich Web Starter
Community
Open Bluemix Catalog
Runtimes
Run an app in the language of your choice
Liberty for Java™
IBM
SDK for Node.js™
IBM
Ruby on Rails
Community
Ruby Sinatra
Community
Bring Your Buildpack
Community

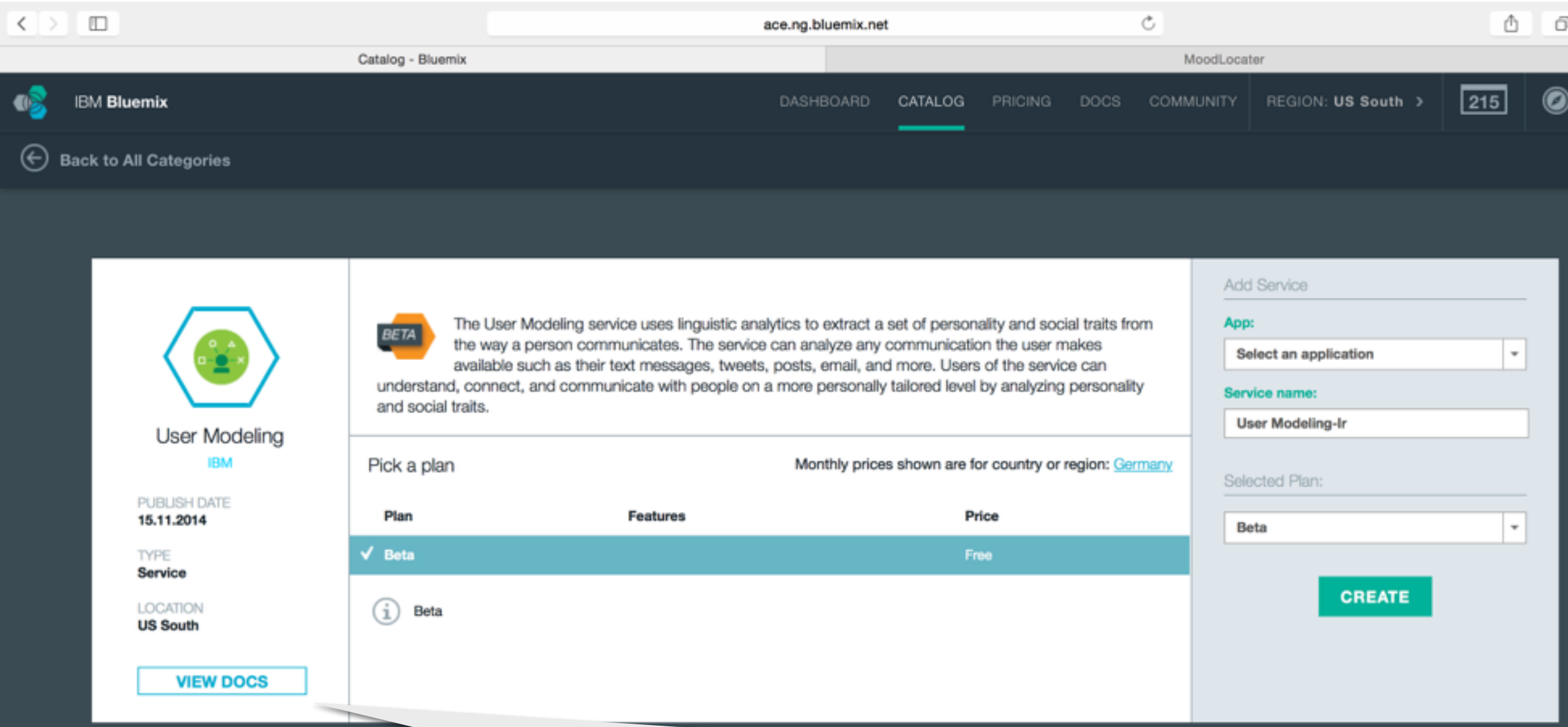
Services


ace.ng.bluemix.net
Catalog - Bluemix
MoodLocater
IBM Bluemix
DASHBOARD
CATALOG
PRICING
DOCS
COMMUNITY
REGION: US South
215
jdihlman@de.i...
Type here to begin
Services // The building blocks of any great app
Category
Watson
Mobile
DevOps
Web and Application
Integration
Data Management
Big Data
Security
Business Analytics
Internet of Things
Support
IBM
Third Party
Experimental
Beta
Watson
Build cognitive apps that help enhance, scale, and accelerate human expertise
Concept Expansion
IBM BETA
Language Identification
IBM BETA
Machine Translation
IBM BETA
Message Resonance
IBM BETA
Question and Answer
IBM BETA
Relationship Extraction
IBM BETA
User Modeli
IBM BETA
Mobile
Quickly get started with your next app
Mobile Application Sec...
IBM
Mobile Data
IBM
DevOps
From development to deployment
Search for Watson

User Modelling

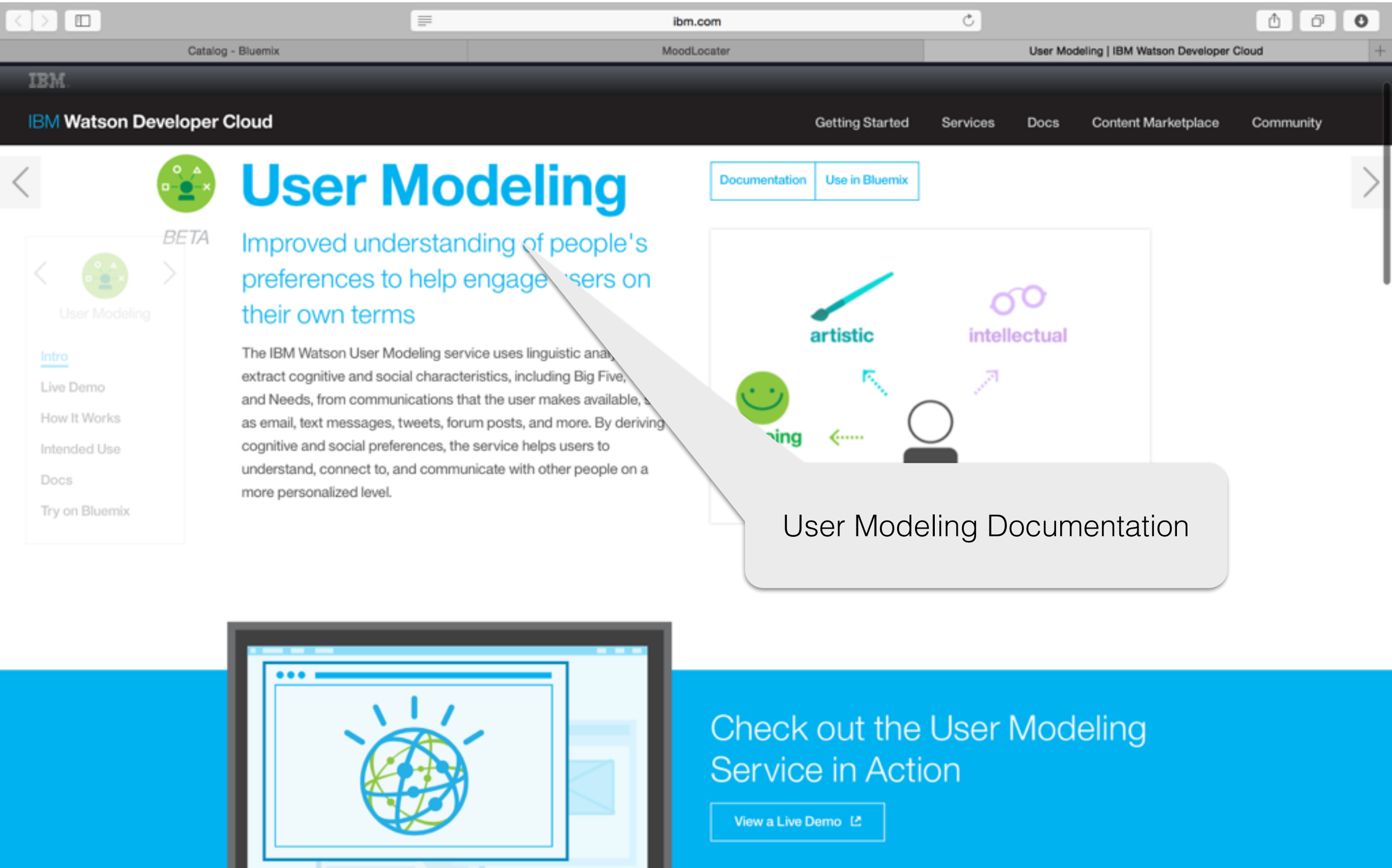


User Modeling Info



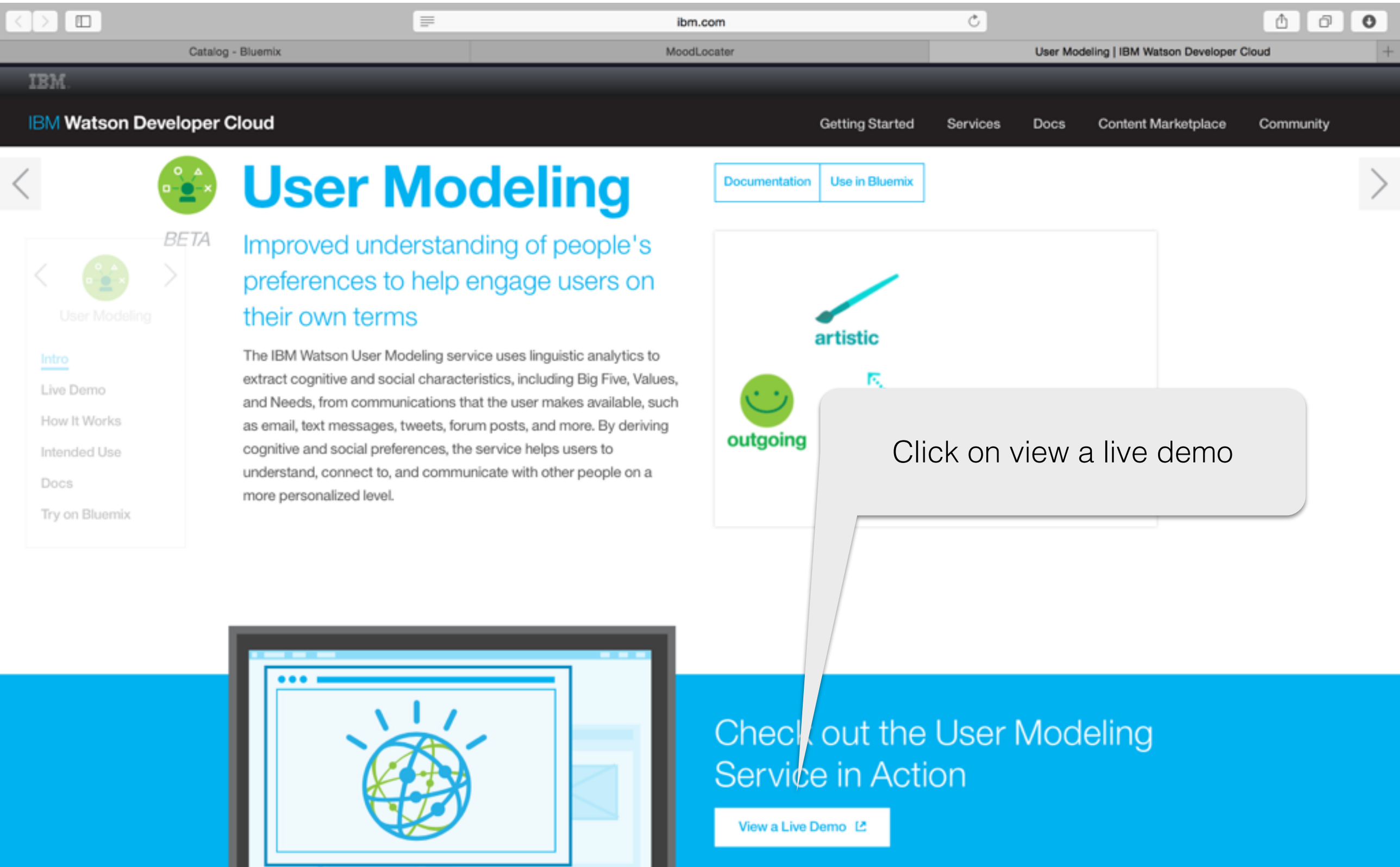
Click on View Docs

User Modeling Docs

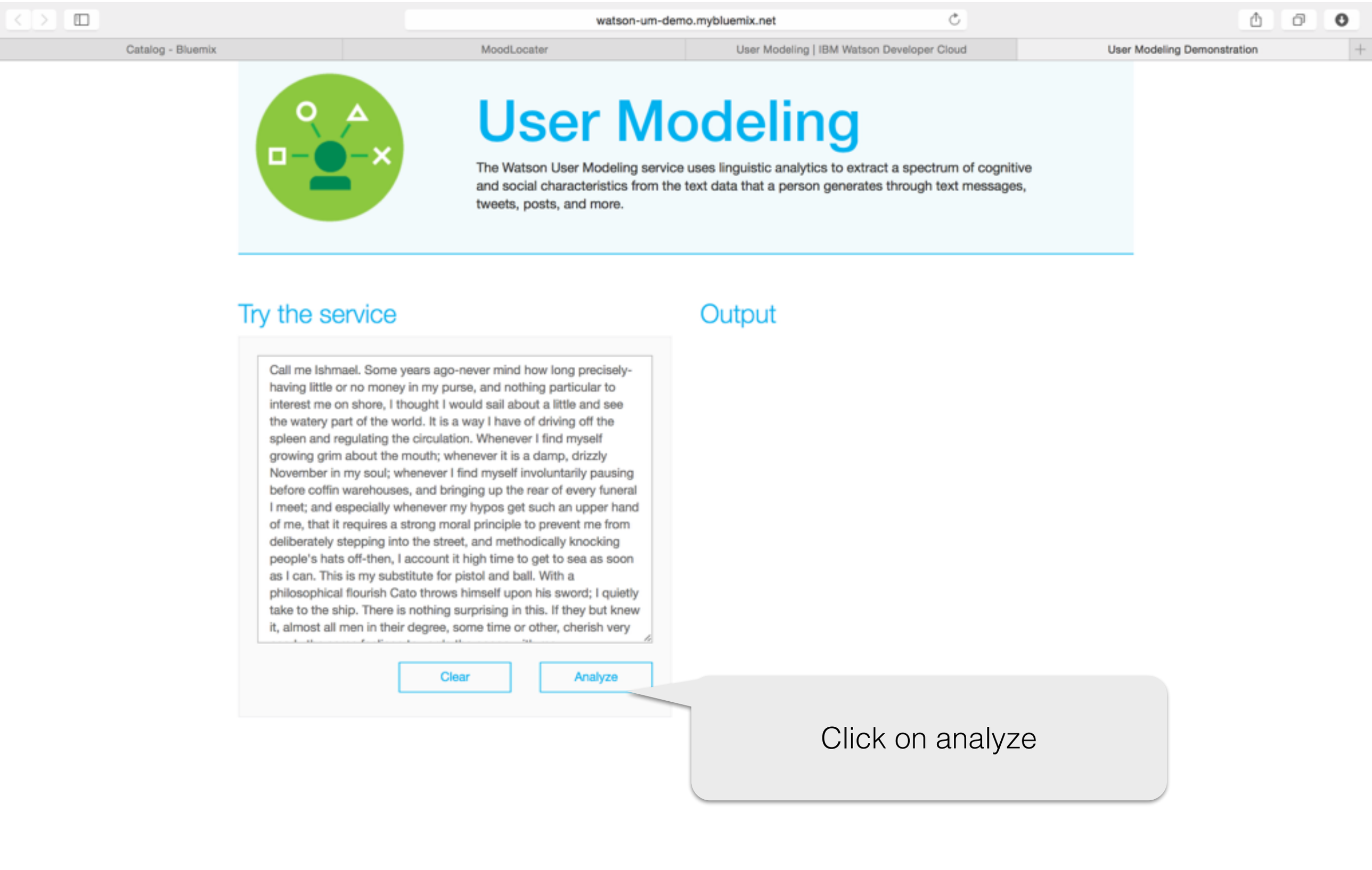


User Modeling Documentation

View a Live Demo



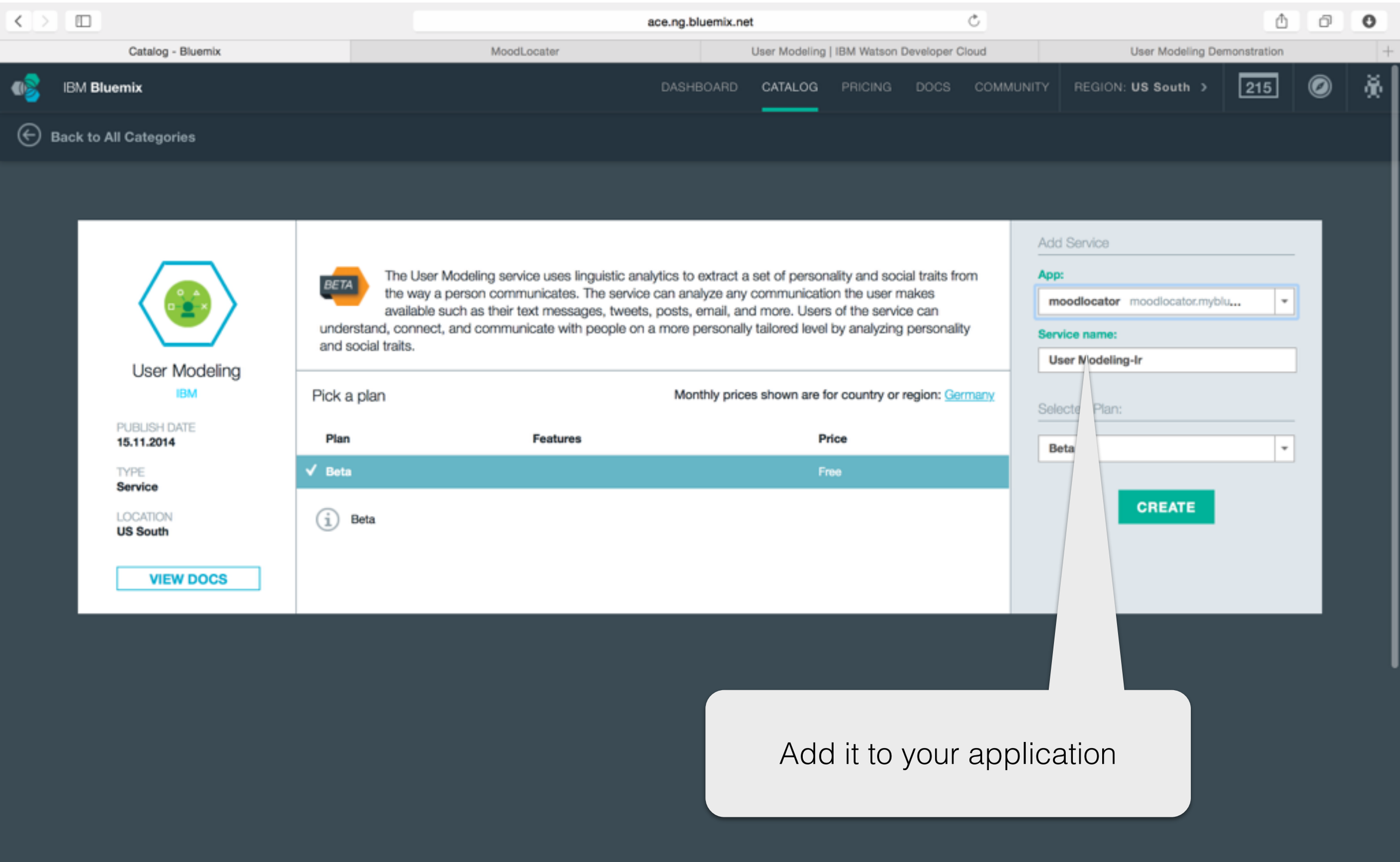
Live Demo


watson-um-demo.mybluemix.net
Catalog - Bluemix
MoodLocater
User Modeling | IBM Watson Developer Cloud
User Modeling Demonstration
User Modeling
The Watson User Modeling service uses linguistic analytics to extract a spectrum of cognitive and social characteristics from the text data that a person generates through text messages, tweets, posts, and more.
Try the service
Call me Ishmael. Some years ago-never mind how long precisely-having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off-then, I account it high time to get to sea as soon as I can. This is my substitute for pistol and ball. With a philosophical flourish Cato throws himself upon his sword; I quietly take to the ship. There is nothing surprising in this. If they but knew it, almost all men in their degree, some time or other, cherish very
Clear
Analyze
Output
Click on analyze

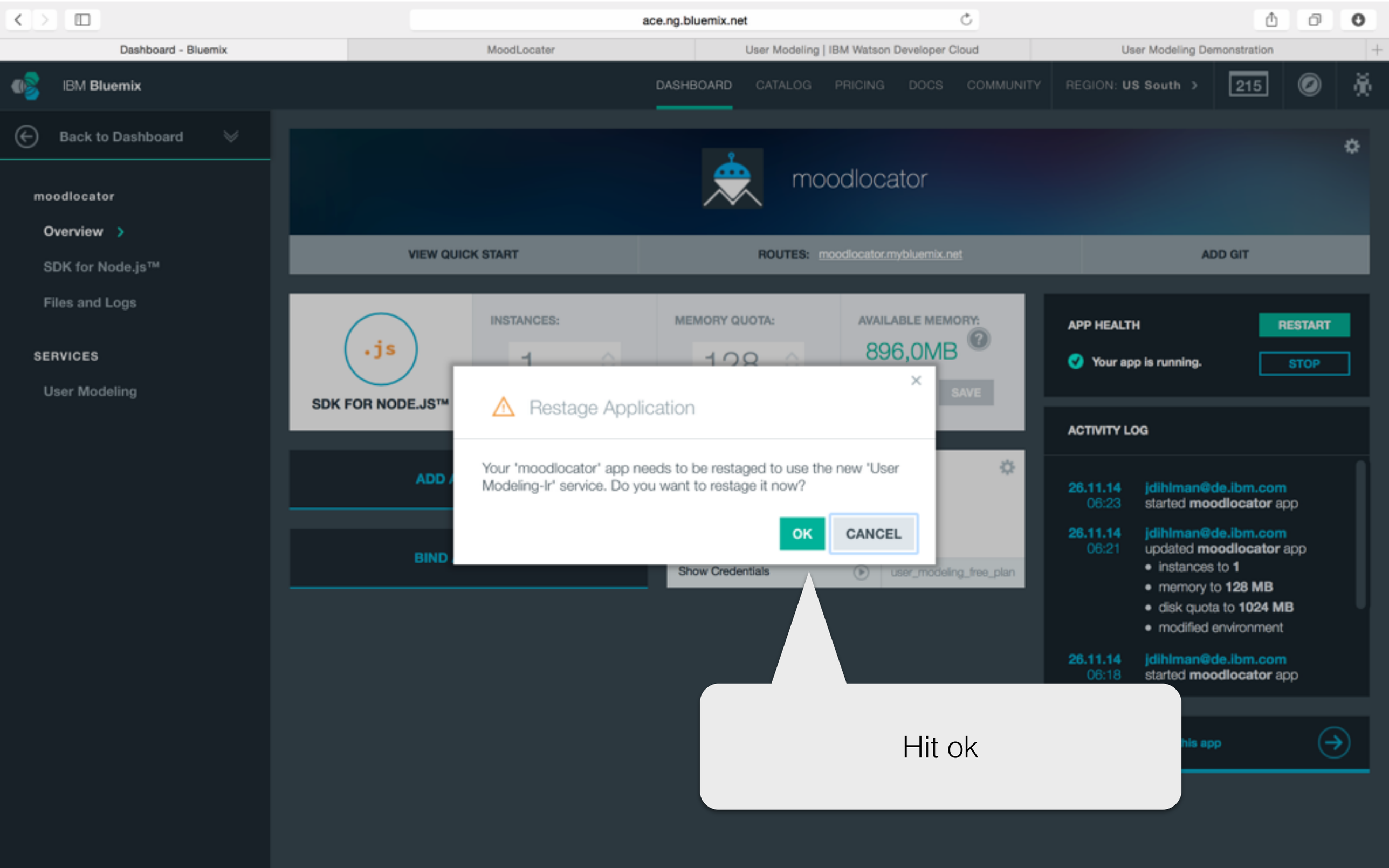
Analyze


watson-um-demo.mybluemix.net
Catalog - Bluemix
MoodLocater
User Modeling | IBM Watson Developer Cloud
User Modeling Demonstration
Values
Needs
Big 5
Openness (94%)
Openness to cha (98%)
Conscientiousness (48%)
Extraversion (17%)
Agreeableness (30%)
Emotional range (43%)
Harmony (94%)
Adventurousness (72%)
Artistic interests (19%)
Emotionality (32%)
Imagination (93%)
Intellect (94%)
Authority-challenging (93%)
Achievement striving (41%)
Cautiousness (98%)
Dutifulness (21%)
Orderliness (28%)
Self-discipline (13%)
Self-efficacy (88%)
Activity level (4%)
Assertiveness (12%)
Cheerfulness (11%)
Excitement-seeking (1%)
Outgoing (10%)
Gregariousness (2%)
Altruism (28%)
Cooperation (92%)
Modesty (4%)
Uncompromising (36%)
Sympathy (100%)
Trust (38%)
Fiery (46%)
Prone to worry (15%)
Melancholy (22%)
Immoderation (18%)
Self-consciousness (39%)
Susceptible to stress (26%)
Challenge (93%)
Closeness (80%)
Curiosity (24%)
Excitement (7%)
Harmony (94%)
Ideal (49%)
Liberty (28%)
Love (74%)
Practicality (35%)
Self-expression (15%)
Stability (21%)
Structure (42%)
Conservation (4%)
Hedonism (32%)
Self-enhancement (41%)
Self-transcendence (68%)

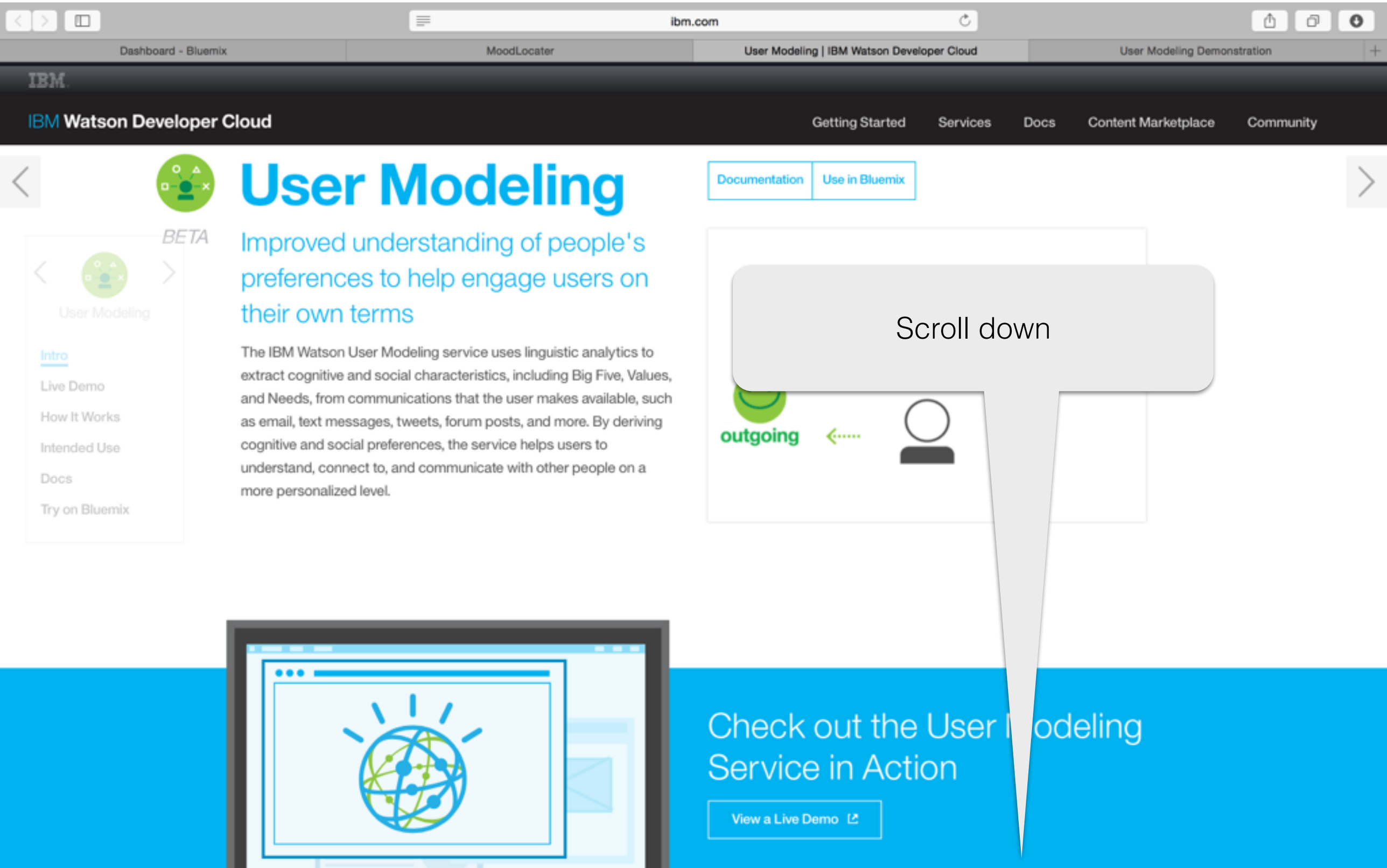
User Modeling Info



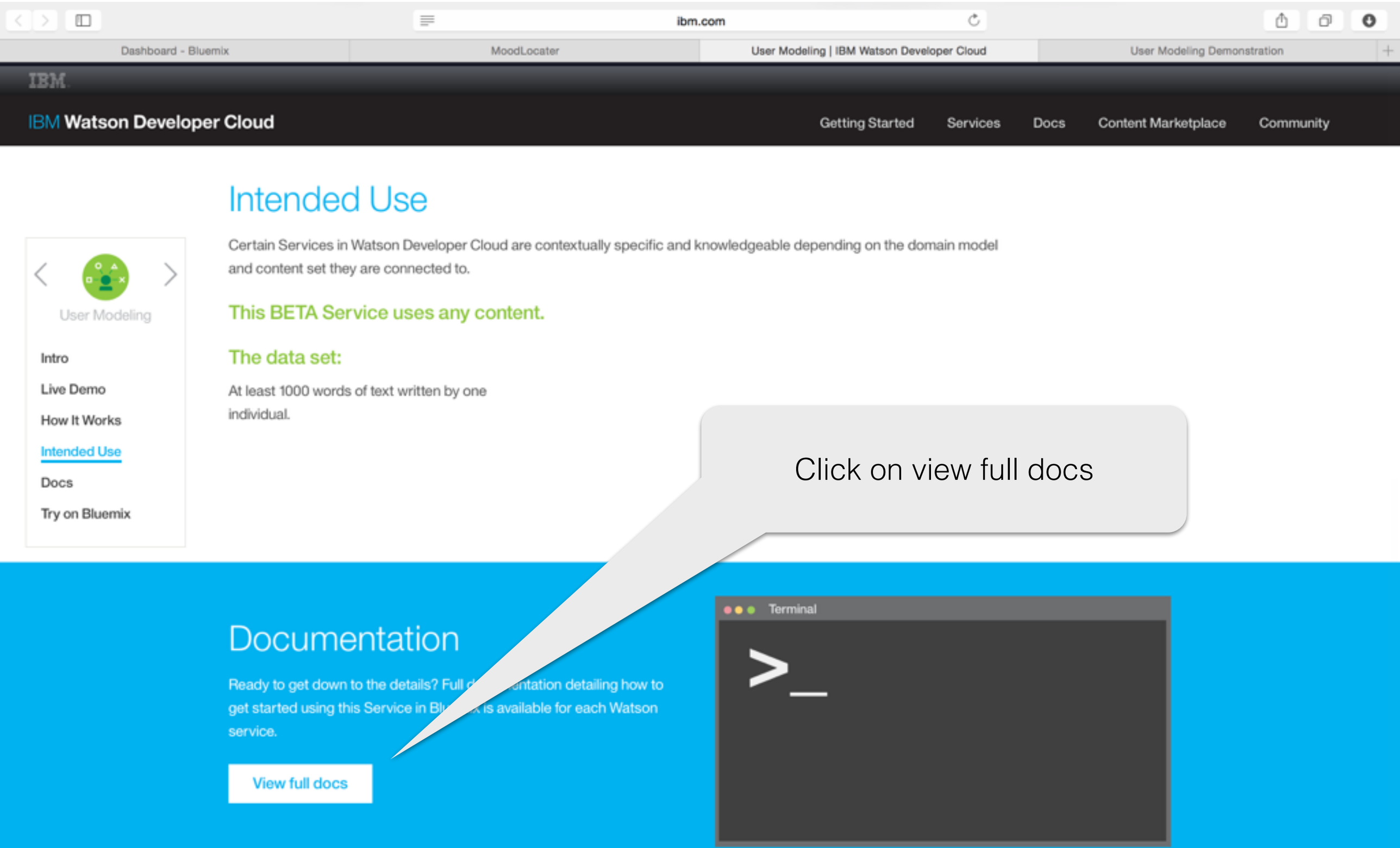
Add Service Process



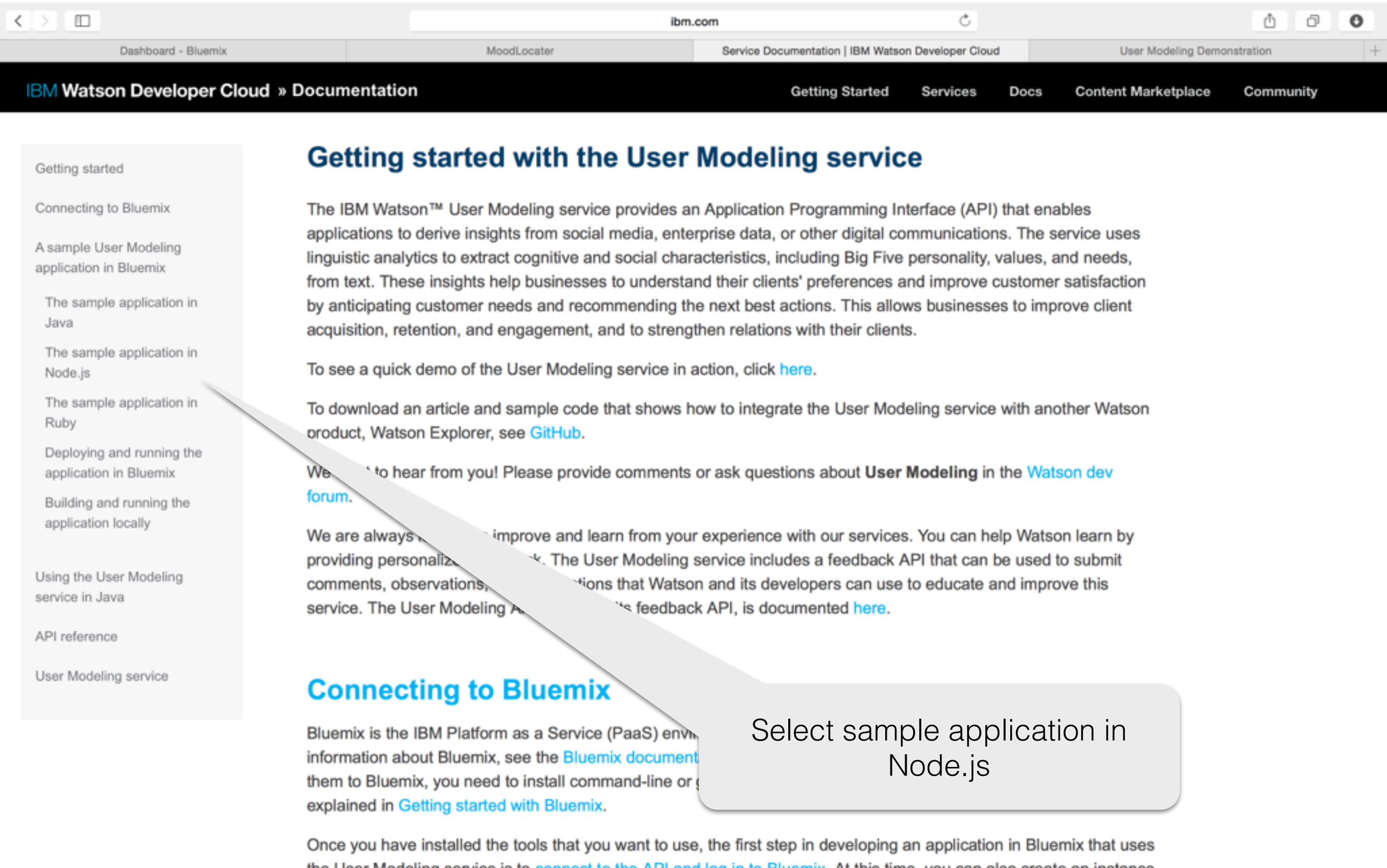
User Modeling Doc



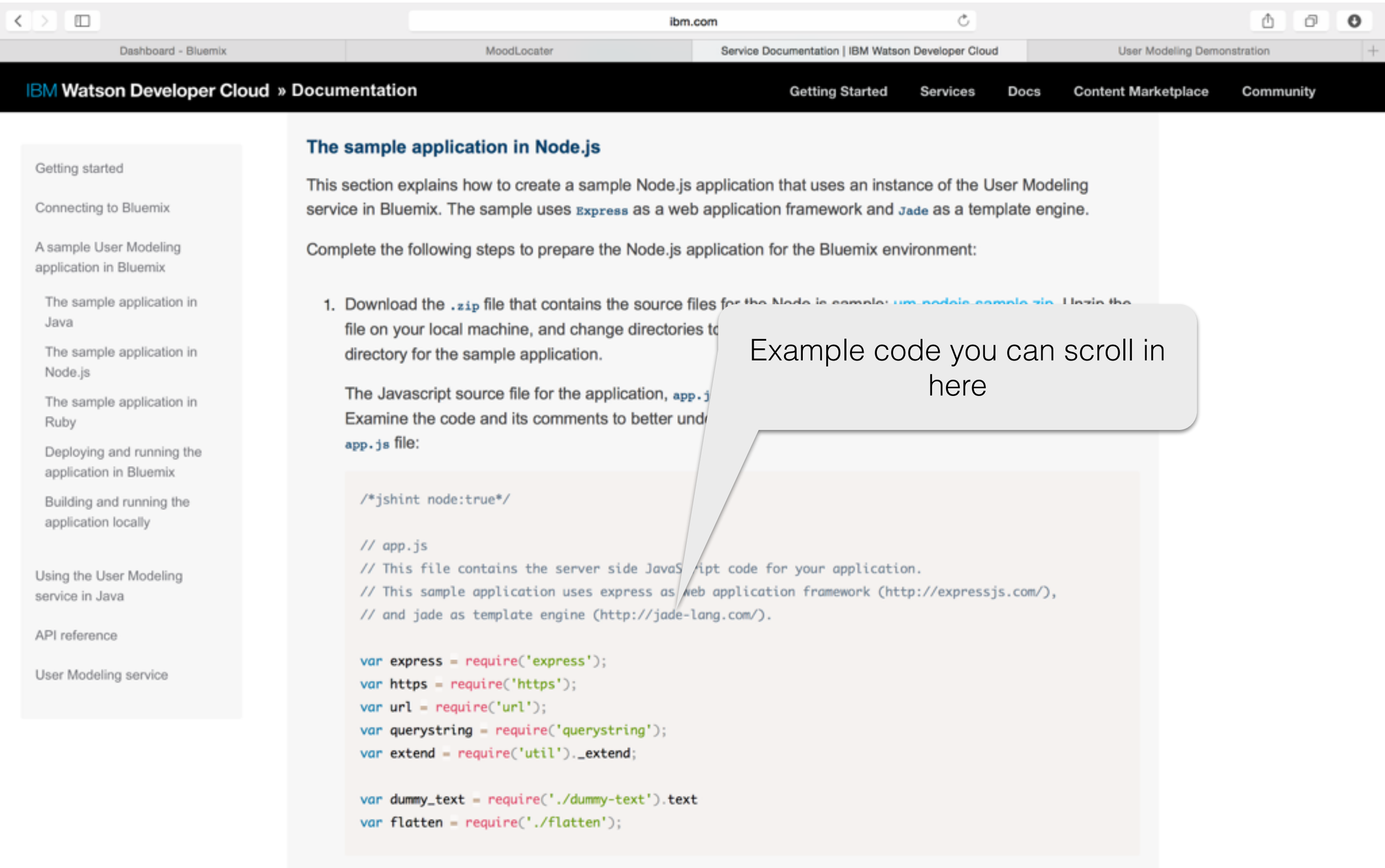
View full Docs



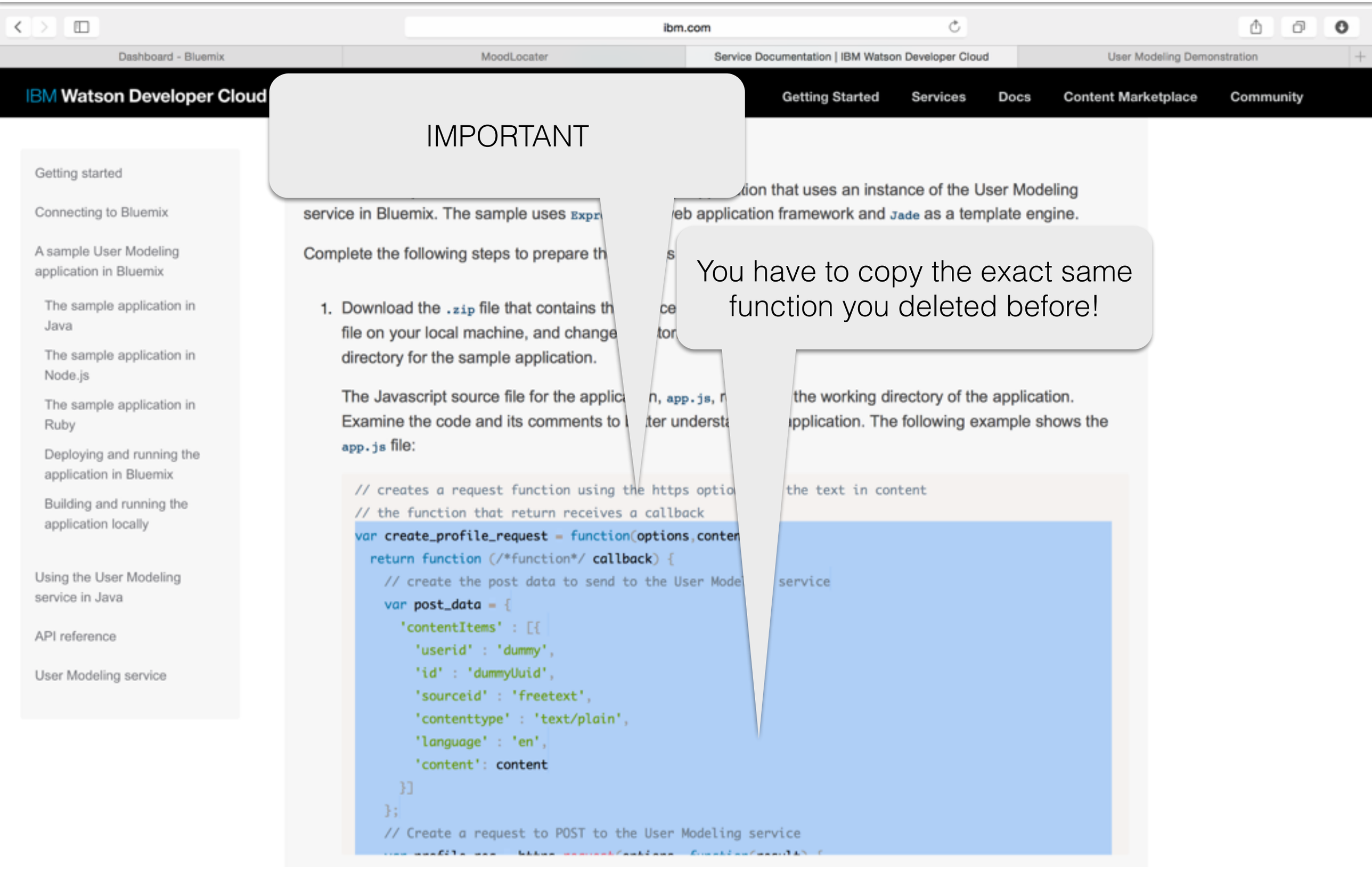
Watson Doc


ibm.com
Dashboard - Bluemix
MoodLocater
Service Documentation | IBM Watson Developer Cloud
User Modeling Demonstration
IBM Watson Developer Cloud » Documentation
Getting Started
Services
Docs
Content Marketplace
Community
Getting started
Connecting to Bluemix
A sample User Modeling application in Bluemix
The sample application in Java
The sample application in Node.js
The sample application in Ruby
Deploying and running the application in Bluemix
Building and running the application locally
Using the User Modeling service in Java
API reference
User Modeling service
Getting started with the User Modeling service
The IBM Watson™ User Modeling service provides an Application Programming Interface (API) that enables applications to derive insights from social media, enterprise data, or other digital communications. The service uses linguistic analytics to extract cognitive and social characteristics, including Big Five personality, values, and needs, from text. These insights help businesses to understand their clients' preferences and improve customer satisfaction by anticipating customer needs and recommending the next best actions. This allows businesses to improve client acquisition, retention, and engagement, and to strengthen relations with their clients.
To see a quick demo of the User Modeling service in action, click here.
To download an article and sample code that shows how to integrate the User Modeling service with another Watson product, Watson Explorer, see GitHub.
We to hear from you! Please provide comments or ask questions about User Modeling in the Watson dev forum.
We are always improve and learn from your experience with our services. You can help Watson learn by providing personali k. The User Modeling service includes a feedback API that can be used to submit comments, observations, tions that Watson and its developers can use to educate and improve this service. The User Modeling A s feedback API, is documented here.
Connecting to Bluemix
Bluemix is the IBM Platform as a Service (PaaS) envi information about Bluemix, see the Bluemix document them to Bluemix, you need to install command-line or explained in Getting started with Bluemix.
Once you have installed the tools that you want to use, the first step in developing an application in Bluemix that uses
Select sample application in Node.js

Sample Application



Copy & Paste



4. Activate the User Modeling Service

COPY THIS PART

You have to copy the exact same thing ——> out of the Watson Documentation into your application

Select it

Copy it

cmd c / strg c

We will undo everything what we did for preparation

```
//For presentation start delete this section

//Function you will find in the User Modeling documentation
var create_profile_request = function(options,content) {
  return function (/*function*/ callback) {
    // create the post data to send to the User Modeling service
    var post_data = {
      'contentItems' : [{
        'userid' : 'dummy',
        'id' : 'dummyUuid',
        'sourceid' : 'freetext',
        'contenttype' : 'text/plain',
        'language' : 'en',
        'content': content
      }]
    };
    // Create a request to POST to the User Modeling service
    var profile_req = https.request(options, function(result) {
      result.setEncoding('utf-8');
      var response_string = '';

      result.on('data', function(chunk) {
        response_string += chunk;
      });

      result.on('end', function() {

        if (result.statusCode != 200) {
          var error = JSON.parse(response_string);
          callback({'message': error.user_message}, null);
        } else
          callback(null,response_string);
      });
    });

    profile_req.on('error', function(e) {
      callback(e,null);
    });

    profile_req.write(JSON.stringify(post_data));
    profile_req.end();
  }
};

//For presentation stop deleting this section
```

Open app.js

Open the app.js file with a text editor you will find it in

/ node / app.js

Scroll to line 238

Paste the whole code at the end of your document, where you deleted it before.

Stay in app.js

Scroll to line 189

You will find something that looks like that, delete this bracket „ /* “ in line 190 and this bracket „ */ “ in line 211. I marked the position red where to add them.

Before

```
189     //For presentation start commenting here
190     /*
191     //Create a profile request with the text and the https options and call it
192     create_profile_request(profile_options,textToWatson)(function(error,profile_string) {
193         if (error) console.log(error.message);
194         else {
195             //Parse the Profile and format it
196             var profile_json = JSON.parse(profile_string);
197             var flat_traits = flatten.flat(profile_json.tree);
198
199             //Sent Watson Data to Client
200             var watsonObj = {
201                 id: msg.id,
202                 watsonJSON: flat_traits
203             }
204
205
206             //Send Watson Data to Client
207             socket.emit('watson message', watsonObj);
208             //console.log(flat_traits);
209         }
210     }); //For presentation stop commenting here
211     */
212 }
```

After

```
189     //For presentation start commenting here
190
191     //Create a profile request with the text and the https options and call it
192     create_profile_request(profile_options,textToWatson)(function(error,profile_string) {
193         if (error) console.log(error.message);
194         else {
195             //Parse the Profile and format it
196             var profile_json = JSON.parse(profile_string);
197             var flat_traits = flatten.flat(profile_json.tree);
198
199             //Sent Watson Data to Client
200             var watsonObj = {
201                 id: msg.id,
202                 watsonJSON: flat_traits
203             }
204
205
206             //Send Watson Data to Client
207             socket.emit('watson message', watsonObj);
208             //console.log(flat_traits);
209         }
210     }); //For presentation stop commenting here
211
212 }
```

Save your changes

Open style.css

Open the style.css file with a text editor you will find it in

/ node / public / stylesheets / style.css

Scroll to the bottom of the page

You will find something that looks like that, add this bracket „ /* “ in line 449 and this bracket „ */ “ in line 452. I marked the position red where to add them.

Before

```
448   /* For Presentation activate oder deactivate the Section below */
449
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452
```

After

```
448   /* For Presentation activate oder deactivate the Section below */
449   /*
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452   */
```

Save your changes. Perfect to understand what it does, it enables or disables the white Watson User Modeling bar at the bottom of the page.

Open style.css

Open the style.css file with a text editor you will find it in

/ node / public / stylesheets / style.css

Scroll to the bottom of the page

You will find something that looks like that, add this bracket „ /* “ in line 449 and this bracket „ */ “ in line 452. I marked the position red where to add them.

Before

```
448   /* For Presentation activate oder deactivate the Section below */
449
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452
```

After

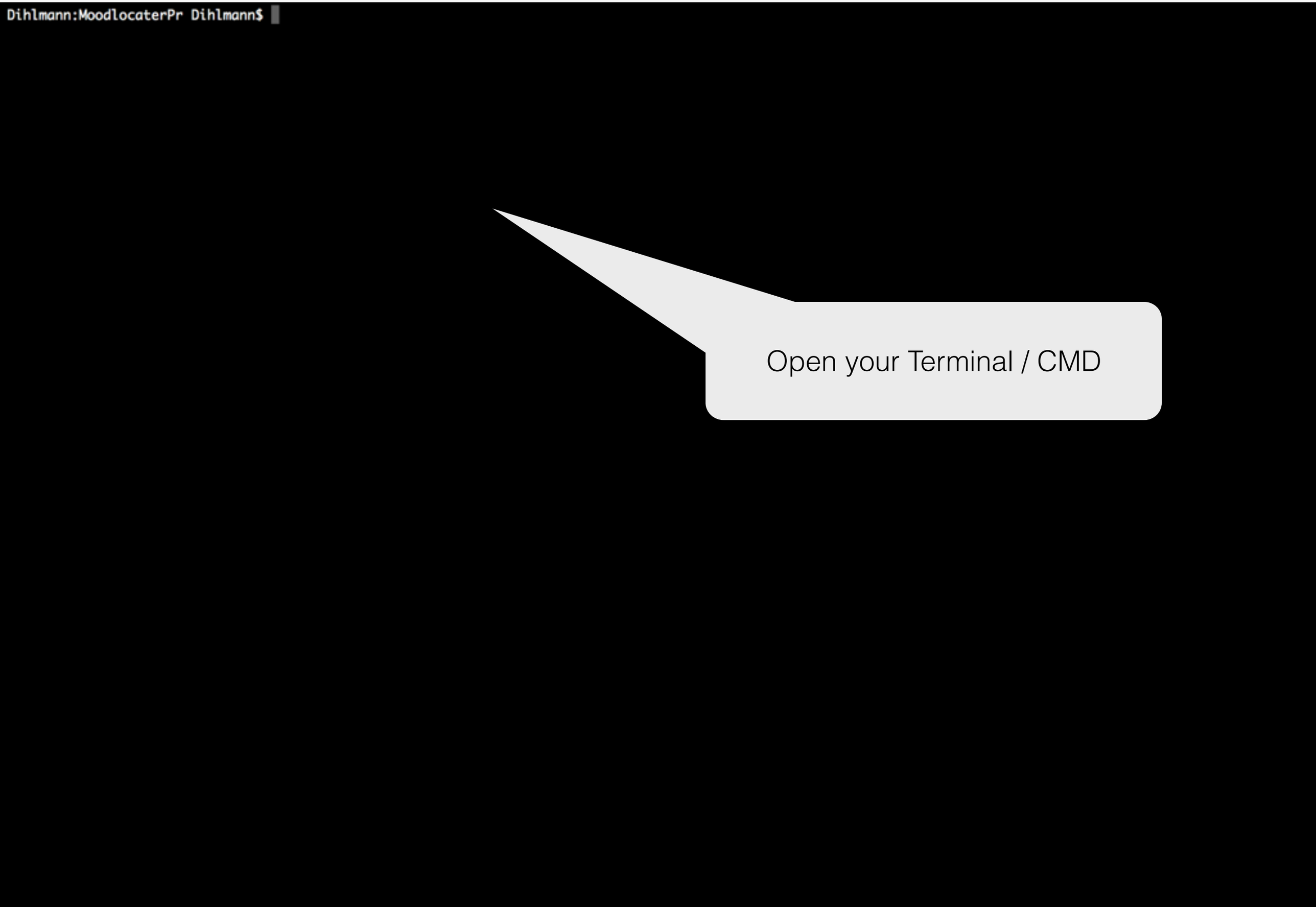
```
448   /* For Presentation activate oder deactivate the Section below */
449   /*
450   #footer{ visibility: hidden;}
451   #map_canvas{ height: 100%;}
452   */
```

Save your changes.

5. Upload the application

Go back to GitHub and take a look at section 6

Terminal



cd MoodLocator

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$
```

Instead of „MoodlocaterPr“ write „node“

Connect to Bluemix

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$
```

Copy & Paste

Login

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf login -u jdihlman@de.ibm.com
API endpoint: https://api.ng.bluemix.net

Password>
Authenticating...
OK

Targeted org jdihlman@de.ibm.com

Targeted space dev


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$
```

Enter the password

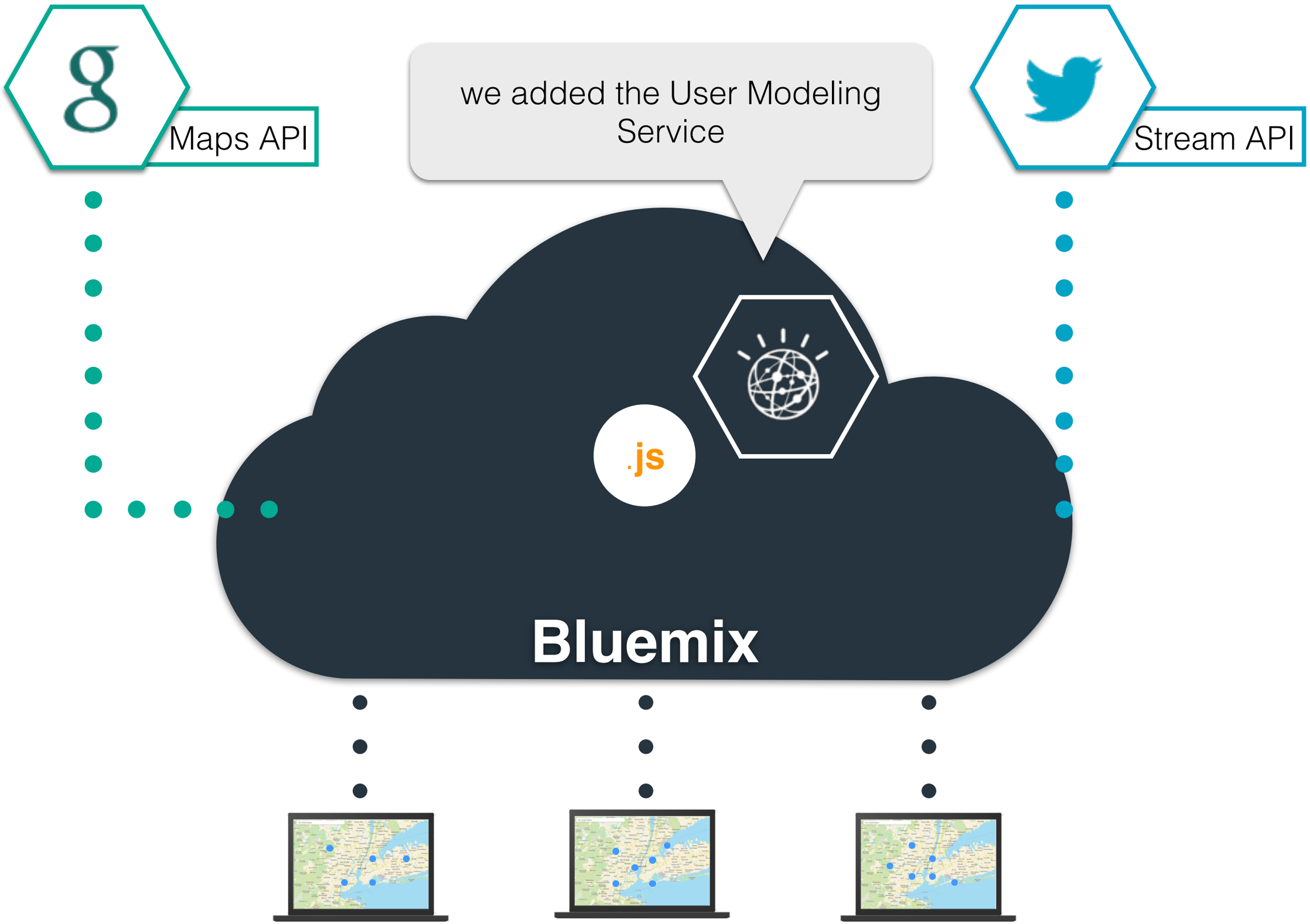
cf push

```
Dihlmann:~ Dihlmann$ cd Desktop/
Dihlmann:Desktop Dihlmann$ cd MoodlocaterPr/
Dihlmann:MoodlocaterPr Dihlmann$ cf api https://api.ng.bluemix.net
Setting api endpoint to https://api.ng.bluemix.net...
OK


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf login -u jdihlman@de.ibm.com
API endpoint: https://api.ng.bluemix.net

Password>
Authenticating...
OK

Targeted org jdihlman@de.ibm.com

Targeted space dev


API endpoint:   https://api.ng.bluemix.net (API version: 2.14.0)
User:           jdihlman@de.ibm.com
Org:            jdihlman@de.ibm.com
Space:          dev
Dihlmann:MoodlocaterPr Dihlmann$ cf push moodlocator
Using manifest file /Users/Dihlmann/Desktop/MoodlocaterPr/manifest.yml

Updating app moodlocator in org jdihlman@de.ibm.com / space dev as jdihlman@de.ibm.com...
OK

```

Instead of „moodlocator“ enter your application name

Your stand alone application is uploading. It may take 1minute

Structure


Maps API
we added the User Modeling Service
Stream API
.js
Bluemix

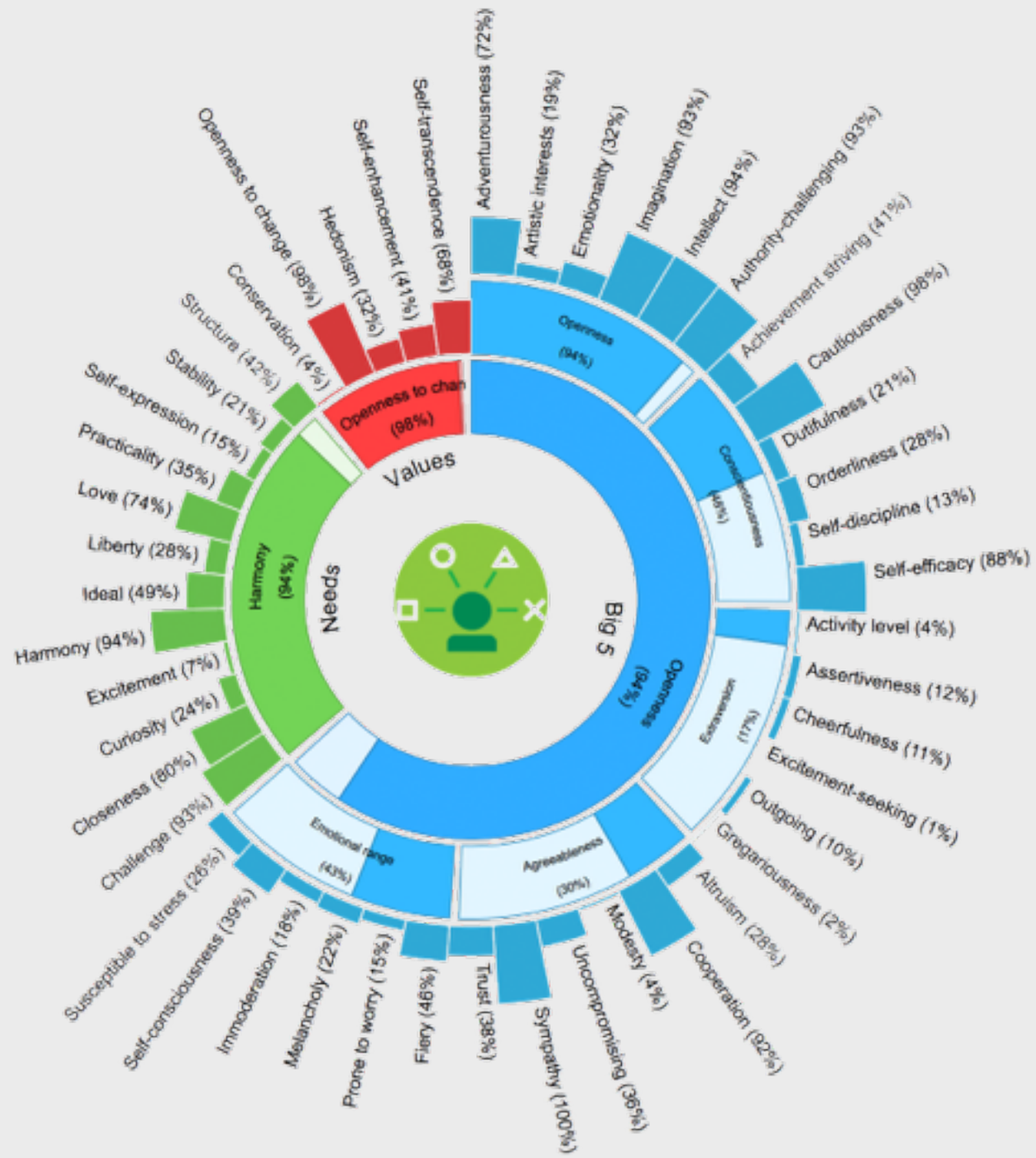
User Modeling Service

Text - at least 100 Words

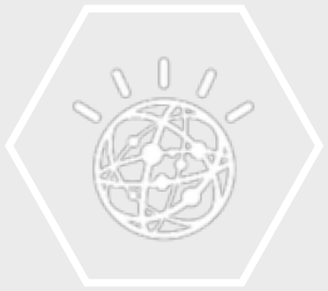
BETA:
Only english
Text

Get a fully character analysis from the author of the text

Possible character of the author



User Modeling Service



All Tweets from one city

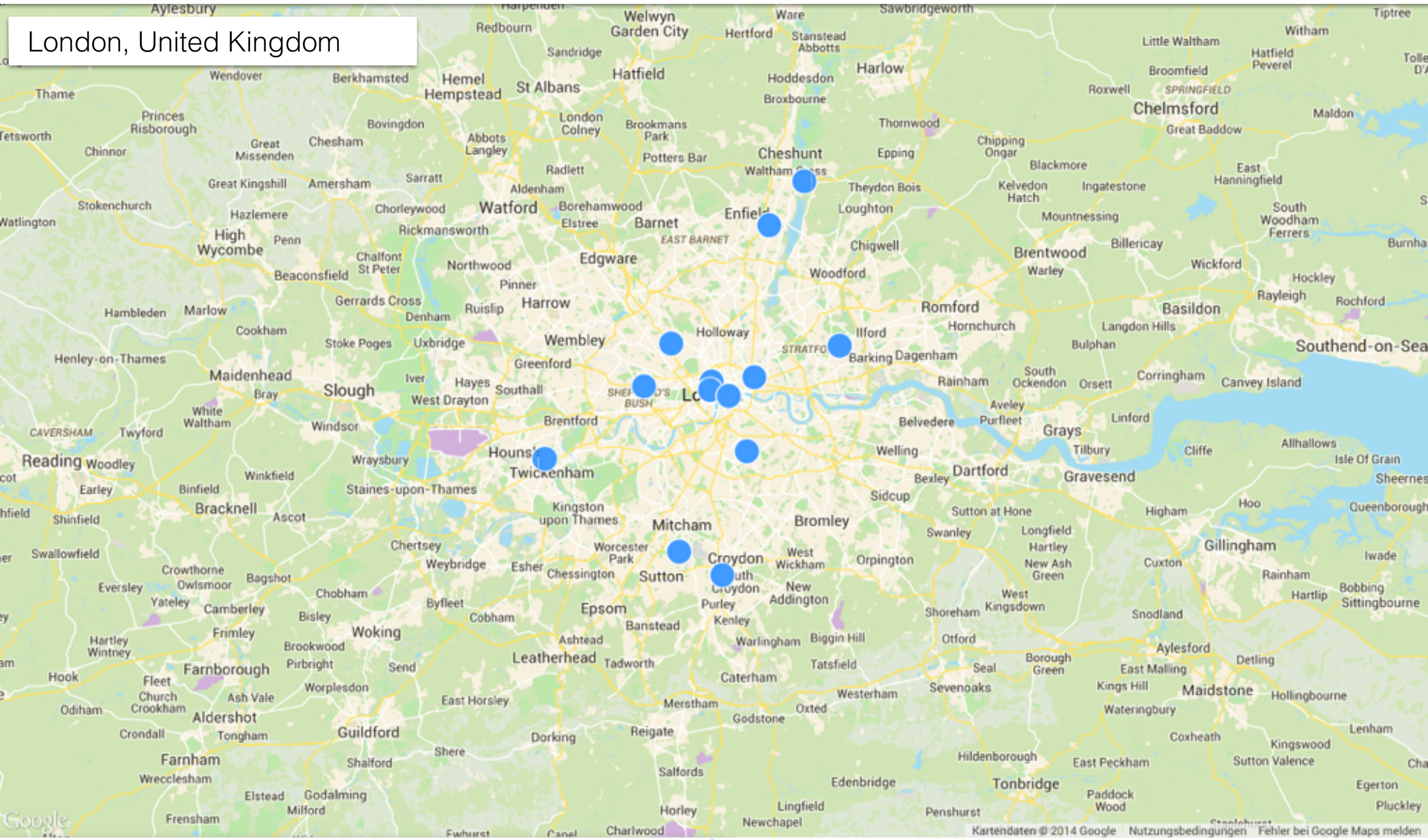
BETA:
Only english Tweets

Get a fully character analysis from all the users combined who are twittering in this city

6. Show how to use the new application

MoodLocator

London, United Kingdom



Openness

London	51%
Choose City	

Openness	Emotionality	Authority-challenging
Adventurousness	Imagination	Conscientiousness
Artistic interests	Intellect	Achievement striving

Tipp how to use

- User Modeling Service is a **BETA** version, is only analyzing english words. Therefore choose a city where people twitter mostly in english.

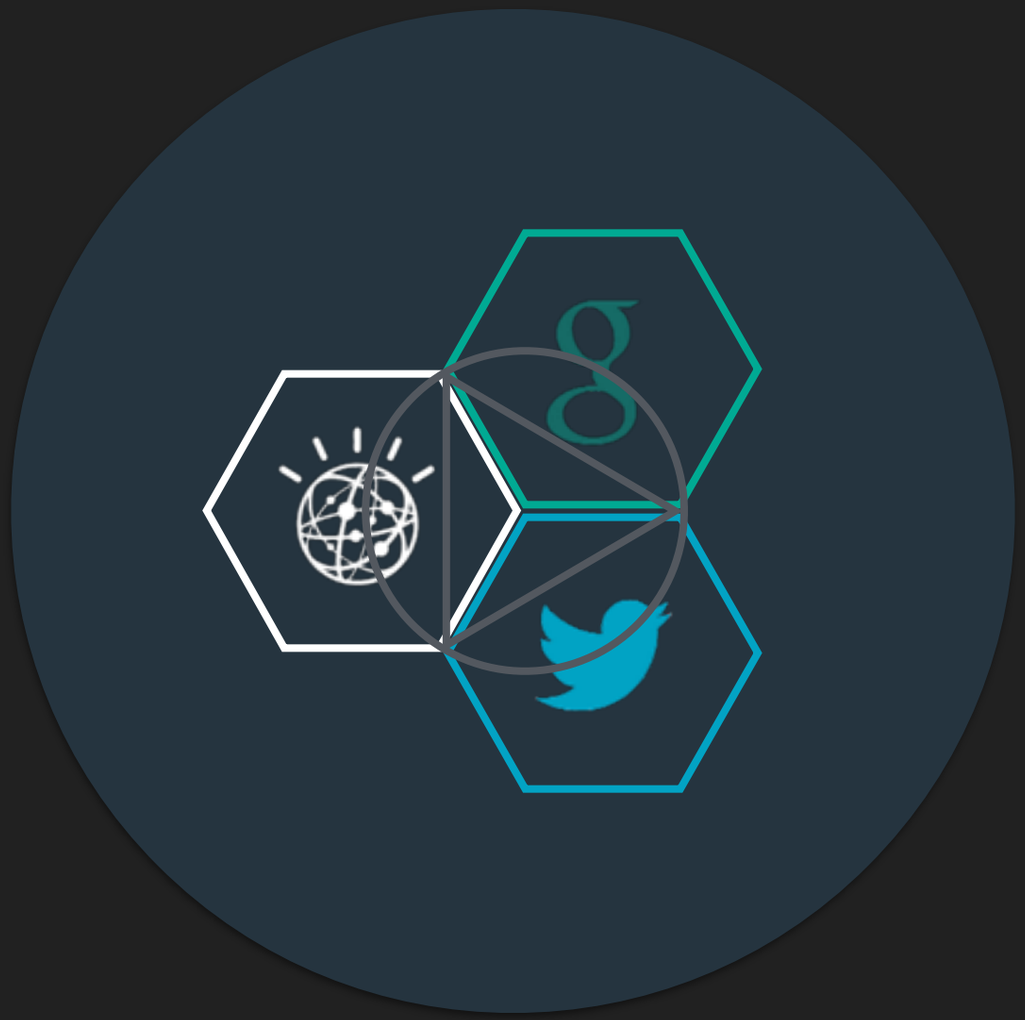
- Choose big cities like London, New York, Johannesburg.

- When you hover over one Tweet, only hover over the next Tweet after the first one closes.

- Test it yourself, enter your current location and send a Tweet with text and geolocation. Try to find it.

  Have fun and enjoy it!

MoodLocator



Bluemix: moodlocator.mybluemix.com

GitHub: /JDihlmann/moodlocator