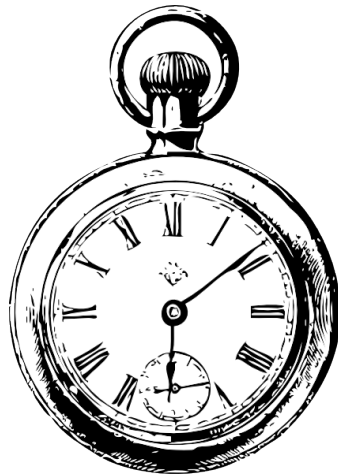


How to Build an API Service in 30 Minutes with *Express*. *js*



@rdegges

@gostormpath

Hey, I'm Randall

Developer Evangelist,
Stormpath



Open Source Dude

<3 Developer Tools

BitStamp (USD)

Apr 19, 2015 - Daily

bitstampUSD

UTC - <http://bitcoincharts.com>



“How much is one BTC
worth right now?”



<https://github.com/rdegges/btc-sms>



Get BTC Rates via SMS

BTC SMS makes it easy to track the value of Bitcoin via SMS. Each time you hit the API service, we'll SMS you the current Bitcoin price in a user-friendly way.

[GET STARTED!](#)

Pricing

2¢ / query

We believe in simple pricing. Everyone pays the same usage-based feeds regardless of size.

Regardless of how many requests you make, BTC exchange rates are updated once per hour.

GET STARTED!

API Documentation

This page contains the documentation for this API service. There is only a single API endpoint available right now, so this document is fairly short.

Questions? Please email support@apiservice.com for help!

REST Endpoints

POST /api/message

Description

This API endpoint takes in a phone number, and sends this phone an SMS message with the current Bitcoin exchange rate.

Input

Field	Type	Required
phoneNumber	String	true

Success Output

Field	Type	Example
phoneNumber	String	"+18182223333"
message	String	"1 Bitcoin is currently worth \$225.42 USD."
cost	Integer	2

Failure Output

Field	Type	Example
error	String	"We couldn't send the SMS message. Try again soon!"

Example Request

```
$ curl -X POST \
  --user 'id:secret' \
  --data '{"phoneNumber": "+18182223333"}' \
  -H 'Content-Type: application/json' \
  'http://apiservice.com/api/message'
```


Log In or [Create Account](#)

Email

Password

Log In

[Resend Verification Email?](#)

Create Account

First Name

Randall

Last Name

Degges

Email

r@rdegges.com

Password

.....

Create Account

[Back to Log In](#)

Your account verification email has been sent!

Before you can log into your account, you need to activate your account by clicking the link we sent to you at your email address: **r@rdegges.com**.

Please check your inbox to continue.

Verify your account



Inbox x



Randall Degges via sendgrid.stormpath.com

to me 

5:26 PM (0 minutes ago) ☆



Hi,
You have been registered for an application that uses Stormpath.

<http://localhost:3000/verified?sptoken=7OHQaRG3h31icXIFkyCQx1>

Once you verify, you will be able to login.

For general inquiries or to request support with your account, please email r@rdegges.com



Click here to [Reply](#) or [Forward](#)

Account Verification Complete!

Your account has been verified, and you have been logged into your account.
You will be redirected back to the site in 5 seconds.

API Key ID: 7DGSVQ9W51Q5DXZ7S18IST64U

API Key Secret: Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA

ANALYTICS

0

**total queries*

BILLING

\$0.00

**current account balance*

Pay with Card

rdegges at Randalls-MacBook-Pro in ~

```
o curl -v -X POST --user '7DGSVQ9W51Q5DXZ7S18IST64UAPI:Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA' -H 'Content-Type: application/json' --data '{"phoneNumber": "+18182179229"}' 'http://127.0.0.1:3000/api/message'
```

* Hostname was NOT found in DNS cache

* Trying 127.0.0.1...

* Connected to 127.0.0.1 (127.0.0.1) port 3000 (#0)

* Server auth using Basic with user '7DGSVQ9W51Q5DXZ7S18IST64UAPI'

> POST /api/message HTTP/1.1

> Authorization: Basic N0RHU1ZROVc1MVE1RFhaN1MxOE1TVDY0VUFQSTpZMnF3NDArMmI1eStnZXE0cVM1bXd6YzJGc1hnZXdlleUIvSFJkdllpNWpB

> User-Agent: curl/7.37.1

> Host: 127.0.0.1:3000

> Accept: */*

> Content-Type: application/json

> Content-Length: 31

>

* upload completely sent off: 31 out of 31 bytes

< HTTP/1.1 401 Unauthorized

< X-Powered-By: Express

< Content-Type: application/json; charset=utf-8

< Content-Length: 36

< ETag: W/"24-c8972ac4"

< Date: Mon, 20 Apr 2015 00:40:13 GMT

< Connection: keep-alive

<

* Connection #0 to host 127.0.0.1 left intact

{"error":"Invalid API credentials."}

```
rdegges at Randalls-MacBook-Pro in ~
o curl -v -X POST --user '7DGSVQ9W51Q5DXZ7S18IST64U:Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA' -H 'Content-Type: application/json' --data '{"phoneNumber": "+18182179229"}' 'http://127.0.0.1:3000/api/message'
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 3000 (#0)
* Server auth using Basic with user '7DGSVQ9W51Q5DXZ7S18IST64U'
> POST /api/message HTTP/1.1
> Authorization: Basic N0RHU1ZR0Vc1MVE1RFhaN1Mx0E1TVDY0VTpZMnF3NDArMmI1eStnZXE0cVM1bXd6YzJGc1hnZXdlleUIvSFJkdllpNWpB
> User-Agent: curl/7.37.1
> Host: 127.0.0.1:3000
> Accept: */*
> Content-Type: application/json
> Content-Length: 31
>
* upload completely sent off: 31 out of 31 bytes
< HTTP/1.1 402 Payment Required
< X-Powered-By: Express
< Content-Type: application/json; charset=utf-8
< Content-Length: 74
< ETag: W/"4a-518e0d4e"
< Date: Mon, 20 Apr 2015 00:41:46 GMT
< Connection: keep-alive
<
* Connection #0 to host 127.0.0.1 left intact
{"error":"Payment required. You need to deposit funds into your account."}%
```


API Key ID: 7DGSVQ9w51Q5DXZ7S18IST64U

API Key Secret: Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA

ANALYTICS

0

**total queries*

BTC SMS

r@rdegges.com

4242 4242 4242 4242



01 / 16

123

Pay \$20.00

BILLING

\$0.00

**current account balance*

Pay with Card

API Key ID: 7DGSVQ9W51Q5DXZ7S18IST64U

API Key Secret: Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA

ANALYTICS

0

**total queries*

BILLING

\$20.00

**current account balance*

Pay with Card

rdegges at Randalls-MacBook-Pro in ~

```
o curl -v -X POST --user '7DGSVQ9W51Q5DXZ7S18IST64U:Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA' -H 'Content-Type: application/json' --data '{"phoneNumber": "+18182179229"}' 'http://127.0.0.1:3000/api/message'
```

* Hostname was NOT found in DNS cache

* Trying 127.0.0.1...

* Connected to 127.0.0.1 (127.0.0.1) port 3000 (#0)

* Server auth using Basic with user '7DGSVQ9W51Q5DXZ7S18IST64U'

> POST /api/message HTTP/1.1

> Authorization: Basic N0RHU1ZR0Vc1MVE1RFhaN1Mx0E1TVDY0VTpZMnF3NDARmMl1eStnZXE0cVM1bXd6YzJGc1hnZXdlleUIvSFJkd1lpNWpB

> User-Agent: curl/7.37.1

> Host: 127.0.0.1:3000

> Accept: */*

> Content-Type: application/json

> Content-Length: 31

>

* upload completely sent off: 31 out of 31 bytes

< HTTP/1.1 200 OK

< X-Powered-By: Express

< Content-Type: application/json; charset=utf-8

< Content-Length: 93

< ETag: W/"5d-da7a22d3"

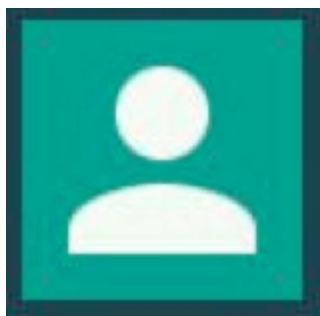
< Date: Mon, 20 Apr 2015 00:44:52 GMT

< Connection: keep-alive

<

* Connection #0 to host 127.0.0.1 left intact

{\"phoneNumber\":\"+18182179229\",\"message\":\"1 Bitcoin is currently worth \$225.13 USD.\",\"cost\":2}



+17126316056



1 Bitcoin is currently worth \$225.13 USD.

Motorola Nexus 6 via Pushbullet at 5:4...



Reply



Dismiss

API Key ID: 7DGSVQ9W51Q5DXZ7S18IST64U

API Key Secret: Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA

ANALYTICS

1

**total queries*

BILLING

\$19.98

**current account balance*

Pay with Card

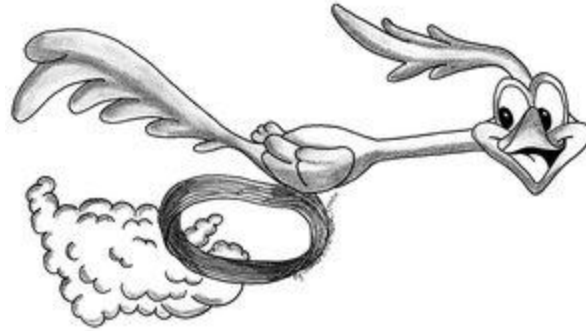
How?



stripe



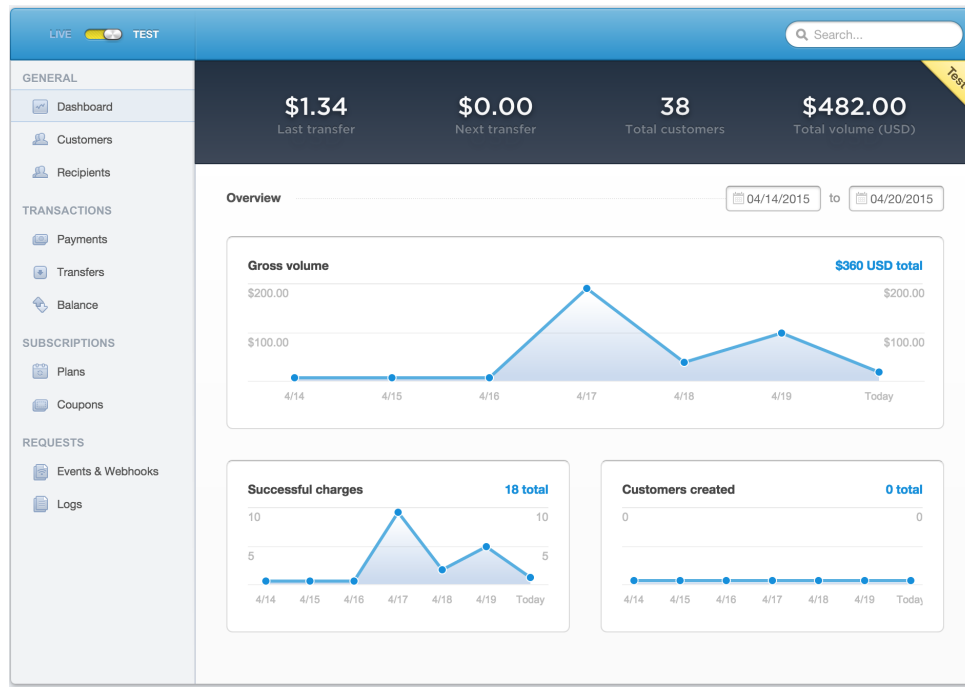
twilio



 **bitcoin charts**

 **Stormpath**

Stripe



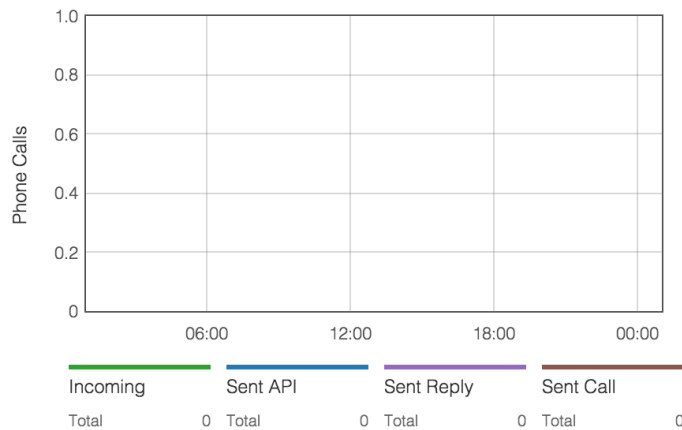

```
var stripe = require('stripe')(process.env.STRIPE_SECRET_KEY);

stripe.charges.create({
  amount: 2000, // $20
  currency: 'usd',
  source: token,
  description: 'One time deposit.'
}, function(err, charge) {
  if (err) throw err;
  console.log('Successfully billed user:', charge.amount);
});
```

Twilio

Voice Analytics

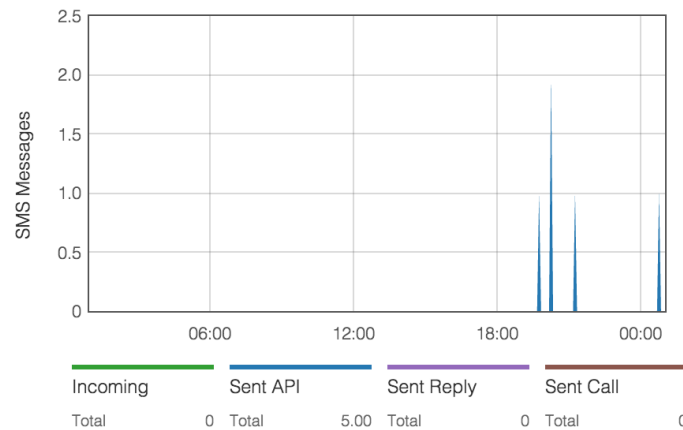
Today's Calls



More

SMS Analytics

Today's SMS Messages



More

```
var twilio = require('twilio')(
  process.env.TWILIO_ACCOUNT_SID,
  process.env.TWILIO_AUTH_TOKEN
);

twilio.sendMessage({
  to: '+18182179229',
  from: '+18882223333',
  body: 'Heyo!'
}, function(err, resp) {
  if (err) throw err;
  console.log('SMS message sent!');
});
```

Bitcoin Charts

```
{  
  - USD: {  
    7d: "225.06",  
    30d: "243.91",  
    24h: "225.14"  
  },  
  - IDR: {  
    7d: "2970699.92",  
    30d: "3167201.79",  
    24h: "2944989.69"  
  },  
  - ILS: {  
    7d: "888.54",  
    30d: "967.96",  
    24h: "872.07"  
  },  
}
```

```
var request = require('request');

request(
  'http://api.bitcoincharts.com/v1/weighted_prices.json',
  function(err, resp, body) {
    if (err) throw err;
    var data = JSON.parse(body);
    console.log('BTC value in USD:', data.USD['24h']);
  }
);
```

Stormpath

- User storage.
- User data storage (as JSON).
- Authentication.
- Authorization.
- API key management.
- API authentication.
- Password reset.
- etc.

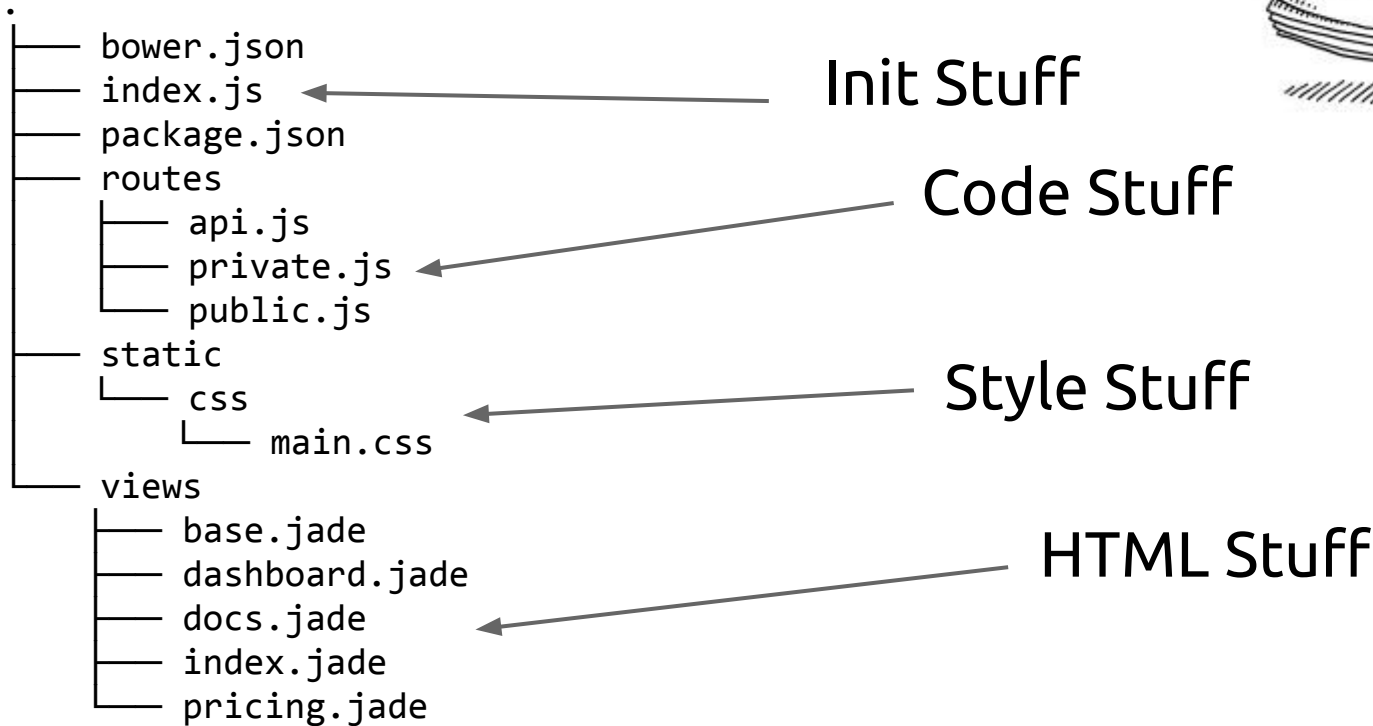


```
var stormpath = require('express-stormpath');
```

```
app.use(stormpath.init(app));
```

```
app.get('/', stormpath.loginRequired, function(req, res) {  
  res.send('Hi ' + req.user.email + '!');  
});
```

The Layout



bower.json

```
{  
  "name": "btc-sms",  
  "main": "index.js",  
  "version": "0.0.0",  
  "dependencies": {  
    "jquery": "~2.1.3",  
    "bootstrap": "~3.3.4",  
    "respond": "~1.4.2",  
    "html5shiv": "~3.7.2",  
    "bootswatch": "~3.3.4+1"  
  }  
}
```



package.json

yey ^^

```
{
  "name": "btc-sms",
  "version": "0.0.0",
  "main": "index.js",
  "dependencies": {
    "async": "^0.9.0",
    "body-parser": "^1.12.3",
    "express": "^4.12.3",
    "express-stormpath": "^1.0.4",
    "jade": "^1.9.2",
    "request": "^2.55.0",
    "stripe": "^3.3.4",
    "twilio": "^2.0.0"
  }
}
```



index.jade

```
extends base
```

```
block vars
```

```
- var title = 'Home'
```

```
block body
```

```
.container.index
```

```
h1.text-center Get BTC Rates via SMS
```

```
.row
```

```
.col-xs-12.col-md-offset-2.col-md-8
```

```
.jumbotron.text-justify
```

```
p.
```

```
#{siteTitle} makes it easy to track the value of Bitcoin via SMS.
```

```
Each time you hit the API service, we'll SMS you the current Bitcoin  
price in a user-friendly way.
```

```
a(href='/register')
```

```
button.btn.btn-lg.btn-primary.center-block(type='button') Get Started!
```



Get BTC Rates via SMS

BTC SMS makes it easy to track the value of Bitcoin via SMS. Each time you hit the API service, we'll SMS you the current Bitcoin price in a user-friendly way.

[GET STARTED!](#)

pricing.jade

```
extends base
```

```
block vars
```

```
- var title = 'Pricing'
```

```
block body
```

```
.container.pricing
```

```
h1.text-center Pricing
```

```
.row
```

```
.col-xs-offset-2.col-xs-8.col-md-offset-4.col-md-4.price-box.text-center
```

```
h2 #{costPerQuery}&cent; / query
```

```
p.text-justify.
```

We believe in simple pricing. Everyone pays the same usage-based feeds regardless of size.

```
p.text-justify.end.
```

<i>Regardless of how many requests you make, BTC exchange rates are updated once per hour.</i>

```
.row
```

```
.col-xs-offset-2.col-xs-8.col-md-offset-4.col-md-4
```

```
a(href='/register')
```

```
button.btn.btn-lg.btn-primary.center-block(type='button') Get Started!
```



Pricing

2¢ / query

We believe in simple pricing. Everyone pays the same usage-based feeds regardless of size.

Regardless of how many requests you make, BTC exchange rates are updated once per hour.

GET STARTED!

docs.jade

```
extends base

block vars
  - var title = 'Docs'

block body
  .container.docs
    h1.text-center API Documentation
    .row
      .col-xs-12.col-md-offset-2.col-md-8
        p.text-justify
          i.
            This page contains the documentation for this API service. There is
            only a single API endpoint available right now, so this document is
            fairly short.
        p.text-justify
          i.
            Questions? Please email <a href="mailto:support@apiservice.com">
support@apiservice.com</a>
            for help!
        h2 REST Endpoints
        h3 POST /api/message
        span Description
        p.description.
          This API endpoint takes in a phone number, and sends this phone an
          SMS message with the current Bitcoin exchange rate.
        span Input
        .table-box
          table.table.table-bordered
            thead
              tr
                th Field
                th Type
                th Required
            tbody
              tr
                td phoneNumber
                td String
                td true
```

```
span Success Output
.table-box
  table.table.table-bordered
    thead
      tr
        th Field
        th Type
        th Example
    tbody
      tr
        td phoneNumber
        td String
        td "+18182223333"
      tr
        td message
        td String
        td "1 Bitcoin is currently worth $225.42 USD."
      tr
        td cost
        td Integer
        td #{costPerQuery}
span Failure Output
.table-box
  table.table.table-bordered
    thead
      tr
        th Field
        th Type
        th Example
    tbody
      tr
        td error
        td String
        td "We couldn't send the SMS message. Try again soon!"
span Example Request
pre.
$ curl -X POST \
  --user 'id:secret' \
  --data '{"phoneNumber": "+18182223333"}' \
  -H 'Content-Type: application/json' \
  'http://apiservice.com/api/message'
```



API Documentation

This page contains the documentation for this API service. There is only a single API endpoint available right now, so this document is fairly short.

Questions? Please email support@apiservice.com for help!

REST Endpoints

POST /api/message

Description

This API endpoint takes in a phone number, and sends this phone an SMS message with the current Bitcoin exchange rate.

Input

Field	Type	Required
phoneNumber	String	true

Success Output

Field	Type	Example
phoneNumber	String	"+18182223333"
message	String	"1 Bitcoin is currently worth \$225.42 USD."
cost	Integer	2

Failure Output

Field	Type	Example
error	String	"We couldn't send the SMS message. Try again soon!"

Example Request

```
$ curl -X POST \
  --user 'id:secret' \
  --data '{"phoneNumber": "+18182223333"}' \
  -H 'Content-Type: application/json' \
  'http://apiservice.com/api/message'
```


dashboard.jade

```
extends base

block vars
  - var title = 'Dashboard'

block body
  .container.dashboard
    .row.api-keys
      ul.list-group
        .col-xs-offset-1.col-xs-10
          li.list-group-item.api-key-container
            .left
              strong API Key ID:
              span.api-key-id #{user.apiKeys.items[0].id}
            .right
              strong API Key Secret:
              span.api-key-secret #{user.apiKeys.items[0].secret}
    .row.widgets
      .col-md-offset-1.col-md-5
        .panel.panel-primary
          .panel-heading.text-center
            h3.panel-title Analytics
          .analytics-content.text-center
```

```
span.total-queries #{user.customData.totalQueries}
br
span
  i.
    *total queries
.col-md-5
  .panel.panel-primary
    .panel-heading.text-center
      h3.panel-title Billing
    .billing-content.text-center
      span.account-balance $#{(user.customData.balance / 100).toFixed(2)}
      br
      span
        i.
          *current account balance
      form(action='/dashboard/charge', method='POST')
        script.stripe-button(
          src = 'https://checkout.stripe.com/checkout.js',
          data-email = '#{user.email}',
          data-key = '#{stripePublishableKey}',
          data-name = '#{siteTitle}',
          data-amount = '2000',
          data-allow-remember-me = 'false'
        )
```



API Key ID: 7DGSVQ9W51Q5DXZ7S18IST64U

API Key Secret: Y2qw40+2b5y+geq4qS5mwzc2FsXgeweyB/HRdvYi5jA

ANALYTICS

1

**total queries*

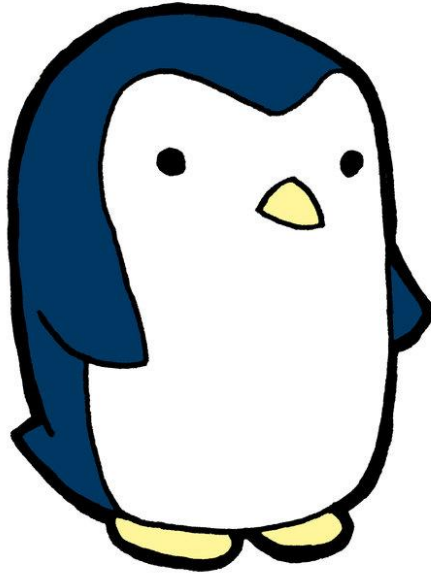
BILLING

\$19.98

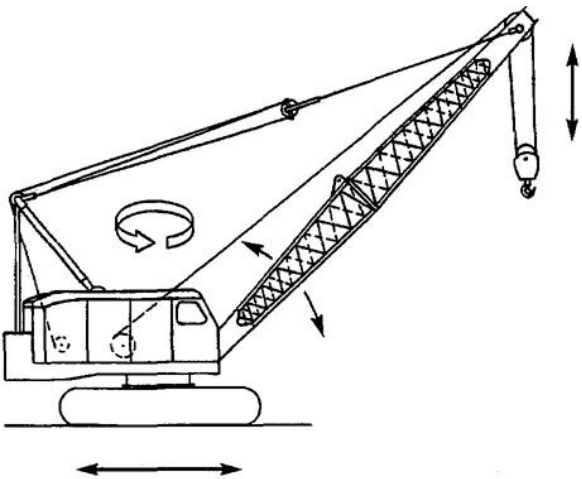
**current account balance*

Pay with Card

Show Me Code



Application Setup



```
var async = require('async');
var express = require('express');
var stormpath = require('express-stormpath');

var apiRoutes = require('./routes/api');
var privateRoutes = require('./routes/private');
var publicRoutes = require('./routes/public');

// Globals
var app = express();

// Application settings
app.set('view engine', 'jade');
app.set('views', './views');

app.locals.costPerQuery = parseInt(process.env.COST_PER_QUERY);
app.locals.siteTitle = 'BTC SMS';
app.locals.stripePublishableKey = process.env.STRIPE_PUBLISHABLE_KEY;
```



#{siteTitle}

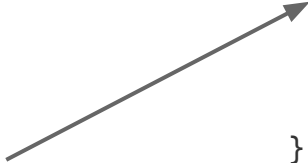
```
// Middlewares
app.use('/static', express.static('./static', {
  index: false,
  redirect: false
}));

app.use('/static', express.static('./bower_components', {
  index: false,
  redirect: false
}));
```

```
app.use(stormpath.init(app, {  
  enableAccountVerification: true,  
  redirectUrl: '/dashboard',  
  secretKey: process.env.SECRET_KEY,  
  postRegistrationHandler: registrationHandler  
}));
```

persist data

```
function(account, req, res, next) {
  async.parallel([
    // Set the user's default settings.
    function(cb) {
      account.customData.balance = 0;
      account.customData.totalQueries = 0;
      account.customData.save(function(err) {
        if (err) return cb(err);
        cb();
      });
    },
    // Create an API key for this user.
    function(cb) {
      account.createApiKey(function(err, key) {
        if (err) return cb(err);
        cb();
      });
    }
  ], function(err) {
    if (err) return next(err);
    next();
  });
}
```




```
li.list-group-item.api-key-container
```

```
.left
```

```
strong API Key ID:
```

```
span.api-key-id #{user.apiKeys.items[0].id}
```

```
.right
```

```
strong API Key Secret:
```

```
span.api-key-secret #{user.apiKeys.items[0].secret}
```



API key info

query info



```
.analytics-content.text-center
```

```
span.total-queries #{user.customData.totalQueries}
```

\$\$\$ info

```
.billing-content.text-center
```

```
span.account-balance ${ (user.customData.balance / 100).toFixed(2) }
```



security middleware

```
// Routes
```

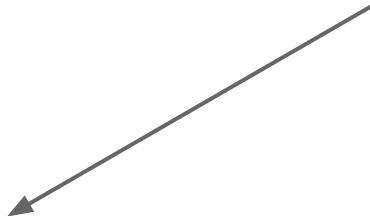
```
app.use('/', publicRoutes);
```

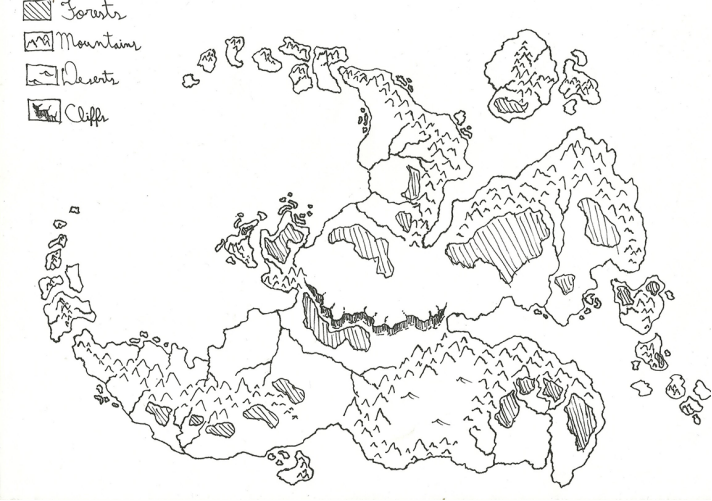
```
app.use('/api', stormpath.apiAuthenticationRequired, apiRoutes);
```

```
app.use('/dashboard', stormpath.loginRequired, privateRoutes);
```

```
// Server
```

```
app.listen(process.env.PORT || 3000);
```





Public Routes

```
var express = require('express');

// Globals
var router = express.Router();

// Routes
router.get('/', function(req, res) {
  res.render('index');
});

router.get('/pricing', function(req, res) {
  res.render('pricing');
});

router.get('/docs', function(req, res) {
  res.render('docs');
});

// Exports
module.exports = router;
```

Private Routes



```
// dashboard.jade
form(action='/dashboard/charge', method='POST')
  script.stripe-button(
    src = 'https://checkout.stripe.com/checkout.js',
    data-email = '#{user.email}',
    data-key = '#{stripePublishableKey}',
    data-name = '#{siteTitle}',
    data-amount = '2000',
    data-allow-remember-me = 'false'
  )
```

```
var bodyParser = require('body-parser');
var express = require('express');
var stormpath = require('express-stormpath');
var stripe = require('stripe')(process.env.
STRIPE_SECRET_KEY);

// Globals
var router = express.Router();

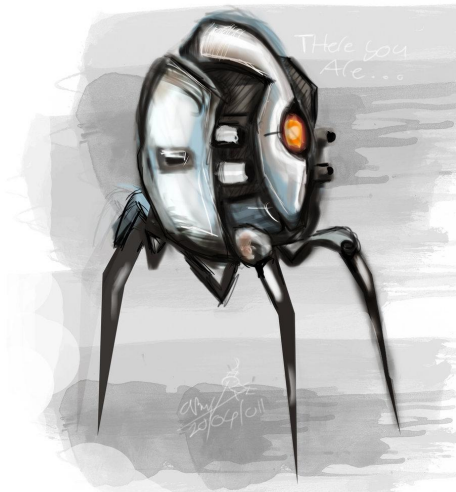
// Middlewares
router.use(bodyParser.urlencoded({ extended: true
}));

// Routes
router.get('/', function(req, res) {
  res.render('dashboard');
});
```

```
router.post('/charge', function(req, res, next) {
  stripe.charges.create({
    amount: 2000,
    currency: 'usd',
    source: req.body.stripeToken,
    description: 'One time deposit for ' +
      req.user.email + '.'
  }, function(err, charge) {
    if (err) return next(err);
    req.user.customData.balance += charge.amount;
    req.user.customData.save(function(err) {
      if (err) return next(err);
      res.redirect('/dashboard');
    });
  });
});

// Exports
module.exports = router;
```

API Routes




```
var bodyParser = require('body-parser');
var express = require('express');
var request = require('request');
var twilio = require('twilio')(
  process.env.TWILIO_ACCOUNT_SID,
  process.env.TWILIO_AUTH_TOKEN
);

// Globals
var router = express.Router();
var BTC_EXCHANGE_RATE;
var COST_PER_QUERY = parseInt(process.env.COST_PER_QUERY);

// Middlewares
router.use(bodyParser.json());
```

// Functions

```
function getExchangeRates() {  
  request('http://api.bitcoincharts.com/v1/weighted_prices.json', function(err, resp, body) {  
    if (err || resp.statusCode !== 200) {  
      console.log('Failed to retrieve BTC exchange rates.');      return;  
    }  
  
    try {  
      var data = JSON.parse(body);  
      BTC_EXCHANGE_RATE = data.USD['24h'];  
      console.log('Updated BTC exchange rate: ' + BTC_EXCHANGE_RATE + '.');    } catch (err) {  
      console.log('Failed to parse BTC exchange rates.');      return;  
    }  
  });  
}
```

// Tasks

```
getExchangeRates();  
setInterval(getExchangeRates, 60000);
```

```
// Routes
router.post('/message', function(req, res) {
  if (!req.body || !req.body.phoneNumber) {
    return res.status(400).json({ error: 'phoneNumber is required.' });
  } else if (!BTC_EXCHANGE_RATE) {
    return res.status(500).json({ error: "We're having trouble getting the exchange rates right now. Try again soon!"
  });
  } else if (req.user.customData.balance < COST_PER_QUERY) {
    return res.status(402).json({ error: 'Payment required. You need to deposit funds into your account.' });
  }

  var message = '1 Bitcoin is currently worth $' + BTC_EXCHANGE_RATE + ' USD.';

  twilio.sendMessage({
    to: req.body.phoneNumber,
    from: process.env.TWILIO_PHONE_NUMBER,
    body: message
  }, function(err, resp) {
    if (err) return res.status(500).json({ error: "We couldn't send the SMS message. Try again soon!" });

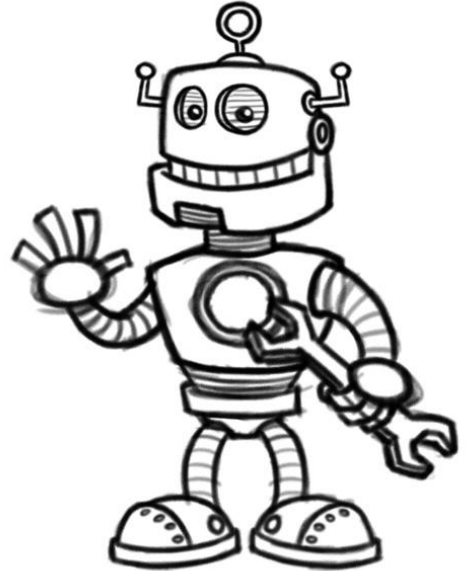
    req.user.customData.balance -= COST_PER_QUERY;
    req.user.customData.totalQueries += 1;
    req.user.customData.save();

    res.json({ phoneNumber: req.body.phoneNumber, message: message, cost: COST_PER_QUERY });
  });
});

// Exports
module.exports = router;
```

Any questions?

Thanks!



@rdegges

@gostormpath