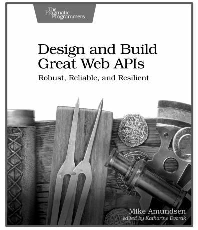
Design and Build Great Web APIs

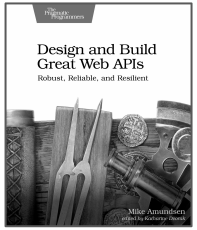
Releasing

@mamund
Mike Amundsen
training.amundsen.com

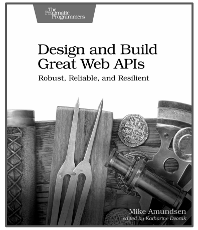

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Releasing

- Testing
  - Happy and sad paths
- Securing
  - You want to do what?
- Deploying
  - Let's do it live

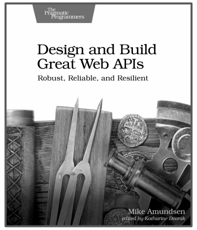

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing APIs

- Testing the Network
- Using **Postman/Newman**


The Pragmatic Programmers

Design and Build Great Web APIs

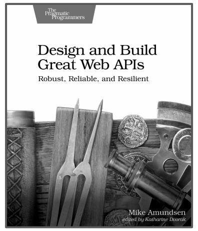
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

Testing APIs - Testing the Network

- Most of the threat for APIs is not in the code
- It's in the network itself
- And other people's APIs
- You can code and test for these cases



The Pragmatic Programmers

Design and Build Great Web APIs

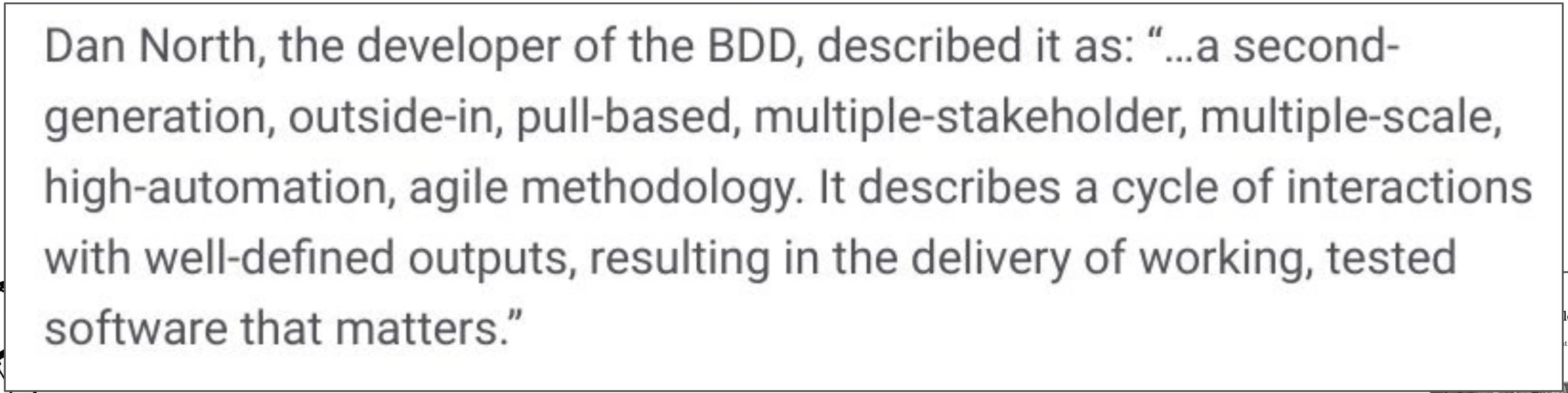
Robust, Reliable, and Resilient

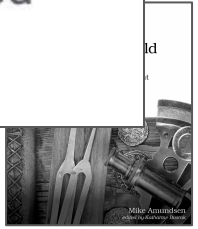
Mike Amundsen

edited by Katharine Dvorak

Testing APIs : BDD

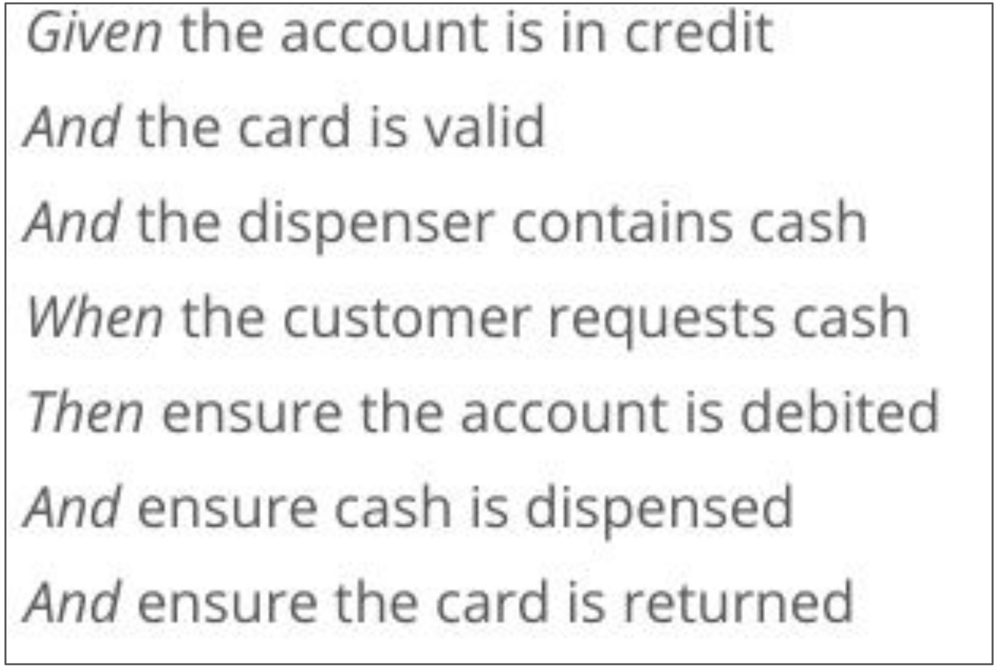
- Behavior-Driven Development (2006)
- Dan North, Thoughtworks
- Outside-in


Dan North, the developer of the BDD, described it as: "...a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters."



Testing APIs : BDD

- Given - When - Then



Given the account is in credit

And the card is valid

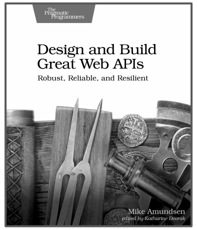
And the dispenser contains cash

When the customer requests cash

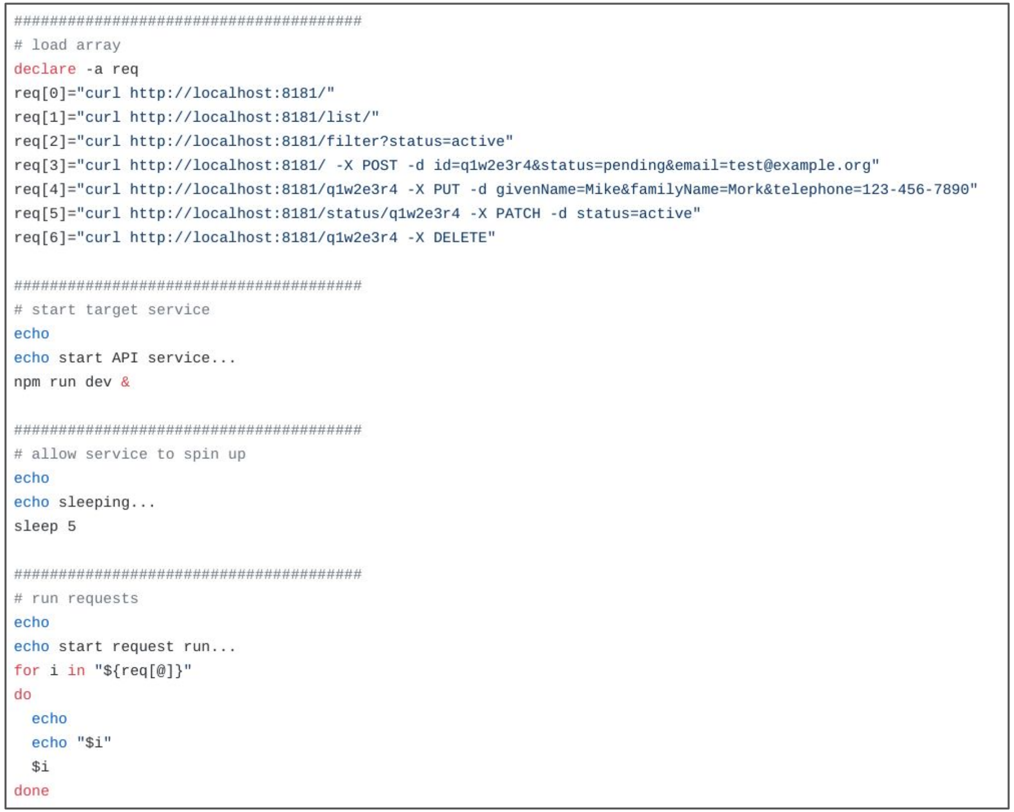
Then ensure the account is debited

And ensure cash is dispensed

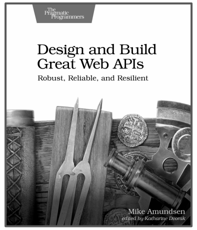
And ensure the card is returned


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing APIs : Quick Requests

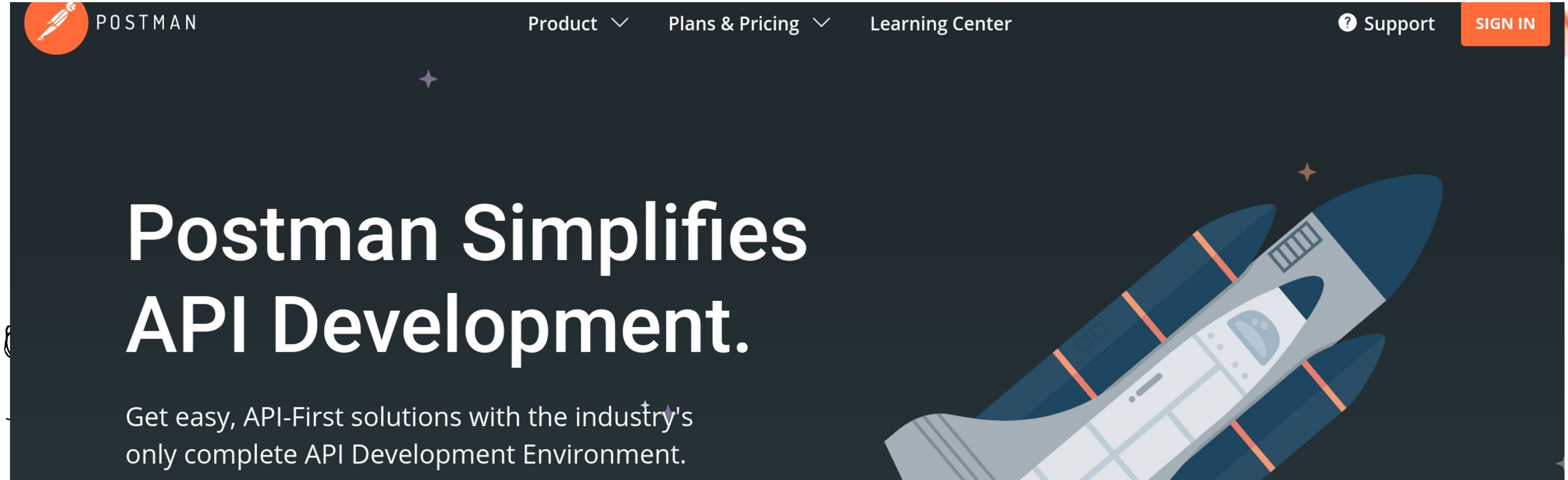


```
#######################################
# load array
declare -a req
req[0]="curl http://localhost:8181/"
req[1]="curl http://localhost:8181/list/"
req[2]="curl http://localhost:8181/filter?status=active"
req[3]="curl http://localhost:8181/ -X POST -d id=q1w2e3r4&status=pending&email=test@example.org"
req[4]="curl http://localhost:8181/q1w2e3r4 -X PUT -d givenName=Mike&familyName=Mork&telephone=123-456-7890"
req[5]="curl http://localhost:8181/status/q1w2e3r4 -X PATCH -d status=active"
req[6]="curl http://localhost:8181/q1w2e3r4 -X DELETE"

#######################################
# start target service
echo
echo start API service...
npm run dev &

#######################################
# allow service to spin up
echo
echo sleeping...
sleep 5

#######################################
# run requests
echo
echo start request run...
for i in "${req[@]}"
do
  echo
  echo "$i"
  $i
done
```


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing APIs - Using Postman

- Postman (2014)
- Working to be a complete API platform
- Focused on testing



Postman Simplifies API Development.

Get easy, API-First solutions with the industry's only complete API Development Environment.

Testing APIs - Using Postman

- Download & Install

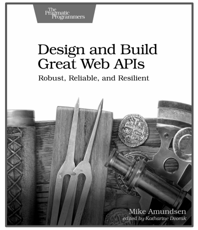
  https://www.getpostman.com/downloads/

- Documentation

  https://learning.getpostman.com/docs/postman/launching_postman/installation_and_updates/

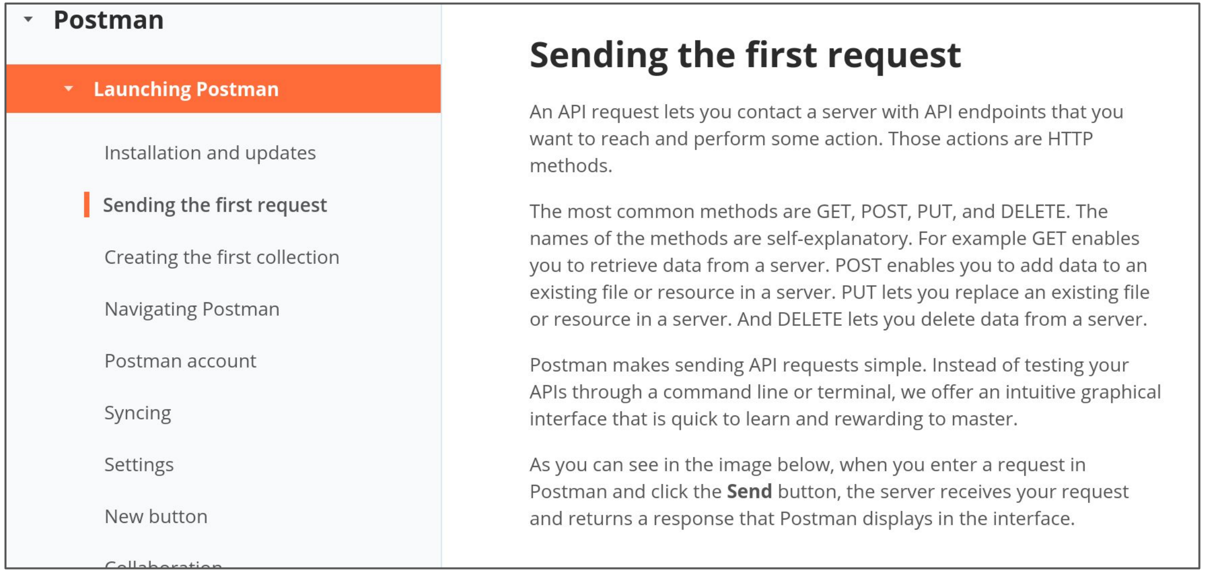
- Create an Account (recommended)

  https://identity.getpostman.com/signup


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing APIs - Using Postman

- Install, launch, and start testing



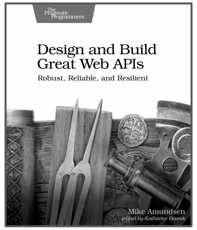
Sending the first request

An API request lets you contact a server with API endpoints that you want to reach and perform some action. Those actions are HTTP methods.

The most common methods are GET, POST, PUT, and DELETE. The names of the methods are self-explanatory. For example GET enables you to retrieve data from a server. POST enables you to add data to an existing file or resource in a server. PUT lets you replace an existing file or resource in a server. And DELETE lets you delete data from a server.

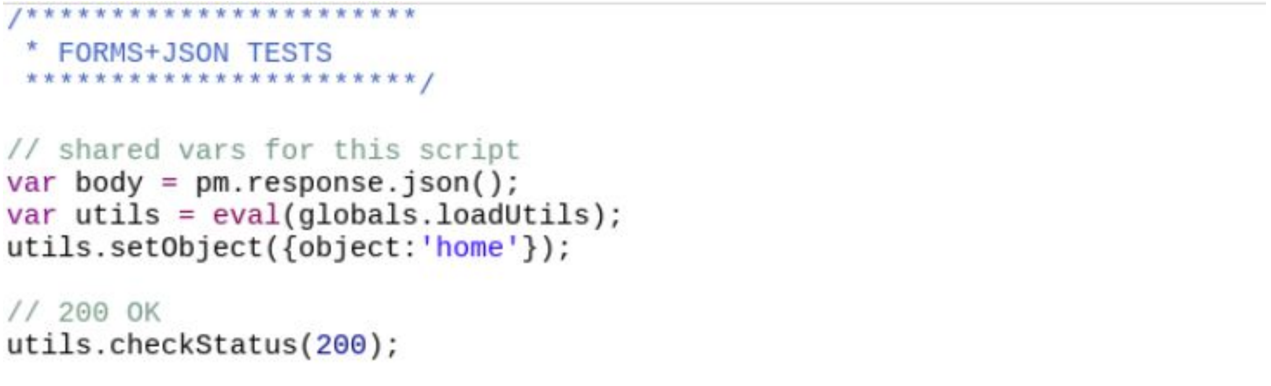
Postman makes sending API requests simple. Instead of testing your APIs through a command line or terminal, we offer an intuitive graphical interface that is quick to learn and rewarding to master.

As you can see in the image below, when you enter a request in Postman and click the **Send** button, the server receives your request and returns a response that Postman displays in the interface.

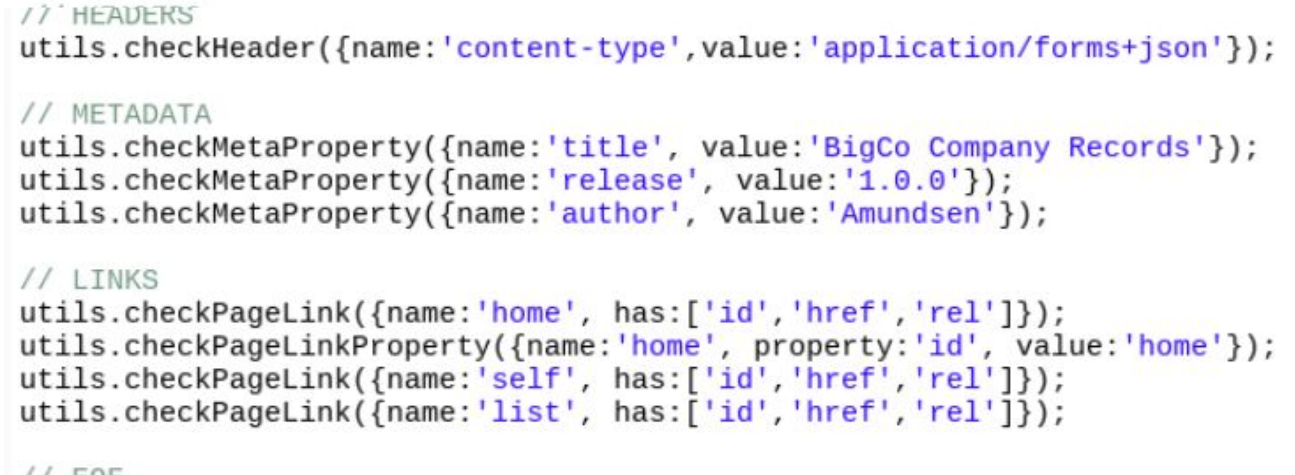


Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

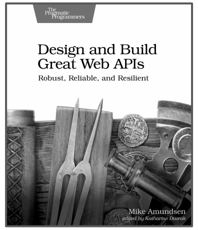
Testing APIs : Postman



```
/************************
 * FORMS+JSON TESTS
 ************************/

// shared vars for this script
var body = pm.response.json();
var utils = eval(globals.loadUtils);
utils.setObject({object:'home'});

// 200 OK
utils.checkStatus(200);
```



```
// HEADERS
utils.checkHeader({name:'content-type',value:'application/forms+json'});

// METADATA
utils.checkMetaProperty({name:'title', value:'BigCo Company Records'});
utils.checkMetaProperty({name:'release', value:'1.0.0'});
utils.checkMetaProperty({name:'author', value:'Amundsen'});

// LINKS
utils.checkPageLink({name:'home', has:['id','href','rel']});
utils.checkPageLinkProperty({name:'home', property:'id', value:'home'});
utils.checkPageLink({name:'self', has:['id','href','rel']});
utils.checkPageLink({name:'list', has:['id','href','rel']});

// EOF
```



The Pragmatic Programmers

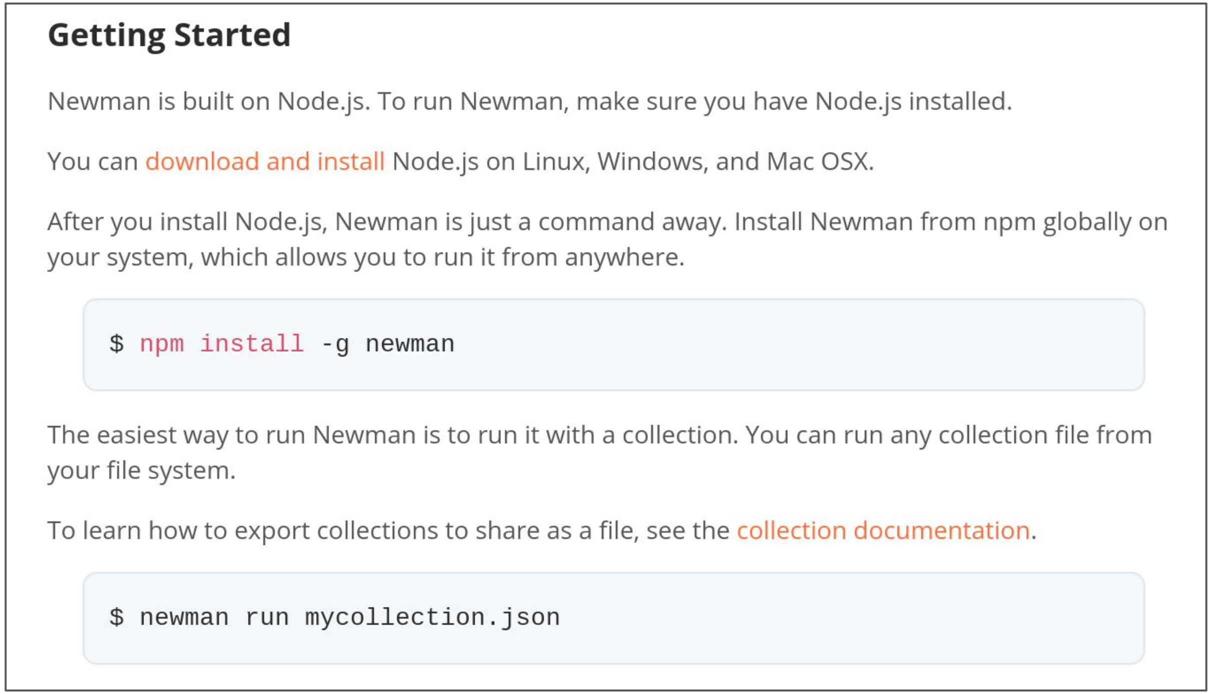
Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen
edited by Katharine Dvorak

Testing APIs - Bonus Utility -- newman

- CLI for running postman tests



Getting Started

Newman is built on Node.js. To run Newman, make sure you have Node.js installed.

You can download and install Node.js on Linux, Windows, and Mac OSX.

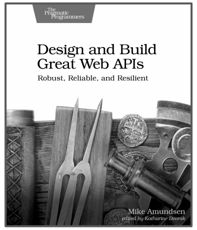
After you install Node.js, Newman is just a command away. Install Newman from npm globally on your system, which allows you to run it from anywhere.

```
$ npm install -g newman
```

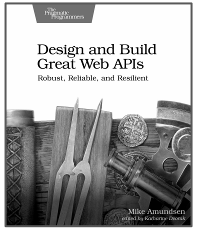
The easiest way to run Newman is to run it with a collection. You can run any collection file from your file system.

To learn how to export collections to share as a file, see the collection documentation.

```
$ newman run mycollection.json
```


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Securing

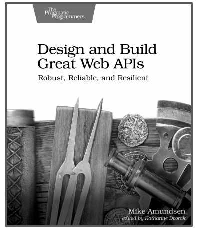

Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen

Securing APIs - Security Basics

- Authentication (Identity)
- Authorization (Access Control)
- TLS/HTTPS (message encoding)
- Encryption (field-level encoding)


The Pragmatic Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient

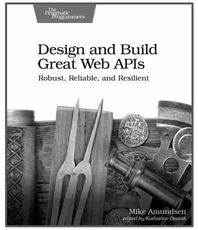
Mike Amundsen

edited by Katharine Dvorak

Securing APIs - Security Basics

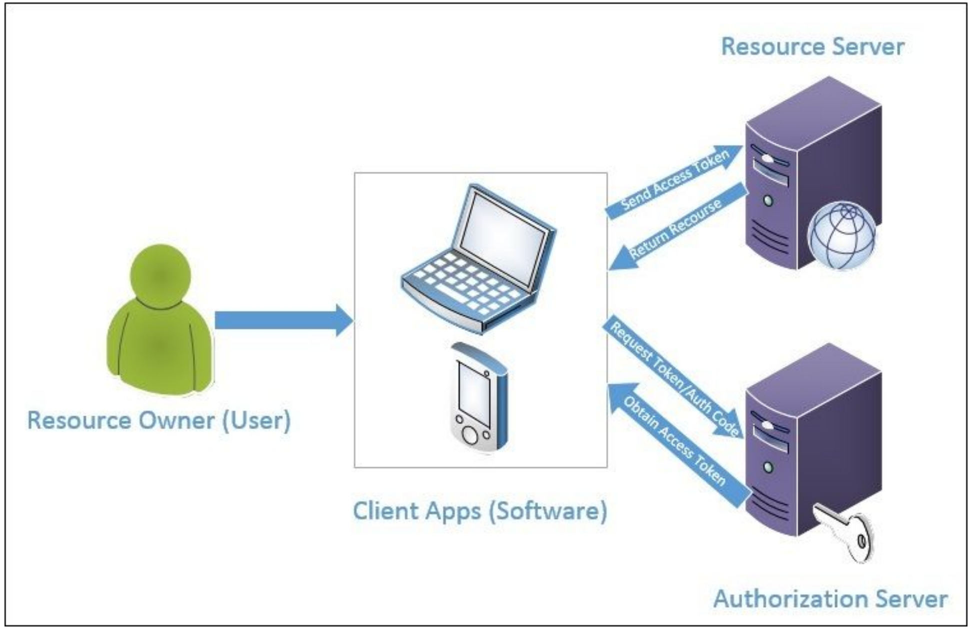
- Securing a Web API is tricky
- Who holds the list of users?
- Who holds the list of access rules?
- Who wrote the server side?
- Who wrote the client side?
- Who can see credentials?

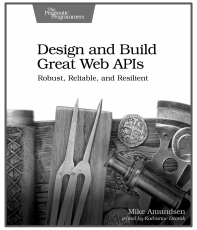
Solution: is "Three-legged Authentication"


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Securing APIs - Security Basics

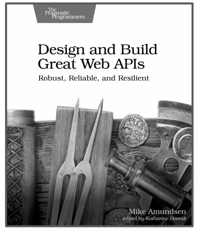
- Three-legged Authentication




The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

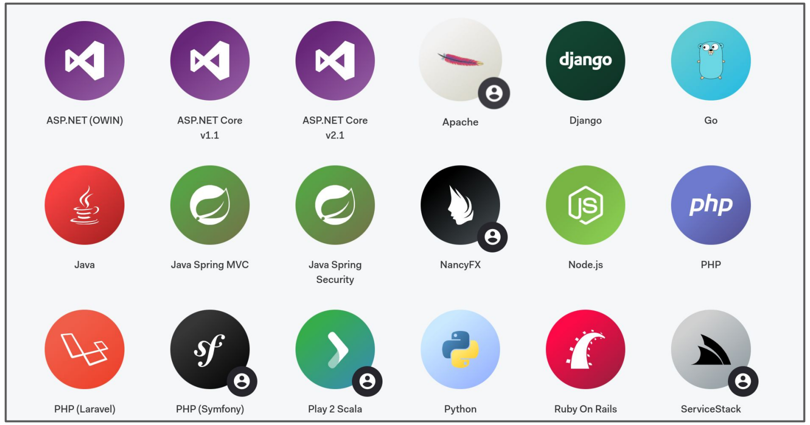
Securing APIs - OAuth

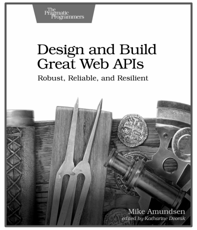
- OAuth was created to solve this problem
- Originally designed for Twitter in 2006
- Moved to IETF Standards in 2008
- OAuth 1.0 released 2010 (RFC5849)
- OAuth 2.0 released in 2012 (RFC6749 & RFC6750)


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Securing APIs - Auth0

- Public OAuth Cloud Service
- Created in 2013
- Supports Mobile, Web, API scenarios
- Lots of SDKs for many platforms




The Pragmatic Programmers

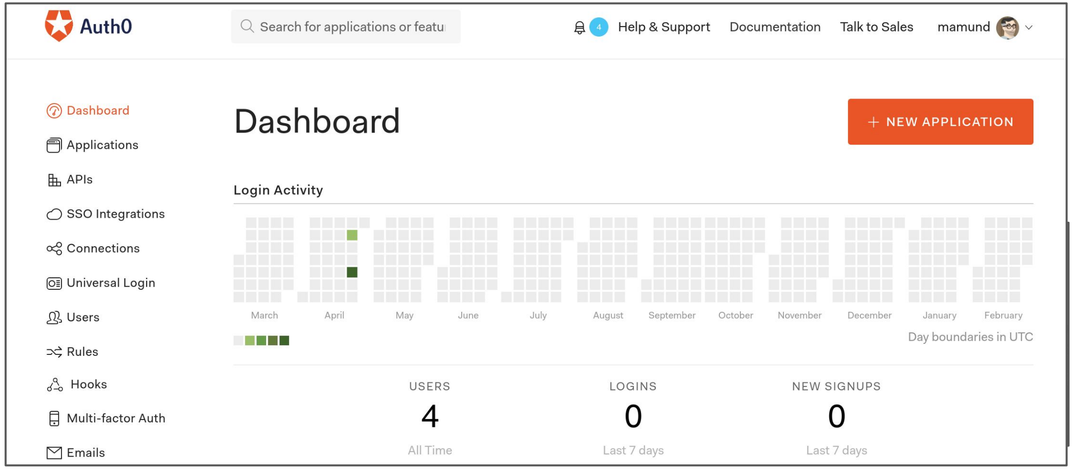
Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen
edited by Katharine Dvorak

Securing APIs - Auth0

- No Download
  https://auth0.com/
- Create Account (required)
  https://auth0.com/signup
- Dashboard
  https://manage.auth0.com/#/



Dashboard

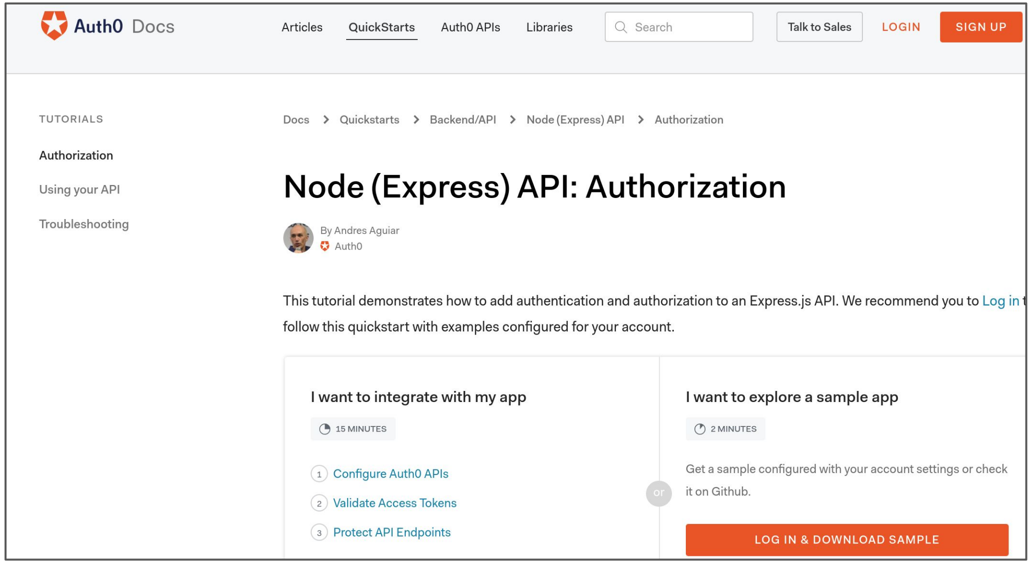
Login Activity

Day boundaries in UTC

USERS	LOGINS	NEW SIGNUPS
4	0	0
All Time	Last 7 days	Last 7 days

Securing APIs - Auth0

- Set up API Authentication for ExpressJS
  https://auth0.com/docs/quickstart/backend/nodejs



TUTORIALS
Authorization
Using your API
Troubleshooting

Docs > Quickstarts > Backend/API > Node (Express) API > Authorization

Node (Express) API: Authorization

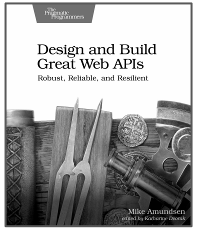
By Andres Aguiar
Auth0

This tutorial demonstrates how to add authentication and authorization to an Express.js API. We recommend you to Log in t
follow this quickstart with examples configured for your account.

I want to integrate with my app
15 MINUTES
1. Configure Auth0 APIs
2. Validate Access Tokens
3. Protect API Endpoints

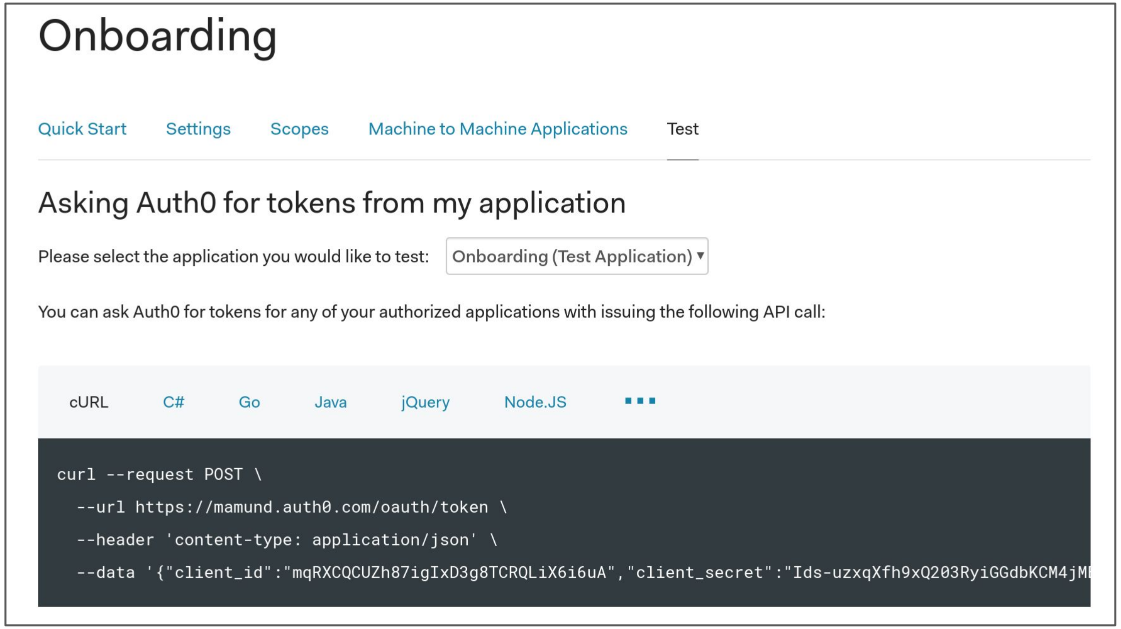
or

I want to explore a sample app
2 MINUTES
Get a sample configured with your account settings or check it on Github.
LOG IN & DOWNLOAD SAMPLE


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Securing APIs - Auth0

- Execute machine-to-machine (API) call via test page



Onboarding

Quick Start · Settings · Scopes · Machine to Machine Applications · Test

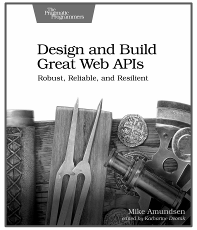
Asking Auth0 for tokens from my application

Please select the application you would like to test: Onboarding (Test Application)

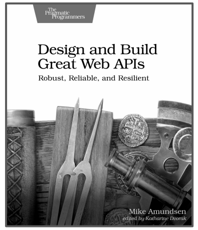
You can ask Auth0 for tokens for any of your authorized applications with issuing the following API call:

cURL · C# · Go · Java · jQuery · Node.JS · ...

```
curl --request POST \
  --url https://mamund.auth0.com/oauth/token \
  --header 'content-type: application/json' \
  --data '{"client_id":"mqRXCQCUZh87igIxD3g8TCRQLiX6i6uA","client_secret":"Ids-uzxqXfh9xQ203RyiGGdbKCM4jM
```

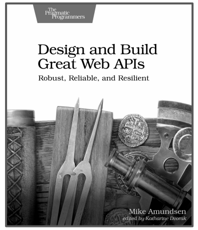

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Deploying


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Deploying APIs

- Git-based Deployment
- Using **Heroku**


The Pragmatic Programmers

Design and Build Great Web APIs

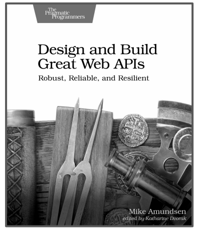
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

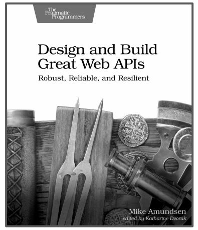
Deploying APIs - Challenges

- Deploying your app can be complicated
- Compatibility
  - Hardware
  - OS
  - Platform
  - Framework
  - Dependencies


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Deploying APIs - DevOps

- DevOps was created to help with all this
- Developers & Operators working together
- Started as a hashtag on twitter #DevOps
- Series of small conferences started in 2009
- Emphasis on automation to improve reliability


The Pragmatic Programmers

Design and Build Great Web APIs

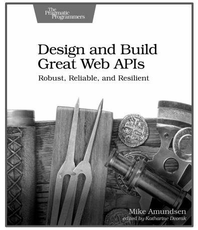
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

Deploying APIs - Tools

- Build tools
- CI/CD pipeline
- Docker (containers)
- Kubernetes (deployment orchestration)



The Pragmatic Programmers

Design and Build Great Web APIs

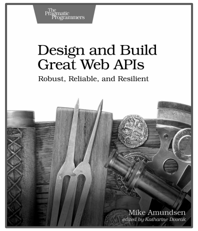
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

Deploying APIs - Using Heroku

- Cloud platform (2007)
- Originally just for Ruby/Rails projects
- Now supports Java, NodeJS, Python, Go, Clojure, Scala
- Acquired by Salesforce in 2011
- Full platform w/ marketplace ecosystem
- Heroku uses proprietary container tech (Dynos)

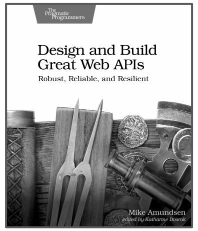

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Deploying APIs - Using Heroku

- Download CLI
  https://devcenter.heroku.com/articles/heroku-cli

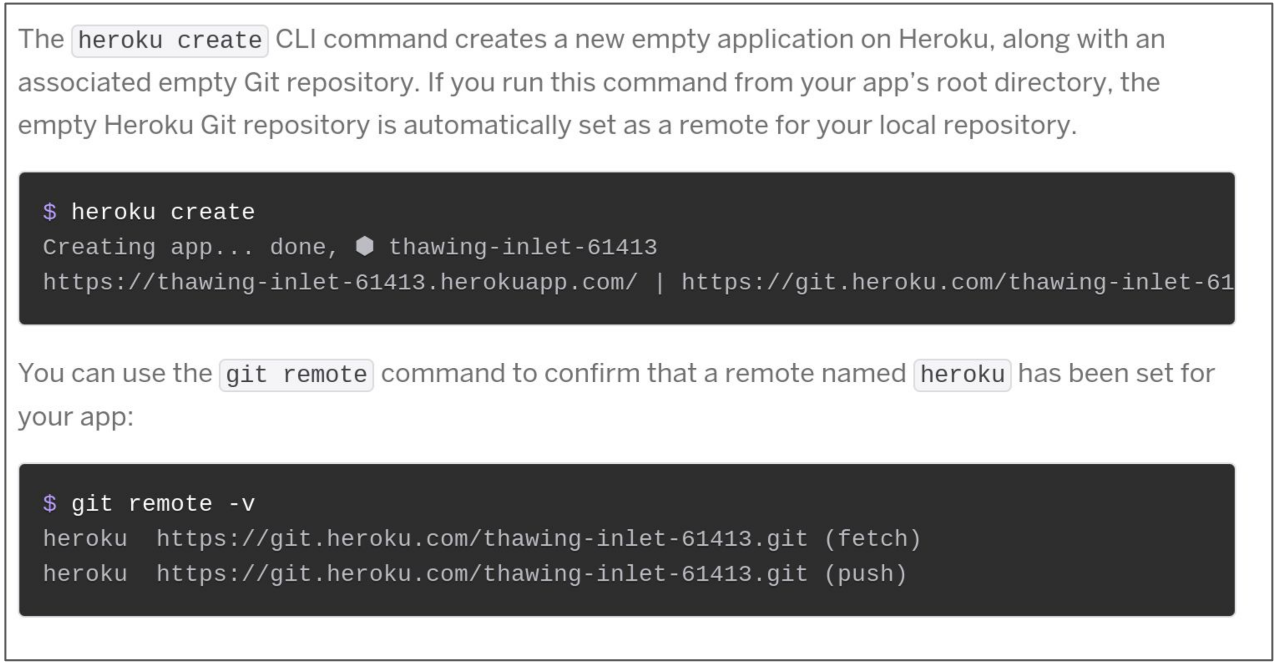
- Documentation
  https://devcenter.heroku.com/articles/using-the-cli

- Create an Account (required)
  https://signup.heroku.com/


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Deploying APIs - Using Heroku

- Git deploy tutorial
  https://devcenter.heroku.com/articles/git

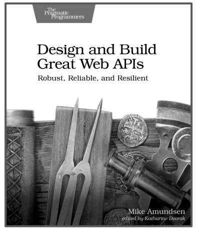


The `heroku create` CLI command creates a new empty application on Heroku, along with an associated empty Git repository. If you run this command from your app's root directory, the empty Heroku Git repository is automatically set as a remote for your local repository.

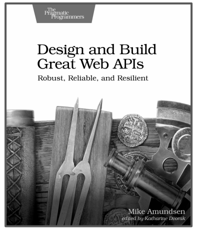
```
$ heroku create
Creating app... done, ⬢ thawing-inlet-61413
https://thawing-inlet-61413.herokuapp.com/ | https://git.heroku.com/thawing-inlet-61
```

You can use the `git remote` command to confirm that a remote named `heroku` has been set for your app:

```
$ git remote -v
heroku  https://git.heroku.com/thawing-inlet-61413.git (fetch)
heroku  https://git.heroku.com/thawing-inlet-61413.git (push)
```

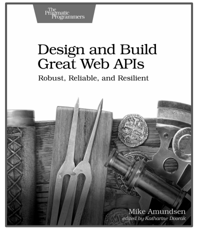

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

Releasing Exercise

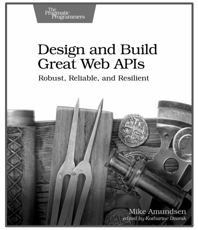

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Releasing Exercise

- Open command window in your project
- `heroku login`
- `heroku create`
- `git push heroku master`

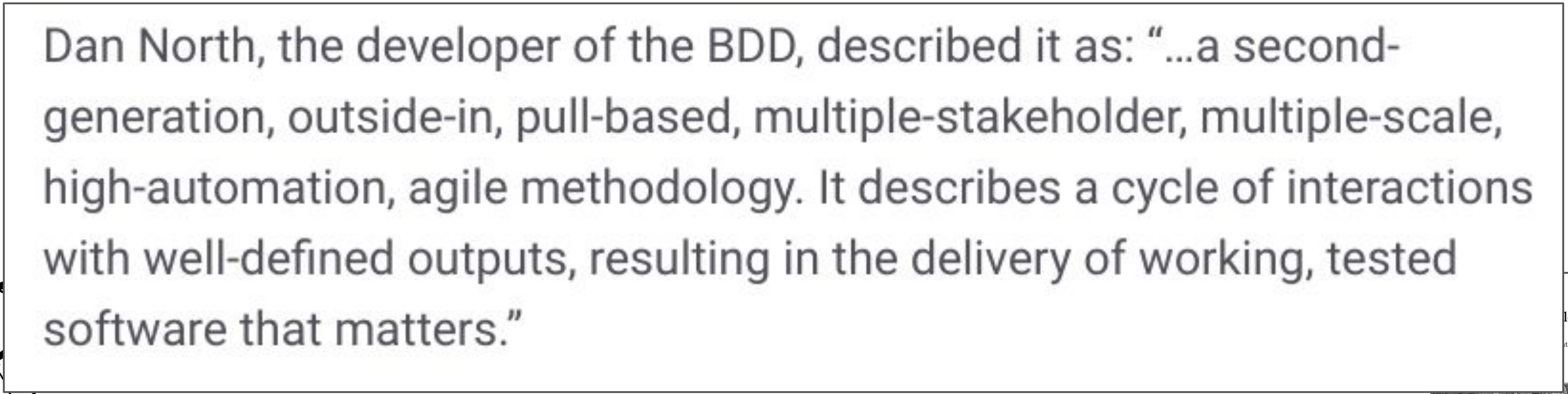

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Summary

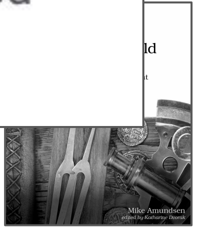

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Testing APIs : BDD

- Behavior-Driven Development (2006)
- Dan North, Thoughtworks
- Outside-in

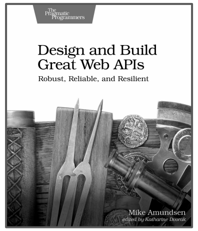


Dan North, the developer of the BDD, described it as: "...a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters."



Securing APIs - Security Basics

- Authentication (Identity)
- Authorization (Access Control)
- TLS/HTTPS (message encoding)
- Encryption (field-level encoding)


The Pragmatic Programmers

Design and Build Great Web APIs

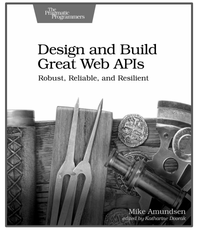
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

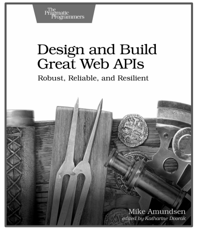
Deploying APIs - Challenges

- Deploying your app can be complicated
- Compatibility
  - Hardware
  - OS
  - Platform
  - Framework
  - Dependencies


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Releasing

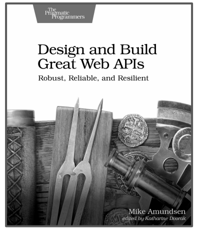
- Testing
  - From request lists to BDD
- Securing
  - Identity and Access Control
- Deploying
  - Automation is your friend


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

Design and Build Great Web APIs

Releasing

@mamund
Mike Amundsen
training.amundsen.com


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak