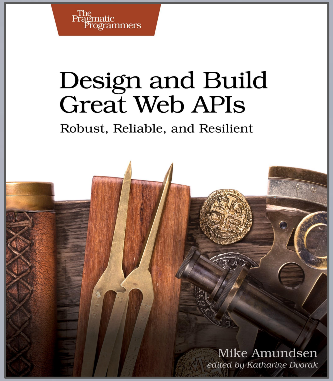
# Design and Build Great APIs

**@mamund**
**Mike Amundsen**
**youtube.com/mamund**



The Pragmatic Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen

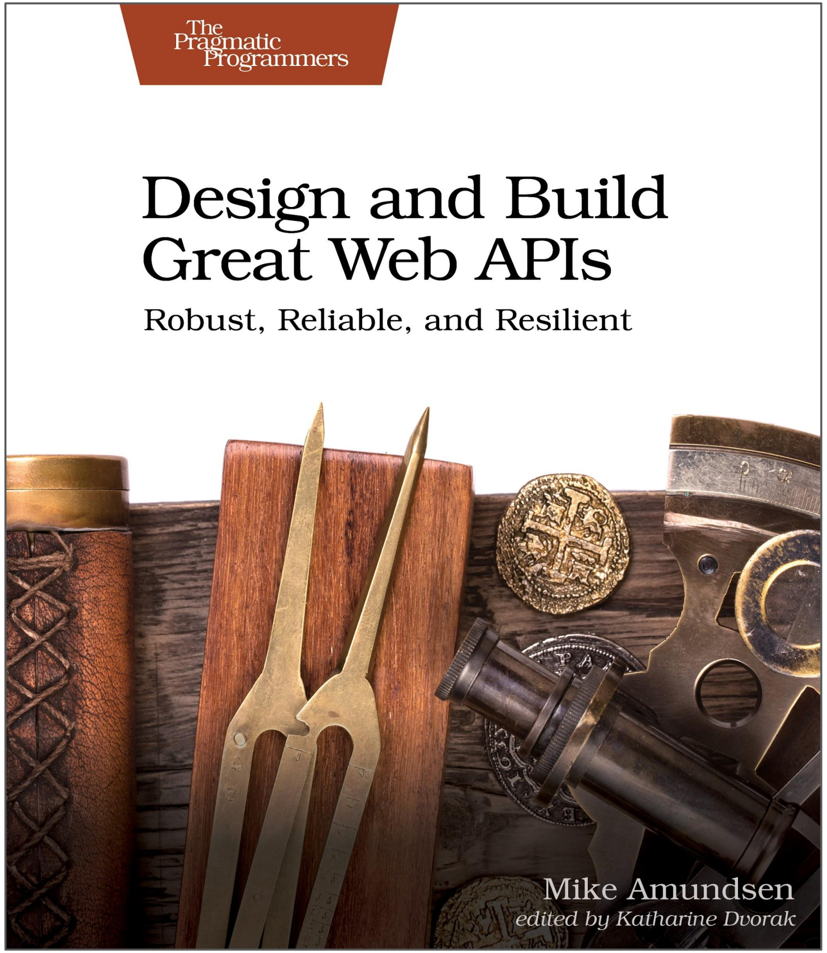
*edited by Katharine Dvorak*

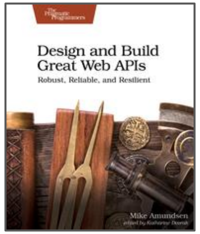
Mike Amundsen
@mamund

## g.mamund.com/GreatWebAPIs

*"From design to code to test to deployment, unlock hidden business value and release stable and scalable web APIs that meet customer needs and solve important business problems in a consistent and reliable manner."*

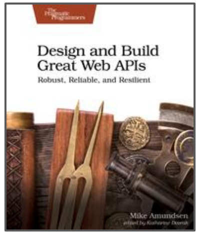
*-- Pragmatic Publishers*


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



## The Big Picture

- Foundations
- Designing
- Building
- Releasing
- And Then...

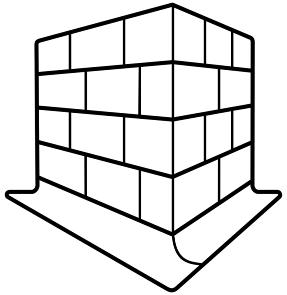

The Pragmatic Programmers

Design and Build Great Web APIs

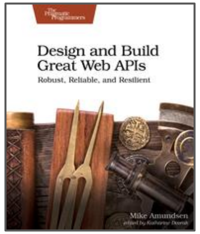
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

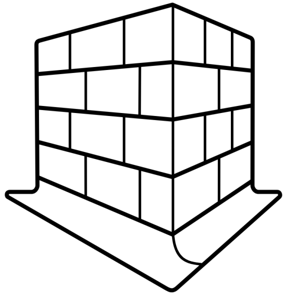


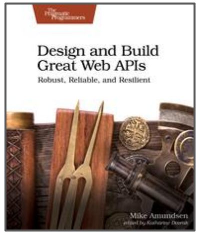
## Foundations


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

## Foundations

- API First
- HTTP, Web, & REST

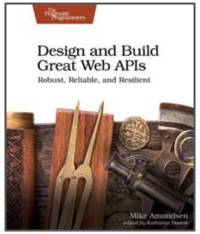



The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

# What is API-First?

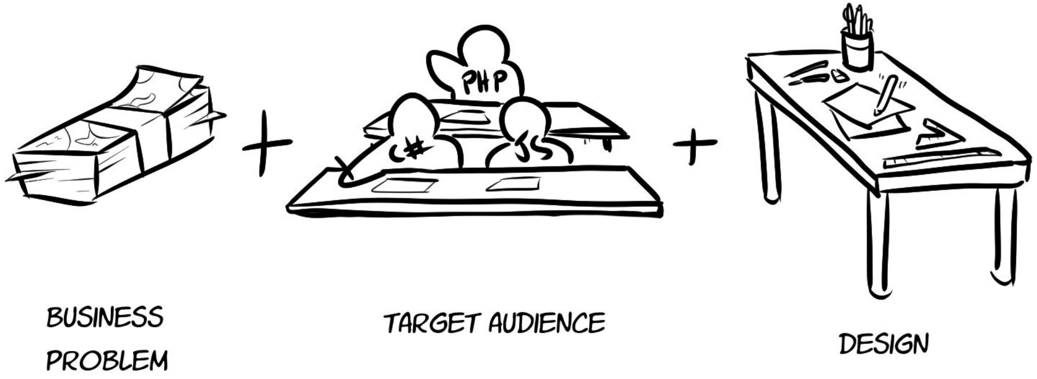
*"API-first design means identifying and/or defining key actors and personas, determining what those actors and personas expect to be able to do with APIs"*

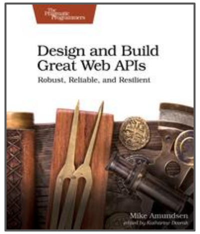
*-- Kas Thomas, 2009*


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

## Foundations

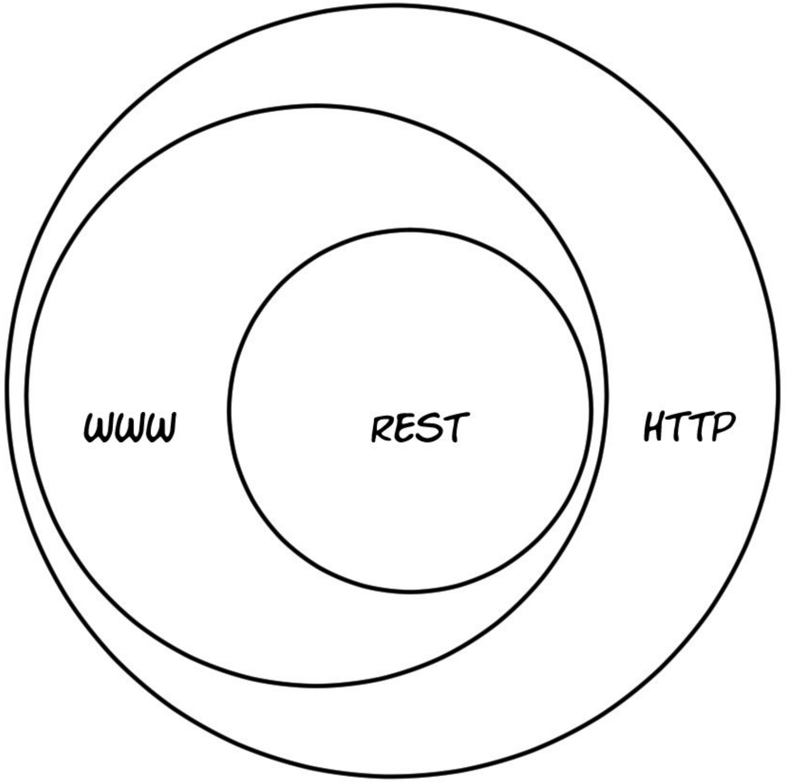
- API First
  - APIs Solve Business Problems
  - Designing APIs for People
  - Design First
- HTTP, Web, & REST

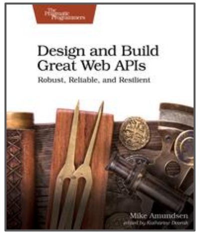



The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Foundations

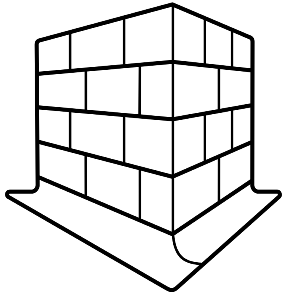
- API First
- HTTP, Web, & REST
  - HTTP is a protocol
    - URLs, Methods, Messages
  - Web is a common practice
    - SoC, links & forms
  - REST is a style
    - properties, requirements, constraints

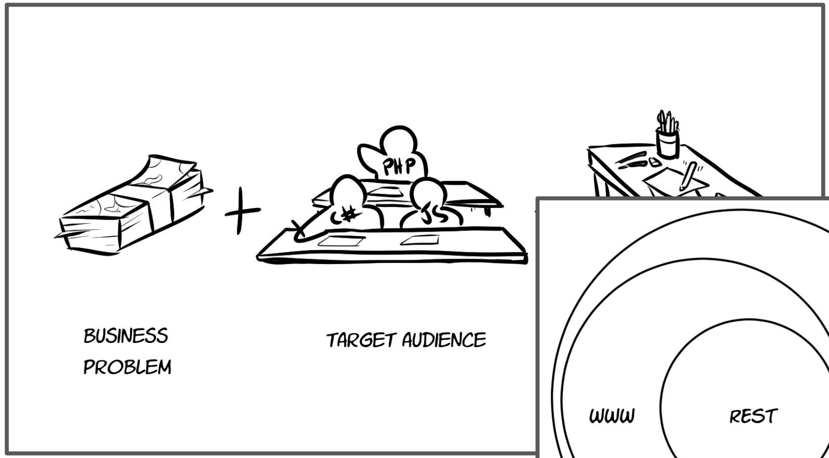


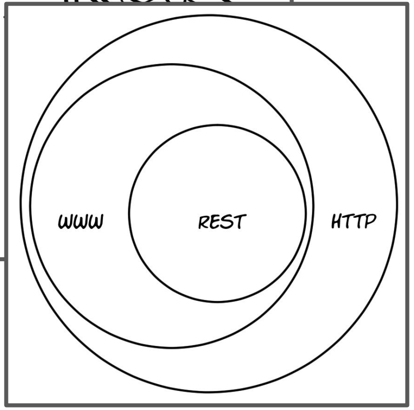


# Foundations

- API First
  - Solving business problems for people
- HTTP, Web, & REST
  - Protocol, practice, style

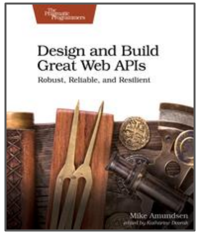






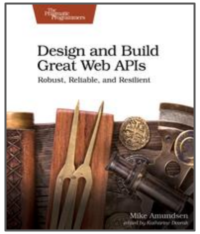
The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Designing APIs


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

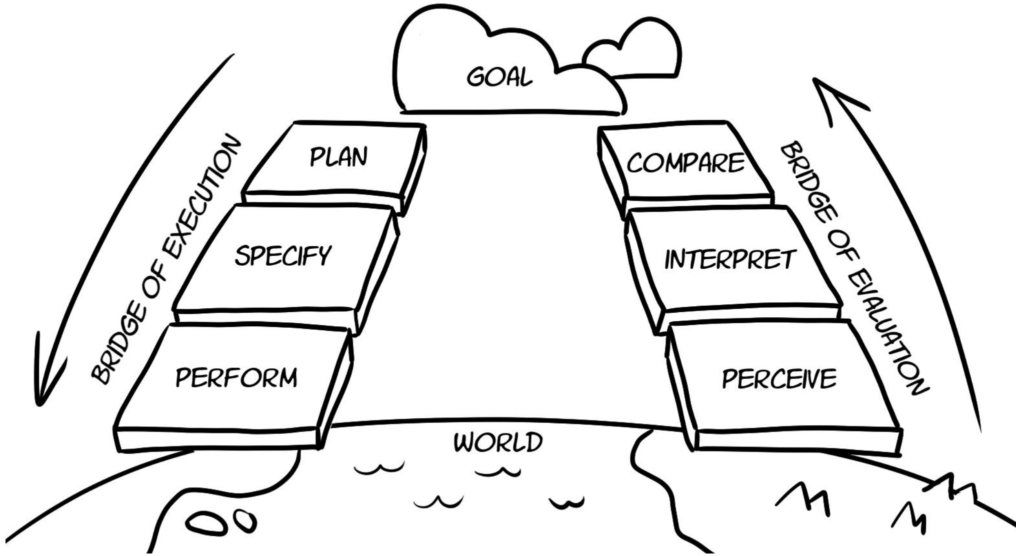
## Designing APIs

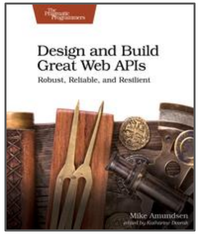
- Model
- Design
- Describe

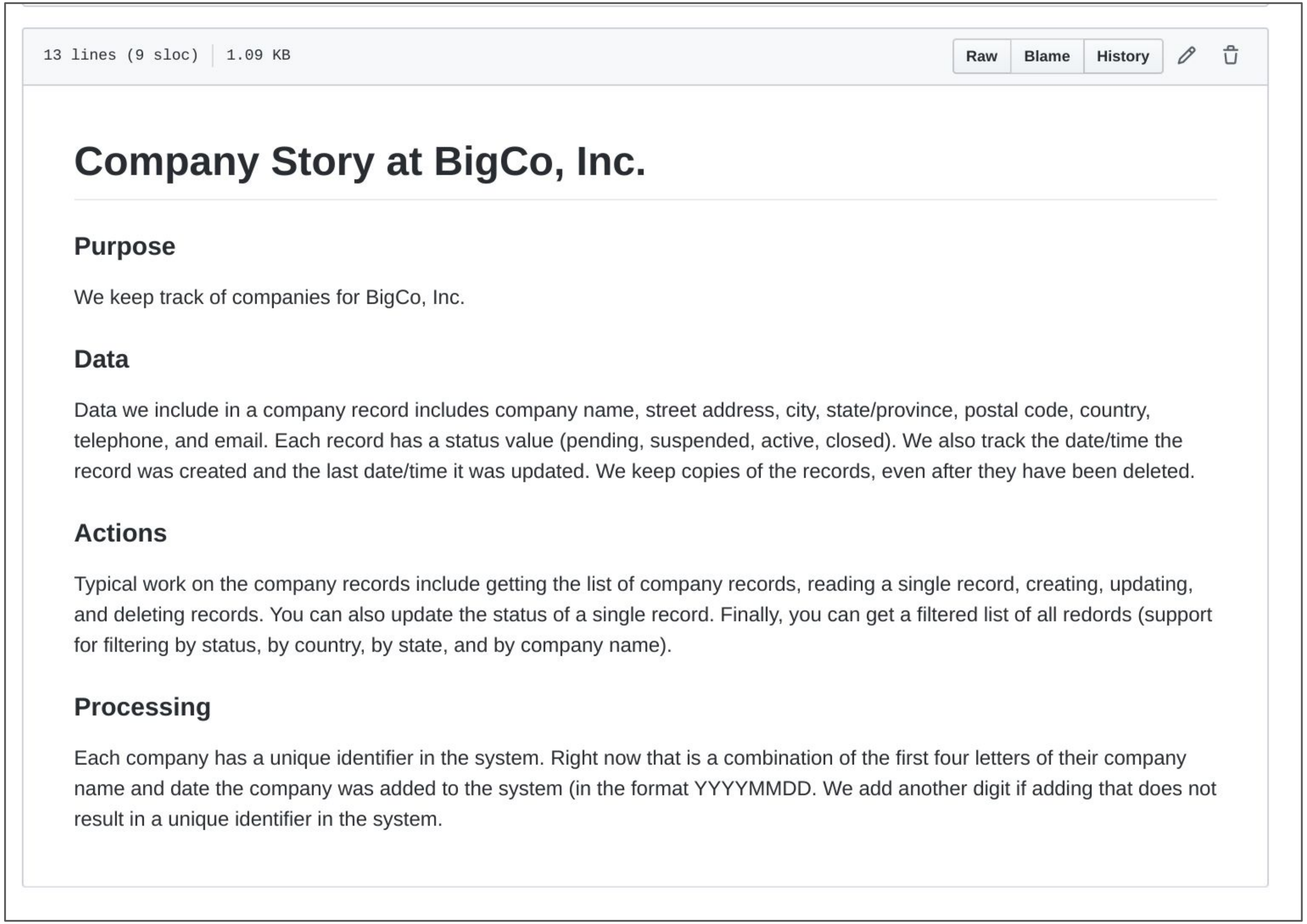

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

## Designing APIs

- Model
  - Norman's Lifecycle
  - There are no straight lines
  - The API Story
- Design
- Describe




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



13 lines (9 sloc) | 1.09 KB

# Company Story at BigCo, Inc.

## Purpose

We keep track of companies for BigCo, Inc.

## Data

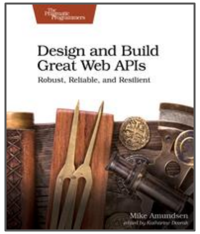
Data we include in a company record includes company name, street address, city, state/province, postal code, country, telephone, and email. Each record has a status value (pending, suspended, active, closed). We also track the date/time the record was created and the last date/time it was updated. We keep copies of the records, even after they have been deleted.

## Actions

Typical work on the company records include getting the list of company records, reading a single record, creating, updating, and deleting records. You can also update the status of a single record. Finally, you can get a filtered list of all redords (support for filtering by status, by country, by state, and by company name).

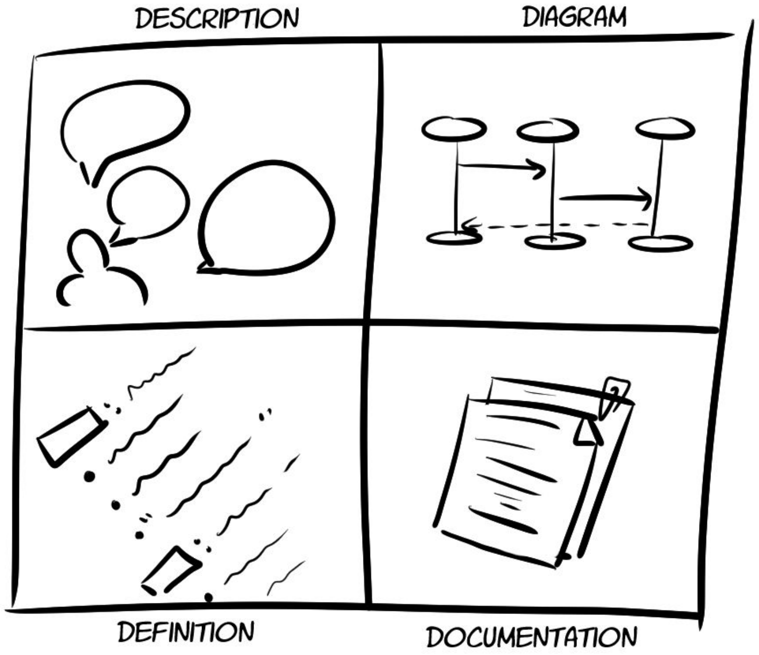
## Processing

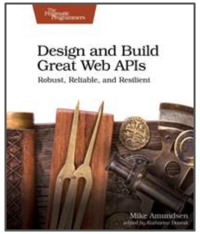
Each company has a unique identifier in the system. Right now that is a combination of the first four letters of their company name and date the company was added to the system (in the format YYYYMMDD. We add another digit if adding that does not result in a unique identifier in the system.

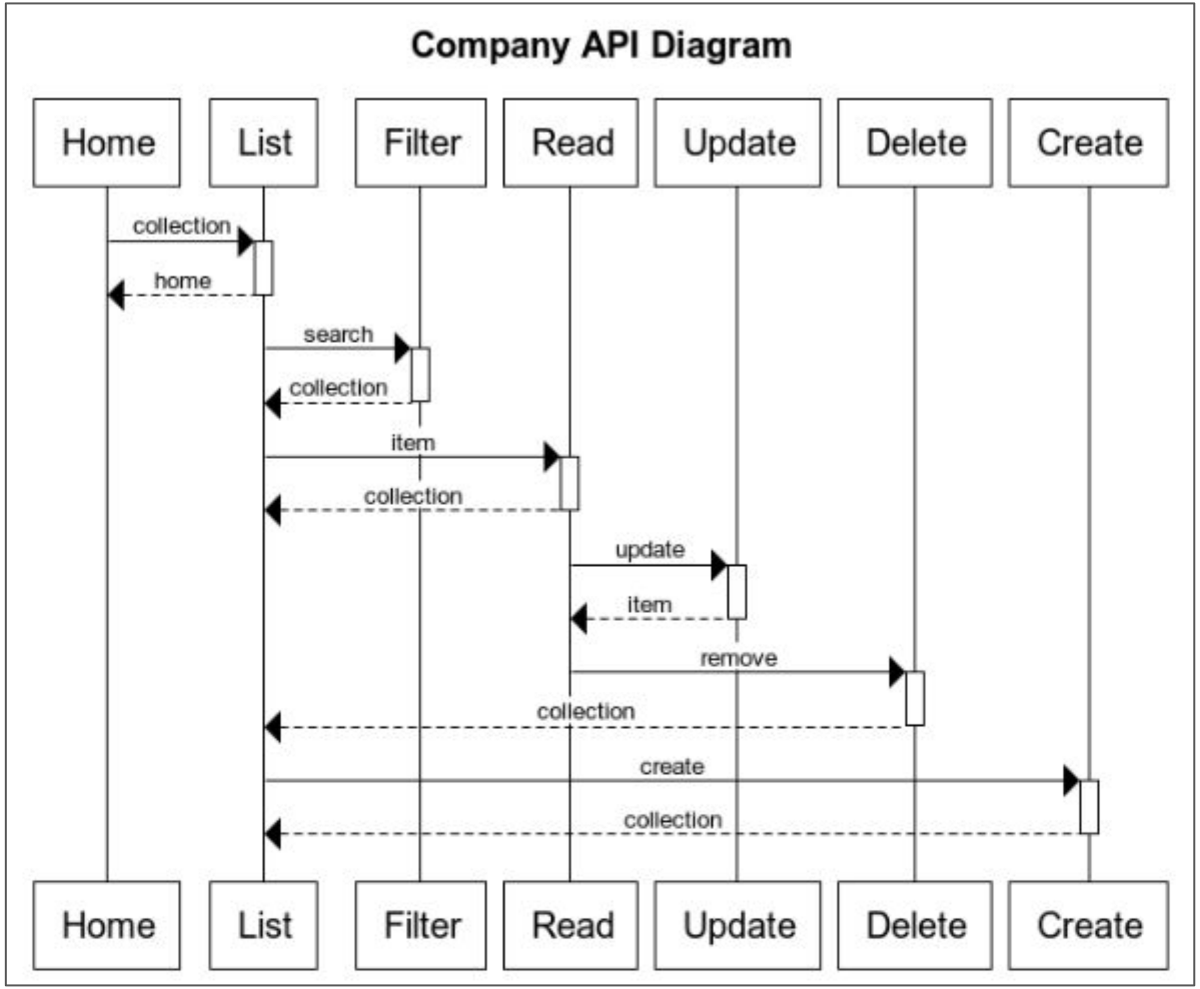

Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

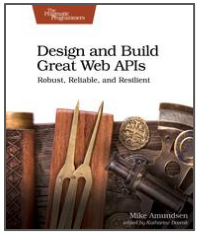
# Designing APIs

- Model
- Design
  - Design Thinking
  - Jobs-to-be-Done
  - The API Diagram
- Describe



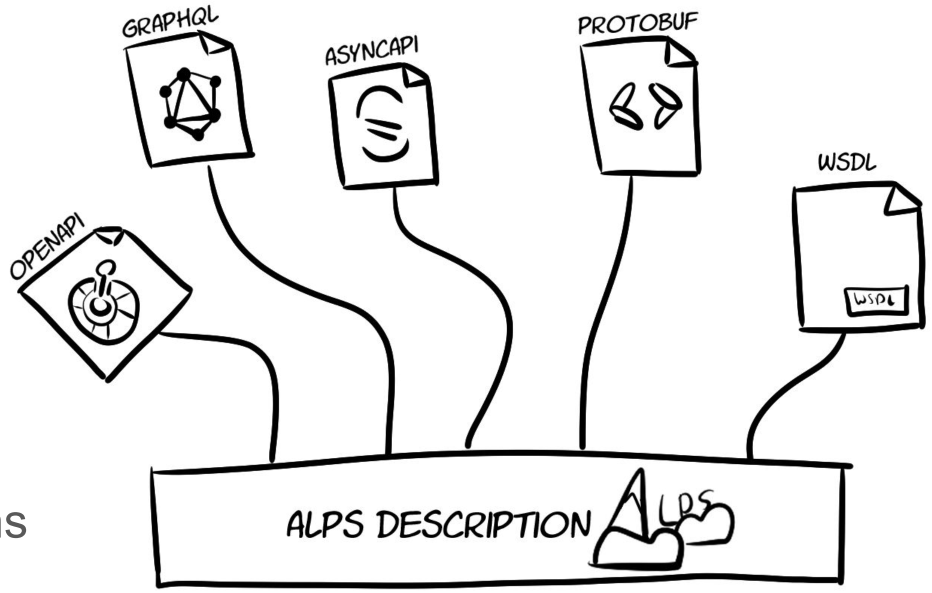

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen


Company API Diagram
Home
List
Filter
Read
Update
Delete
Create
collection
home
search
collection
item
collection
update
item
remove
collection
create
collection
Home
List
Filter
Read
Update
Delete
Create

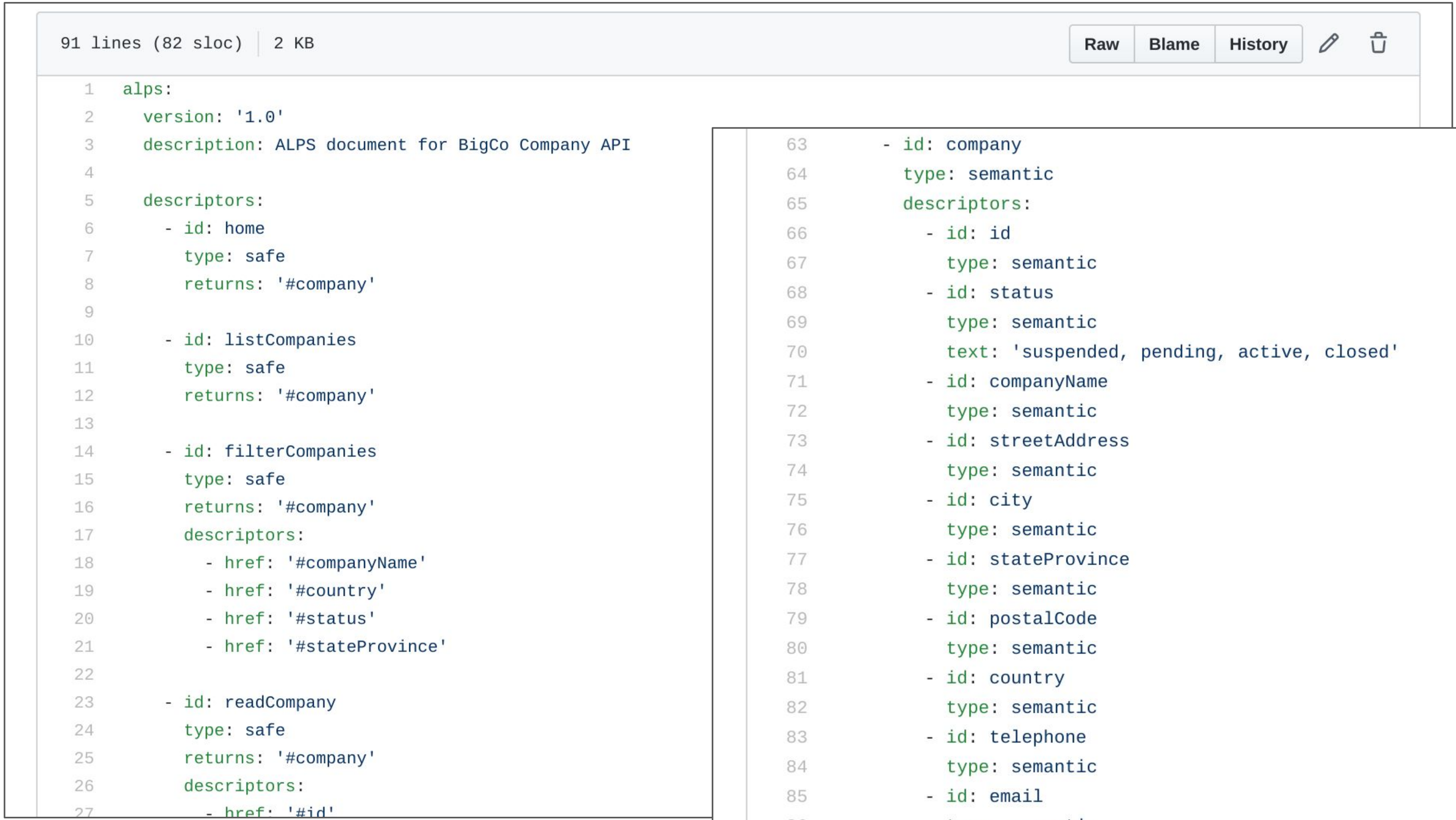

Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Designing APIs

- Model
- Design
- Describe
  - Technology agnostic
  - Details on properties & actions
  - ALPS (2014)
  - **`https://github.com/mamund/2020-04-unified-api-design`**

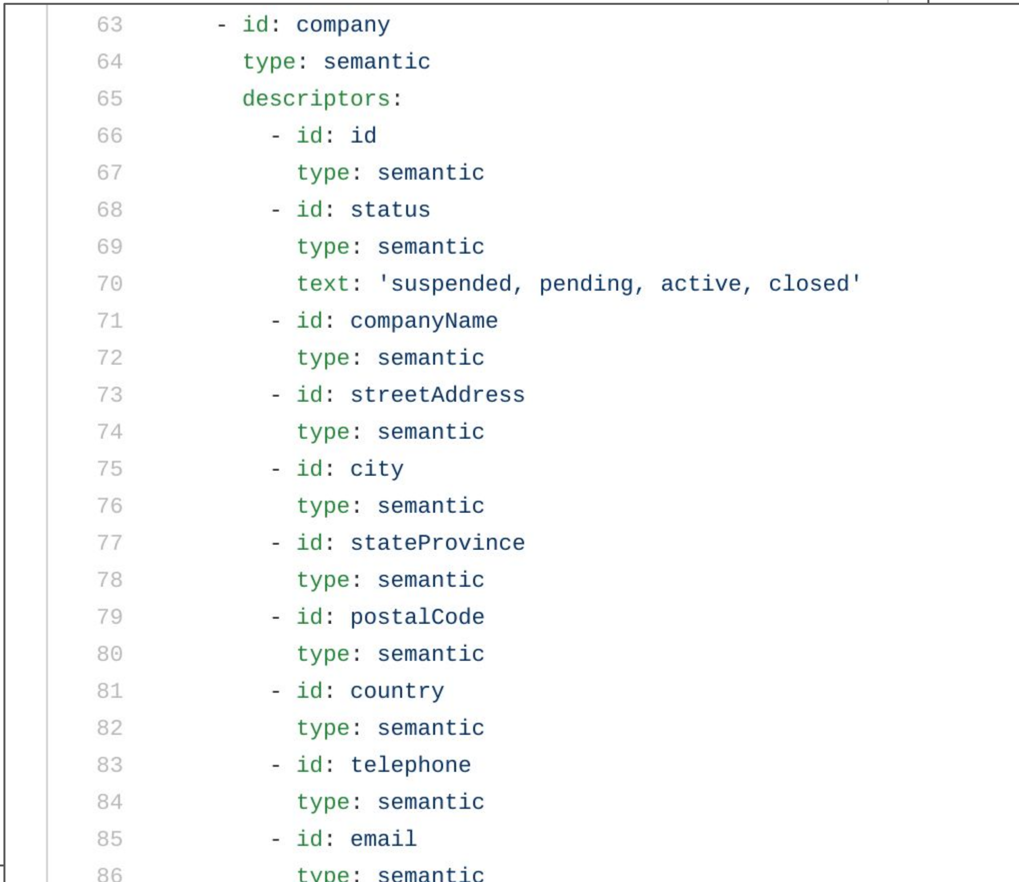


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen



91 lines (82 sloc) | 2 KB

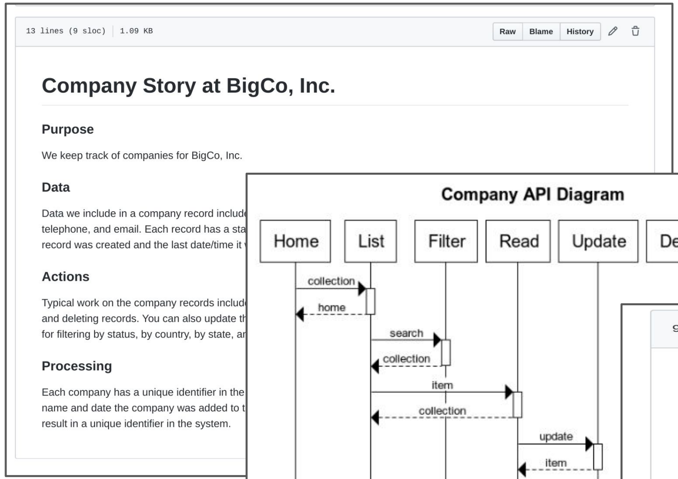
```yaml
alps:
  version: '1.0'
  description: ALPS document for BigCo Company API

  descriptors:
    - id: home
      type: safe
      returns: '#company'

    - id: listCompanies
      type: safe
      returns: '#company'

    - id: filterCompanies
      type: safe
      returns: '#company'
      descriptors:
        - href: '#companyName'
        - href: '#country'
        - href: '#status'
        - href: '#stateProvince'

    - id: readCompany
      type: safe
      returns: '#company'
      descriptors:
        - href: '#id'
```



```yaml
    - id: company
      type: semantic
      descriptors:
        - id: id
          type: semantic
        - id: status
          type: semantic
          text: 'suspended, pending, active, closed'
        - id: companyName
          type: semantic
        - id: streetAddress
          type: semantic
        - id: city
          type: semantic
        - id: stateProvince
          type: semantic
        - id: postalCode
          type: semantic
        - id: country
          type: semantic
        - id: telephone
          type: semantic
        - id: email
          type: semantic
```

# Designing APIs

- Model
  - API Story
- Design
  - API Diagram
- Describe
  - API Description



13 lines (9 sloc) | 1.09 KB

**Company Story at BigCo, Inc.**

**Purpose**

We keep track of companies for BigCo, Inc.

**Data**

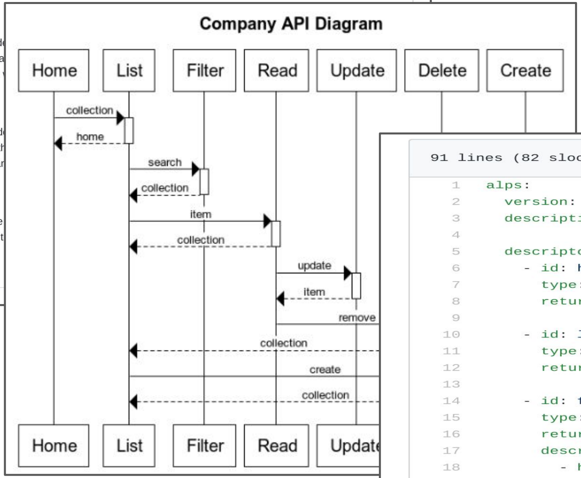
Data we include in a company record include
telephone, and email. Each record has a sta
record was created and the last date/time it

**Actions**

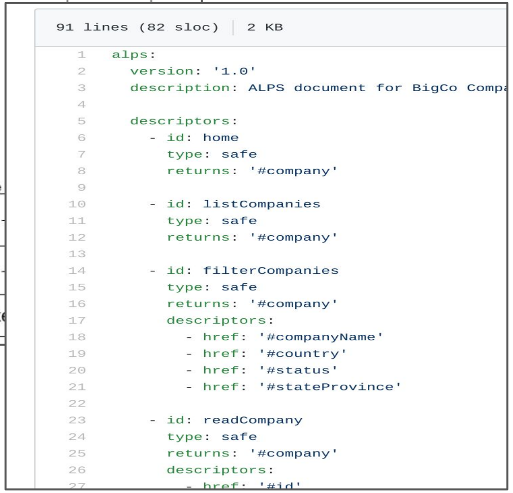
Typical work on the company records includ
and deleting records. You can also update th
for filtering by status, by country, by state, ar

**Processing**

Each company has a unique identifier in the
name and date the company was added to t
result in a unique identifier in the system.

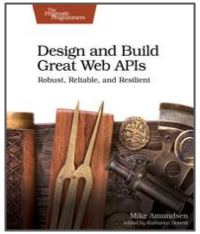


**Company API Diagram**



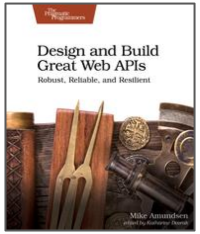
91 lines (82 sloc) | 2 KB

```yaml
alps:
  version: '1.0'
  description: ALPS document for BigCo Comp

  descriptors:
    - id: home
      type: safe
      returns: '#company'

    - id: listCompanies
      type: safe
      returns: '#company'

    - id: filterCompanies
      type: safe
      returns: '#company'
      descriptors:
        - href: '#companyName'
        - href: '#country'
        - href: '#status'
        - href: '#stateProvince'

    - id: readCompany
      type: safe
      returns: '#company'
      descriptors:
        - href: '#id'
```



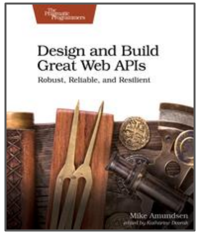
The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Building APIs


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

## Building APIs

- Sketching
- Prototyping
- Building


The Pragmatic Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient

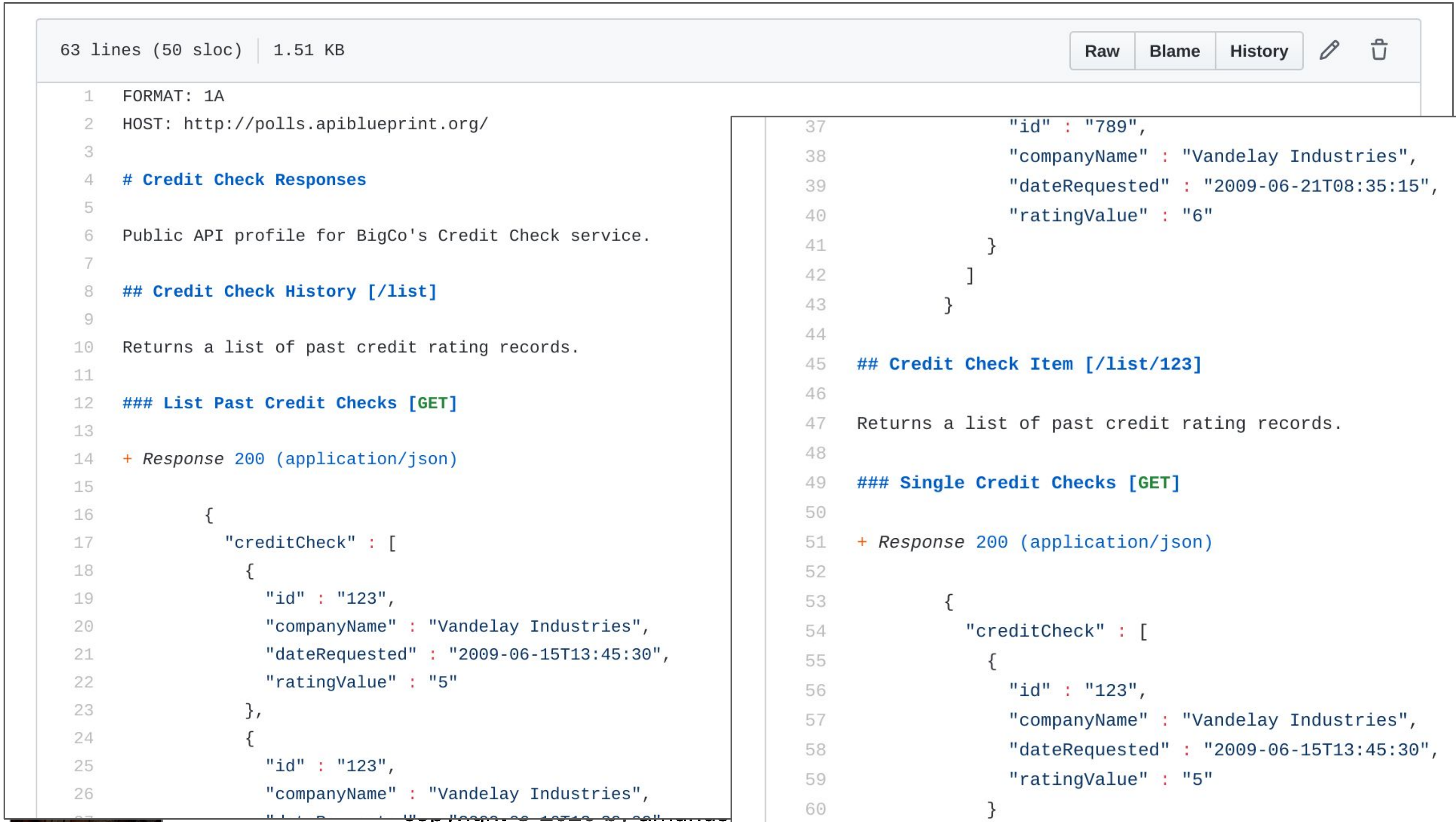
Mike Amundsen

edited by Katharine Dvorak

## Building APIs

- Sketching
  - Frank Gehry (via Ronnie Mitra)
  - Experiment w/ possibilities
  - Apiary Blueprint (APIB)
  - Sketches are meant to be thrown away
- Prototyping
- Building

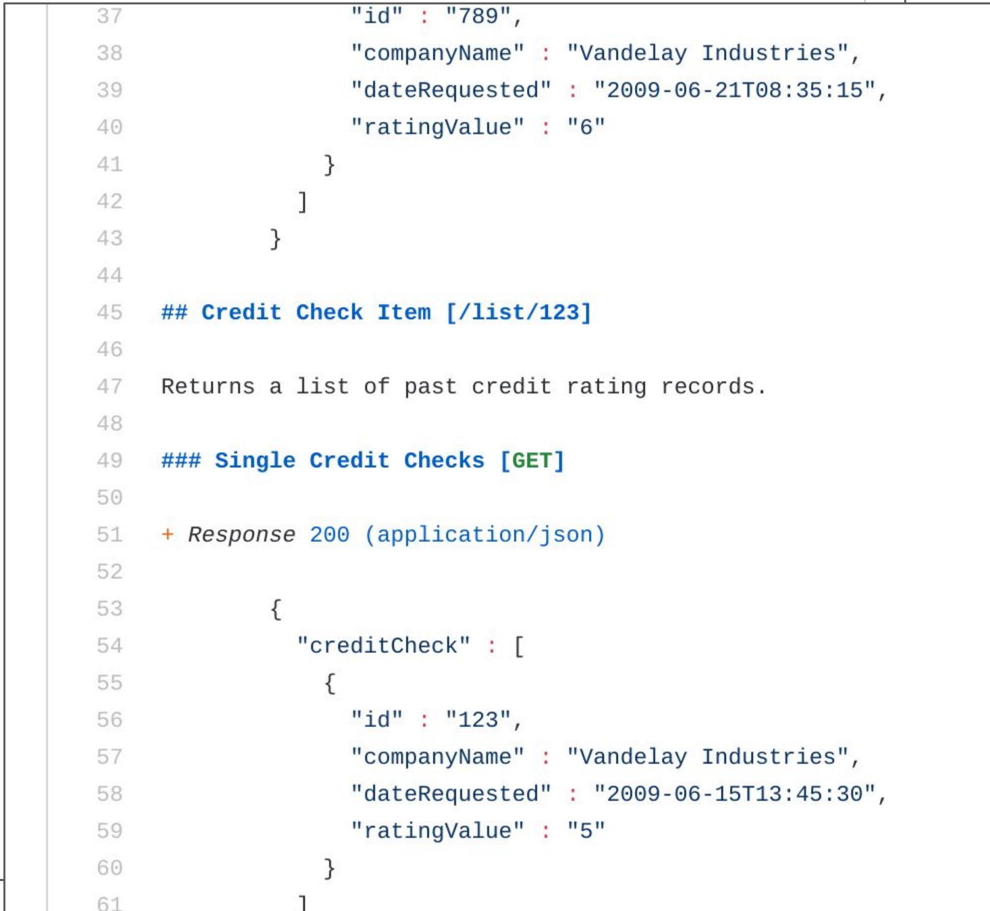
The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak



63 lines (50 sloc) | 1.51 KB

Raw | Blame | History

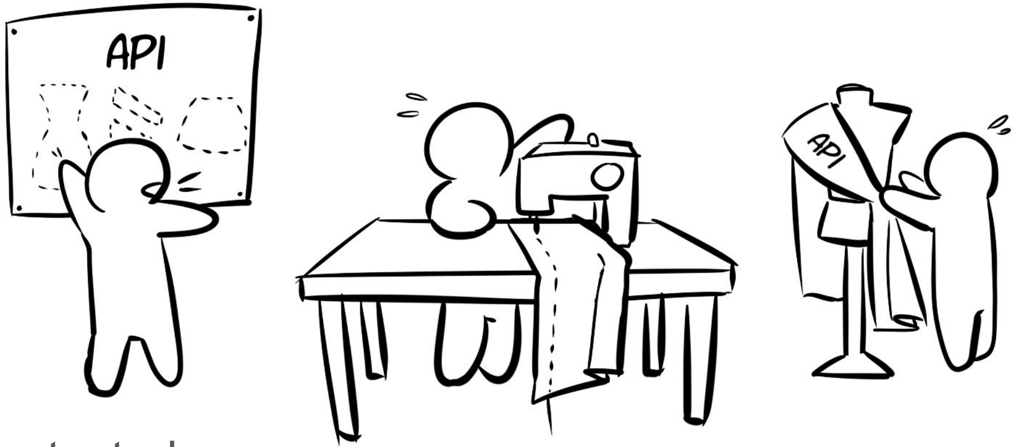
```
FORMAT: 1A
HOST: http://polls.apiblueprint.org/

# Credit Check Responses

Public API profile for BigCo's Credit Check service.

## Credit Check History [/list]

Returns a list of past credit rating records.

### List Past Credit Checks [GET]

+ Response 200 (application/json)

        {
          "creditCheck" : [
            {
              "id" : "123",
              "companyName" : "Vandelay Industries",
              "dateRequested" : "2009-06-15T13:45:30",
              "ratingValue" : "5"
            },
            {
              "id" : "123",
              "companyName" : "Vandelay Industries",
```

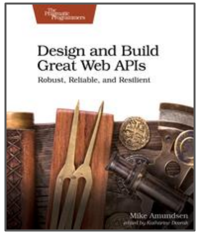


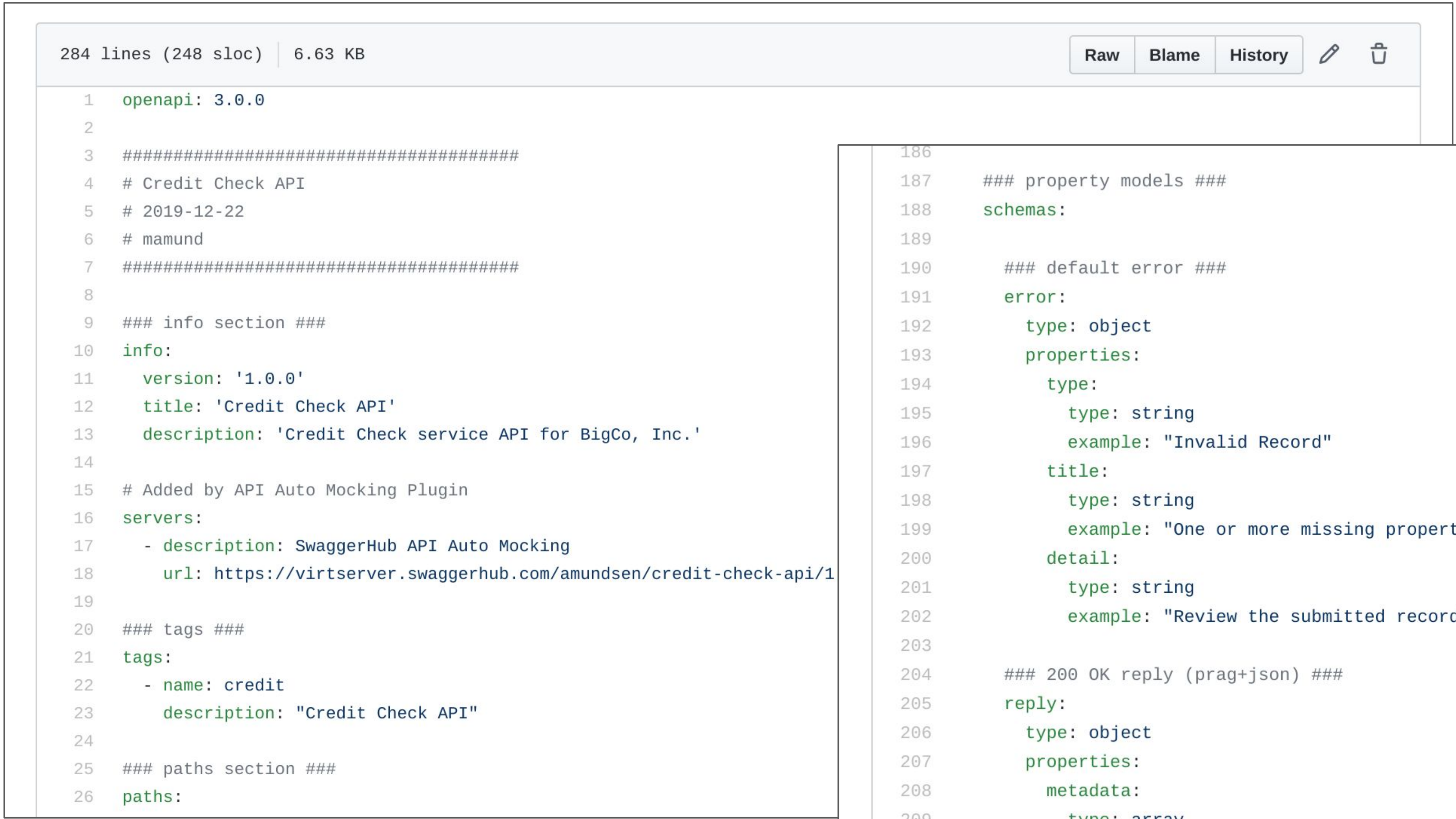
```
              "id" : "789",
              "companyName" : "Vandelay Industries",
              "dateRequested" : "2009-06-21T08:35:15",
              "ratingValue" : "6"
            }
          ]
        }

## Credit Check Item [/list/123]

Returns a list of past credit rating records.

### Single Credit Checks [GET]

+ Response 200 (application/json)

        {
          "creditCheck" : [
            {
              "id" : "123",
              "companyName" : "Vandelay Industries",
              "dateRequested" : "2009-06-15T13:45:30",
              "ratingValue" : "5"
            }
          ]
```

## Building APIs

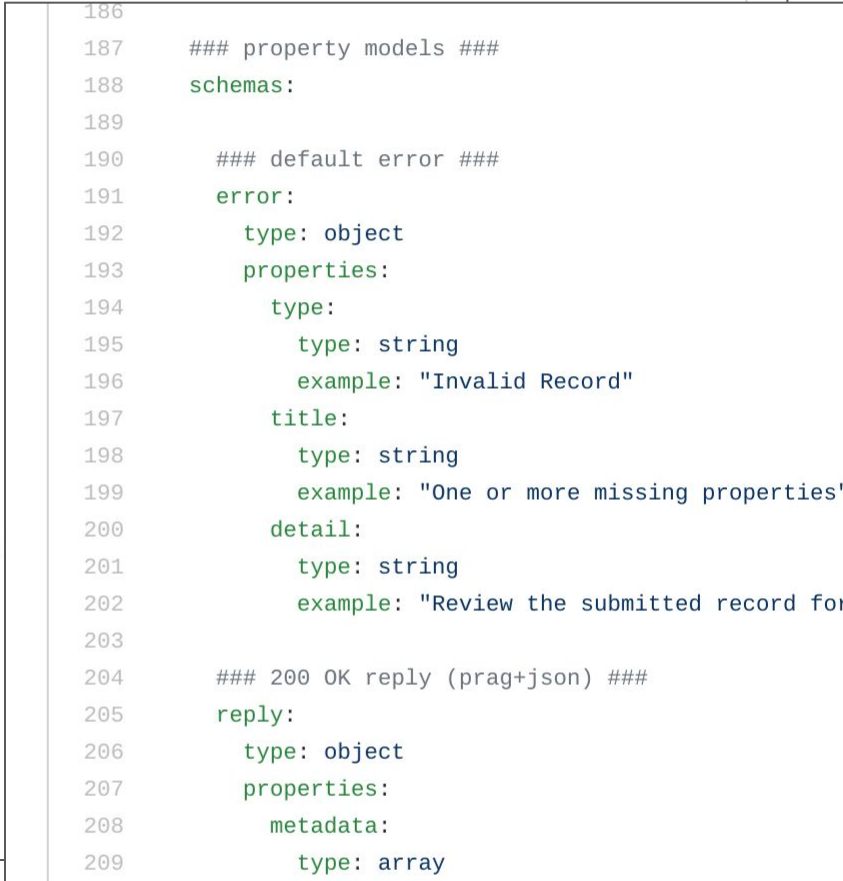
- Sketching
- Prototyping
  - Making a Toile
  - Explore the details
  - OpenAPI (OAS)
  - Prototypes are made to be tested
- Building




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



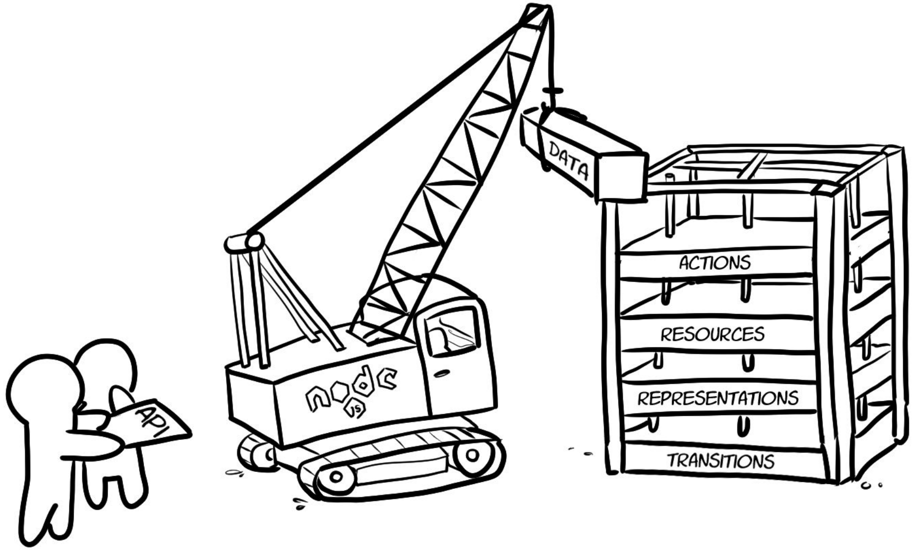
```yaml
openapi: 3.0.0

########################################
# Credit Check API
# 2019-12-22
# mamund
########################################

### info section ###
info:
  version: '1.0.0'
  title: 'Credit Check API'
  description: 'Credit Check service API for BigCo, Inc.'

# Added by API Auto Mocking Plugin
servers:
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/amundsen/credit-check-api/1

### tags ###
tags:
  - name: credit
    description: "Credit Check API"

### paths section ###
paths:
```

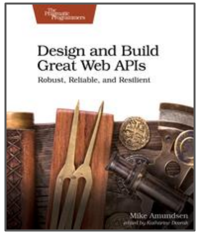


```yaml

### property models ###
schemas:

  ### default error ###
  error:
    type: object
    properties:
      type:
        type: string
        example: "Invalid Record"
      title:
        type: string
        example: "One or more missing properties'
      detail:
        type: string
        example: "Review the submitted record fo

  ### 200 OK reply (prag+json) ###
  reply:
    type: object
    properties:
      metadata:
        type: array
```

## Building APIs

- Sketching
- Prototyping
- Building
    - Repeatable process
    - Convert prototype/design into code
    - DARRT
        - Data, Actions, Resources, Representations, Transitions

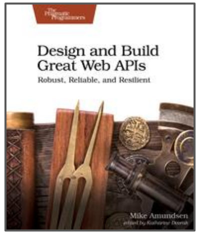



The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Building APIs : DARRT

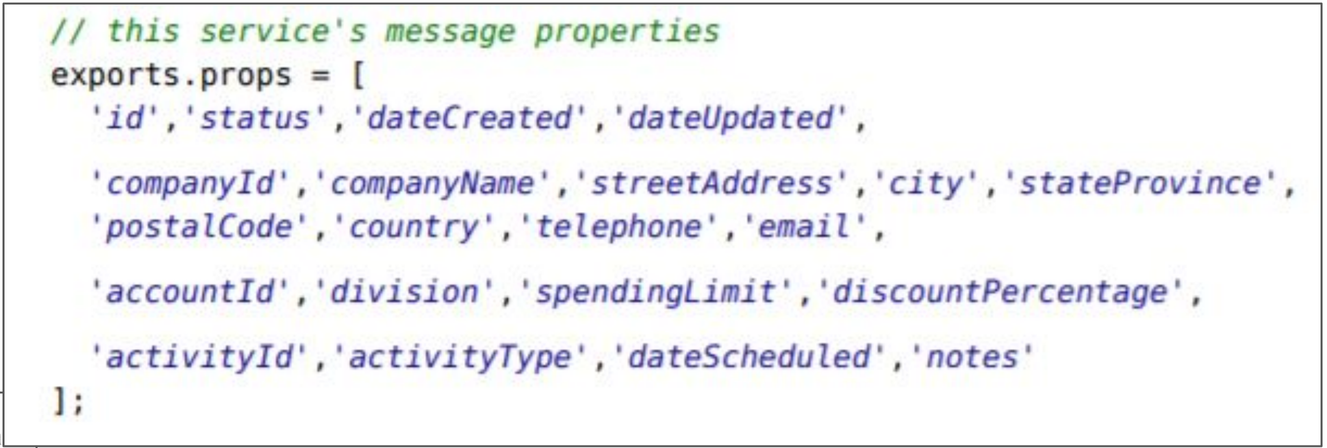
A simple, repeatable process for publishing API interfaces

- Data
- Actions
- Resources
- Representations
- Transitions

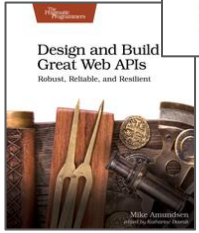

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Building APIs : DARRT : Data

- The state properties to pass in messages
  - **`properties, requireds, enums, defaults`**

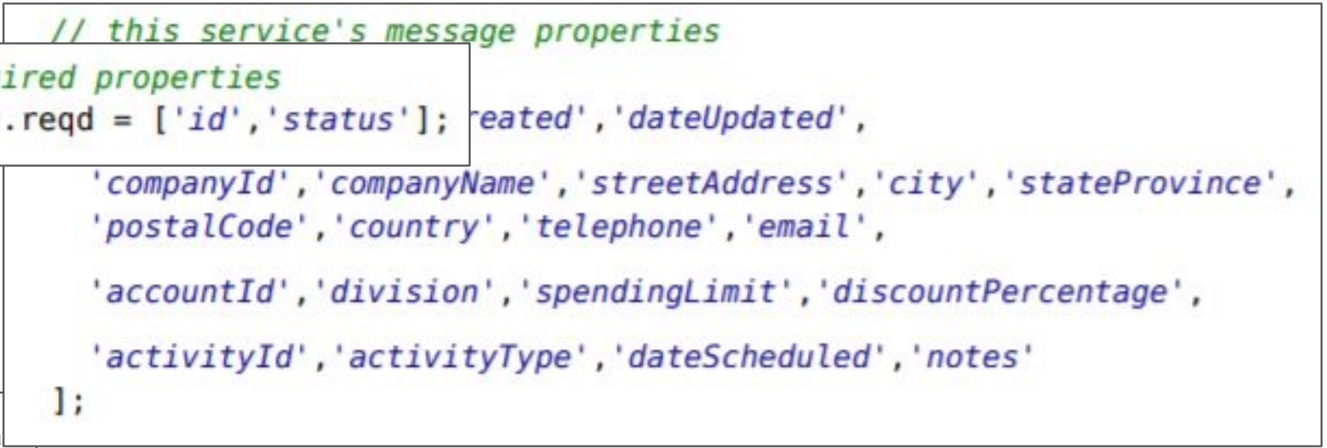


```
// this service's message properties
exports.props = [
  'id','status','dateCreated','dateUpdated',

  'companyId','companyName','streetAddress','city','stateProvince',
  'postalCode','country','telephone','email',

  'accountId','division','spendingLimit','discountPercentage',

  'activityId','activityType','dateScheduled','notes'
];
```

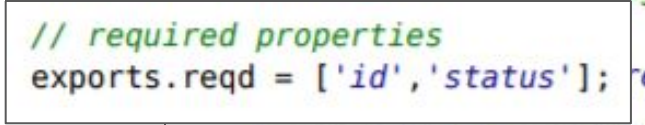


# Building APIs : DARRT : Data

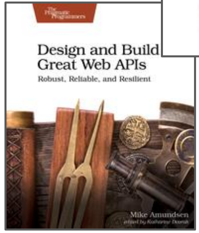
- The state properties to pass in messages
  - **`properties, requireds, enums, defaults`**



```
// this service's message properties

eated','dateUpdated',
    'companyId','companyName','streetAddress','city','stateProvince',
    'postalCode','country','telephone','email',
    'accountId','division','spendingLimit','discountPercentage',
    'activityId','activityType','dateScheduled','notes'
];
```

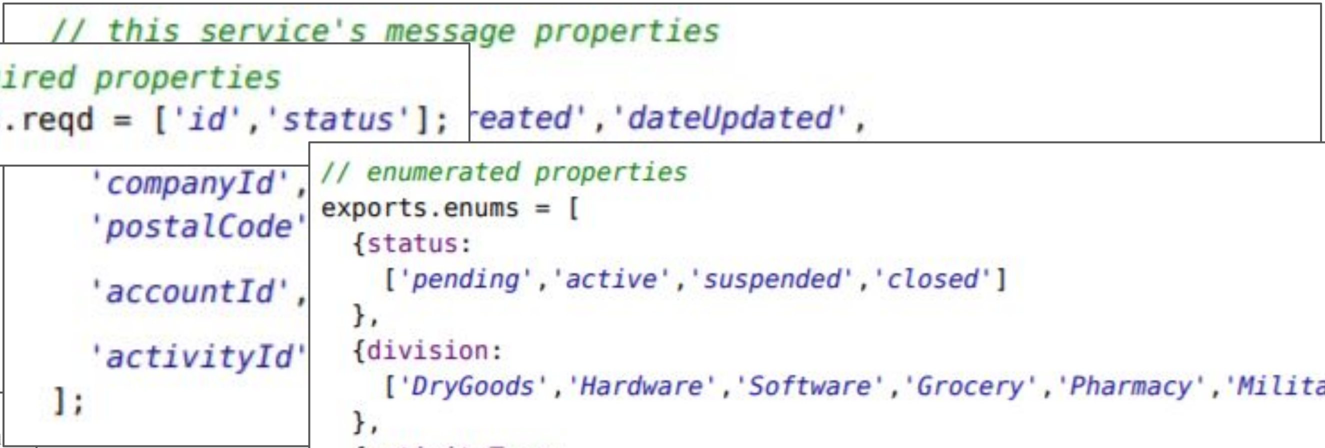


```
// required properties
exports.reqd = ['id','status'];
```

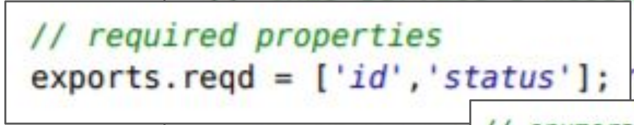

Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Building APIs : DARRT : Data

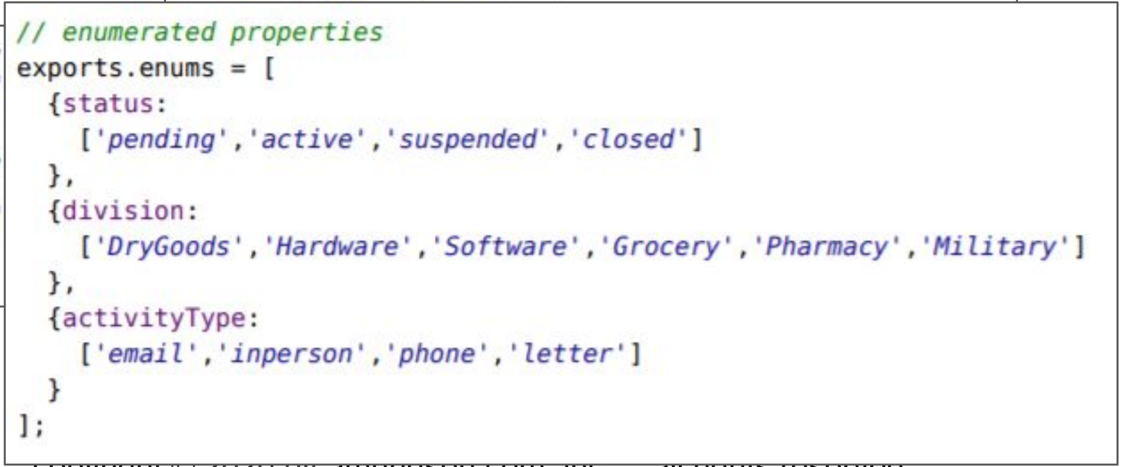
- The state properties to pass in messages
  - **`properties, requireds, enums, defaults`**



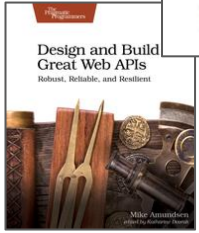
```
// this service's message properties
                                        reated','dateUpdated',
    'companyId',
    'postalCode'

    'accountId',

    'activityId'
];
```



```
// required properties
exports.reqd = ['id','status'];
```

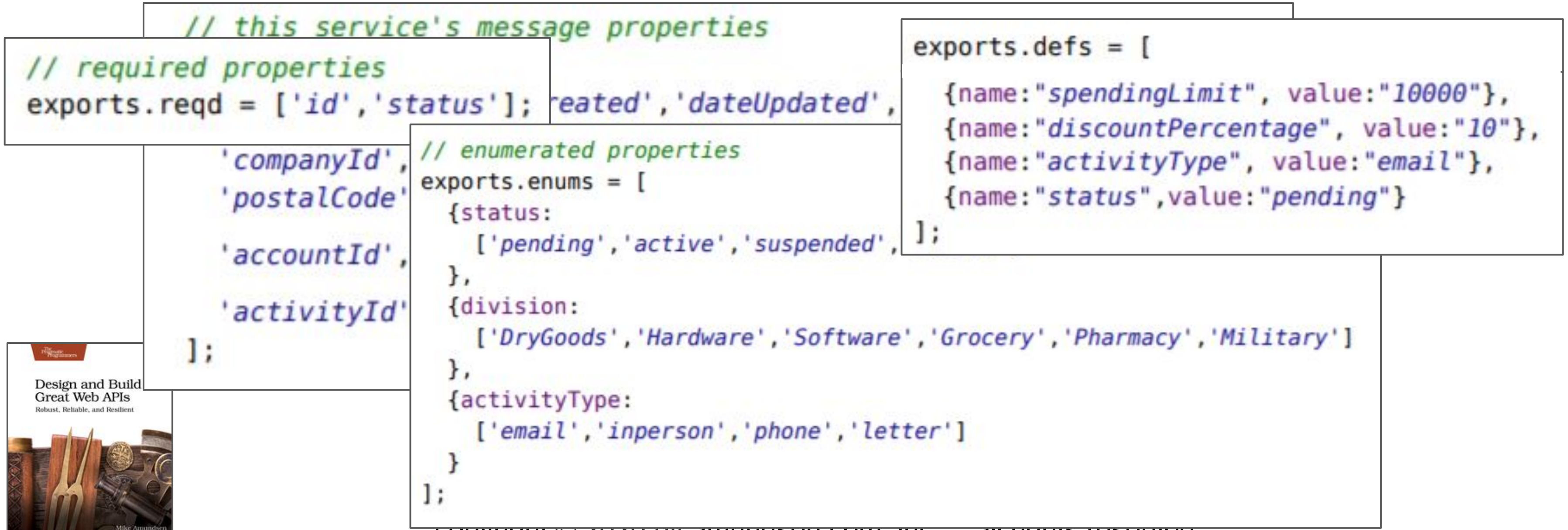


```
// enumerated properties
exports.enums = [
  {status:
    ['pending','active','suspended','closed']
  },
  {division:
    ['DryGoods','Hardware','Software','Grocery','Pharmacy','Military']
  },
  {activityType:
    ['email','inperson','phone','letter']
  }
];
```


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

# Building APIs : DARRT : Data

- The state properties to pass in messages
  - **`properties, requireds, enums, defs`**

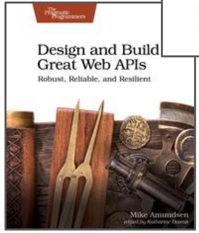


```
// this service's message properties
                                   eated','dateUpdated',
      'companyId',
      'postalCode'

      'accountId',

      'activityId'
  ];
```

```
// required properties
exports.reqd = ['id','status'];
```

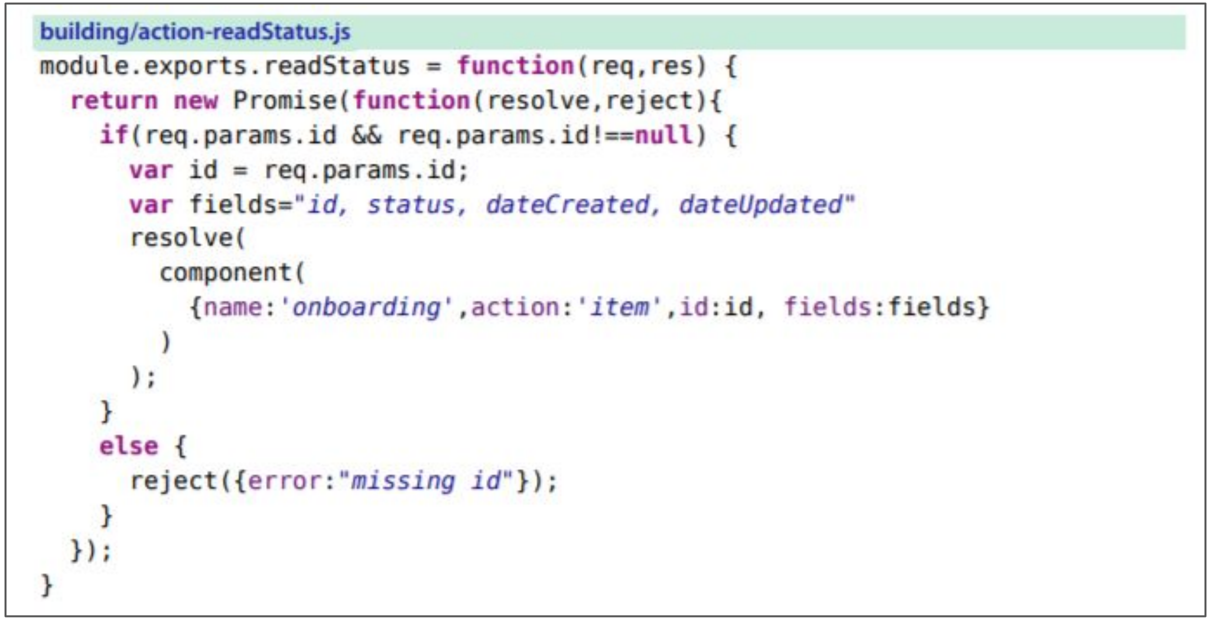
```
// enumerated properties
exports.enums = [
  {status:
    ['pending','active','suspended',
  },
  {division:
    ['DryGoods','Hardware','Software','Grocery','Pharmacy','Military']
  },
  {activityType:
    ['email','inperson','phone','letter']
  }
];
```

```
exports.defs = [

  {name:"spendingLimit", value:"10000"},
  {name:"discountPercentage", value:"10"},
  {name:"activityType", value:"email"},
  {name:"status",value:"pending"}
];
```


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

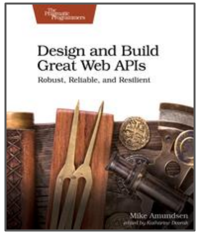
## Building APIs : DARRT : Actions

- The actual internal operations for the interface
  - **`approvePayroll, updateCustomer, setStatus`**



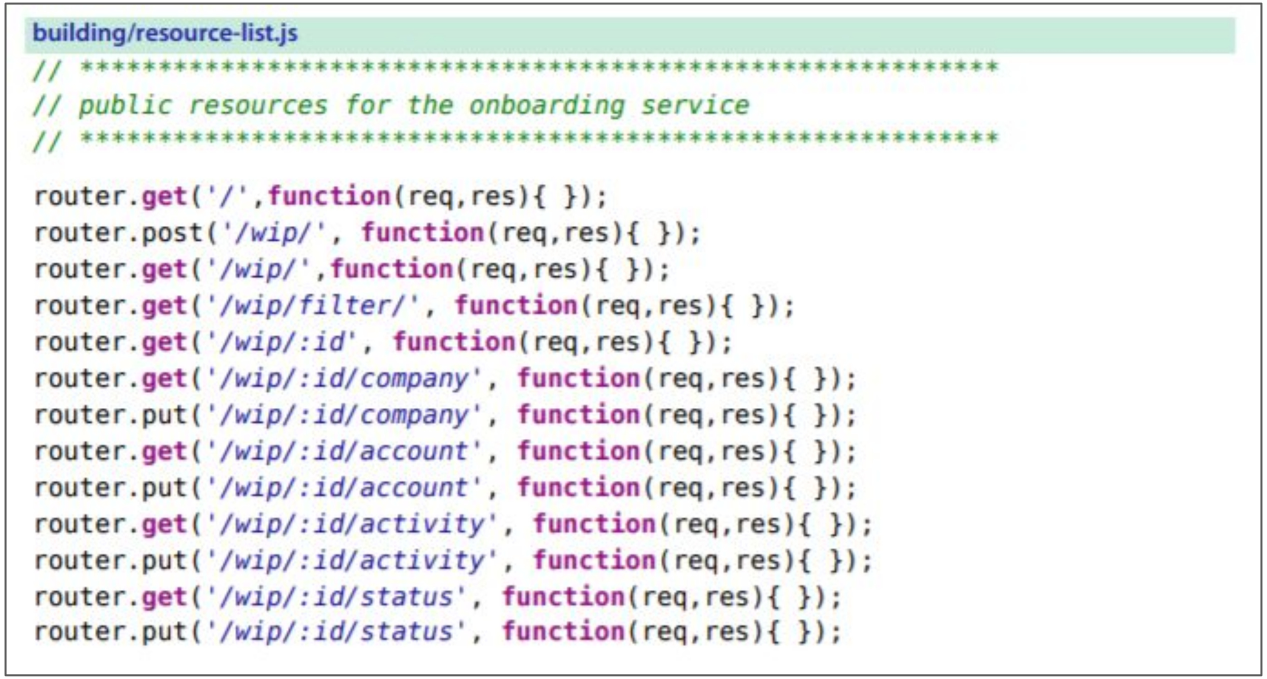
building/action-readStatus.js

```
module.exports.readStatus = function(req,res) {
  return new Promise(function(resolve,reject){
    if(req.params.id && req.params.id!==null) {
      var id = req.params.id;
      var fields="id, status, dateCreated, dateUpdated"
      resolve(
        component(
          {name:'onboarding',action:'item',id:id, fields:fields}
        )
      );
    }
    else {
      reject({error:"missing id"});
    }
  });
}
```



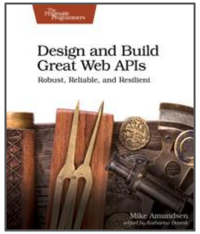
## Building APIs : DARRT : Resources

- The public HTTP resources to access the operations



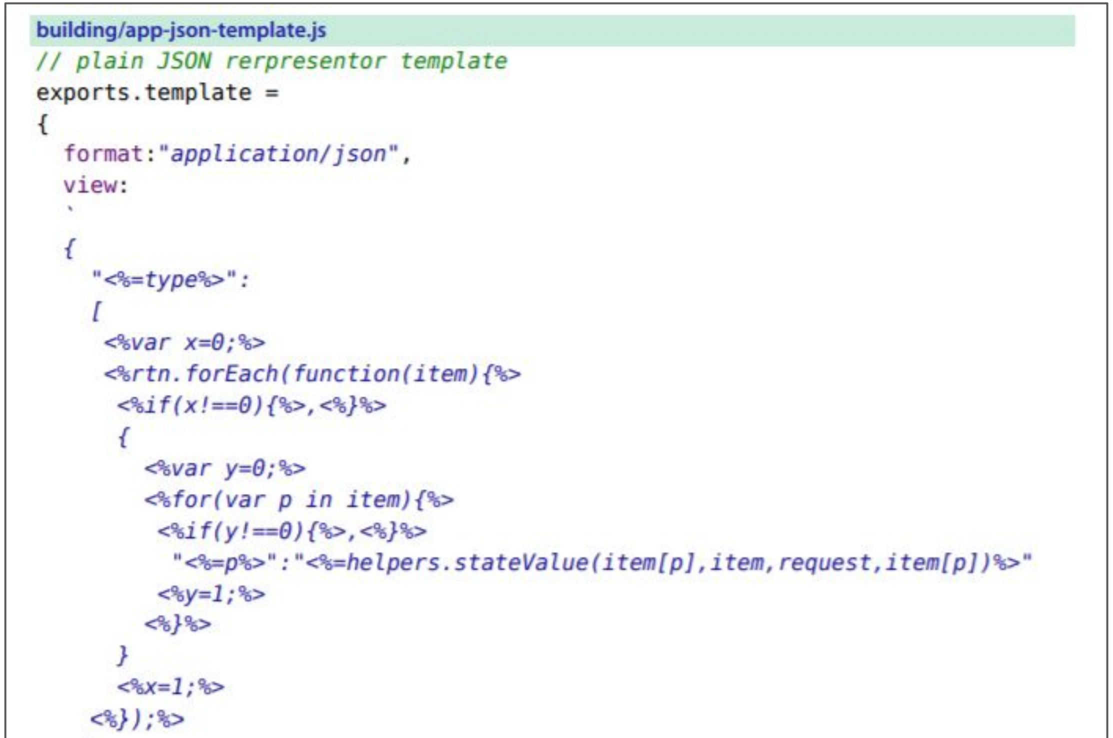
building/resource-list.js

```
// **************************************************************
// public resources for the onboarding service
// **************************************************************

router.get('/',function(req,res){ });
router.post('/wip/', function(req,res){ });
router.get('/wip/',function(req,res){ });
router.get('/wip/filter/', function(req,res){ });
router.get('/wip/:id', function(req,res){ });
router.get('/wip/:id/company', function(req,res){ });
router.put('/wip/:id/company', function(req,res){ });
router.get('/wip/:id/account', function(req,res){ });
router.put('/wip/:id/account', function(req,res){ });
router.get('/wip/:id/activity', function(req,res){ });
router.put('/wip/:id/activity', function(req,res){ });
router.get('/wip/:id/status', function(req,res){ });
router.put('/wip/:id/status', function(req,res){ });
```


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

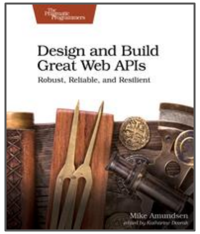
# Building APIs : DARRT : Representations

- Produce the requested format for resource responses



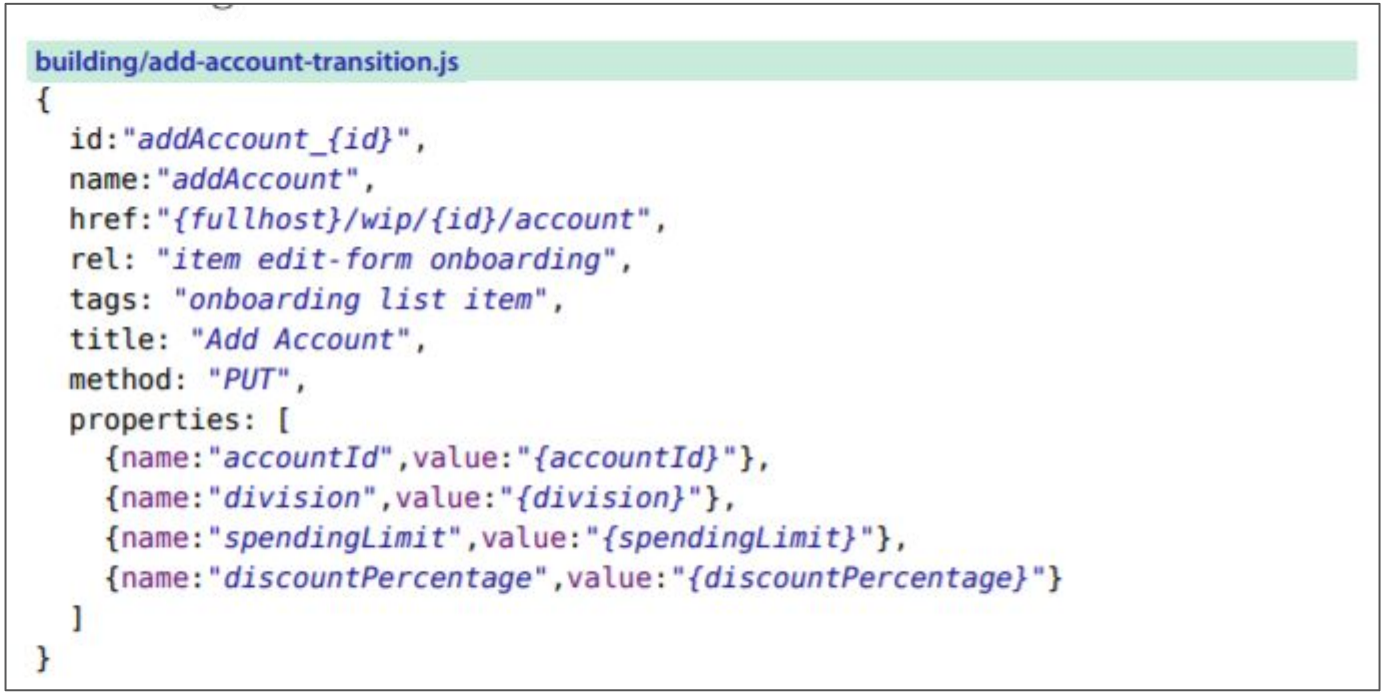
building/app-json-template.js

```
// plain JSON rerpresentor template
exports.template =
{
  format:"application/json",
  view:
  `
  {
    "<%=type%>":
    [
     <%var x=0;%>
     <%rtn.forEach(function(item){%>
      <%if(x!==0){%>,<%}%>
      {
        <%var y=0;%>
        <%for(var p in item){%>
         <%if(y!==0){%>,<%}%>
          "<%=p%>":"<%=helpers.stateValue(item[p],item,request,item[p])%>"
         <%y=1;%>
        <%}%>
      }
      <%x=1;%>
    <%});%>
```


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

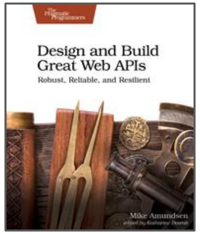
## Building APIs : DARRT : Transitions

- The list of public actions as expressed in HTTP



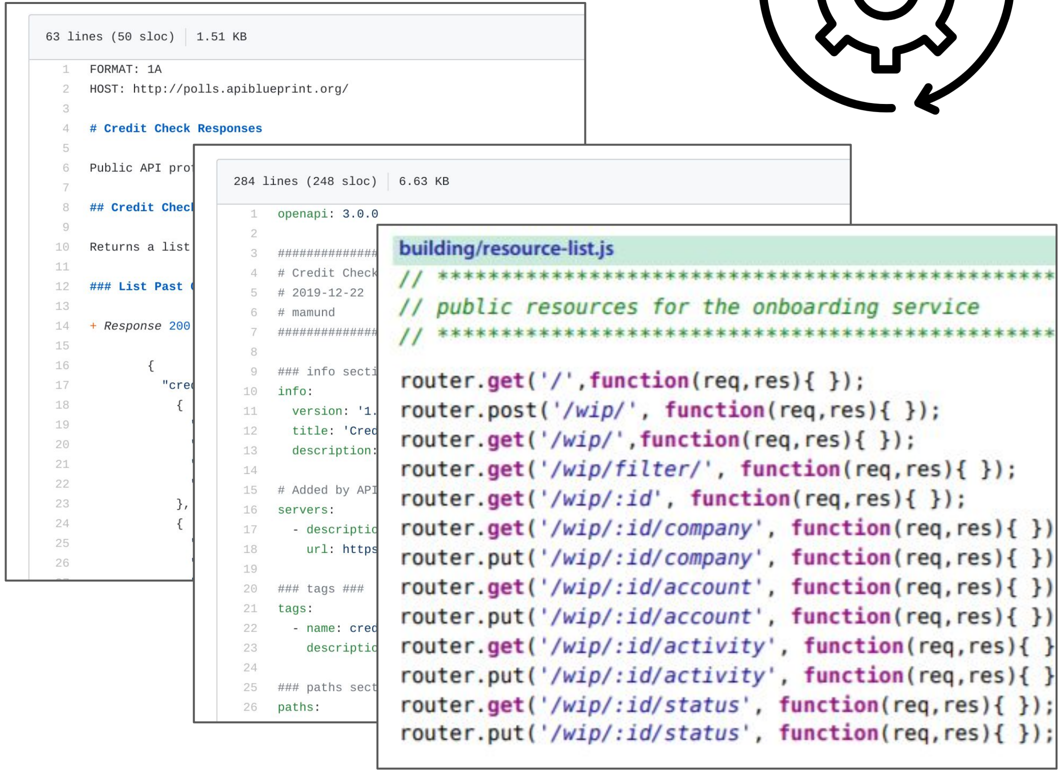
building/add-account-transition.js

```
{
  id:"addAccount_{id}",
  name:"addAccount",
  href:"{fullhost}/wip/{id}/account",
  rel: "item edit-form onboarding",
  tags: "onboarding list item",
  title: "Add Account",
  method: "PUT",
  properties: [
    {name:"accountId",value:"{accountId}"},
    {name:"division",value:"{division}"},
    {name:"spendingLimit",value:"{spendingLimit}"},
    {name:"discountPercentage",value:"{discountPercentage}"}
  ]
}
```



# Building APIs

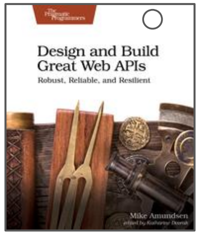
- Sketching
  - Experiment, throw away
  - APIB
- Prototyping
  - Explore, test
  - OpenAPI
- Building
  - Translate, repeat
  - NodeJS/DARRT



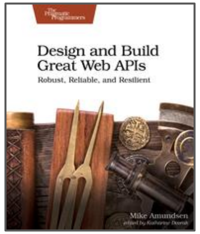
```
63 lines (50 sloc) | 1.51 KB
FORMAT: 1A
HOST: http://polls.apiblueprint.org/

# Credit Check Responses

Public API pro

## Credit Chec

Returns a list

### List Past

+ Response 200

        {
          "cre
            {



            },
            {

```

```
284 lines (248 sloc) | 6.63 KB
openapi: 3.0.0

##############
# Credit Check
# 2019-12-22
# mamund
##############

### info secti
info:
  version: '1.
  title: 'Cre
  description:

# Added by API
servers:
  - descriptio
    url: https

### tags ###
tags:
  - name: cred
    descriptio

### paths sect
paths:
```

```
building/resource-list.js
// *********************************************
// public resources for the onboarding service
// *********************************************

router.get('/',function(req,res){ });
router.post('/wip/', function(req,res){ });
router.get('/wip/',function(req,res){ });
router.get('/wip/filter/', function(req,res){ });
router.get('/wip/:id', function(req,res){ });
router.get('/wip/:id/company', function(req,res){ })
router.put('/wip/:id/company', function(req,res){ })
router.get('/wip/:id/account', function(req,res){ })
router.put('/wip/:id/account', function(req,res){ })
router.get('/wip/:id/activity', function(req,res){ }
router.put('/wip/:id/activity', function(req,res){ }
router.get('/wip/:id/status', function(req,res){ });
router.put('/wip/:id/status', function(req,res){ });
```

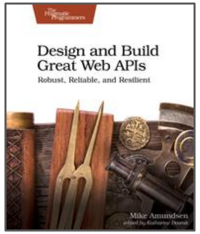

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

## Releasing APIs


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

## Releasing APIs

- Testing
- Security
- Deployment


The Pragmatic Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient

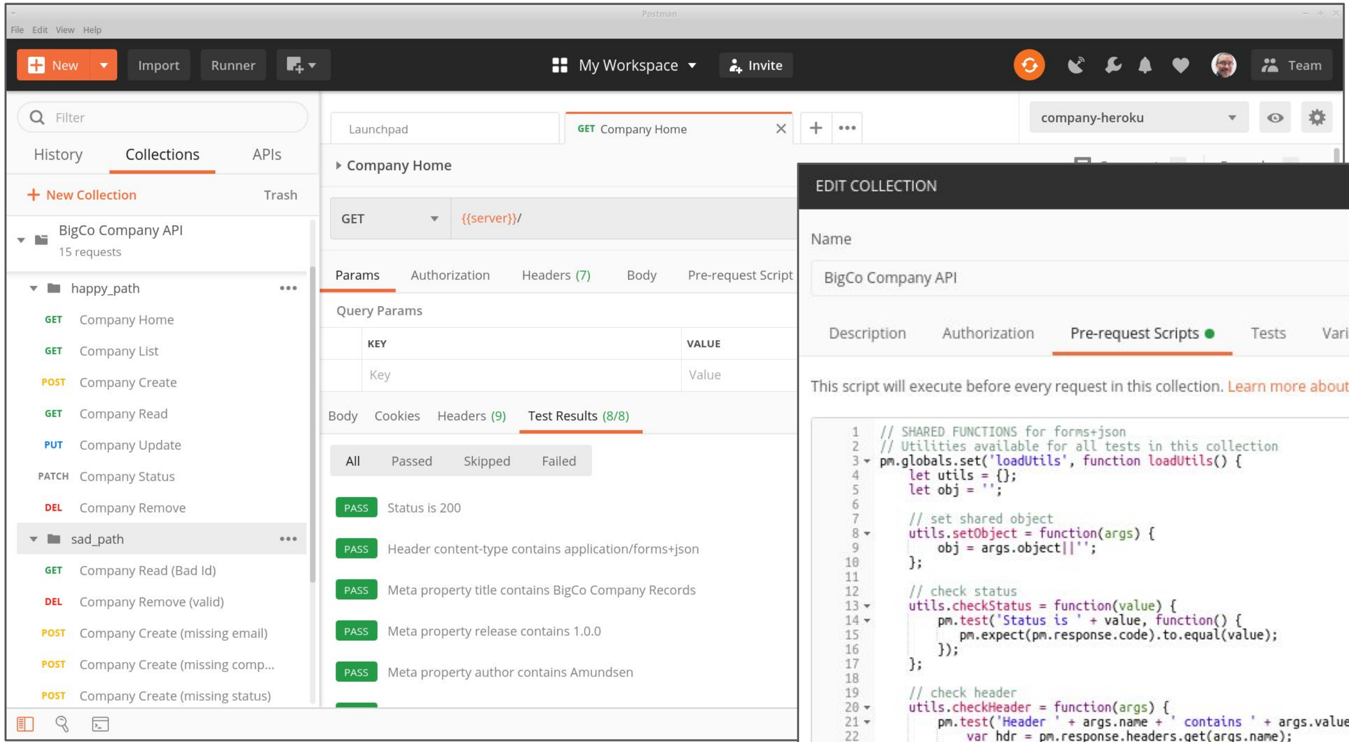
Mike Amundsen

edited by Katharine Dvorak

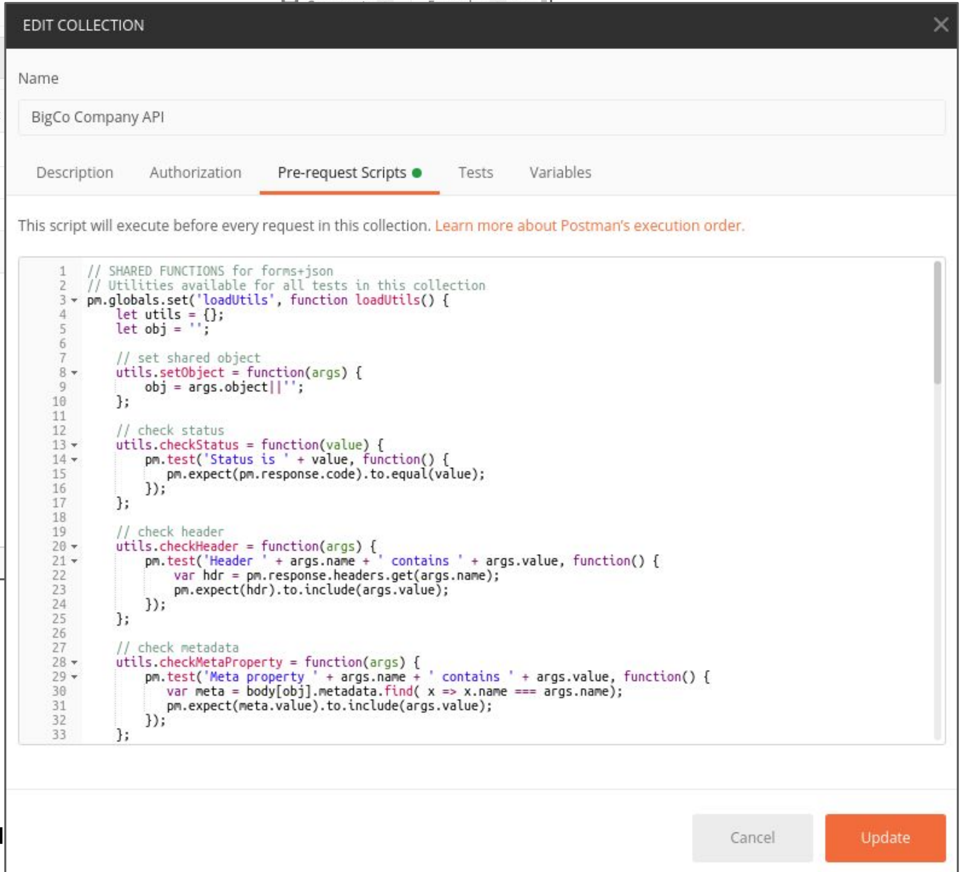
# Releasing APIs

- Testing
  - Testing Interface, not Code
  - Happy-path (200) & sad-path (400)
  - Simple Request Tests (SRTs)
  - Postman/Newman for BDD
- Security
- Deployment

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



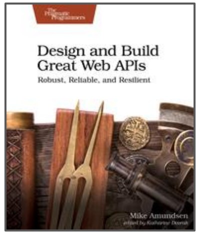
PASS Status is 200
PASS Header content-type contains application/forms+json
PASS Meta property title contains BigCo Company Records
PASS Meta property release contains 1.0.0
PASS Meta property author contains Amundsen


EDIT COLLECTION
Name
Description Authorization Pre-request Scripts Tests Variables
Cancel
Update

Name: BigCo Company API

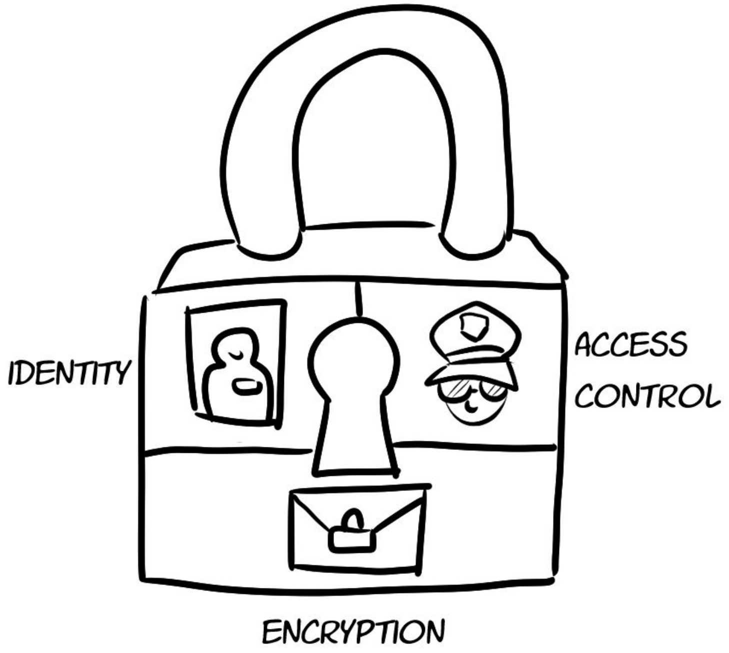
This script will execute before every request in this collection. Learn more about Postman's execution order.

```
// SHARED FUNCTIONS for forms+json
// Utilities available for all tests in this collection
pm.globals.set('loadUtils', function loadUtils() {
    let utils = {};
    let obj = '';

    // set shared object
    utils.setObject = function(args) {
        obj = args.object||'';
    };

    // check status
    utils.checkStatus = function(value) {
        pm.test('Status is ' + value, function() {
           pm.expect(pm.response.code).to.equal(value);
        });
    };

    // check header
    utils.checkHeader = function(args) {
        pm.test('Header ' + args.name + ' contains ' + args.value, function() {
            var hdr = pm.response.headers.get(args.name);
            pm.expect(hdr).to.include(args.value);
        });
    };

    // check metadata
    utils.checkMetaProperty = function(args) {
        pm.test('Meta property ' + args.name + ' contains ' + args.value, function() {
           var meta = body[obj].metadata.find( x => x.name === args.name);
           pm.expect(meta.value).to.include(args.value);
        });
    };
```

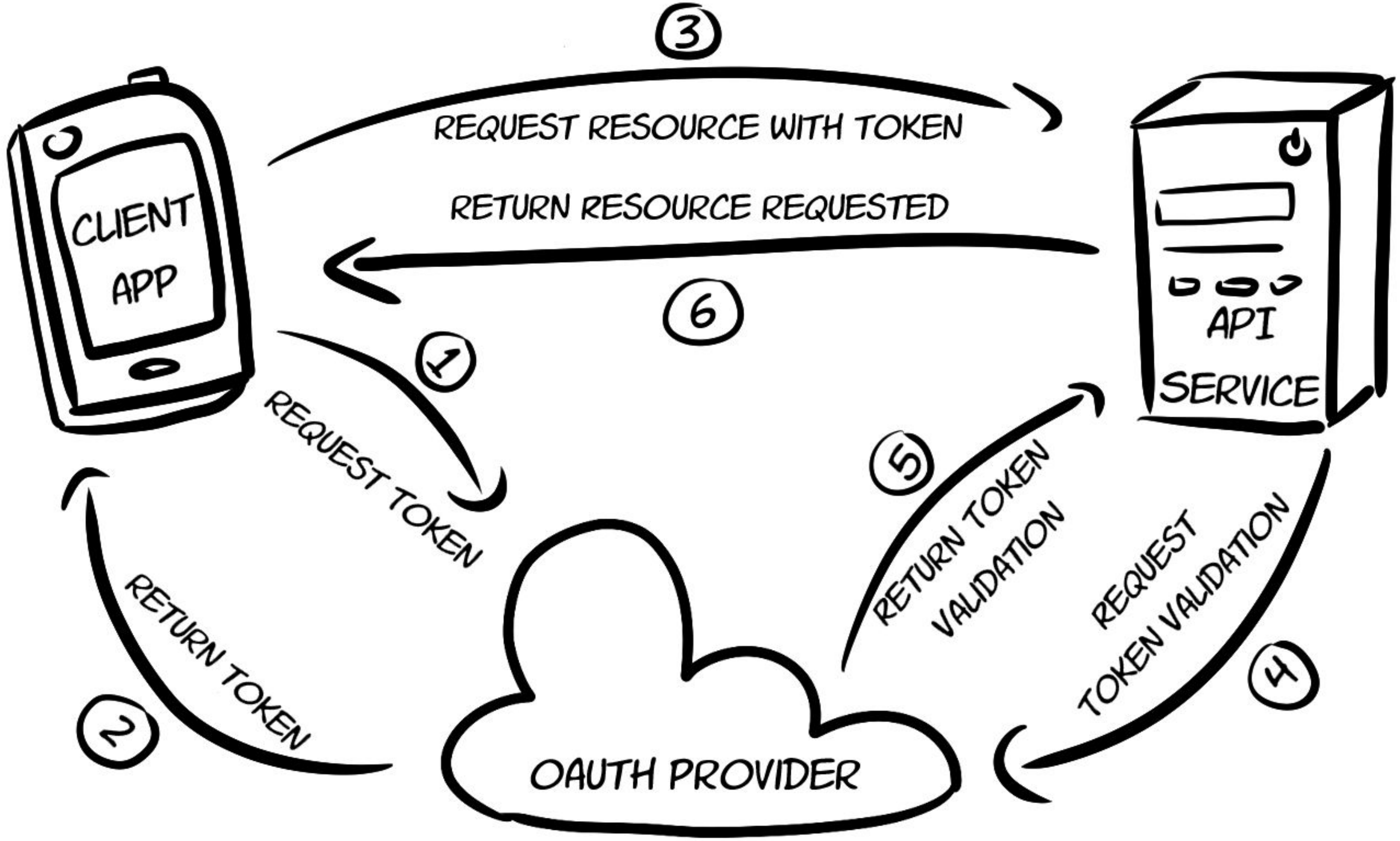

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

# Releasing APIs

- Testing
- Security
  - Encryption, Identity, Access Control
  - M2M Authorization w/ JWTs
  - Auth0 as provider
- Deployment

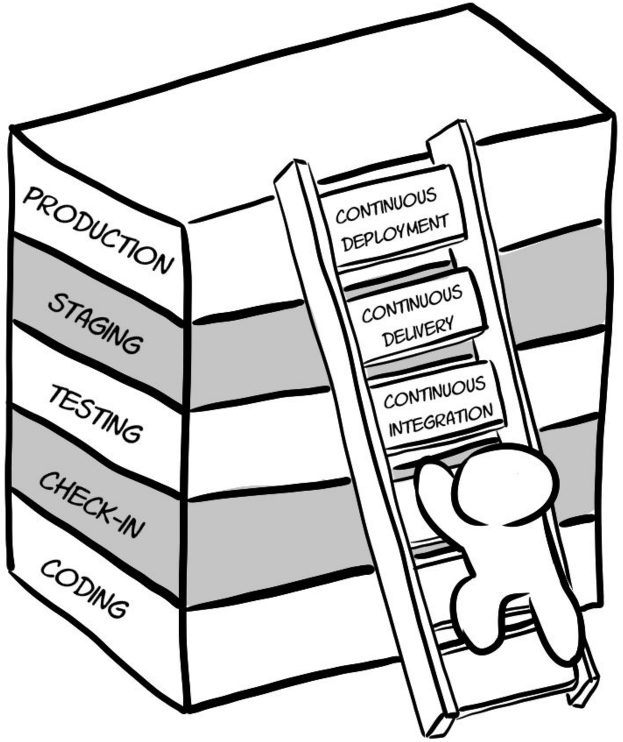


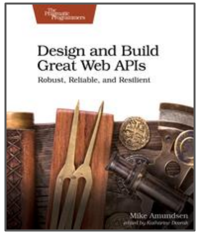
The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

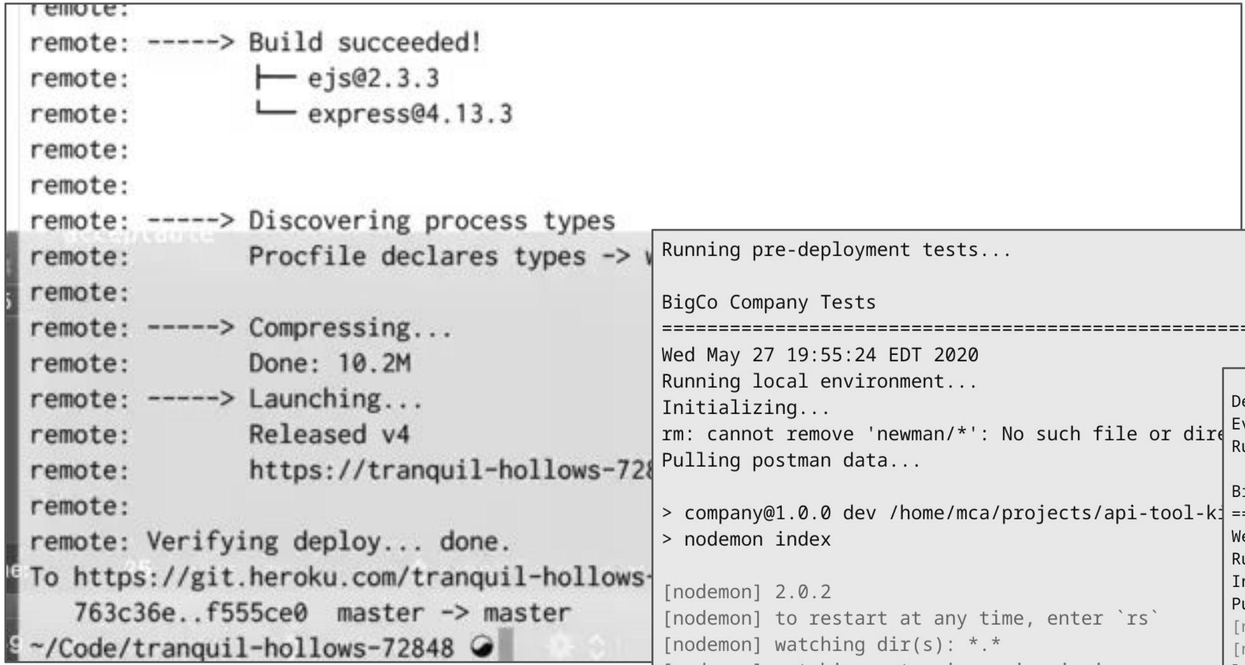

3
REQUEST RESOURCE WITH TOKEN
RETURN RESOURCE REQUESTED
6
CLIENT APP
API SERVICE
1
REQUEST TOKEN
2
RETURN TOKEN
5
RETURN TOKEN VALIDATION
REQUEST TOKEN VALIDATION
4
OAUTH PROVIDER

## Releasing APIs

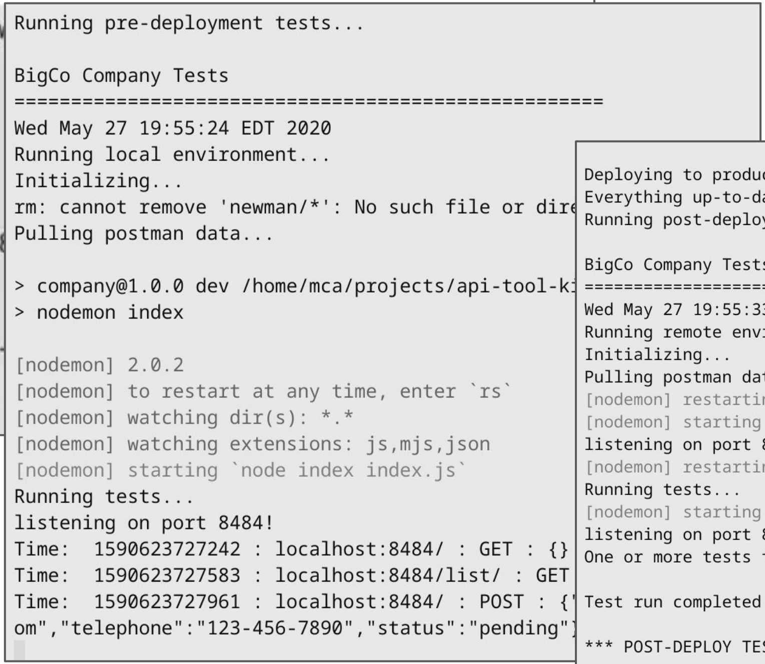
- Testing
- Security
- Deployment
  - Integration, delivery, deployment
  - Automation improves safety
  - Heroku via git push



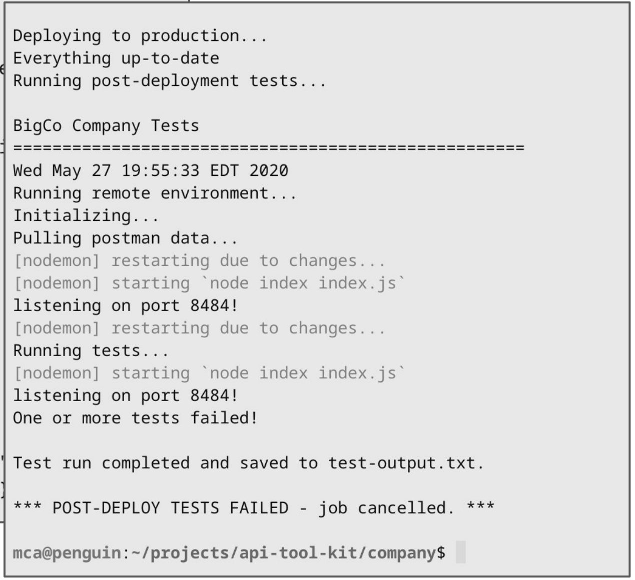

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen



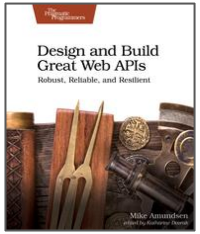
```
remote:
remote: -----> Build succeeded!
remote:        ├── ejs@2.3.3
remote:        └── express@4.13.3
remote:
remote:
remote: -----> Discovering process types
remote:        Procfile declares types -> w
remote:
remote: -----> Compressing...
remote:        Done: 10.2M
remote: -----> Launching...
remote:        Released v4
remote:        https://tranquil-hollows-728
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/tranquil-hollows-
   763c36e..f555ce0  master -> master
~/Code/tranquil-hollows-72848
```



```
Running pre-deployment tests...

BigCo Company Tests
=====================================================
Wed May 27 19:55:24 EDT 2020
Running local environment...
Initializing...
rm: cannot remove 'newman/*': No such file or dire
Pulling postman data...

> company@1.0.0 dev /home/mca/projects/api-tool-ki
> nodemon index

[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index index.js`
Running tests...
listening on port 8484!
Time:  1590623727242 : localhost:8484/ : GET : {}
Time:  1590623727583 : localhost:8484/list/ : GET
Time:  1590623727961 : localhost:8484/ : POST : {"
om","telephone":"123-456-7890","status":"pending"}
```

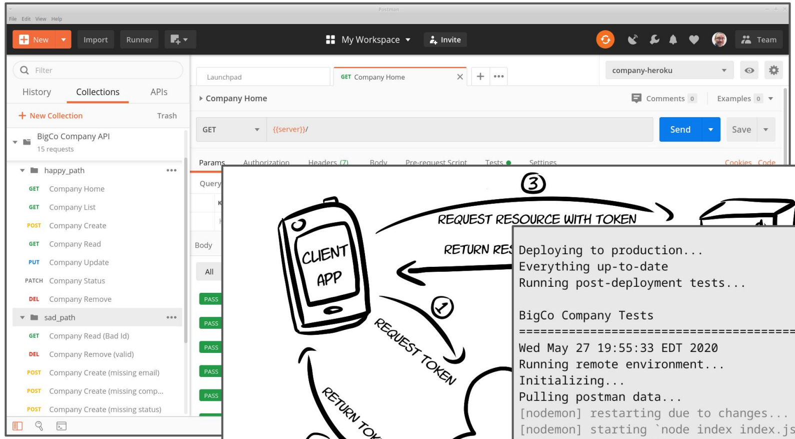


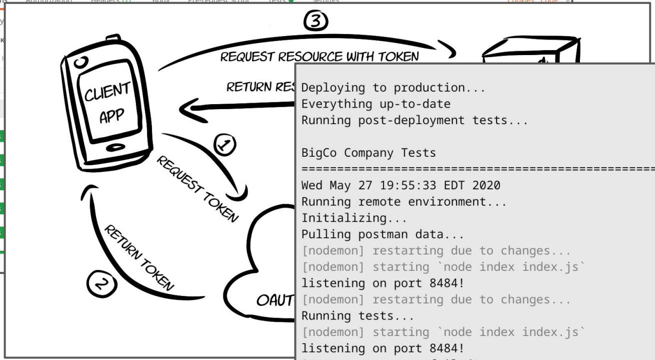
```
Deploying to production...
Everything up-to-date
Running post-deployment tests...

BigCo Company Tests
=====================================================
Wed May 27 19:55:33 EDT 2020
Running remote environment...
Initializing...
Pulling postman data...
[nodemon] restarting due to changes...
[nodemon] starting `node index index.js`
listening on port 8484!
[nodemon] restarting due to changes...
Running tests...
[nodemon] starting `node index index.js`
listening on port 8484!
One or more tests failed!

Test run completed and saved to test-output.txt.

*** POST-DEPLOY TESTS FAILED - job cancelled. ***

mca@penguin:~/projects/api-tool-kit/company$
```

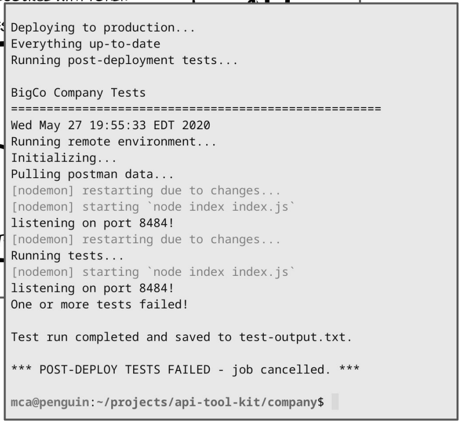

Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

# Releasing APIs

- Testing
  - Test interface and behavior
  - Postman/Newman
- Security
  - APIs use client_credentials w/ JWTs
  - Auth0 as a provider
- Deployment
  - Automate for safety
  - Heroku git push

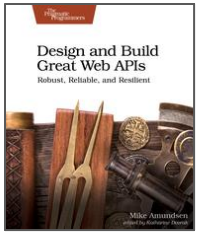




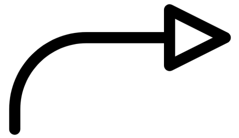

```
Deploying to production...
Everything up-to-date
Running post-deployment tests...

BigCo Company Tests
===================================================
Wed May 27 19:55:33 EDT 2020
Running remote environment...
Initializing...
Pulling postman data...
[nodemon] restarting due to changes...
[nodemon] starting `node index index.js`
listening on port 8484!
[nodemon] restarting due to changes...
Running tests...
[nodemon] starting `node index index.js`
listening on port 8484!
One or more tests failed!

Test run completed and saved to test-output.txt.

*** POST-DEPLOY TESTS FAILED - job cancelled. ***

mca@penguin:~/projects/api-tool-kit/company$
```

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

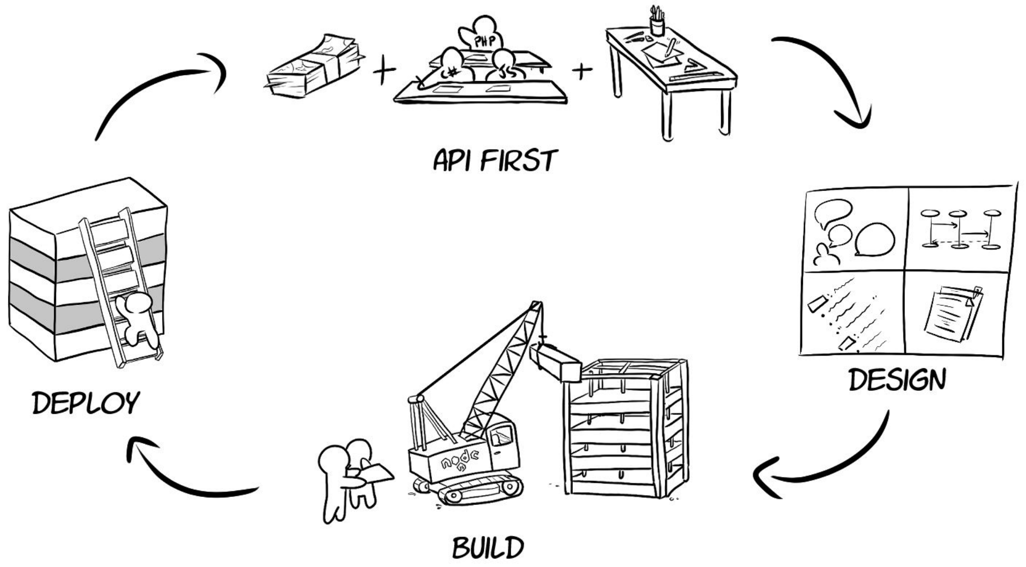
## And Then...


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

## Modifying APIs



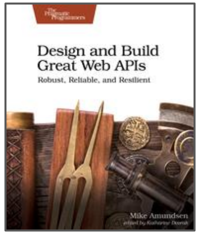
- Principles for Updating APIs
- Designing Updates
- Testing Updates
- Releasing Updates
- Shutting down APIs



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

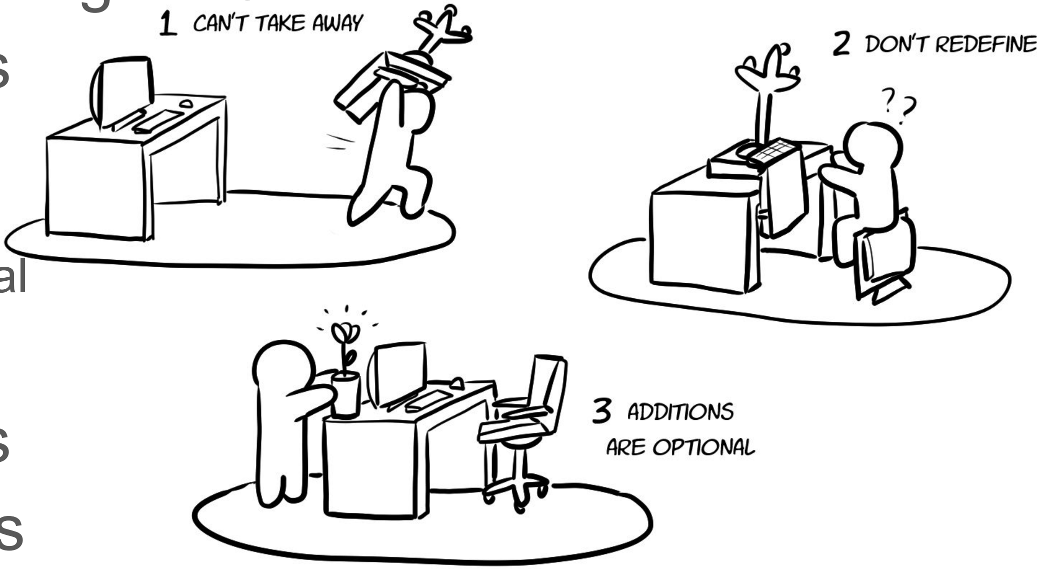
## Modifying APIs

- Principles for Updating APIs
  - First, do no harm
  - Fork your API
  - Know when to say "No"
- Designing Updates
- Testing Updates
- Releasing Updates
- Shutting down APIs


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

## Modifying APIs

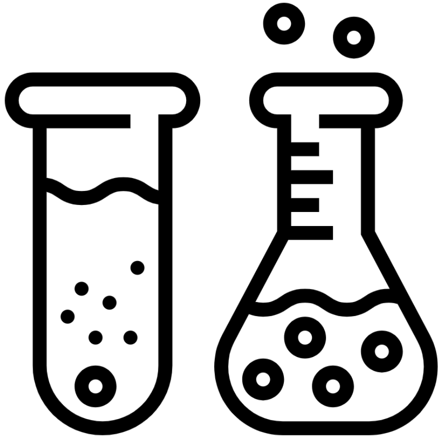
- Principles for Updating APIs
- Designing Updates
  - Take nothing away
  - Don't redefine
  - Additions are optional
- Testing Updates
- Releasing Updates
- Shutting down APIs

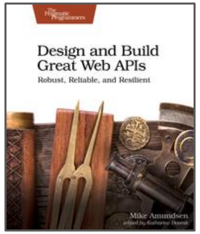


The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen
edited by Katharine Dvorak

# Modifying APIs

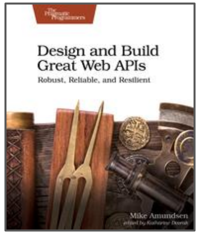
- Principles for Updating APIs
- Designing Updates
- Testing Updates
  - Use all existing tests
  - Add new tests for each release
- Releasing Updates
- Shutting down APIs




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

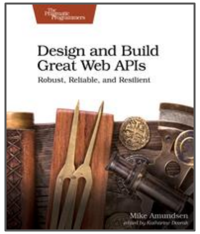
## Modifying APIs

- Principles for Updating APIs
- Designing Updates
- Testing Updates
- Releasing Updates
  - Reversibility/Re-entry First
  - Side-by-Side Releases
  - Overwriting Releases
- Shutting down APIs


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

# Modifying APIs

- Principles for Updating APIs
- Designing Updates
- Testing Updates
- Releasing Updates
- Shutting down APIs
  - Place Code in Public Domain
  - Open Source the Interface
  - Offer Recoverable Data
  - Mark the API 410 Gone (w/ a pointer)


The Pragmatic Programmers

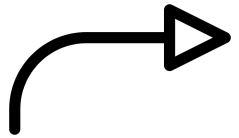
Design and Build Great Web APIs

Robust, Reliable, and Resilient

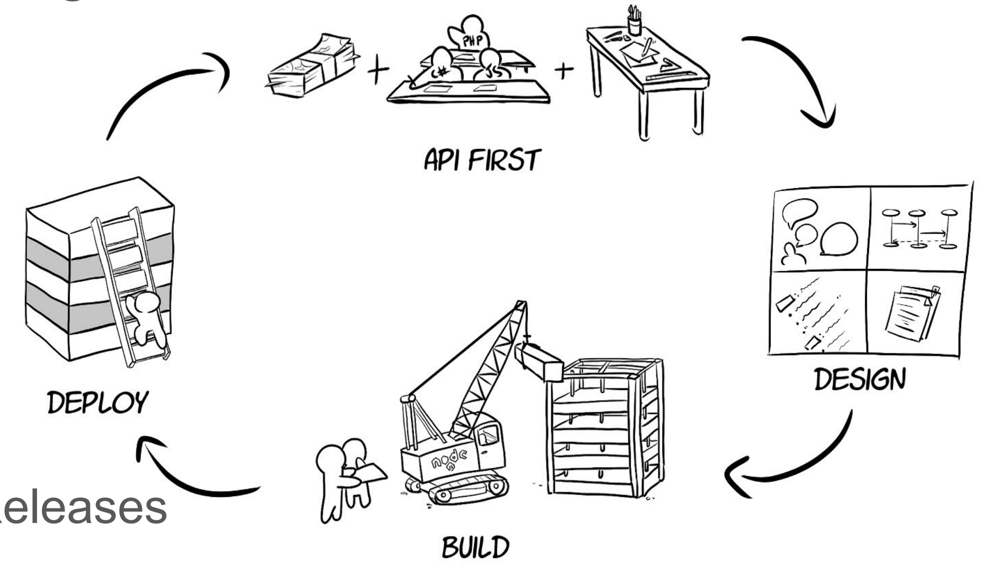
Mike Amundsen

edited by Katharine Dvorak

# Modifying APIs

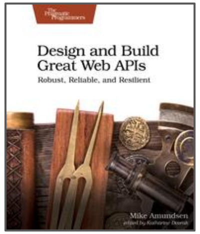


- Principles for Updating APIs
  - First, Do no harm
- Designing Updates
  - No breaking changes
- Testing Updates
  - Use all the old tests
- Releasing Updates
  - Favor Side-by-Side Releases
- Shutting down APIs
  - Responsible shutdowns

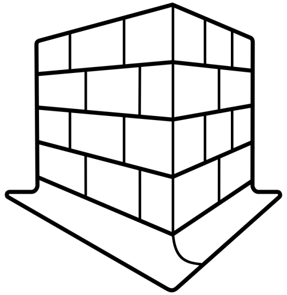


Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

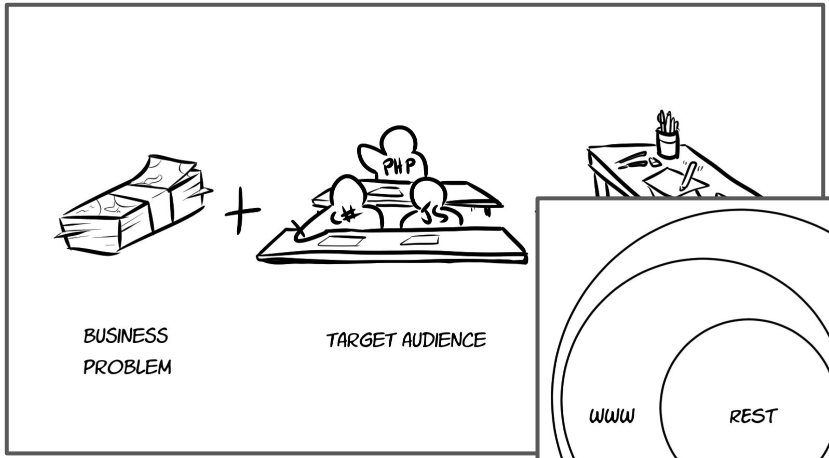
*So….*

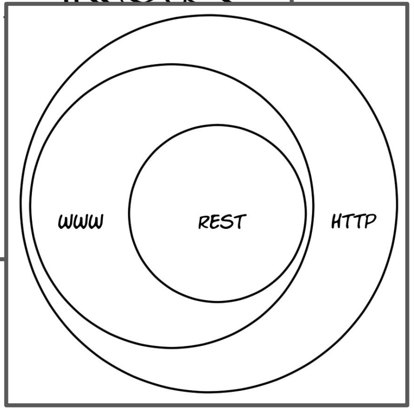

The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

## Foundations



- API First
  - Solving business problems for people
- HTTP, Web, & REST
  - Protocol, practice, style

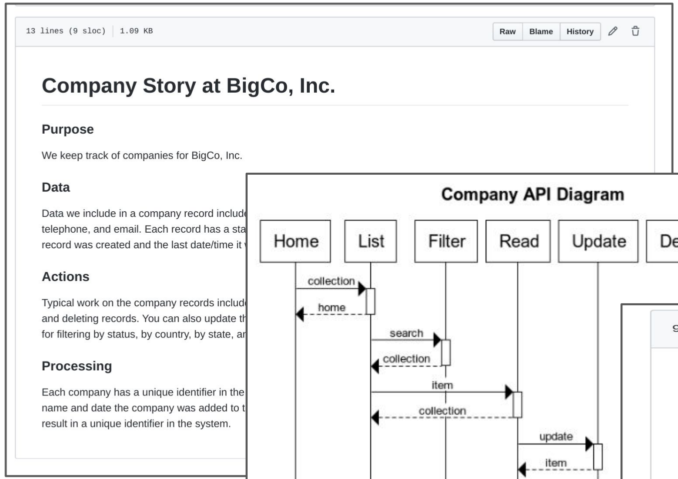




The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

# Designing APIs

- Model
  - API Story
- Design
  - API Diagram
- Describe
  - API Description



13 lines (9 sloc) | 1.09 KB

Raw Blame History

**Company Story at BigCo, Inc.**

**Purpose**

We keep track of companies for BigCo, Inc.

**Data**

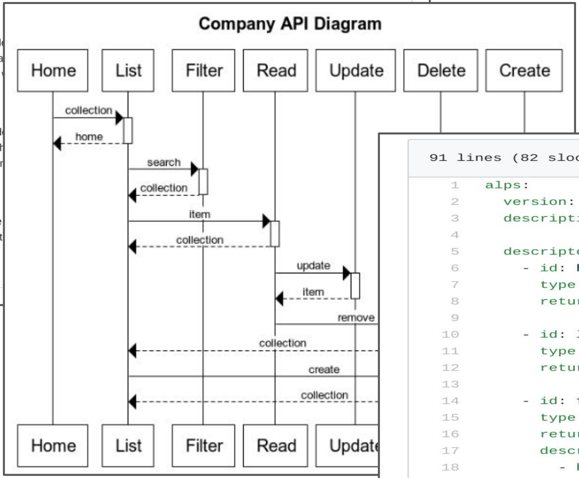
Data we include in a company record includ
telephone, and email. Each record has a sta
record was created and the last date/time it

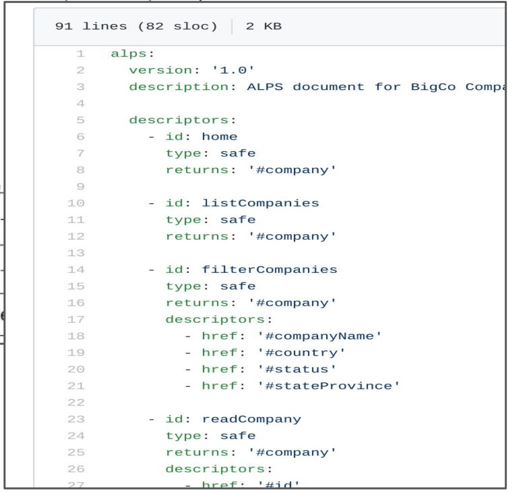
**Actions**

Typical work on the company records includ
and deleting records. You can also update th
for filtering by status, by country, by state, ar

**Processing**

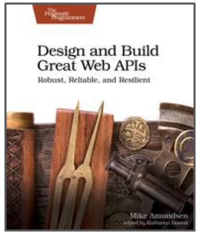
Each company has a unique identifier in the
name and date the company was added to t
result in a unique identifier in the system.





91 lines (82 sloc) | 2 KB

```
alps:
  version: '1.0'
  description: ALPS document for BigCo Compa

  descriptors:
    - id: home
      type: safe
      returns: '#company'

    - id: listCompanies
      type: safe
      returns: '#company'

    - id: filterCompanies
      type: safe
      returns: '#company'
      descriptors:
        - href: '#companyName'
        - href: '#country'
        - href: '#status'
        - href: '#stateProvince'

    - id: readCompany
      type: safe
      returns: '#company'
      descriptors:
        - href: '#id'
```



The Pragmatic Programmers

Design and Build Great Web APIs

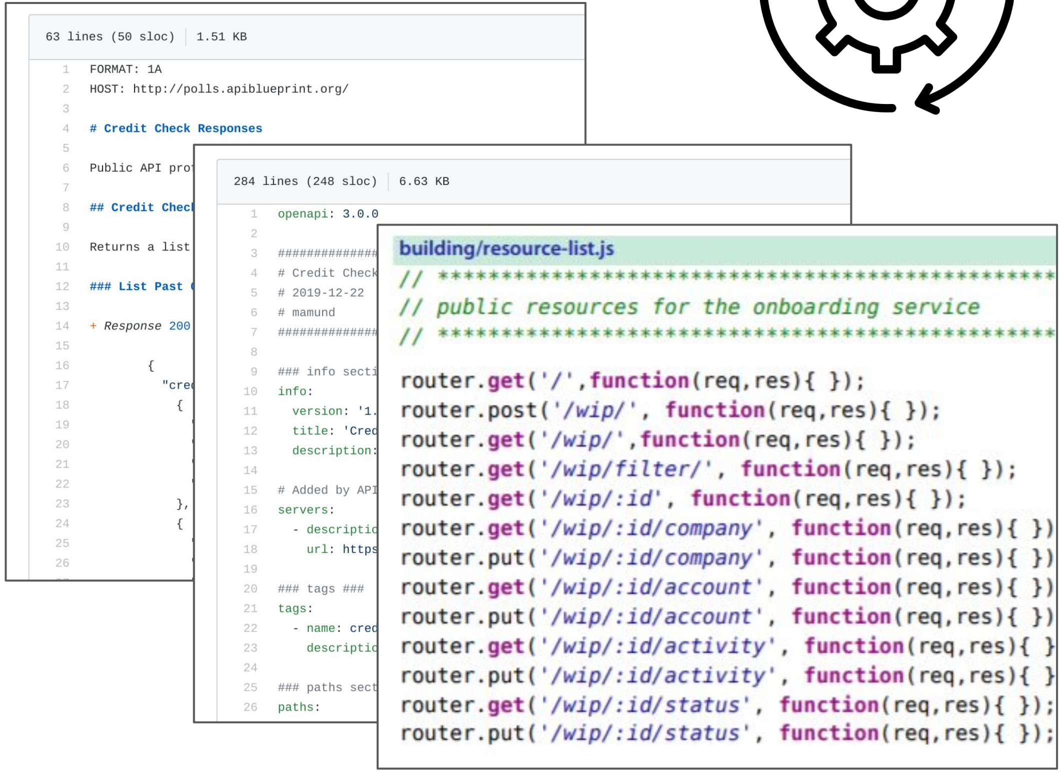
Robust, Reliable, and Resilient

Mike Amundsen

edited by Katharine Dvorak

# Building APIs

- Sketching
  - Explore, throw away
  - APIB
- Prototyping
  - Details, test
  - OpenAPI
- Building
  - Translate, repeat
  - NodeJS/DARRT



```
63 lines (50 sloc) | 1.51 KB

FORMAT: 1A
HOST: http://polls.apiblueprint.org/

# Credit Check Responses

Public API pro

## Credit Checl

Returns a list

### List Past (

+ Response 200

        {
          "cre
            {



          },
            {
```

```
284 lines (248 sloc) | 6.63 KB

openapi: 3.0.0

##############
# Credit Check
# 2019-12-22
# mamund
##############

### info secti
info:
  version: '1.
  title: 'Cred
  description:

# Added by API
servers:
  - descriptio
    url: https

### tags ###
tags:
  - name: cred
    descriptio

### paths sect
paths:
```

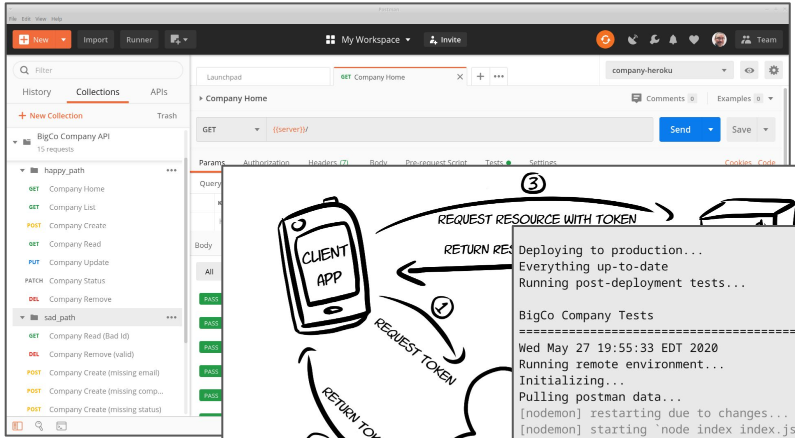
building/resource-list.js

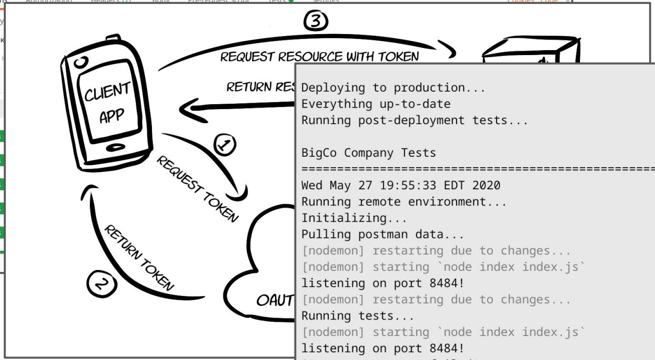
```
// ********************************************
// public resources for the onboarding service
// ********************************************

router.get('/',function(req,res){ });
router.post('/wip/', function(req,res){ });
router.get('/wip/',function(req,res){ });
router.get('/wip/filter/', function(req,res){ });
router.get('/wip/:id', function(req,res){ });
router.get('/wip/:id/company', function(req,res){ })
router.put('/wip/:id/company', function(req,res){ })
router.get('/wip/:id/account', function(req,res){ })
router.put('/wip/:id/account', function(req,res){ })
router.get('/wip/:id/activity', function(req,res){ }
router.put('/wip/:id/activity', function(req,res){ }
router.get('/wip/:id/status', function(req,res){ });
router.put('/wip/:id/status', function(req,res){ });
```

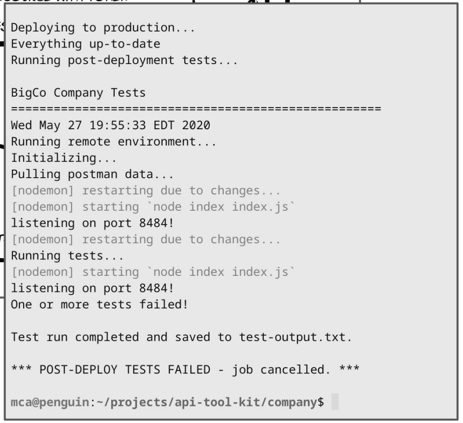
The Pragmatic Programmers
Design and Build Great Web APIs
Robust, Reliable, and Resilient
Mike Amundsen

# Releasing APIs

- Testing
  - Test interface and behavior
  - Postman/Newman
- Security
  - APIs use client_credentials w/ JWTs
  - Auth0 as a provider
- Deployment
  - Automate for safety
  - Heroku git push





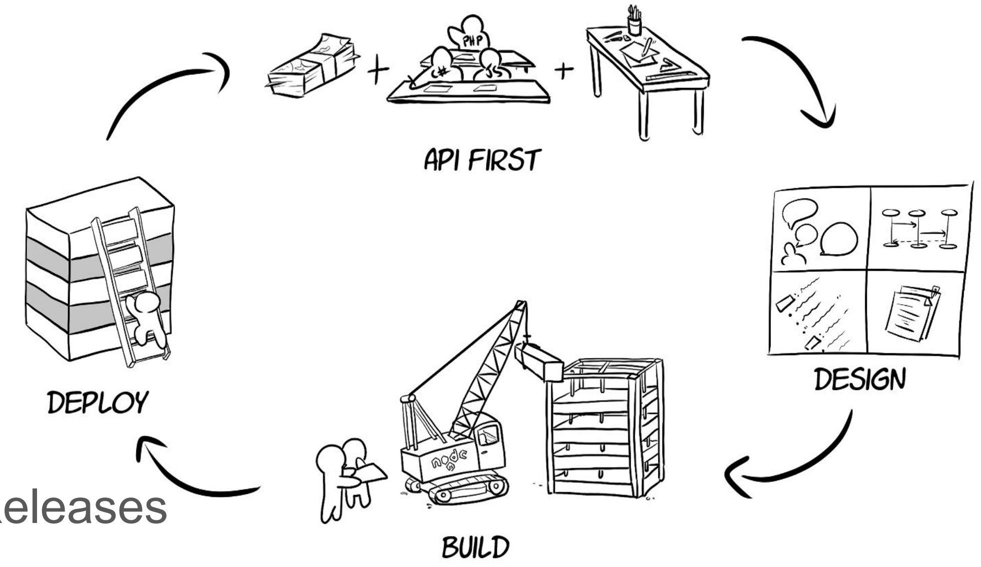


```
Deploying to production...
Everything up-to-date
Running post-deployment tests...

BigCo Company Tests
==================================================
Wed May 27 19:55:33 EDT 2020
Running remote environment...
Initializing...
Pulling postman data...
[nodemon] restarting due to changes...
[nodemon] starting `node index index.js`
listening on port 8484!
[nodemon] restarting due to changes...
Running tests...
[nodemon] starting `node index index.js`
listening on port 8484!
One or more tests failed!

Test run completed and saved to test-output.txt.

*** POST-DEPLOY TESTS FAILED - job cancelled. ***

mca@penguin:~/projects/api-tool-kit/company$
```

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

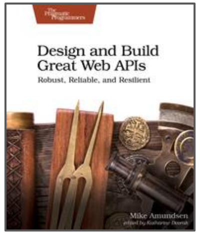
## Modifying APIs

- Update Principles
  - First, Do no harm
- Design Updates
  - No breaking changes
- Test Updates
  - Use all the old tests
- Release Updates
  - Favor Side-by-Side Releases
- API Shutdown
  - Responsible shutdowns



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

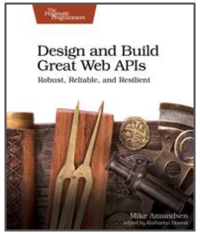
*And, don't forget ...*


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

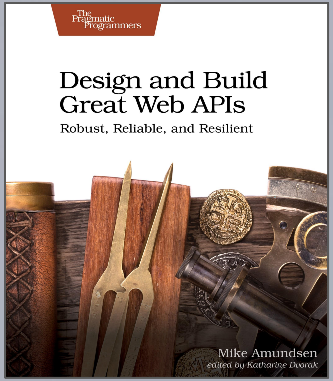
## What is API-First?

*"API-first design means **identifying** and/or defining key actors and personas, **determining** what those actors and personas expect to be able **to do** with **APIs**"*

*-- Kas Thomas, 2009*


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

# Design and Build Great APIs



The Pragmatic Programmers

Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen

*edited by Katharine Dvorak*

**g.mamund.com/2020-07-apidays**

**@mamund**

**Mike Amundsen**

**youtube.com/mamund**