

Workshop:

API-first Redesign of a Legacy Application

Chris Busse, *APIvista*

Laptops out & online, please!

Hello!

Chris Busse

CTO, APIvista

...work with a wide variety of tech stacks and legacy apps (and organizations) undergoing “digital transformation”...



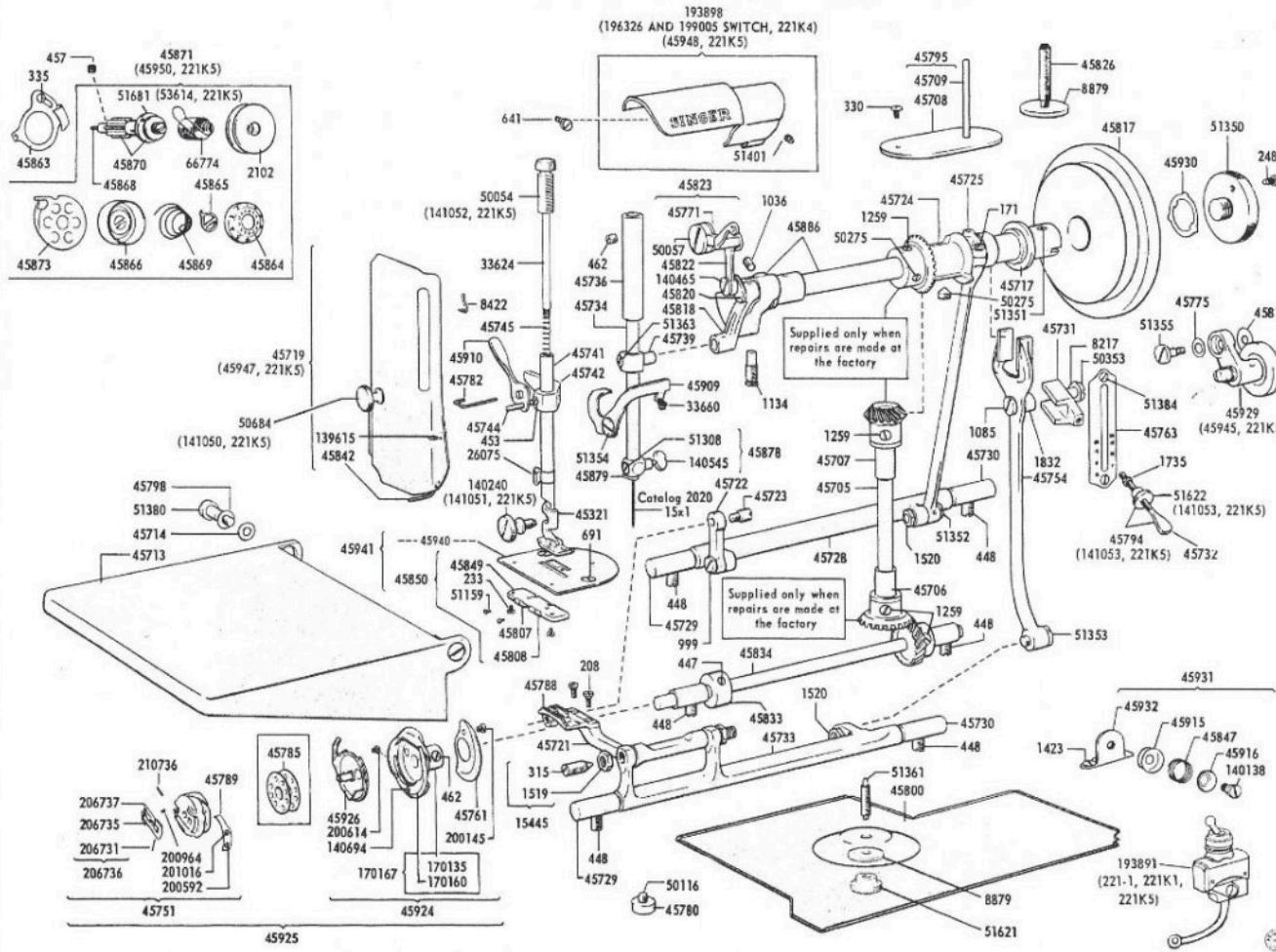
API-first Redesign of a Legacy App
Chris Busse | @busse

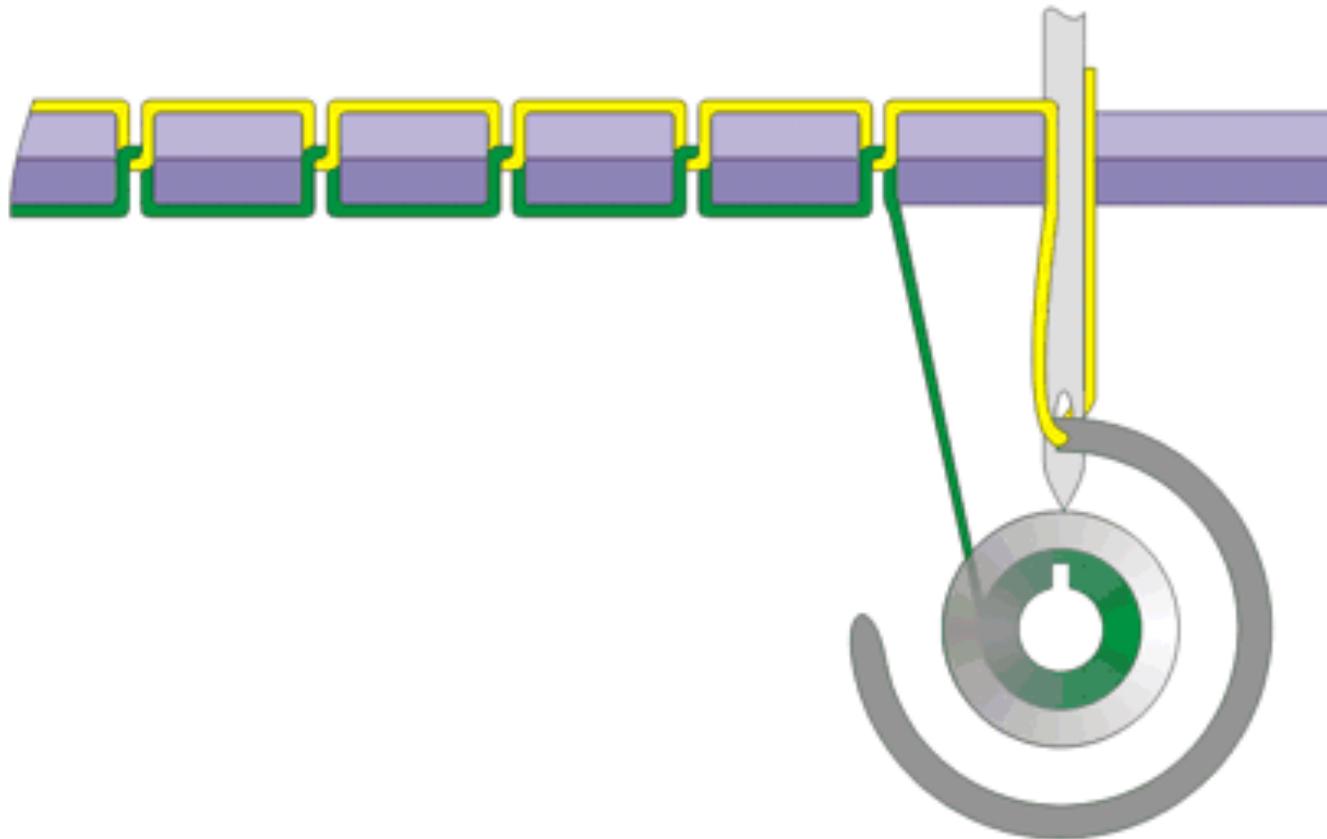






List of Parts-Machines Nos. 221-1, 221K1, 221K4 and 221K5





What is a REST API?



API-first Redesign of a Legacy App
Chris Busse | @busse

API-first Redesign of a Legacy Application

“Resource manipulation through representations”

- REST is Stateless – Flow Control needs to be handled by the client
- No business logic in the client that would enable game cheats (this has real-world implications too)
- Keep the User Experience in mind
- Keep the Developer Experience in mind
- Think like Spock 

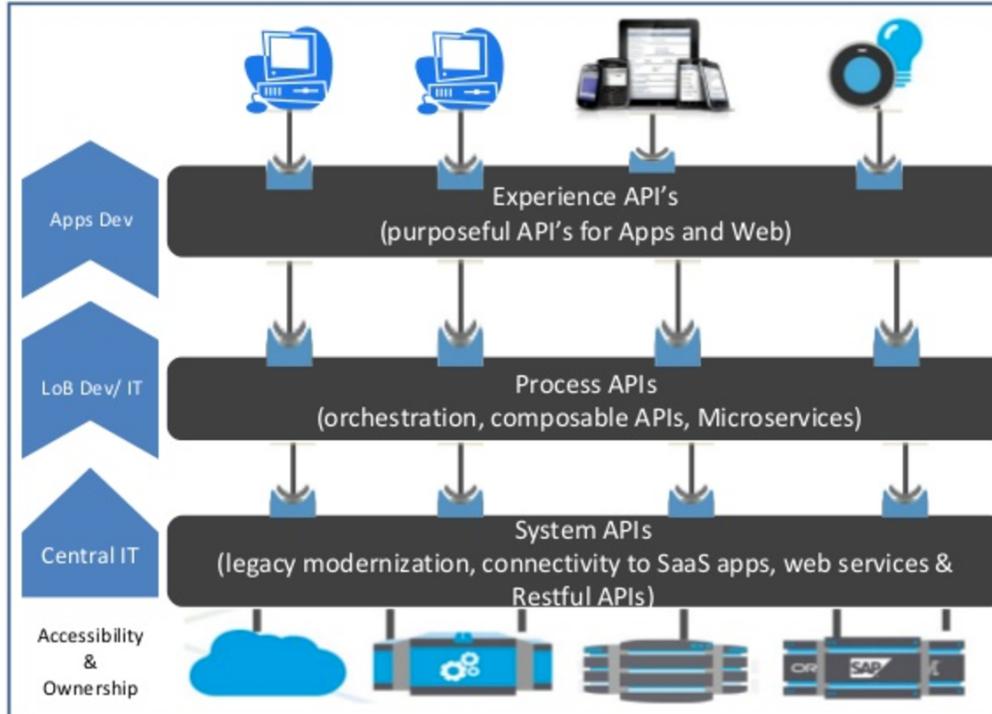


Abstractions



API-first Redesign of a Legacy App
Chris Busse | @busse

Three Tier API Layer Architecture



(the Mulesoft way)

Source: <https://www.slideshare.net/HarishKumar544/three-layer-api-design-architecture>



API-first Redesign of a Legacy App
Chris Busse | @busse

YAGNI

(“You aren’t going to need it”)



API-first Redesign of a Legacy App
Chris Busse | @busse

YAGNIN

(“You aren’t going to need it... now...”)



API-first Redesign of a Legacy App
Chris Busse | @busse

Conway's Law

“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.” — *Mel Conway*

“Organizations often produce web sites with a content and structure which mirrors the internal concerns of the organization rather than the needs of the users of the site.” – *Nigel Bevin*



Conway's Law, Example

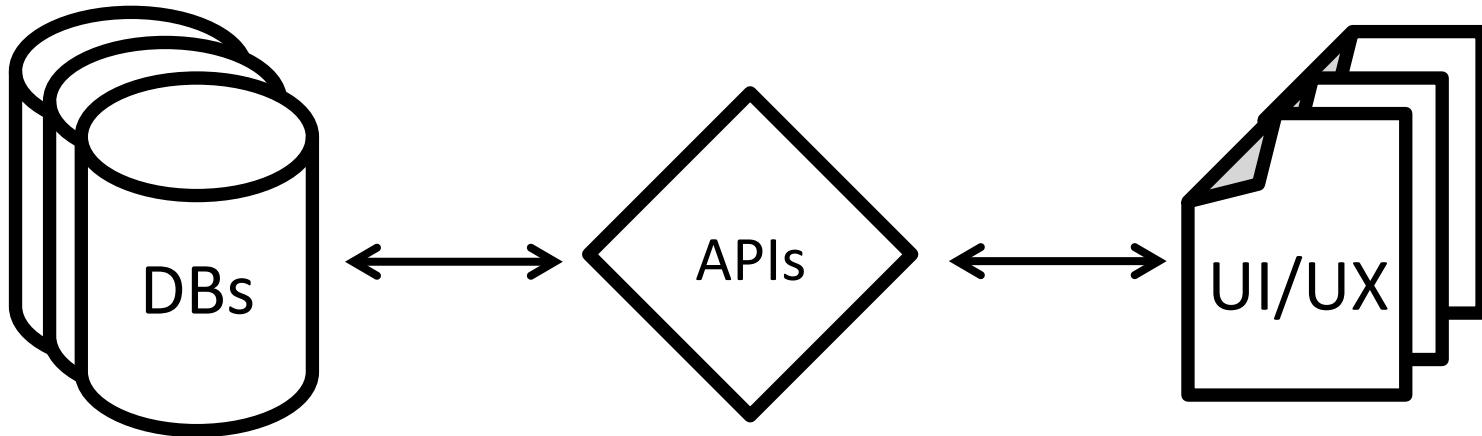
A large bank conglomerate may think of relating to people as its Lines of Business, ex: Consumer Checking & Savings, Auto Loan, Home Loan, Credit Card.

The bank's back-end systems may be designed to be separate between each line of business (sometimes for technical reasons, sometimes for regulatory reasons).

However, a customer of the bank expects to see the bank as one entity where they use several products, and doesn't want to concern themselves with the various "departments".

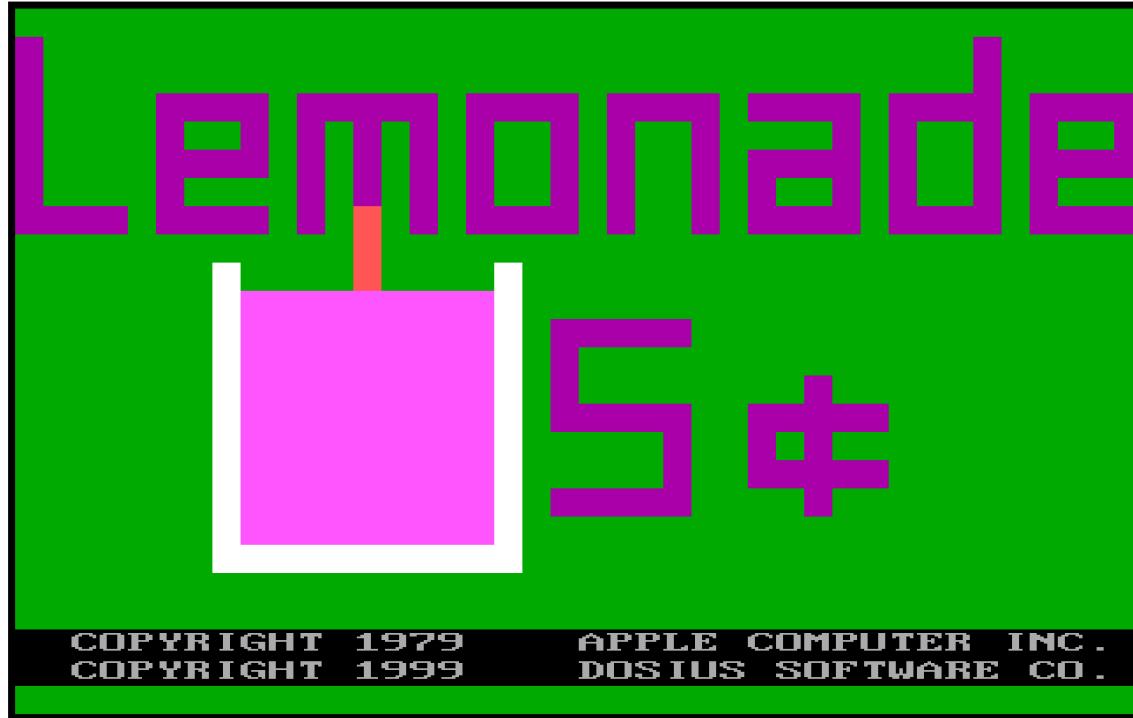


DB first or UI/UX first?



Which should the API give precedence to in its design?

Lemonade Stand



API-first Redesign of a Legacy App
Chris Busse | @busse

(Switch to live Lemonade Stand demo)



API-first Redesign of a Legacy App
Chris Busse | @busse

Your turn!

<http://busse.io>

then click “Lemonade Stand Links” at the top
(permalink: <http://busse.io/Lemonade-Stand-Workshop/>)

pick in-browser or download Mac or PC

After 10 Turns, write your team name & ending \$ amount on a Post-It



What are the “rules of the game”?

(aka the “business logic”)



API-first Redesign of a Legacy App
Chris Busse | @busse

Rules given to user...

TO MANAGE YOUR LEMONADE STAND, YOU WILL NEED TO MAKE THESE DECISIONS EVERY DAY:

1. HOW MANY GLASSES OF LEMONADE TO MAKE (ONLY ONE BATCH IS MADE EACH MORNING)
2. HOW MANY ADVERTISING SIGNS TO MAKE (THE SIGNS COST FIFTEEN CENTS EACH)
3. WHAT PRICE TO CHARGE FOR EACH GLASS

YOU WILL BEGIN WITH \$2.00 CASH (ASSETS). BECAUSE YOUR MOTHER GAVE YOU SOME SUGAR, YOUR COST TO MAKE LEMONADE IS TWO CENTS A GLASS (THIS MAY CHANGE IN THE FUTURE).

YOUR EXPENSES ARE THE SUM OF THE COST OF THE LEMONADE AND THE COST OF THE SIGNS.

YOUR PROFITS ARE THE DIFFERENCE BETWEEN THE INCOME FROM SALES AND YOUR EXPENSES.

THE NUMBER OF GLASSES YOU SELL EACH DAY DEPENDS ON THE PRICE YOU CHARGE, AND ON THE NUMBER OF ADVERTISING SIGNS YOU USE.

KEEP TRACK OF YOUR ASSETS, BECAUSE YOU CAN'T SPEND MORE MONEY THAN YOU HAVE!

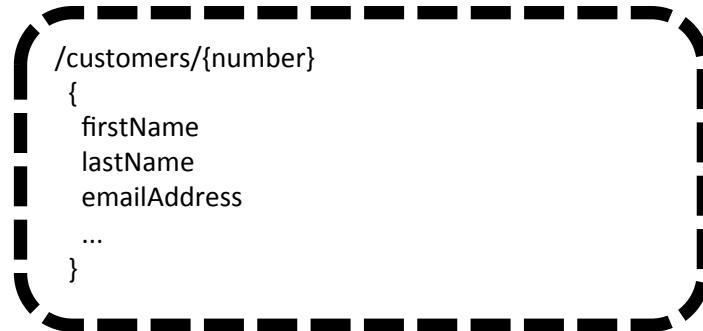
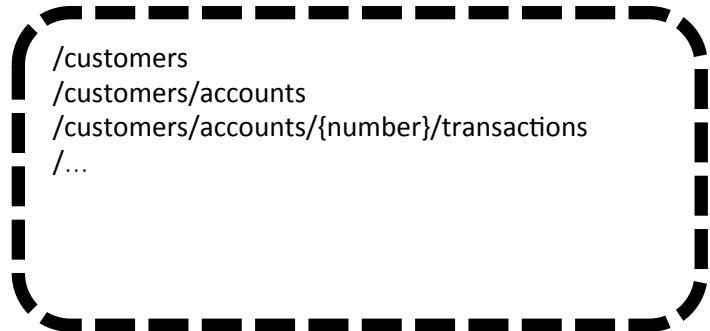


Do we have enough information to redesign this legacy app API-first?



(Re-)Design

- **Resources**
- URL / URI
- Ontology
- **Payloads**
- Request / Responses
- Taxonomies



<http://lemonade-stand/???>



/customers
/customers/accounts
/customers/accounts/{number}/transactions
/...



Swagger Editor File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ Switch back to previous editor

```
1 swagger: '2.0'
2 info:
3   title: Lemonade Stand API
4   description: >
5     This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in real life, is a bit messy.
6   version: 0.0.1
7 host: lemonade-stand.local
8 schemes:
9   - http
10 basePath: /v1
11 produces:
12   - application/json
13 paths:
14   '/lemonade':
15     get:
16       tags:
17         - supplies
18       summary: Placeholder for Lemonade Top level
19       operationId: getLemonade
20       responses:
21         '200':
22           description: OK
23           schema:
24             type: array
25             items:
26               $ref: '#/definitions/Lemonade'
27
28   '/signs':
29     get:
30       tags:
31         - supplies
32       summary: Placeholder for Signs Top level
33       operationId: getSigns
34       responses:
35         '200':
36           description: OK
37           schema:
38             type: array
39             items:
40               $ref: '#/definitions/Signs'
41
42
43
44
45
46
47
```

Lemonade Stand API 0.0.1

[Base URL: lemonade-stand.local/v1]

This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in real life, is a bit messy.

Schemes

HTTP

supplies ▼

GET /lemonade Placeholder for Lemonade Top level

GET /signs Placeholder for Signs Top level

Models

Lemonade ↵ { ↵ quantity integer }

Signs ↵ { ↵ quantity integer }



API-first Redesign of a Legacy App
Chris Busse | @busse

Define Payloads

10 minutes

```
/customers/{number}  
{  
    firstName  
    lastName  
    emailAddress  
    ...  
}
```



Swagger Editor File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ Switch back to previous editor

```
1 swagger: '2.0'
2 info:
3   title: Lemonade Stand API
4   description: >
5     This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in
       real life, is a bit messy.
6   version: 0.0.1
7 host: lemonade-stand.local
8 schemes:
9   - http
10 basePath: /v1
11 produces:
12   - application/json
13 paths:
14   '/lemonade':
15     get:
16       tags:
17         - supplies
18       summary: Placeholder for Lemonade Top level
19       operationId: getLemonade
20       responses:
21         '200':
22           description: OK
23           schema:
24             type: array
25             items:
26               $ref: '#/definitions/Lemonade'
27   '/lemonade/{turn}':
28     post:
29       tags:
30         - supplies
31       summary: Set how many cups of lemonade to make that turn
32       operationId: addLemonade
33       parameters:
34         - name: turn
35           in: path
36           description: The turn to set the quantity for
37           required: true
38           type: string
39         - name: quantity
40           description: How many cups to make
41           in: body
42           required: true
43           schema:
44             $ref: '#/definitions/Lemonade'
45       responses:
46         '201':
47           description: Quantity set
```

Lemonade Stand API 0.0.1

[Base URL: lemonade-stand.local/v1]

This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in real life, is a bit messy.

Schemes

HTTP

supplies ▼

GET /lemonade Placeholder for Lemonade Top level

POST /lemonade/{turn} Set how many cups of lemonade to make that turn

GET /signs Placeholder for Signs Top level

POST /signs/{turn} Set how many cups of lemonade to make that turn

Models

Lemonade ↳ { ↲

quantity	integer
----------	---------



API-first Redesign of a Legacy App
Chris Busse | @busse

Swagger Editor File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ Switch back to previous editor

```
1 swagger: '2.0'
2 info:
3   title: Lemonade Stand API
4   description: >
5     This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in
       real life, is a bit messy.
6   version: 0.0.1
7 host: lemonade-stand.local
8 schemes:
9   - http
10 basePath: /v1
11 produces:
12   - application/json
13 paths:
14   '/turn/{turn}':
15     get:
16       tags:
17         - supplies
18       summary: Placeholder for Turn top level
19       operationId: getTurn
20       responses:
21         '200':
22           description: OK
23           schema:
24             type: array
25             items:
26               $ref: '#/definitions/Turn'
27
28     post:
29       tags:
30         - supplies
31       summary: Placeholder for Turn top level
32       operationId: playTrun
33       parameters:
34         - name: turn
35         in: path
36         description: The turn to set the quantity for
37         required: true
38         type: string
39         - name: choices
40         description: The choices the player made this turn
41         in: body
42         required: true
43         schema:
44           $ref: '#/definitions/Turn'
45       responses:
46         '201':
47           description: Turn is ready
```

Lemonade Stand API v0.0.1

[Base URL: lemonade-stand.local/v1]

This is an API for the Lemonade Stand game. It is used in a workshop and like making lemonade in real life, is a bit messy.

Schemes

HTTP

supplies ▼

GET /turn/{turn} Placeholder for Turn top level

POST /turn/{turn} Placeholder for Turn top level

Models

Turn ↳ { ↲

```
  lemonade      integer
  signs         integer
}
```



API-first Redesign of a Legacy App
Chris Busse | @busse

What else?

- Deviation from the “happy path”
- Exception handling: *What should be in the client, what would be in the API?*
- Great time to embrace defensive design



```
890 PRINT "HOW MANY GLASSES OF LEMONADE DO YOU"
895 PRINT "WISH TO MAKE ";
900 INPUT L(I)
901 IF L(I) < 0 OR L(I) > 1000 THEN 903
902 GOTO 906
903 PRINT "COME ON, LET'S BE REASONABLE NOW!!!"
904 PRINT "TRY AGAIN"
905 GOTO 890
906 IF L(I) < > INT (L(I)) THEN 903
910 IF L(I) * C1 < = A(I) THEN 950
920 PRINT "THINK AGAIN!!! YOU HAVE ONLY ";STI$
930 PRINT "IN CASH AND TO MAKE ";L(I);" GLASSES OF"
932 PRINT "LEMONADE YOU NEED $";L(I) * C1;" IN CASH."
940 GOTO 890
950 PRINT
951 PRINT "HOW MANY ADVERTISING SIGNS (";S3 * 100;" CENTS"
952 PRINT "EACH) DO YOU WANT TO MAKE ";
960 INPUT S(I) 961 IF S(I) < 0 OR S(I) > 50 THEN 963
962 GOTO 965
963 PRINT "COME ON, BE REASONABLE!!! TRY AGAIN."
964 GOTO 950
965 IF S(I) < > INT (S(I)) THEN 963
970 IF S(I) * S3 < = A(I) - L(I) * C1 THEN 1010
975 PRINT 980 STI = A(I) - L(I) * C1: GOSUB 4000
985 PRINT "THINK AGAIN, YOU HAVE ONLY ";STI$
990 PRINT "IN CASH LEFT AFTER MAKING YOUR LEMONADE."
```

1

2

3

4



Add a cookie!

(...not the browser kind)

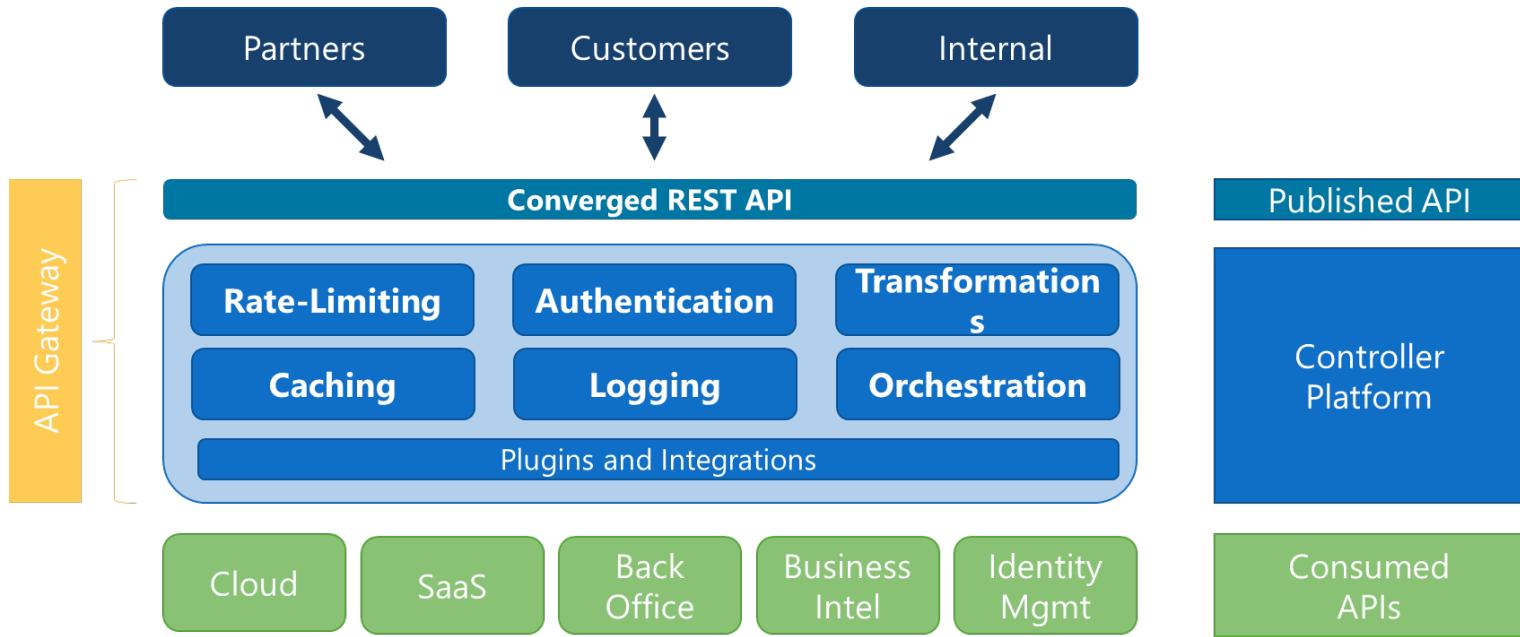
- You are going to sell cookies in the game now – what changes in your API design?
- Which approach (items as resources, or turns) is easier to handle this kind of change?



What about the “external” factors?

- Weather
- Price increase
- Street Crews closing the street





Q & A & Discussion



API-first Redesign of a Legacy App
Chris Busse | @busse

Links



Lemonade Stand (DOS Emulator)

<https://classicreload.com/lemonade-stand.html>



Lemonade Stand (Original Source Code)

6502: <https://github.com/jefftranter/6502/blob/master/basic/LemonadeStand/lemonade.bas>

REALBasic: <http://www.codenautics.com/lemonade/>

Thank you!

Chris Busse

chris.busse@apivista.com
<http://apivista.com>

We're Hiring!
API & DevOps
Engineers

@busse

<http://busse.io>



API-first Redesign of a Legacy App
Chris Busse | @busse

```
10 REM     <<< LEMONADE STAND >>>
15 REM
20 REM     FROM AN ORIGINAL PROGRAM
30 REM     BY BOB JAMISON, OF THE
40 REM     MINNESOTA EDUCATIONAL
50 REM     COMPUTING CONSORTIUM
60 REM             * * *
70 REM     MODIFIED FOR THE APPLE
80 REM     FEBRUARY, 1979
90 REM     BY CHARLIE KELLNE
```



Special thanks to Brandee Shin for helping me develop an early prototype of this workshop at REST Fest 2016