goto;

GOTO
CHICAGO 2023

#GOTOchgo

Programming the Web with HyperLANG and HyperCLI

Mike Amundsen
@mamund

The latest blockbuster movie from Hollywood – This year's E.T.

DAILY EXPRESS

Is it a game, or is it real?

WARGAMES PG

A Leonard Goldberg Production A John Badham Film "WARGAMES" MATTHEW BRODERICK DABNEY COLEMAN JOHN WOOD ALLY SHEEDY Written by LAWRENCE LASKER & WALTER F. PARKES Director of Photography WILLIAM A. FRAKER A.S.C.

Music by ARTHUR B. RUBINSTEIN Executive Producer LEONARD GOLDBERG Produced by HAROLD SCHNEIDER Directed by JOHN BADHAM Produced in association with Sherwood Productions Panavision Metrocolor DOLBY STEREO

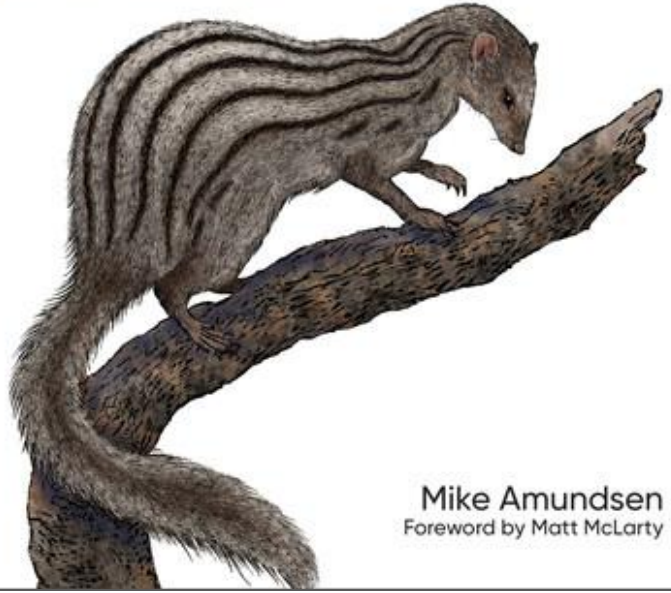
Distributed by UIP Read the Penguin paperback


Mike Amundsen
@mamund

O'REILLY

RESTful Web API Patterns & Practices Cookbook

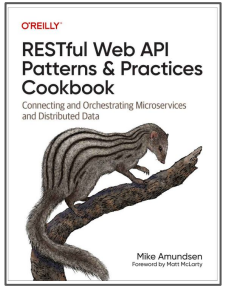
Connecting and Orchestrating Microservices and Distributed Data



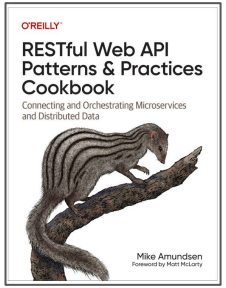
Mike Amundsen
Foreword by Matt McLarty

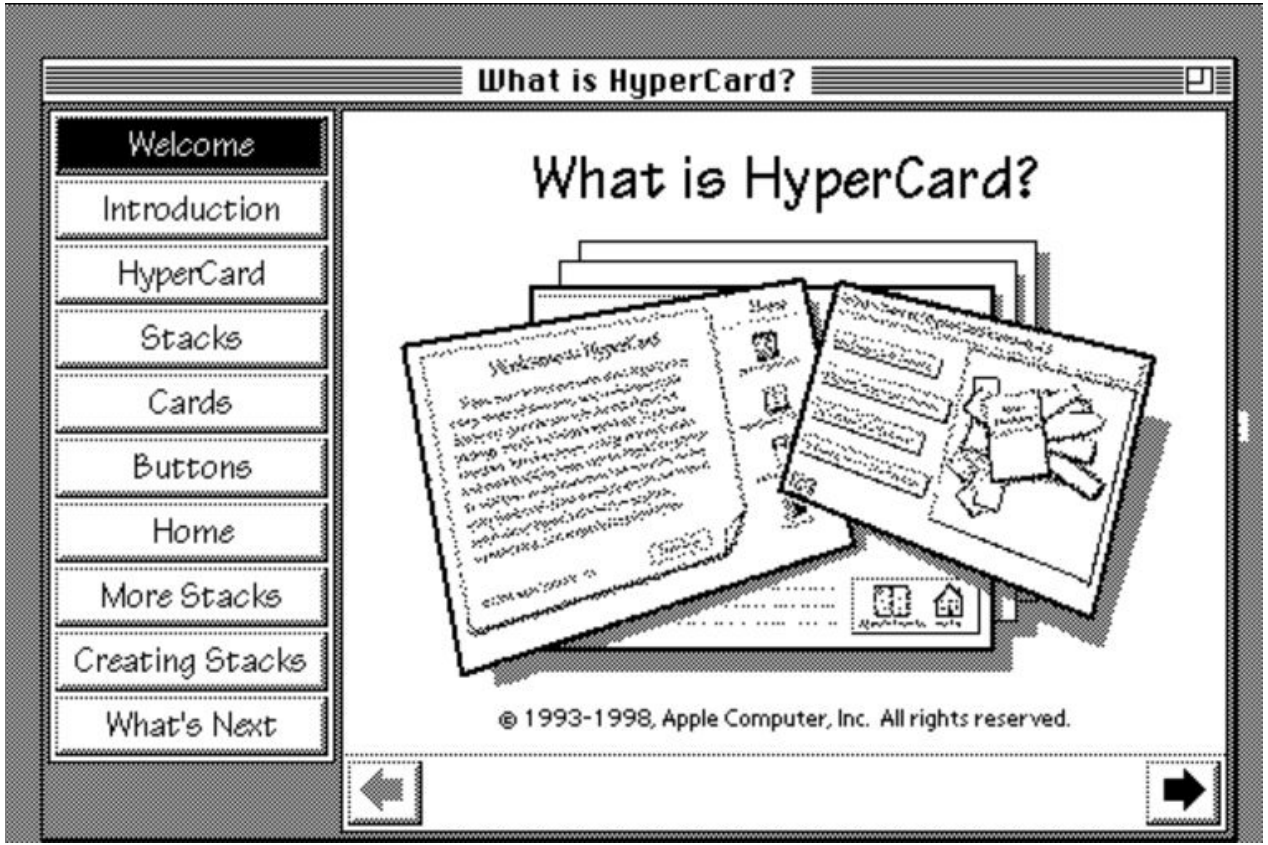
Programming the Web …

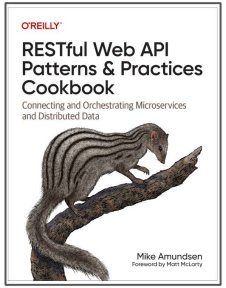
- Welcome to HyperWORLD
- HyperLANG Basics
- Doing More with HyperCLI
- The Future of HyperLANG



Welcome to HyperWORLD

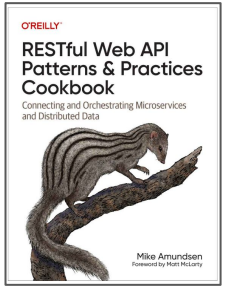



What is HyperCard?
Welcome
Introduction
HyperCard
Stacks
Cards
Buttons
Home
More Stacks
Creating Stacks
What's Next
What is HyperCard?
© 1993-1998, Apple Computer, Inc. All rights reserved.

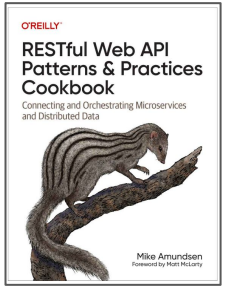

O'REILLY
RESTful Web API Patterns & Practices Cookbook
Connecting and Orchestrating Microservices and Distributed Data
Mike Amundsen
Foreword by Matt McLarty

Welcome to HyperWORLD

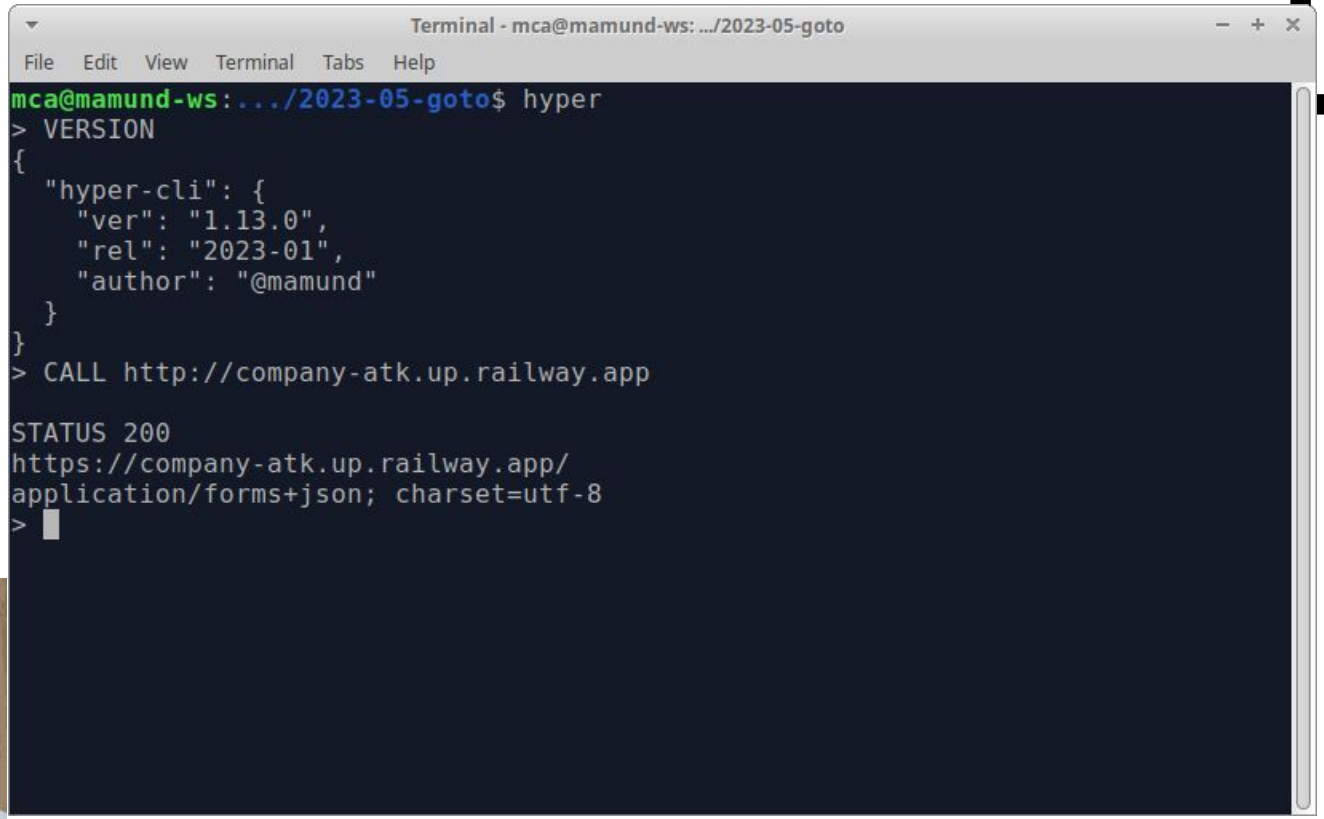
- HyperCLI is a REPL
  - Like CURL but even more
- HyperLANG is a DSL
  - Like programming but without the pain

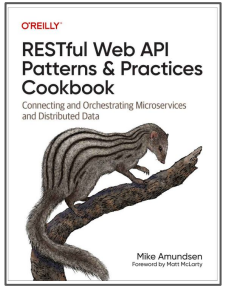


HyperCLI is a REPL



Welcome : HyperCLI is a REPL

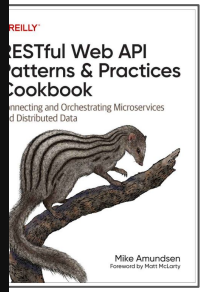




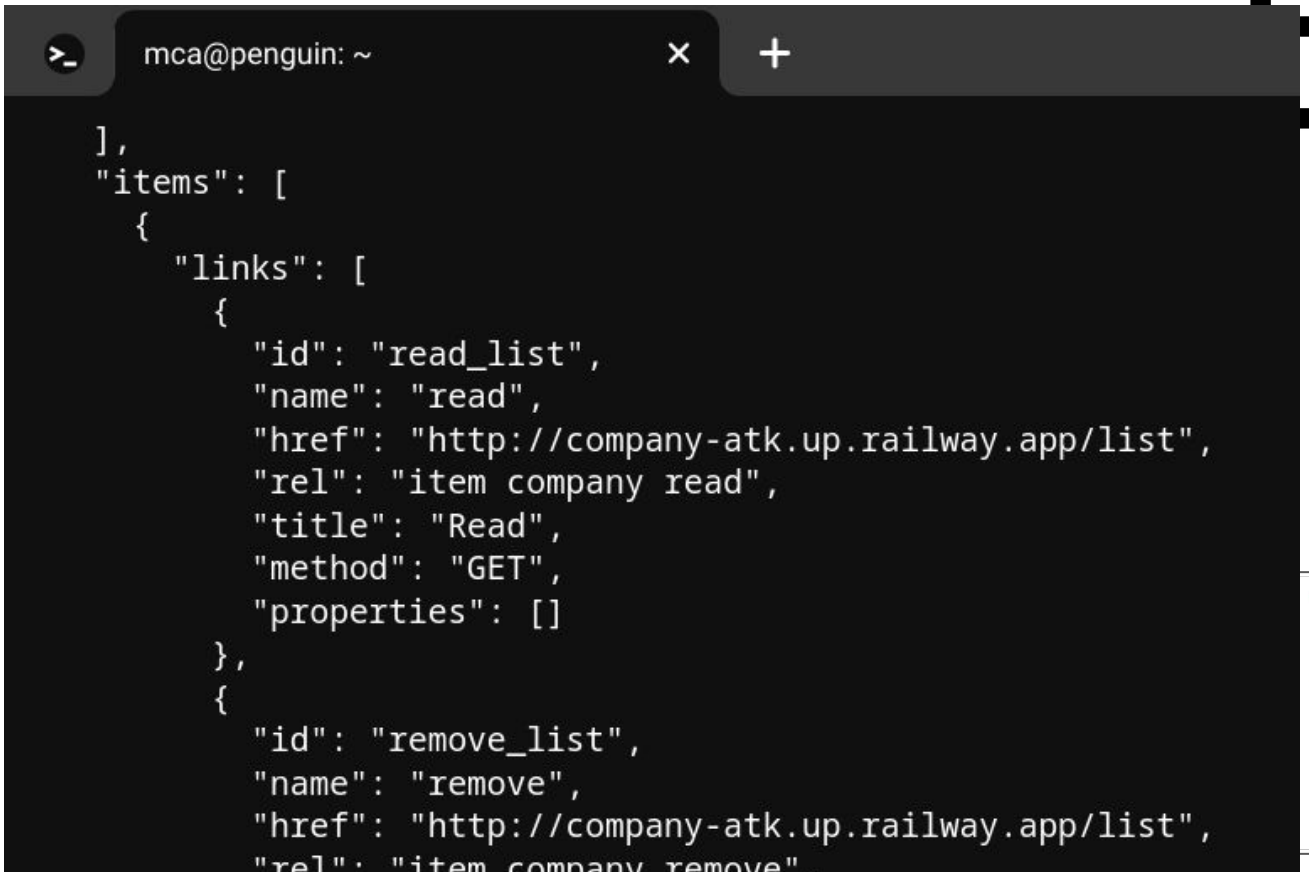
HyperCLI is a smart HTTP client

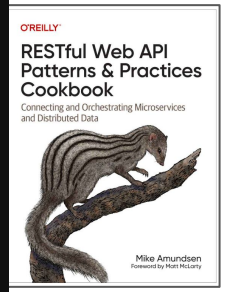
mca@penguin: ~

```
> SHOW REQUEST
{
  "url": "http://company-atk.up.railway.app",
  "method": "GET",
  "query": {},
  "headers": {
    "user-agent": "hyper-cli"
  },
  "body": ""
}
> SHOW METADATA
{
  "url": "https://company-atk.up.railway.app/",
  "statusCode": 200,
  "headers": {
    "x-powered-by": "Express",
    "access-control-allow-origin": "*",
```



HyperCLI lets you inspect responses

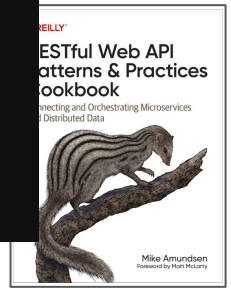




HyperCLI is format-aware (forms+json)

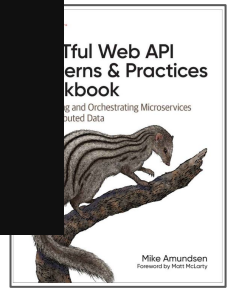
mca@penguin: ~

```
> FJ META
[
  {
    "name": "title",
    "value": "BigCo Company Records"
  },
  {
    "name": "author",
    "value": "Mike Amundsen"
  },
  {
    "name": "release",
    "value": "1.0.0"
  }
]
```

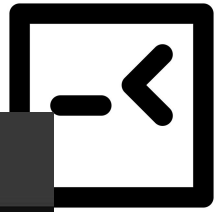


HyperCLI is format-aware

```
> FJ LINKS
[
  {
    "id": "self",
    "name": "self",
    "href": "http://company-atk.up.railway.app/",
    "rel": "self colllection company",
    "tags": "collection company self home list item",
    "title": "Self",
    "method": "GET",
    "properties": []
  },
  {
    "id": "home",
    "name": "home",
    "href": "http://company-atk.up.railway.app/",
    "rel": "collection company",
```

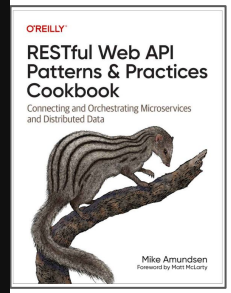


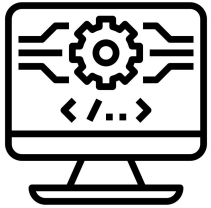
HyperCLI has a memory (stack-based)



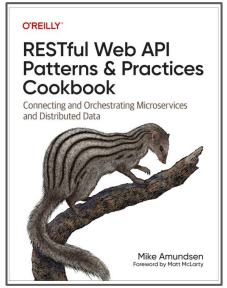
mca@penguin: ~

```
> DISPLAY LEN
4
> DISPLAY
{
  "home": {
    "metadata": [
      {
        "name": "title",
        "value": "BigCo Company Records"
      },
      {
        "name": "author",
        "value": "Mike Amundsen"
      },
      {
        "name": "release",
```



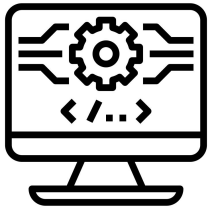


HyperLANG is a DSL



```
(0)  INPUT INVENTORY FILE-A PRICE FILE-B ; OUTPUT PRICED-INV FILE-C UNPRICED-
     INV FILE-D ; HSP D .

(1)  COMPARE PRODUCT-NO (A) WITH PRODUCT-NO (B) ; IF GREATER GO TO OPERATION
     10 ; IF EQUAL GO TO OPERATION 5 ; OTHERWISE GO TO OPERATION 2 .

(2)  TRANSFER A TO D .

(3)  WRITE-ITEM D .

(4)  JUMP TO OPERATION 8 .

(5)  TRANSFER A TO C .

(6)  MOVE UNIT-PRICE (B) TO UNIT-PRICE (C) .

(7)  WRITE-ITEM C .

(8)  READ-ITEM A ; IF END OF DATA GO TO OPERATION 14 .

(9)  JUMP TO OPERATION 1 .

(10) READ-ITEM B ; IF END OF DATA GO TO OPERATION 12 .

(11) JUMP TO OPERATION 1 .

(12) SET OPERATION 9 TO GO TO OPERATION 2 .

(13) JUMP TO OPERATION 2 .

(14) TEST PRODUCT-NO (B) AGAINST ZZZZZZZZZZZZ ; IF EQUAL GO TO OPERATION 16 ;
     OTHERWISE GO TO OPERATION 15 .

(15) REWIND B .

(16) CLOSE-OUT FILES C , D .

(17) STOP . (END) Space Fill to End of Block.
```

FLOW-MATIC programming language (1955)



```
(0)  INPUT INVENTORY FILE-A PRICE FILE-B ; OUTPUT PRICED-INV FILE-C UNPRICED-
     INV FILE-D ; HSP D .

(1)  COMPARE PRODUCT-NO (A) WITH PRODUCT-NO (B) ; IF GREATER GO TO OPERATION
     10 ; IF EQUAL GO TO OPERATION 5 ; OTHERWISE GO TO OPERATION 2 .

(2)  TRANSFER A TO D .

(3)  WRITE-ITEM D .

(4)  JUMP TO OPERATION 8 .

(5)  TRANSFER A TO C .

(6)  MOVE UNIT-PRICE (B) TO UNIT-PRICE (C) .

(7)  WRITE-ITEM C .

(8)  READ-ITEM A ; IF END OF DATA GO TO OPERATION 14 .

(9)  JUMP TO OPERATION 1 .

(10) READ-ITEM B ; IF END OF DATA GO TO OPERATION 12 .

(11) JUMP TO OPERATION 1 .

(12) SET OPERATION 9 TO GO TO OPERATION 2 .

(13) JUMP TO OPERATION 2 .

(14) TEST PRODUCT-NO (B) AGAINST ZZZZZZZZZZZZ ; IF EQUAL GO TO OPERATION 16 ;
     OTHERWISE GO TO OPERATION 15 .

(15) REWIND B .

(16) CLOSE-OUT FILES C , D .

(17) STOP . (END) Space Fill to End of Block.
```

"I've always been more interested in the future than in the past."

-- Grace Hopper (1906 - 1992)

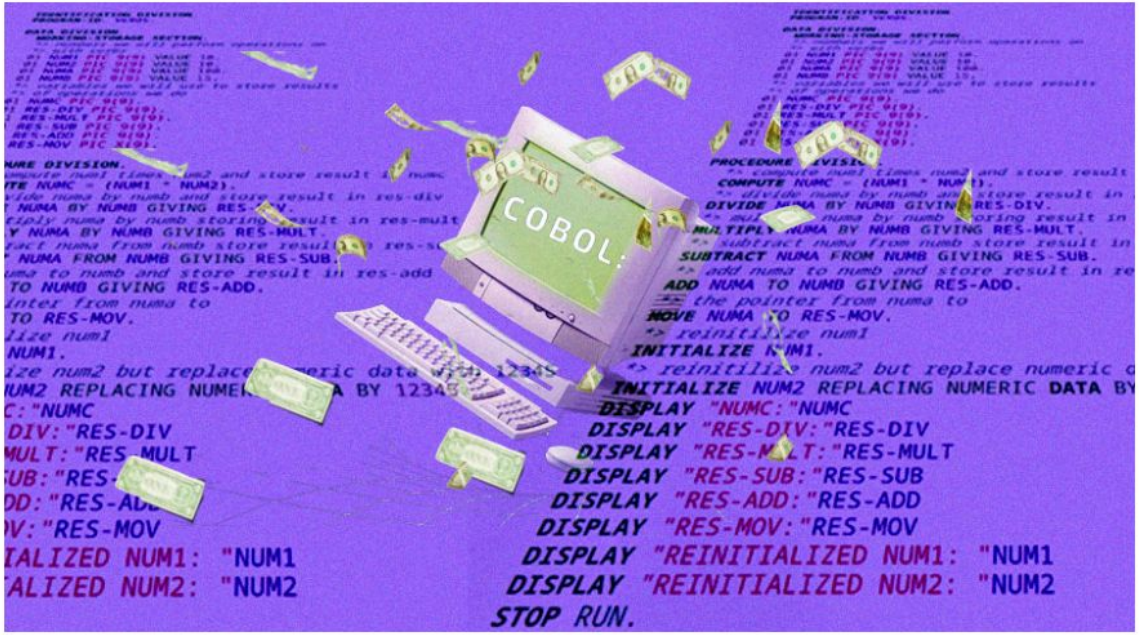
FLOWMATIC becomes COBOL

```
000024
000025        PROCEDURE DIVISION.
000026        0001-MAIN.
000027            INSPECT FUNCTION REVERSE(STR-1)
000028                   TALLYING WS-LEN1 FOR LEADING SPACES.
000029            COMPUTE  WS-LEN = LENGTH OF STR-1 - WS-LEN1.
000030            DISPLAY WS-LEN.
000031            MOVE 1 TO I.
000032            MOVE WS-LEN TO J.
000033            PERFORM REV-PARA WS-LEN TIMES.
000034            DISPLAY STR-1.
000035            DISPLAY STR-2.
000036            GOBACK.
000037         REV-PARA.
000038            MOVE STR-1(J:1) TO STR-2(I:1).
000039            SUBTRACT 1 FROM J.
000040            ADD 1 TO I.
000041            EXIT.
****** ************************** Bottom of Data ****************************
```

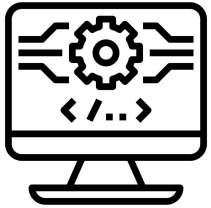
COBOL is Still a High-Paying Job Option for Many Developers

Veda

June 16, 2022 . 2 mins read



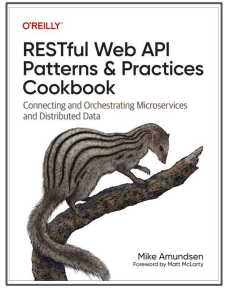
HyperLANG is a DSL



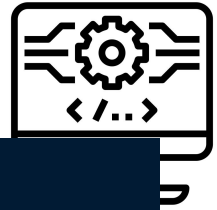
Terminal - mca@mamund-ws: .../2023-05-goto

File Edit View Terminal Tabs Help

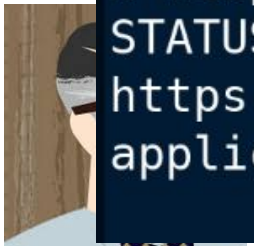
```
> HELP

  ACTIVATE|A|GO|GOTO|CALL|REQUEST|REQ -- synonyms
    WITH-URL <url|$#>
    WITH-REL <string|$#>
    WITH-NAME <string|$#>
    WITH-ID <string|$#>
    WITH-PATH <json-path-string|$#> (applies JSONPath that returns URL)
    WITH-OAUTH <string|$#> (sets the HTTP authorization header from named OAUTH config)
    WITH-BASIC <string|$#> (uses username and password stored at <string> for basic auth)
    WITH-ACCEPT <string|$#> (sets the HTTP accept header directly)
    WITH-FORMAT (uses config.accept property)
    WITH-PROFILE (uses confg.profile property)
    WITH-QUERY <{n:v,...}|$#>
    WITH-BODY <name=value&..|{"name":"value",...}|$#>
    WITH-HEADERS <{"name":"value",...}|$#>
    WITH-ENCODING <string|$#>
    WITH-METHOD <string>
    WITH-FORM <form-identifier-string|#>
    WITH-STACK (uses top stack item for input/query values)
    WITH-DATA <{n:v,...}|$#> (uses JSON object for input/query values)
  CLEAR
  VERSION (returns version of hyper repl)
  SHELL command-string <== "Here be dragons!"
    LS || DIR [folder-string]
  PLUGINS (returns list of loaded plug-in modules)

  See also: STACK HELP, CONFIG HELP, DISPLAY HELP and <PLUGIN> HELP

  Use $>hyper < command-file > output-file to execute HyperLANG scripts
  Use $>hyper "line1;line2;line3" to execute HyperLANG from command line
>
```

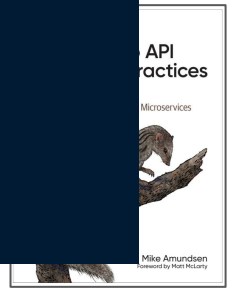


HyperLANG speaks HTTP

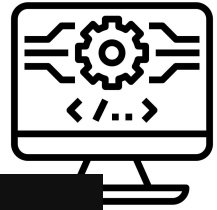


```
# request
CALL http://company-atk.up.railway.app
  WITH-OAUTH gotoMike
  WITH-METHOD post
  WITH-DATA {"companyName":"goto-co","status":"pending"}
  WITH-CONTENT-TYPE application/x-www-form-urlencoded
  WITH-ACCEPT application/json
  WITH-HEADERS {"x-transaction":"q1w2e3"}

# response
STATUS 200
https://company-atk.up.railway.app/
application/forms+json; charset=utf-8
```

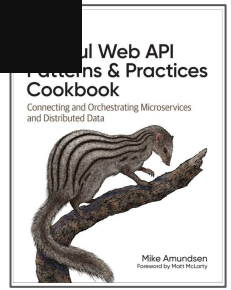




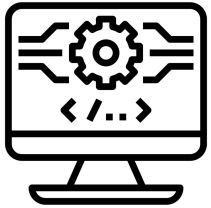
HyperLANG understands responses



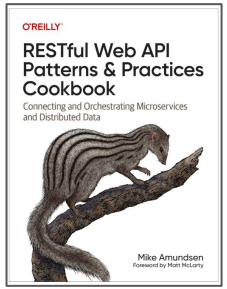
```
DISPLAY|SHOW|HISTORY (synonyms)
  ALL : returns the complete interaction (request, response metadata, response body)
  REQUEST : returns the details of the request (URL, headers, querystring, method, body)
  METADATA|META : returns the response metadata (URL, status, & headers)
  URL|HREF : returns the URL of the current response
  STATUS|STATUS-CODE : returns the HTTP status code of the current response
  CONTENT-TYPE : returns the content-type of the current response
  HEADERS : returns the HTTP headers of the current response
  PEEK : displays the most recent response on the top of the stack (non-destructive)
  POP : pops off [removes] the top item on the response stack
  LENGTH|LEN : returns the count of the responses on the response stack
  CLEAR|FLUSH : clears the response stack
  PATH <jsonpath|xmlpath|$#> : based on response content type, applies supplied query (JSON/XML)
  JPATH <jsonpath-string|$#> : applies the JSON Path query to the response at the top of the stack
  XPATH <xmlpath-string|$#> : applies the XPATH query to the response at the top of the stack
```



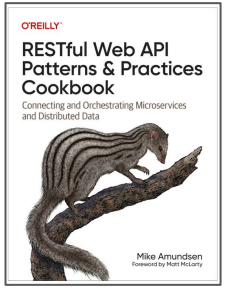
HyperLANG supports OAUTH (via plug-ins)

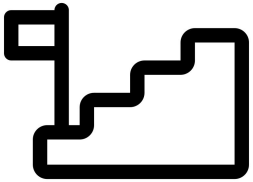


```
> OAUTH HELP
OAUTH
    LIST [name|$#] (lists existing OAUTH definitions using optional name filter
    DEFINE <name> <{"url":"...","id":"...","secret":"...","audience":"...","type":"..."}> (creates a new definition)
    UPDATE <name|$#> <{"n":"...","v":"..."}> (updates existing <name> definition)
    REMOVE <name|$#> (removes <name> from OAUTH collection
    GENERATE <auth-name|$#> (gets a token from OAUTH provider and loads it into the <name> definition)
    SAVE [file-name|$#] saves entire configuration set to disk (defaults to oauth.env)
    LOAD [file-name|$#] reads entire configuration set from disk (defaults to oauth.env)
```

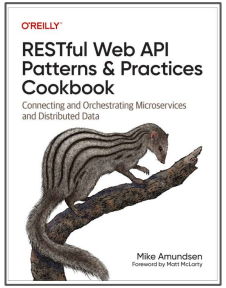


"OK, let's see some examples"

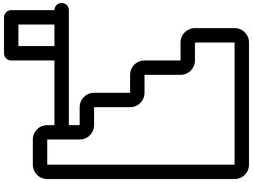




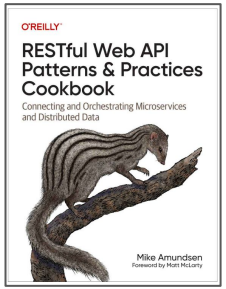
HyperLANG Basics



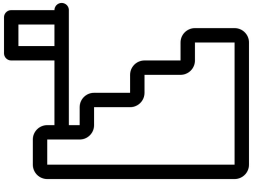
HyperLANG Basics



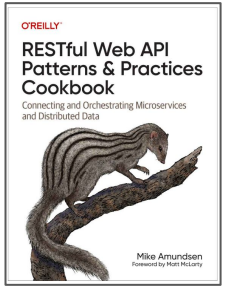
- Calling Services
  - The Anatomy of HyperLANG calls
- Navigating Responses
  - FORMAT DSLs & XPATH/JPATH
- Writing Data with HyperLANG
  - Leveraging auto-complete FORMs



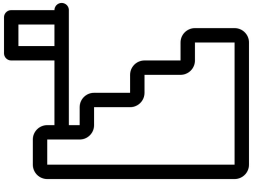
Making a CALL in HyperLANG



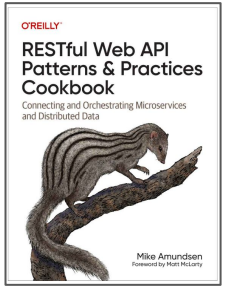
```
> CALL https://company-atk.up.railway.app

STATUS 200
https://company-atk.up.railway.app
application/forms+json; charset=utf-8
```



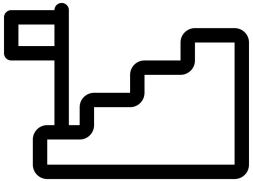
Using the WITH-QUERY keyword



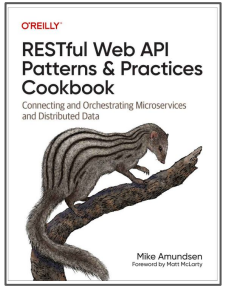
```
> CALL https://company-atk.up.railway.app/filter
  WITH-QUERY {"companyName":"mike-co","status":"pending"}

STATUS 200
https://company-atk.up.railway.app/filter?companyName=mike-co&status=pending
application/forms+json; charset=utf-8
>
```



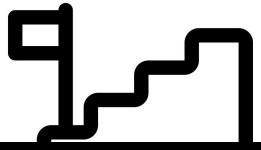
Using the WITH-METHOD keyword



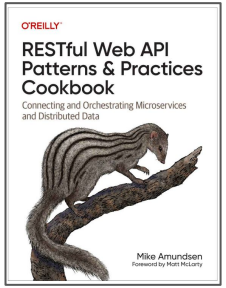
```
CLEAR
CONFIG SET {"company":"http://localhost:8484/filter","accept":"application/forms+json"}

# START

# get my list of records
CALL WITH-URL $$company$$ WITH-METHOD GET WITH-QUERY {"companyName":"mike-co"}
SHOW
```



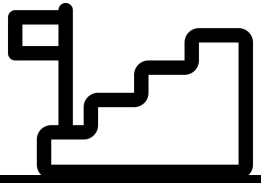
Using the WITH-NAME keyword



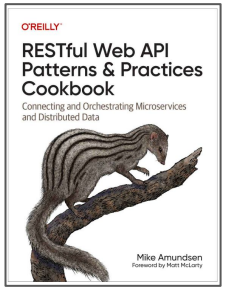
```
CLEAR
CONFIG SET {"company":"http://localhost:8484/","accept":"application/forms+json"}

# START

CALL WITH-URL $$company$$
CALL WITH-NAME list
CALL WITH-NAME filter WITH-QUERY {"companyName":"mike-co"}
SHOW
```



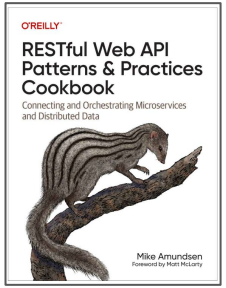
Using the FORM and STACK keywords



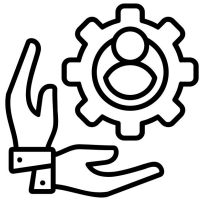
```
CLEAR
CONFIG SET {"company":"http://localhost:8484/","accept":"application/forms+json"}
STACK PUSH {"companyName":"mike-co"}

# START

CALL WITH-URL $$company$$
CALL WITH-NAME list
CALL WITH-FORM filter WITH-STACK
SHOW PATH $.company.items[*].[id,companyName,status]
```



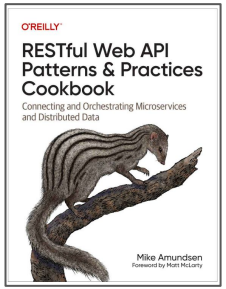
"Are you sure we should be programming in HTTP?"



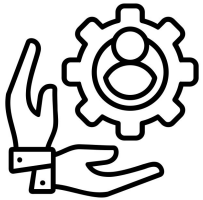
HTTP is a great interface language



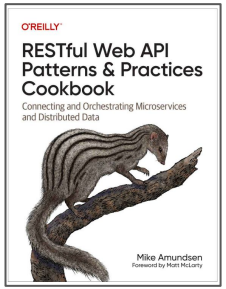
- HTTP was designed to be extremely loosely coupled
- Just about any service can be expressed via HTTP
- HTTP never crashes (services do, tho!)
- A smart client can compose multiple services into a solution



HTTP is a great interface language



- HTTP was designed to be extremely loosely coupled
  - HTTP REST was defined in 2000
- Just about any service can be expressed via HTTP
- HTTP never crashes (services do, tho!)
- A smart client can compose multiple services into a solution



HTTP and REST are still here

UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Styles and
the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

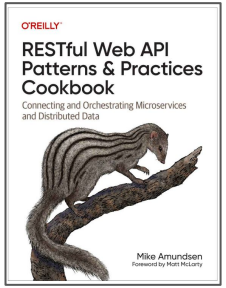
DOCTOR OF PHILOSOPHY

in Information and Computer Science

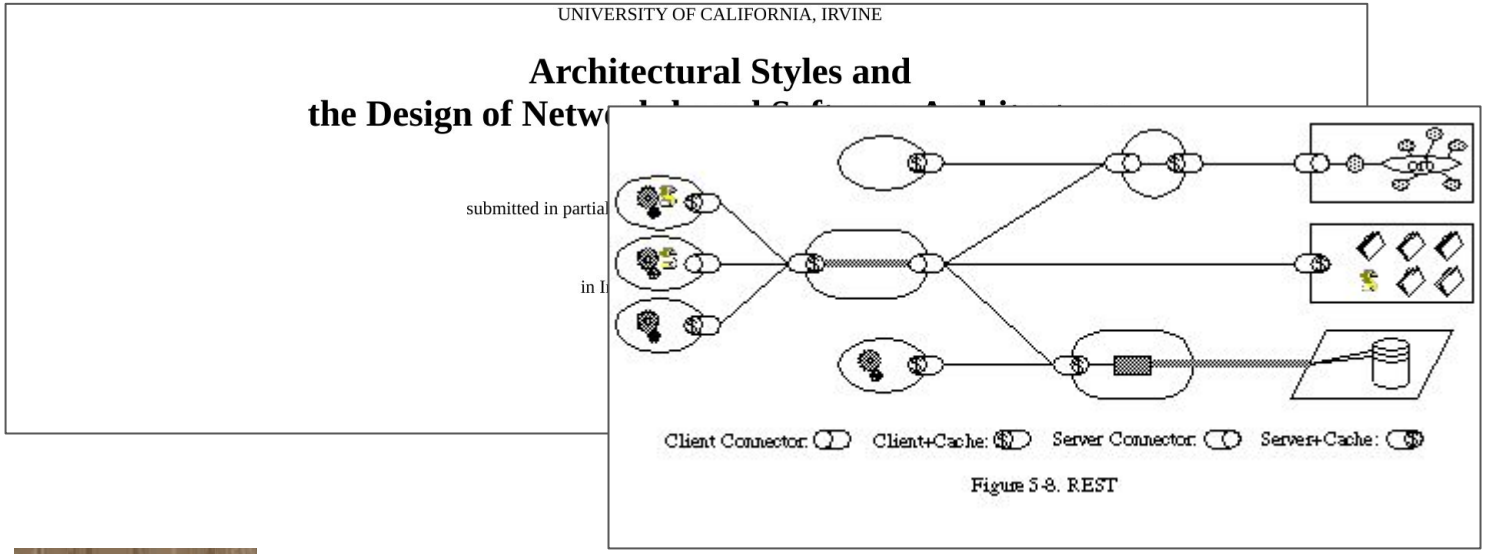
by

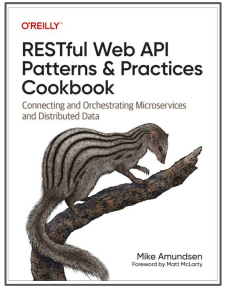
Roy Thomas Fielding

2000

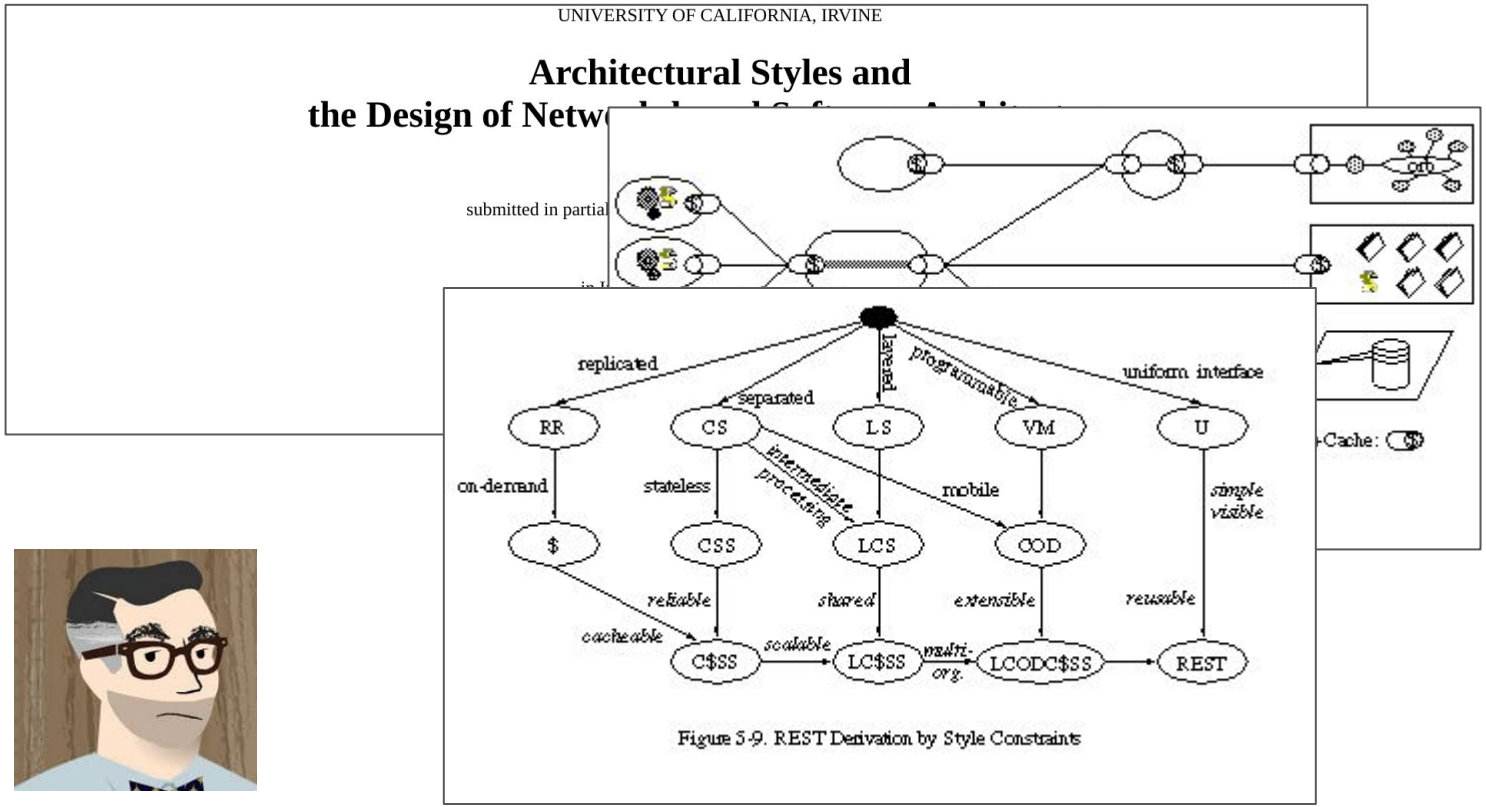


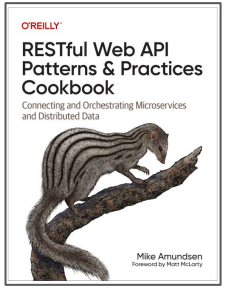
HTTP and REST are still here



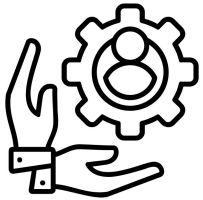


HTTP and REST are still here

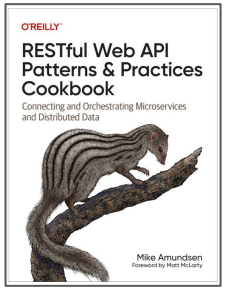




HTTP is a great interface language



- HTTP was designed to be extremely loosely coupled
- Just about any service can be expressed via HTTP
  - Business, science, entertainment
- HTTP never crashes (services do, tho!)
- A smart client can compose multiple services into a solution



APIs are everywhere

Gaming APIs

Gaming related APIs and Resources

Search

Collections

- Recommended APIs
- Popular APIs
- Free Public APIs for Developers
- Top AI Based APIs
- Tax APIs
- View All Collections

Categories

Trivia by API-Ninjas

Access endless trivia question and answers. See more info at https://api-ninjas.com/api/trivia.

9.7 294 ms 100%

Rocket League

Ranks, stats, news & more, provided by the fastest and most powerful API for Rocket League.

9.7 81 ms 100%

League of Legends Champions

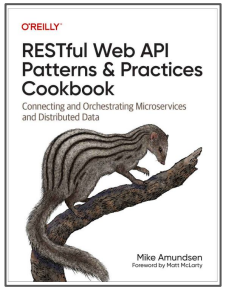
You can get all of league of legends champions list. With more than 140 champions, you'll find the

9.6 16,234 ms 97%

Minecraft Server Status

Get player count, MOTD, status and more for Java or Bedrock servers.

9.5 376 ms 100%



APIs are everywhere

Gaming APIs

Gaming related APIs and Resources

Search

Collections

- Recommended APIs
- Popular APIs
- Free Public APIs for Developer
- Top AI Based APIs
- Tax APIs
- View All Collections

Categories

Synapse

Schedule Consulta

NeoBanking

Disrupt traditional banking and deliver new services like credit builder loans and branded cards to new audiences.

Lending & Credit

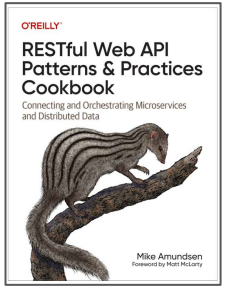
Bring credit to the masses, including under-represented groups. Credit Hub can enable the democratization of credit and a path to the American dream of homeownership.

Card Solutions

Bring credit to the masses, including under-represented groups. Credit Hub can enable the democratization of credit and a path to the American dream of homeownership.

Gig Economy

Wealth Management



APIs are everywhere

Gaming APIs

Gaming related APIs and Resources

Search

Collections

Recommended APIs

Popular APIs

Free Public APIs for Developer

Top AI Based APIs

Tax APIs

View All Collections

Categories

Synapse

NeoBanking

Disrupt traditional banking and deliver new services like credit builder loans and branded cards to new audiences.

Gig Economy

Wealth Management

COVER GENIUS

PLATFORMS SOLUTIONS INDUSTRIES INSURANCE RESEARCH COMPANY

FOLLOWING

EMBEDDED INSURAN

FOR SEAMLESS PROT

Cover Genius is the insurtech for embedde

protect the global customers of the world's

– including Booking Holdings, Intuit, Hopp

Airlines, Descartes, ShipRush, Zip and SeatGeek. Our global

PLATFORMS

XCOVER
The award-winning global platform for any line of protection

RENTALCOVER
The worldwide #1 distribution platform for rental car protection

XCOVER GO
Warranties & shipping protection via an SDK or eCommerce plugins

MICROSERVICES

XCLAIM
The API for complete claims management & instant payment

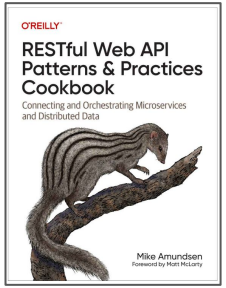
BRIGHTWRITE
Real-time price optimization & product recommendations

CHECK OUT OUR RESTFUL APIS

XClaim API

XCover API

RentalCover API



APIs are everywhere

Gaming APIs

Gaming related APIs and Resources

Search

Collections

Recommended APIs

Popular APIs

Free Public APIs for Developer

Top AI Based APIs

Tax APIs

View All Collections

Categories

Synapse

NeoBanking

Disrupt traditional banking and deliver new services like credit builder loans and branded cards to new audiences.

Gig Econom

COVER GENIUS

PLATFORMS SOLUTIONS INDUSTRIES INSURANCE RESEARCH COMPANY

FOLLOWING

EMBEDDED INSURAN

FOR SEAMLESS PROT

Cover Genius is the insurtech for embedde

PLATFORMS

XCOVER
The award-winning global platform for any line of protection

RENTALCOVER
The worldwide #1 distribution platform for rental car protection

XCOVER GO
Warranties & shipping protection via an SDK or eCommerce plugins

MICROSERVICES

XCLAIM
The API for complete claims management & instant payment

BRIGHTWRITE
Real-time price optimization & product recommendations

zeomega

Who We Serve Solutions Resources Company

Request Demo

Commercial Health Plans

A one-stop shop for care management, benefits administration, process automation, organizational optimization, and efficiency across the enterprise.

Learn More

Health Systems

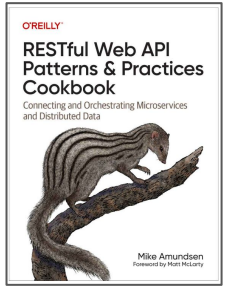
Exceed your value-based care benchmarks leveraging combined clinical and claims data to reduce risk, close gaps, and ensure optimal outcomes.

Learn More

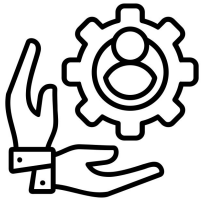
Accountable Care Organizations

From risk-stratification to gap identification and closure strategies, our capabilities are unmatched, and built in SDOH criteria helps you reach at-risk communities with effective programs.

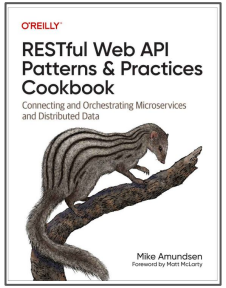
Learn More

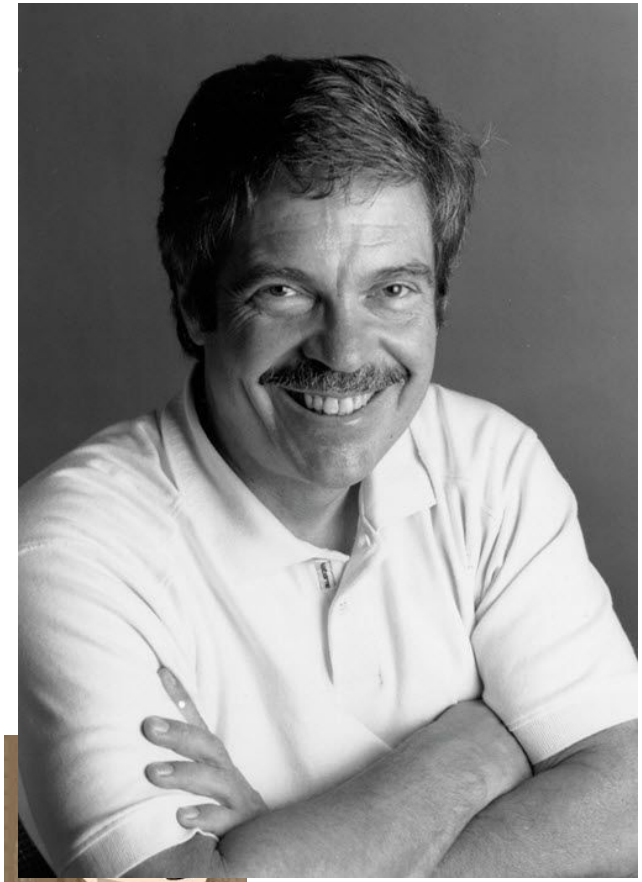


HTTP is a great interface language



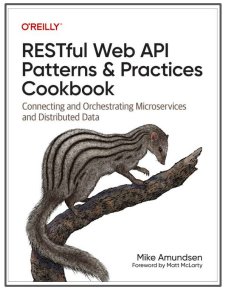
- HTTP was designed to be extremely loosely coupled
- Just about any service can be expressed via HTTP
- HTTP never crashes (services do, tho!)
  - HTTP has been "up and running" for almost 30 years
- A smart client can compose multiple services into a solution



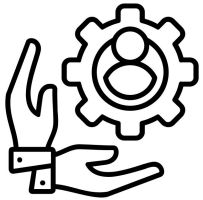


"The Internet was done so well that most people think of it as a natural resource like the Pacific Ocean, rather than something that was man-made. When was the last time a technology with a scale like that was so error-free?"

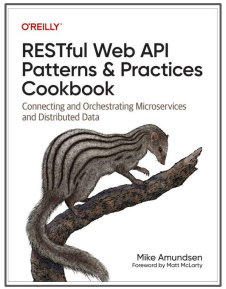
-- Alan Kay, 2012



HTTP is a great interface language

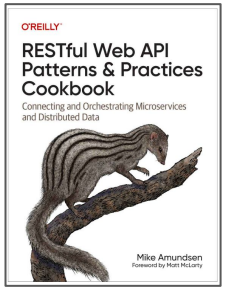


- HTTP was designed to be extremely loosely coupled
- Just about any service can be expressed via HTTP
- HTTP never crashes (services do, tho!)
- A smart client can compose multiple services into a solution
  - Services don't need to "know about each other"

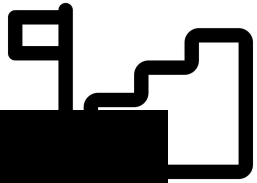


"I had to define a system that could withstand decades of change produced by people spread all over the world. [D]ecades of use while the system continued to evolve, in independent and orthogonal directions, without ever needing to be shut down or redeployed."

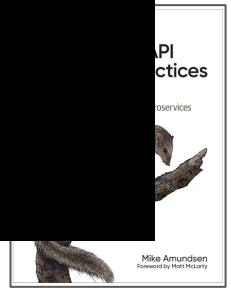
-- Roy Fielding, 2014



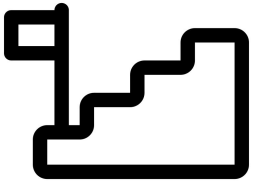
Creating new records with HyperCLI



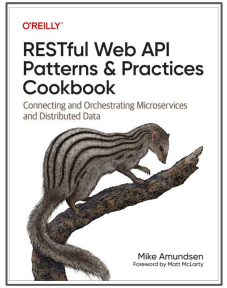
```
# SETUP
CLEAR
CONFIG SET {"company":"http://localhost:8484","accept":"application/forms+js
STACK PUSH {"id":"a1b2c3","companyName":"GOTO-CHI","streetAddress":"123 Main
IL","postalCode":"12345","country":"USA","email":"goto@example.org","status"

# START

# navigate to create form
CALL WITH-URL $$company$$
CALL WITH-NAME home
CALL WITH-NAME list

# execute the create step
CALL WITH-FORM createCompany WITH-STACK

# show the results
CALL WITH-FORM filter WITH-QUERY {"companyName":"GOTO-CHI"}
SHOW PATH $.company.items[0]
```



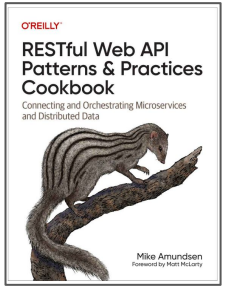
Updating records with HyperCLI



```
# navigate to create form
CALL WITH-URL $$company$$
CALL WITH-NAME home
CALL WITH-NAME list

# pull the target record
CALL WITH-FORM filter WITH-DATA {"companyName":"GOTO-CHI"}

# save record to the stack
STACK PUSH WITH-PATH $.company.items[0]

# update the stack record
STACK SET {"telephone":"789-123-4560"}

# write modified data to server using the stack record
CALL WITH-FORM update WITH-STACK

# show the results
CALL WITH-FORM filter WITH-STACK
SHOW PATH $.company.items[0]
```



Yeah, but can it do SOAP?

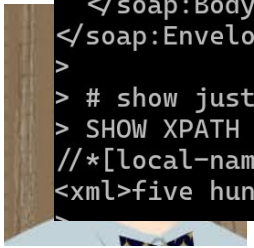


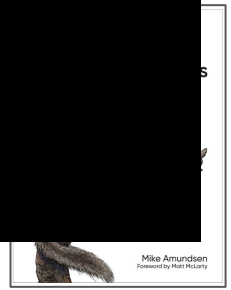
HyperLANG does SOAP!

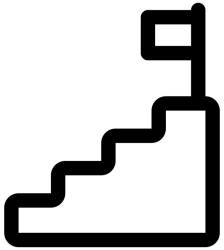
```
# pull WSDL
GOTO WITH-URL https://www.dataaccess.com/webservicesserver/NumberConversion.wso?WSDL

# get the names of possible operations
SHOW XPATH //*[local-name(.)='operation']/@name

# get the documentation nodes
SHOW XPATH //*[local-name(.)='operation']/*[local-name(.)='documentation']

# just show me the text of the documentation nodes
SHOW XPATH //*[local-name(.)='operation']/*[local-name(.)='documentation']/text()

# call the NumberToWords operation
GOTO WITH-URL https://www.dataaccess.com/webservicesserver/NumberConversion.wso WITH-BODY [%
  <?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <NumberToWords xmlns="http://www.dataaccess.com/webservicesserver/">
        <ubiNum>500</ubiNum>
      </NumberToWords>
    </soap:Body>
  </soap:Envelope>
%] WITH-METHOD post WITH-ENCODING text/xml

# show full response
SHOW RESPONSE
```

HyperLANG does SOAP!

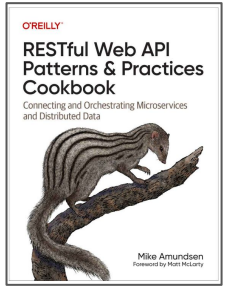
```
> # call the NumberToWords operation
> GOTO WITH-URL https://www.dataaccess.com/webservicesserver/NumberConversion.wso WITH-BODY [% <?xml version="1
0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><NumberT
ords xmlns="http://www.dataaccess.com/webservicesserver/"><ubiNum>500</ubiNum></NumberToWords></soap:Body></soa
:Envelope> %] WITH-METHOD post WITH-ENCODING text/xml

STATUS 200
https://www.dataaccess.com/webservicesserver/NumberConversion.wso
text/xml; charset=utf-8
>
> # show full response
> SHOW RESPONSE
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <m:NumberToWordsResponse xmlns:m="http://www.dataaccess.com/webservicesserver/">
      <m:NumberToWordsResult>five hundred </m:NumberToWordsResult>
    </m:NumberToWordsResponse>
  </soap:Body>
</soap:Envelope>
>
> # show just the results
> SHOW XPATH //*[local-name(.)='NumberToWordsResult']/text()
//*[local-name(.)='NumberToWordsResult']/text()
<xml>five hundred </xml>
```



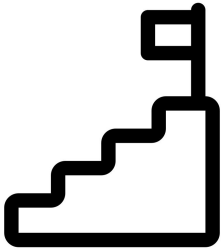




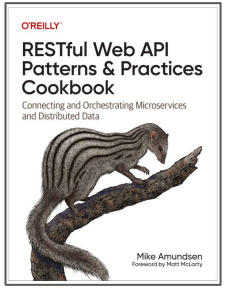
Doing More with HyperCLI

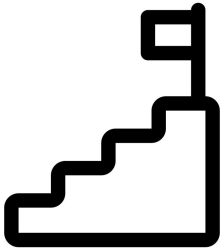


Doing More with HyperCLI

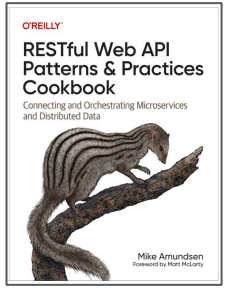


- Security
  - Supporting OAUTH
- Plug-Ins
  - Understanding the plug-in model
- Scripting
  - accessing *.hyper, config, and memory files

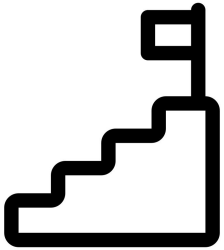




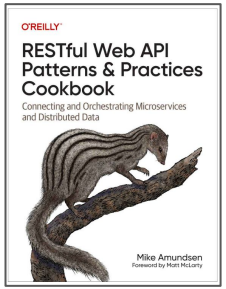
Security


O'REILLY
RESTful Web API Patterns & Practices Cookbook
Connecting and Orchestrating Microservices and Distributed Data
Mike Amundsen
Foreword by Matt McLarty

Supporting OAUTH

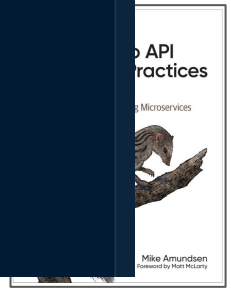


- Use DEFINE to create new OAUTH profiles
- Use GENERATE to get a fresh token
- Use WITH-OAUTH to send encoded header
- Also supports BASIC authorization

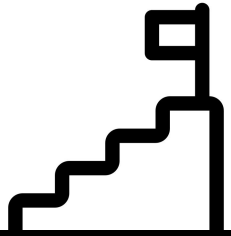


Secure Connections with HyperLANG

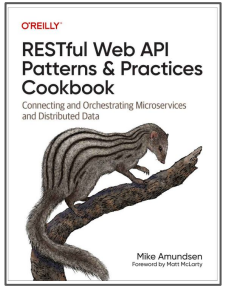
```
# ************************************************
# for details see:
# https://developer.capitalone.com/documentation/retrieve-consumer-bank-products
#
# Before running this script, you need to have :
# - a valid capital one API account,
# - a registered app,
# - oAuth client_id and client_secret
#
# use OAUTH DEFINE & OAUTH SAVE to create a persistent profile: devExchange
#
# OAUTH DEFINE devExchange {
#  "url":"https://api-sandbox.capitalone.com/oath2/token",
#  "id":"...",
#  "secret":"...",
#  "content-type":"application/x-www-form-urlencoded"}
#
# OAUTH SAVE
#
# ************************************************
```



Secure Connections with HyperLANG



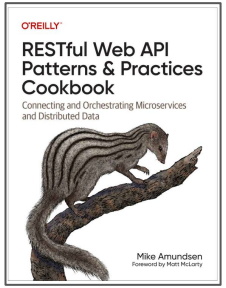
```
CONFIG SET {"devExDepositsAPI":"https://api-sandbox.capitalone.com/deposits/products/~/search"}

OAUTH LOAD
OAUTH GENERATE devExchange

CALL WITH-URL $$devExDepositsAPI$$ WITH-METHOD post WITH-ACCEPT application/json;v=5 WITH-OAUTH devExchange

SHOW PATH $..productName

CONFIG REMOVE devExDepositsAPI

EXIT
```



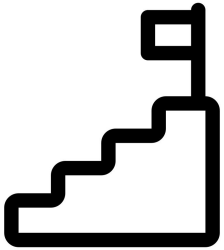
Secure Connections with HyperLANG

```
CONFIG SET {"devExDepositsAPI":"https://api-sandbox.capitalone.com/deposits/products/~/search"}

OAUTH LOAD
OAUTH GENERATE devExchange

CALL WITH-URL $$devExDepositsAPI$$ WITH-METHOD post WITH-ACCEPT application/json;v=5 WITH-OAUTH devExchange

SHOW PATH $..productName

CONFIG REMOVE devExDepositsAPI

EXIT
```

```
> SHOW PATH $..productName
$..productName
[
  "360 Savings",
  "Kids Savings Account",
  "CD",
  "360 Performance Savings",
  "360 Checkings",
  "MONEY"
]
>
```



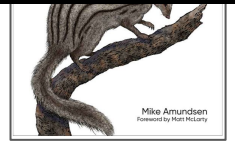
Secure Connections with HyperLANG

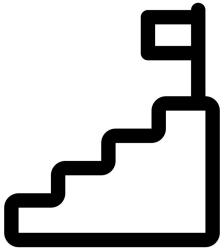


```
> SHOW REQUEST
{
  "url": "https://api-sandbox.capitalone.com/deposits/products/~/search",
  "method": "post",
  "query": {},
  "headers": {
    "authorization": "Bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2Iiwia2lkIjoicjRxIiwidHY
..p9MzZBM7-jqqFX2h-xOYxQ.Y6tVEvC3fE1VaQK0KbMFpuNDND3y1vXhlWLThDdpHyUABnoUJ7Z2l9LG6eGNVaFSBfgKwPU
5gk4RxrkqCNG8rkOOZrrrTtjU0lHsDdIzIYwhQAzqy9ruGsu9unOccyWkHmT6OCuBQUta9sfzIoiSxYj6aoKc8xNwwQzccoD
lx174rmY81lCFiJ85BTRwBBczm97OzTsprvlhJgs8K-k-GA8YKfY9d9NodJuPxGUeeeK6kpJi_erlVw.eXk9x_rH8MuR_ZSr
    "accept": "application/json;v=5",
    "user-agent": "hyper-cli"
  },
  "body": ""
}
>
```

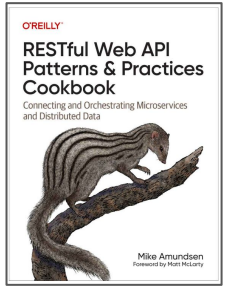
HyperLANG also supports BASIC AUTH

```
#
# basic auth example for github access
#

# clear display & load internal oauth info
CLEAR
OAUTH LOAD

# get my user profile
GOTO WITH-URL https://api.github.com/users/mamund WITH-BASIC github-basic
SHOW BODY

EXIT
```

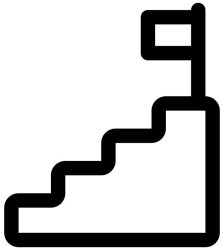




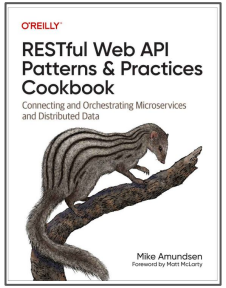
Plug-Ins


O'REILLY
RESTful Web API Patterns & Practices Cookbook
Connecting and Orchestrating Microservices and Distributed Data
Mike Amundsen
Foreword by Matt McLarty

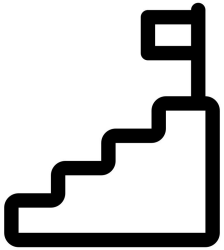
Plug-Ins



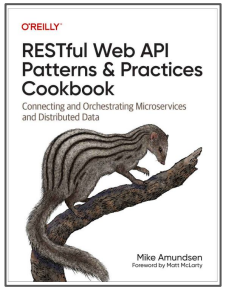
- Extend HyperLANG/HyperCLI with NodeJS
- All HyperLANG features available to plug-in authors
- HyperCLI loads plug-ins automatically



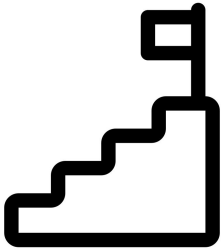
HyperLANG Plug-Ins



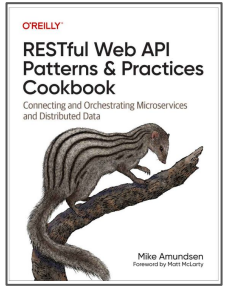
```
> PLUGINS
["CJ","FJ","HAL","OAUTH","PHTAL","SIREN","WSTL"]
> CJ HELP
CJ
    HREF (returns the top-level HREF for this response)
    METADATA (returns the metadata collection)
    LINKS (returns the links collection)
    ITEMS (returns the items collection)
    QUERIES (returns the queries collection)
    TEMPLATE (returns the cj template)
    RELATED (returns a related collection)
    ERRORS|ERROR (returns any error object/array in the response)
    ID|REL|NAME|TAG|FORM <string|$#> (returns matching nodes)
    IDS|RELS|NAMES|TAGS|FORMS (returns simple list)
    PATH <jsonpath-string|$#>

    When using WITH-FORM, the following reserved words are supported:
      - TEMPLATE, TEMPLATE-POST, TEMPLATE-CREATE, CREATE
      - TEMPLATE-PUT, TEMPLATE-UPDATE, UPDATE
      - TEMPLATE-DELETE, DELETE

    As in:
      GOTO WITH-FORM TEMPLATE-UPDATE WITH-STACK

    CJ Templates ALWAYS use the response's collection.href as the URL
```



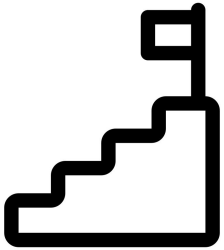
Creating a MATH plug-in



```
// mainline support
// <name of this file (e.g. WSTL.js == WSTL) invokes this plugin
// args: {responses:responses,dataStack:dataStack,config:config,words:words}
// NOTE: sample code below is an old version of WSTL support
function main(args) {
  config = args.config;
  responses = args.responses;
  dataStack = args.dataStack;
  var words = args.words;
  var rt = {};
  var index = 0;
  var token = words[1]||"";
  var response;
  var thisWord = "";
  var path = "";

  switch (token.toUpperCase()) {
    case "HELP":
      rt = showHelp(words[2]||"");
      break;
    case "ADD":
      var x,y;
      x = parseInt(words[2]);
```

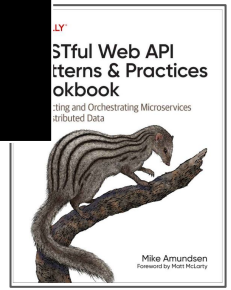


Creating a MATH plug-in

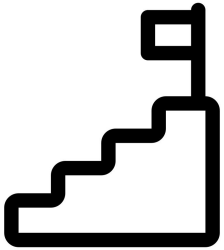


```
// mainline support
// <name of this file (e.g. WSTL.js == WSTL) invokes this plugin
// args: {responses:responses,dataStack:dataStack,config:config,words:words}
// NOTE: sample code below is an old version of WSTL support
function main(args) {
  config = args.config;
  responses = args.responses;
  dataStack = args.dataStack;
  var words = args.words;
  var rt = {};
  var index = 0;
  var token = words[1]||"";
  var response;
  var thisWord = "";
  var path = "";

  switch (token.toUpperCase()) {
    case "HELP":
      rt = showHelp(words[2]||"");
      break;
    case "ADD":
      var x,y;
      x = parseInt(words[2]);
```

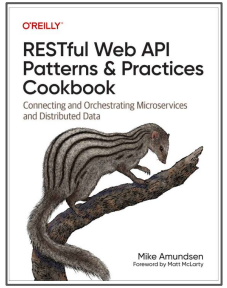
```
// show help text
function showHelp(thisWord) {
  var rt = ""
  rt =
 `MATH
    ADD x y (x plus y)
    SUB|SUBTRACT x y (x minus y)
    MUL|MULTIPLY (x y (x times y)
    DIV|DIVIDE x y (x devided by y)`;

    console.log(rt);

  return "";
}
```

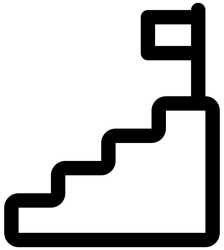


Creating a MATH plug-in

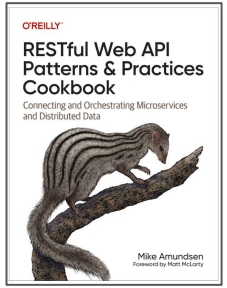


```
> MATH HELP
MATH
    ADD x y (x plus y)
    SUB|SUBTRACT x y (x minus y)
    MUL|MULTIPLY (x y (x times y)
    DIV|DIVIDE x y (x devided by y)
""
> MATH MUL 5 4
"20"
> MATH DIV 5 2
"2.5"
>
```

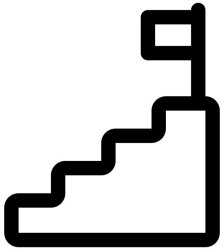




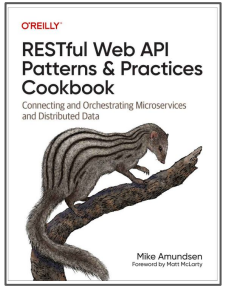
Scripting


O'REILLY
RESTful Web API Patterns & Practices Cookbook
Connecting and Orchestrating Microservices and Distributed Data
Mike Amundsen
Foreword by Matt McLarty

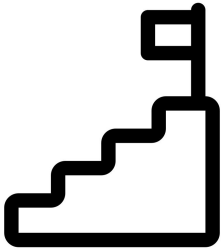
Scripting with HyperLANG



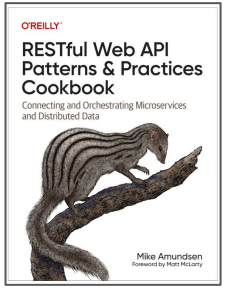
- Supports OS pipes for input / output
- Supports inline commands from shell
- Chaining scripts (pending)



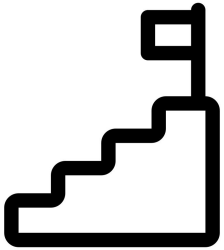
Scripting with HyperLANG



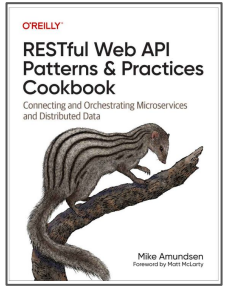
```
mamund-ws$ hyper
```



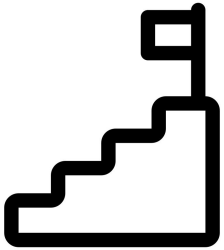
Scripting with HyperLANG



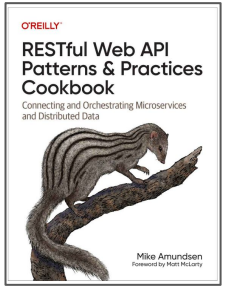
```
mamund-ws$ hyper

mamund-ws$ hyper "CALL https://company-atk.up.railway.app;CALL WITH-NAME list;SHOW ITEMS;"
```



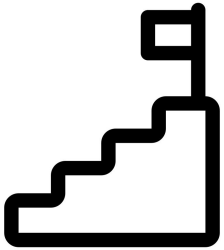
Scripting with HyperLANG



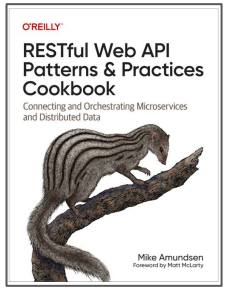
```
mamund-ws$ hyper

mamund-ws$ hyper "CALL https://company-atk.up.railway.app;CALL WITH-NAME list;SHOW ITEMS;"

mamund-ws$ hyper < company-listing.script
```



Scripting with HyperLANG



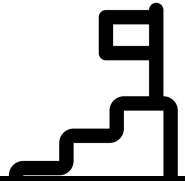
```
mamund-ws$ hyper

mamund-ws$ hyper "CALL https://company-atk.up.railway.app;CALL WITH-NAME list;SHOW ITEMS;"

mamund-ws$ hyper < company-listing.script

mamund-ws$ hyper < company-listing.script > company-listing.output
```



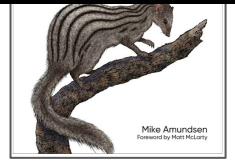
CONFIG support

```
> CONFIG
{
  verbose: 'false',
  accept: 'application/forms+json',
  profile: 'http://api.example.org/profiles/user',
  debug: 'false',
  company: 'http://localhost:8484'
}
> CONFIG HELP

  CONFIG
    READ (returns all the config values)
    SET <{"name":"value", ... }> (writes one or more config values to the list)
    REMOVE <string> (removes the named config value from the list)
    CLEAR (removes all settings)
    RESET (resets to default settings : "hyper.config")
    FILE|LOAD [file-string] : defaults to "hyper.config"
    SAVE|WRITE [file-string] : defaults to "hyper.config"

    Using $$name$$ macros in commands replaces the placeholder with correspond config value
```

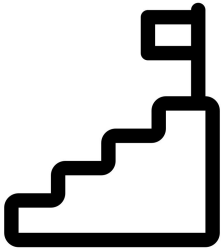
STACK support



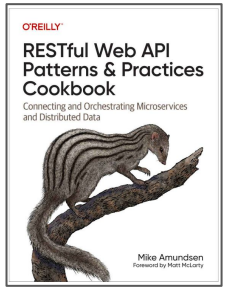
```
> STACK HELP
STACK
     PEEK (returns the top item on the stack)
     PUSH <{"n":"v", ... }> (adds a new item to the stack)
     PUSH WITH-RESPONSE (adds the most recent response to the stack)
     PUSH WITH-PATH <json-path-string> (adds the result of the JSON query to the stack)
     POP (removes the top item from the stack)
     SET <{"n":"v", ... }> (updates the top item of the stack)
     EXPAND-ARRAY [name] : expands array on the stop of the stack using _name_
     LOAD|FILE [file-string] : defaults to hyper.stack
     SAVE|WRITE [file-string] : defaults to hyper.stack
     DUMP [file-string] : defaults to hyper.dump
     FILL [file-string] : defaults to hyper.dump
     CLEAR|FLUSH (removes all the items from the stack)
     LEN|LENGTH (returns the number of items on the stack)

     Using ##name## replaces the macro with the corresponding stack value in the current record.
```



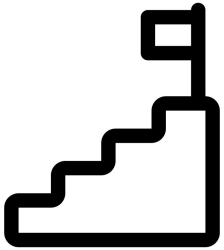
STACK support



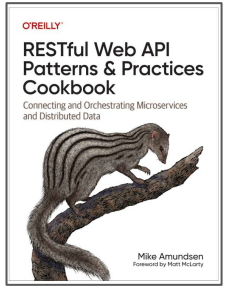
```
> STACK PUSH {"companyURL":"http://localhost:8484"}
{
  "companyURL": "http://localhost:8484"
}
> STACK PUSH {"companyURL":"http://company-atk.up.railway.app"}
{
  "companyURL": "http://company-atk.up.railway.app"
}
> CALL WITH-URL ##companyURL##
```



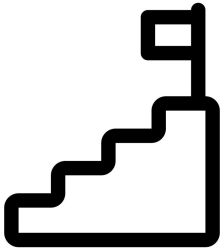
STACK support



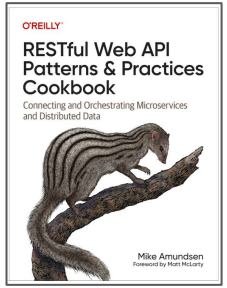
```
> STACK PUSH {"companyURL":"http://localhost:8484"}
{
  "companyURL": "http://localhost:8484"
}
> STACK PUSH {"companyURL":"http://company-atk.up.railway.app"}
{
  "companyURL": "http://company-atk.up.railway.app"
}
> CALL WITH-URL ##companyURL##

STATUS 200
https://company-atk.up.railway.app/
application/forms+json; charset=utf-8
```



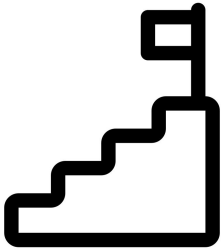
STACK support



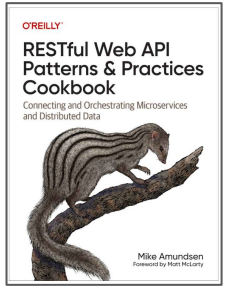
```
> STACK PUSH {"companyURL":"http://localhost:8484"}
{
  "companyURL": "http://localhost:8484"
}
> STACK PUSH {"companyURL":"http://company-atk.up.railway.app"}
{
  "companyURL": "http://company-atk.up.railway.app"
}
> CALL WITH-URL ##companyURL##

STATUS 200
https://company-atk.up.railway.app/
application/forms+json; charset=utf-8
> STACK POP
"OK"
> CALL WITH-URL ##companyURL##

STATUS 200
http://localhost:8484
application/forms+json; charset=utf-8
> STACK POP
"OK"
>
```



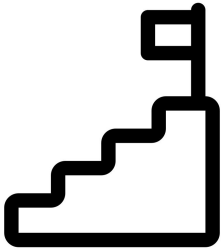
Scripting with HyperLANG



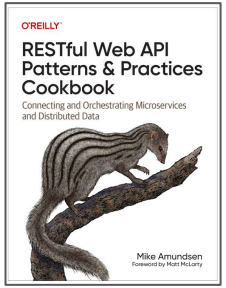
```
#
# testing support for config vars
#

# load coinfig values
CONFIG SET {"user-url":"http://rwcbook10.up.railway.app/user/"}
CONFIG SET {"user-body":"nick=mamund&name=mamunda&password=m@m*nd&email=mam
CONFIG SET {"encoding":"application/x-www-form-urlencoded"}
CONFIG SET {"user-filter":{"email":"mamund@example.org"}}

# connect to user service
# ECHO $$user-url$$
# ACTIVATE WITH-URL $$user-url$$

# create test user
# ECHO $$user-url$$
# ECHO $$user-body$$
# ECHO $$encoding$$
# CONFIG SET {"verbose":"true"}
# ACTIVATE WITH-URL $$user-url$$ WITH-BODY $$user-body$$ WITH-METHOD POST W


# confirm user exists
ECHO $$user-url$$
ECHO $$user-filter$$
CONFIG SET {"verbose":"true"}
```

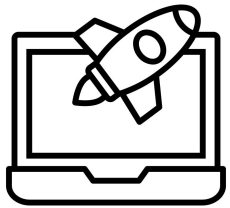


Doing More with HyperCLI

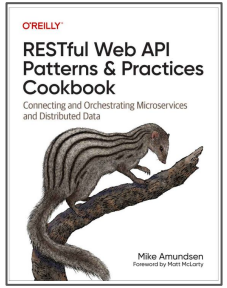


- Security
  - Supporting OAUTH
- Plug-Ins
  - Understanding the plug-in model
- Scripting
  - Accessing *.hyper, config, and memory files

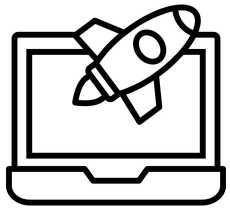




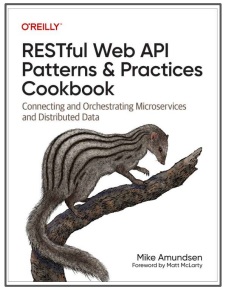
The Future of HyperLANG



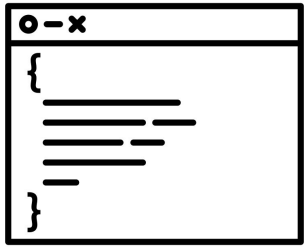
The Future of HyperLANG



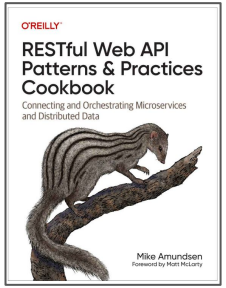
- Current status: Proof of concept
  - Next Level: Product-level work
- Key features to add
  - More ways to code/embed HyperLANG
- Expand Plug-In library
  - Beyond formats to domains



Proof of Concept

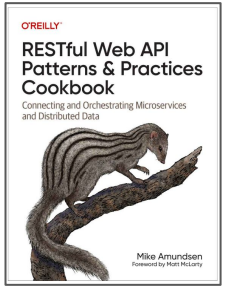


- Harden the code
- Improve error-handling
- Increase modularization



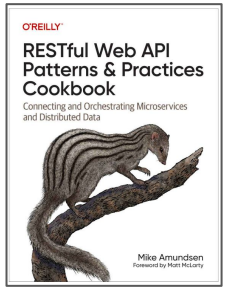
Key Features to Add

- Implement a module/library for inline support
- Publish a document object model (DOM)
- Improve flow control (IF-THEN, REPEAT, etc.)

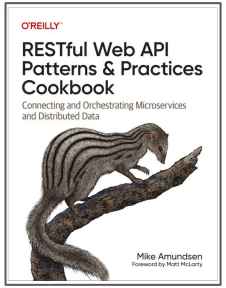


Expand Plug-In Library

- Improve plug-in support internally
- Continue format-driven efforts (HTML, HAL, Cj, SIREN, etc.)
- Add domain-centric plug-ins (PSD2, ACORD, HL7, etc.)

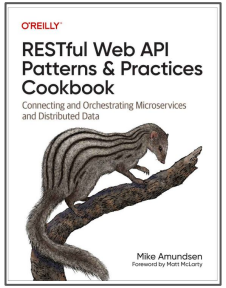


And So…


O'REILLY
RESTful Web API Patterns & Practices Cookbook
Connecting and Orchestrating Microservices and Distributed Data
Mike Amundsen
Foreword by Matt McLarty

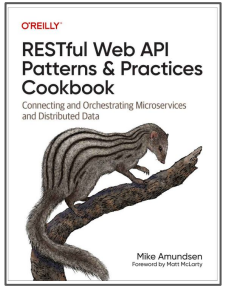
Programming the Web …

- Welcome to HyperWORLD
  - DSL programming platform



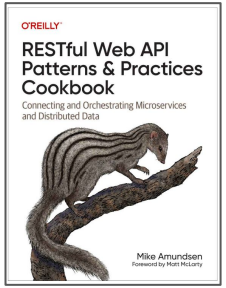
Programming the Web …

- Welcome to HyperWORLD
  - DSL programming platform
- HyperLANG
  - Read/Write for all HTTP services



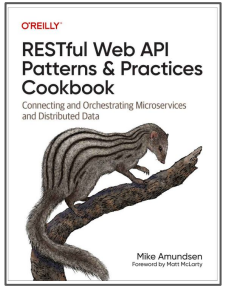
Programming the Web …

- Welcome to HyperWORLD
  - DSL programming platform
- HyperLANG
  - Read/Write for all HTTP services
- HyperCLI
  - Security, plug-ins, and scripting

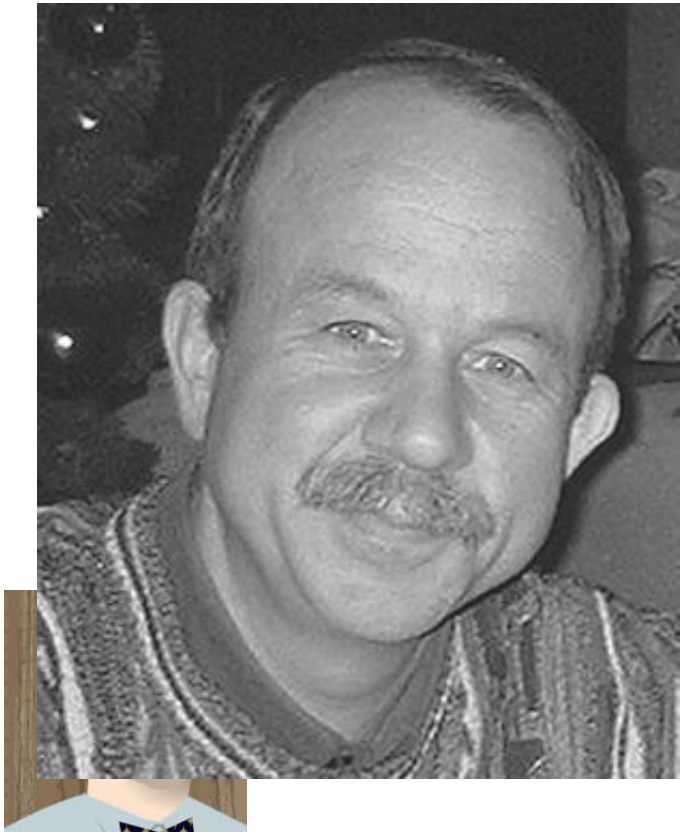


Programming the Web …

- Welcome to HyperWORLD
  - DSL programming platform
- HyperLANG
  - Read/Write for all HTTP services
- HyperCLI
  - Security, plug-ins, and scripting
- The Future of HyperLANG
  - More npm modules, other languages, ec.

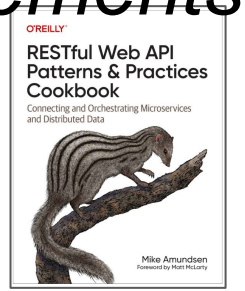


The Best Software Architecture



"The best software architecture 'knows' what changes often and makes that easy."

- Paul Clements



Programming the Web with HyperLANG and HyperCLI

Mike Amundsen
@mamund

goto;

Don’t forget to
vote for this session
in the **GOTO Guide app**