NorthwestOracleUsersGroup

Hitch your wagon to a star! Integrating Oracle Fusion Apps using Go!

The Hackett Group
World Class Defined and Enabled

NWOUG 2019 Portland Training Day
October 22, 2019

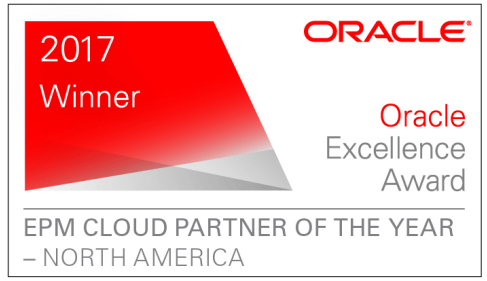
Prepared by Jeff Kayser
The Hackett Group

The Hackett Group - Overview

- Strategic Advisory and Technology Consulting Firm
- The Hackett Group has completed more than 16,500 benchmarking studies with major corporations and government agencies
  - 93% of the Dow Jones Industrials
  - 89% of the Fortune 100
  - 83% of the DAX 30
- Intellectual property-based strategic consultancy
- Best practices digital transformation
  - Robotic Process Automation (RPA)
  - Enterprise cloud application implementations
  - Enterprise analytics
  - Working Capital Management
  - Global Business Services

The Hackett Group - Oracle Awards, Certifications and Cloud Status

- Cloud Excellence Implementer
- Cloud Partner: Cloud Premier - North America
- Cloud Marketplace: Applications
- Cloud Marketplace: Services
- Specializations
- Oracle Excellence Award 2018 Specialized Partner of the Year: SaaS Innovation Solution of the Year - North America
- Oracle Excellence Award 2017 Specialized Partner of the Year: EPM Cloud - Global
- Oracle Excellence Award 2017 Specialized Partner of the Year: EPM Cloud - North America



Cloud Excellence Implementer

This partner has demonstrated ongoing expertise, a successful track record, and superior customer satisfaction in Oracle Cloud implementations.

Learn more about Cloud Excellence Implementers

Enterprise Performance Management

Enterprise Planning and Budgeting (view authorized geographies)

Financial Consolidation and Close (view authorized geographies)

Enterprise Resource Planning

Financial Management - General Ledger (view authorized geographies)

Human Capital Management

Benefit (view authorized geographies)

Human Resources (Core HR) (view authorized geographies)

Payroll (view authorized geographies)

Talent Management (view authorized geographies)

Supply Chain Management

Procurement (view authorized geographies)

8 CEI Certifications

The Hackett Group

ORACLE Platinum Partner
Cloud Premier
North America

Hackett Cloud Experience

Cloud Premier

95+	Oracle EPM Cloud
30+	Oracle ERP Cloud
25+	Oracle HCM Cloud

The Hackett Group

Jeff Kayser

- Oracle E-Business Suite Administrator for 30 years now.
  - (Remember RDBMS version 5? EBS Release 6? Those were the days).
- Like to stay on cutting edge of new technologies, including Google's Go language.
- Special expertise in security.
- Credited in Oracle Critical Patch Update Advisory notes for discovering and reporting security vulnerabilities to Oracle.
  - CERT Vulnerability Note VU#826463
  - Potential Logging of E-Business Suite Passwords (Doc ID 1579709.1)
  - Oracle Critical Patch Update Advisory - July 2013
  - Oracle Critical Patch Update Advisory - October 2008
  - Oracle Critical Patch Update - October 2006
- Work at companies in the Portland metro area, including:
  - Oregon Health and Sciences University
  - Nike (1st 10.7 NCA installation in the world. Woo-hoo!)

Agenda

- Why integrate?
- How to integrate
- An overview of Go
- Web Service Calls with Go
- Real World Examples using Go for Cloud Data Integration
- A few Go odds and ends
- Go with AWS.
- Q & A

Why Integrate?

Have you ever seen a system that was perfect, complete, and needed no integrations?

- Neither have I.
- Moving to the cloud doesn't make integrations go away.
- It just make integrating … different.

Developing for Oracle Cloud Apps

- Fusion Applications Developer Relations
  - https://blogs.oracle.com/fadevrel/
- Direct Database Access In Fusion Cloud Applications
  - https://blogs.oracle.com/fadevrel/direct-data-access-in-fusion-cloud-applications
  - https://blogs.oracle.com/fadevrel/logical-and-relational-data-model-diagrams-for-fusion-applications
- The bad news:
  - No direct database access. No SQL*Plus, No SQL*Developer, etc.
  - No direct server access. No PuTTY.
- The good news: You can use:
  - Web Service calls (mostly SOAP, limited REST and ATOM support)
  - BI Publisher (brush up on your BI Publisher skills)
    - SELECT and PL/SQL blocks that do SELECTs
  - File-based loaders (spreadsheet loaders, HCM Data Loader, etc).

How to Integrate

Lots of integration platforms to choose from ($$$)

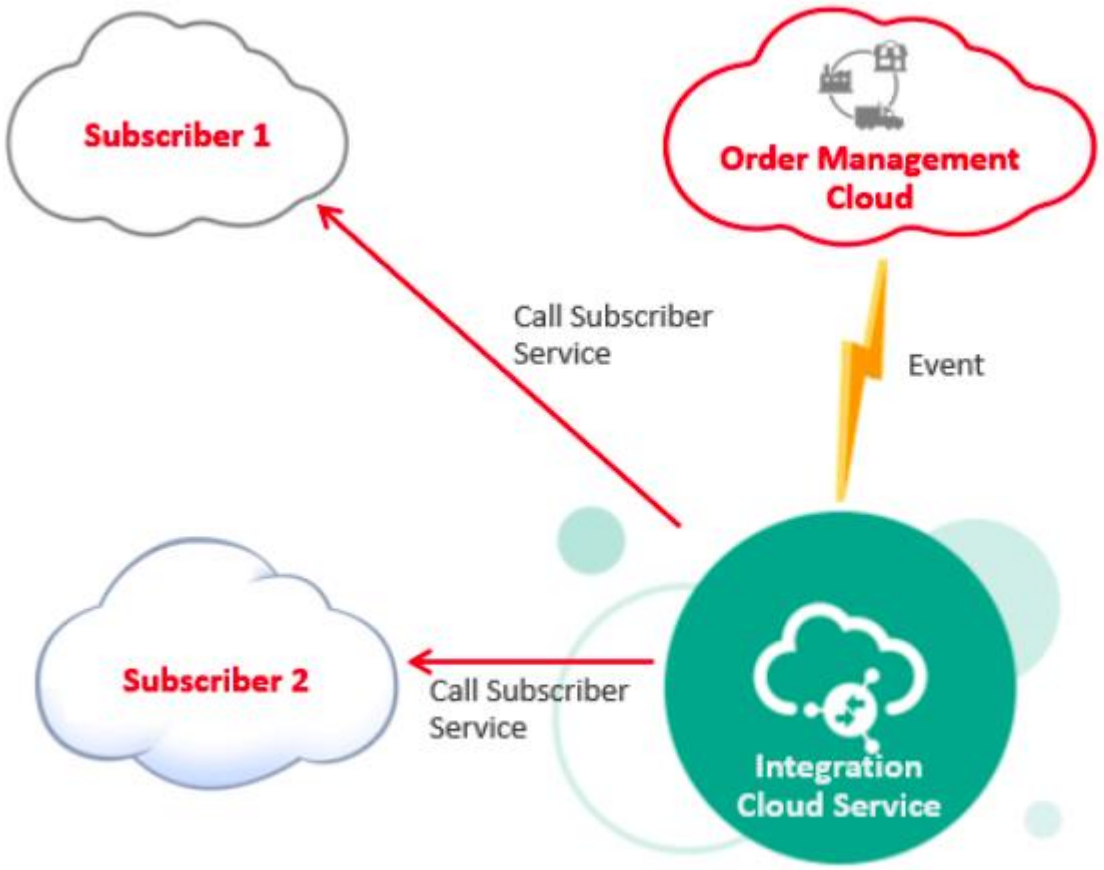
Figure 1. Magic Quadrant for Data Integration Tools

Source: Gartner (July 2018)

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

…Including Oracle Integration Cloud Service ($$$)



The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Is there anything… Better? Cheaper? Faster?

The Hackett Group

ORACLE Platinum Partner
Cloud Premier
North America

Cloud technology… a sea-change towards web services…

- What language is good for calling web services?

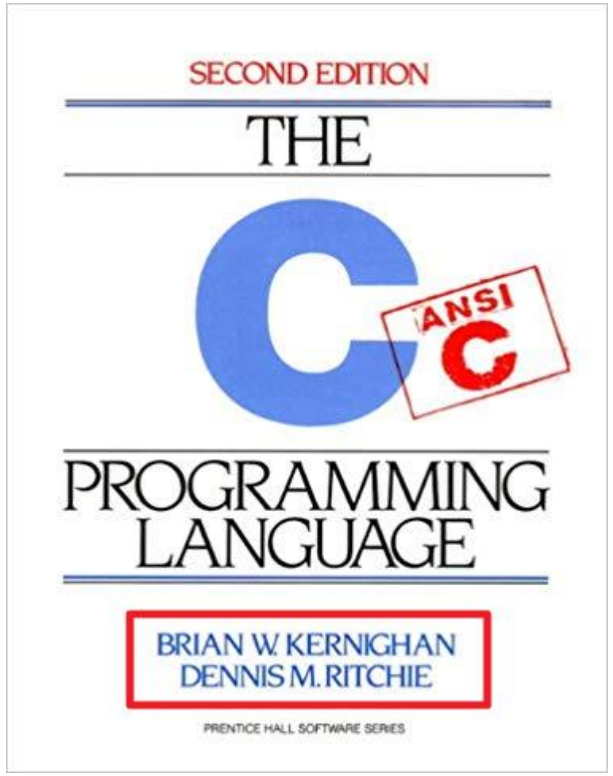
The Hackett Group

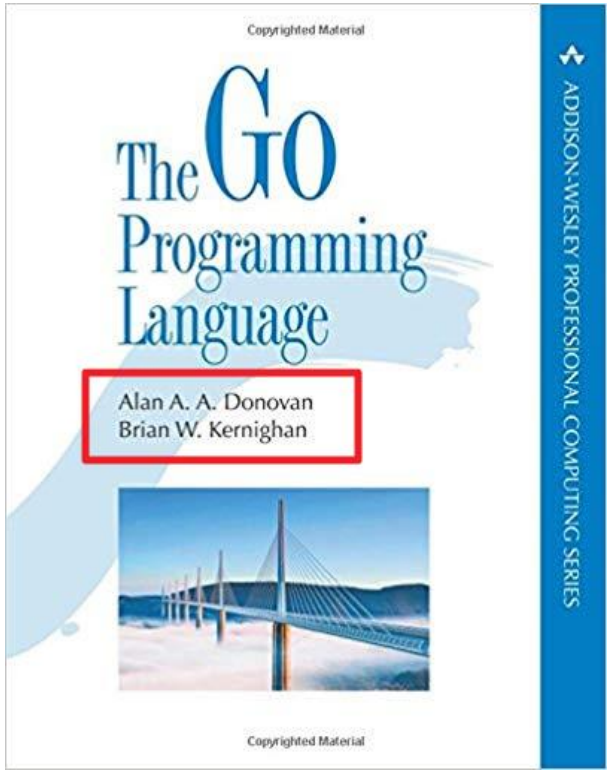
ORACLE Platinum Partner Cloud Premier North America

An overview of Go

Google's Go Language! (aka Golang)

- https://golang.org/
- Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.
- The original C programming language: The new C programming language:





Comparing Go to C wasn't quite fair…

- Go syntax is like C, but…
- Go has implicit memory management with a fast garbage collector.
- Go has static typing implemented by a FAST compiler.
- Go compiles to machine code (no intermediate byte code).
- Go generates binaries that are self-contained (they include the runtime).
- Go has a GREAT standard library. Lots of pre-built functionality you can use right away!
  - https://golang.org/pkg/
- Go is cross-platform (e.g. on Linux, you can generate a Windows binary!).
- Go has GREAT support for the web (hey, it originated at Google).
- Go is aggressive in supporting emerging web standards (HTTP/2, TLS 1.3, etc.)
- Go has built-in language primitives and library support for concurrency.
- Go is sort of object oriented (you can attach functions to structures), but…
- Go does not support inheritance (you can use composition instead).
- Go has interfaces that are implicitly satisfied (no need to specify interfaces explicitly).

And Go has a really cute mascot… the Gopher…

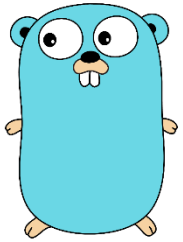
Lots of Companies are using Go (I ran out of room ☺)

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Go also helps power the Internet

GOLANG



kubernetes

docker

HashiCorp Terraform

HEROKU

CLOUDFLARE

elasticsearch

Istio

Connect, secure, control, and observe services.

etcd

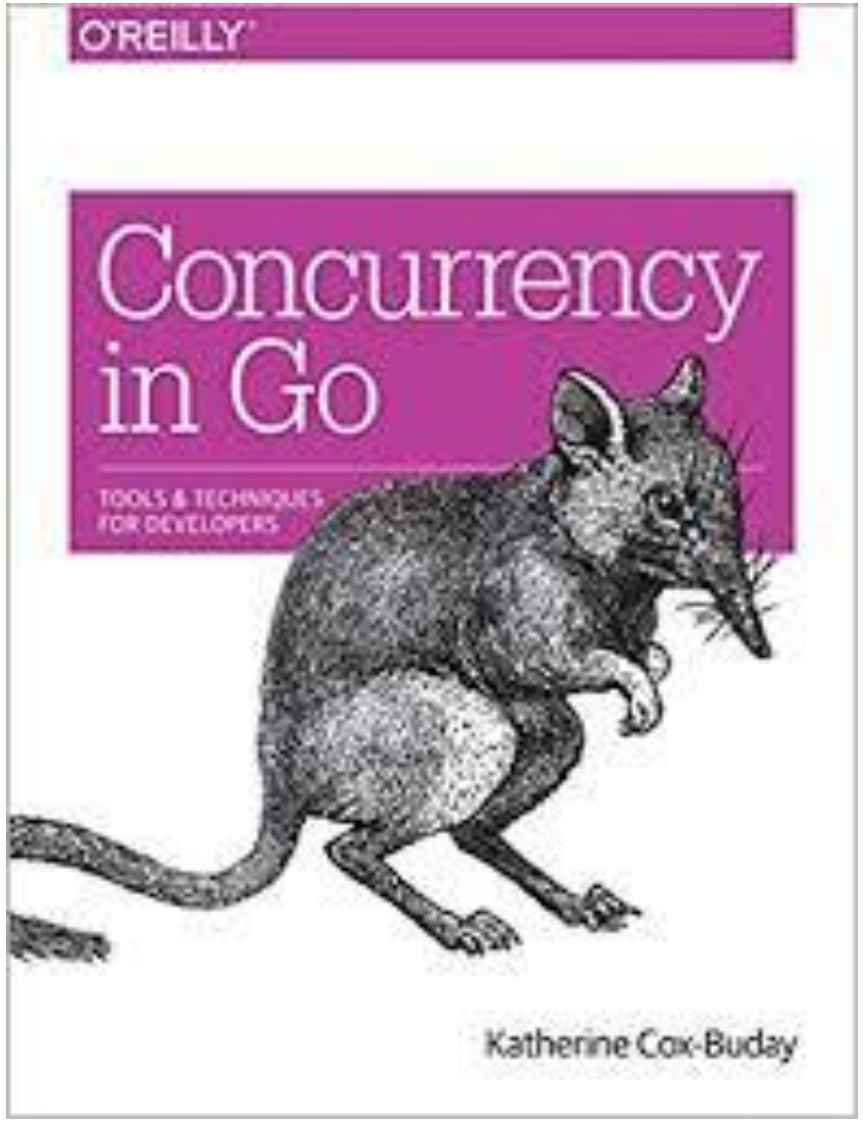
InfluxDB

mongoDB

Couchbase

Cockroach DB

Why Go? It has INCREDIBLE support for Concurrency!



The Hackett Group

ORACLE Platinum Partner
Cloud Premier
North America

Google this: “Why we rewrote in Golang”

- **Iron I/O: How We Went from 30 Servers to 2: Go (from Ruby on Rails)**
  - https://blog.iron.io/how-we-went-from-30-servers-to-2-go/
- **Uber: Code Migration in Production: Rewriting the Sharding Layer of Uber’s Schemaless Datastore (from Python)**
  - https://eng.uber.com/schemaless-rewrite/
- **Getstream I/O: Why we switched from Python to Go (30 times faster than Python)**
  - https://getstream.io/blog/switched-python-go/
- **Parseplatform: How We Moved Our API From Ruby to Go and Saved Our Sanity**
  - https://blog.parseplatform.org/learn/how-we-moved-our-api-from-ruby-to-go-and-saved-our-sanity/
- **Movio.co: Making The Move From Scala To Go, And Why We’re Not Going Back**
  - https://movio.co/blog/migrate-Scala-to-Go/

- **And some of my own experience…**
  - https://github.com/jeffkayser2/learningGo/wiki

IEEE Spectrum: The Top Programming Languages 2018
Languages that are trending…

Language Types (click to hide)

Web | Mobile | Enterprise | Embedded

Language Rank	Types	Trending Ranking
1. Python	Web, Enterprise, Embedded	100.0
2. C++	Mobile, Enterprise, Embedded	96.7
3. Java	Web, Mobile, Enterprise	94.6
4. C	Mobile, Enterprise, Embedded	93.7
5. Go	Web, Enterprise	85.5
6. JavaScript	Web, Mobile	80.8

Without further ado…
The Go version of Hello, World!

```
package main
import (
    "fmt"
)
func main() {
    fmt.Printf("Hello, world!\n")
}
```

Go compiles into a statically-linked stand-alone binary, that includes the Go runtime environment.

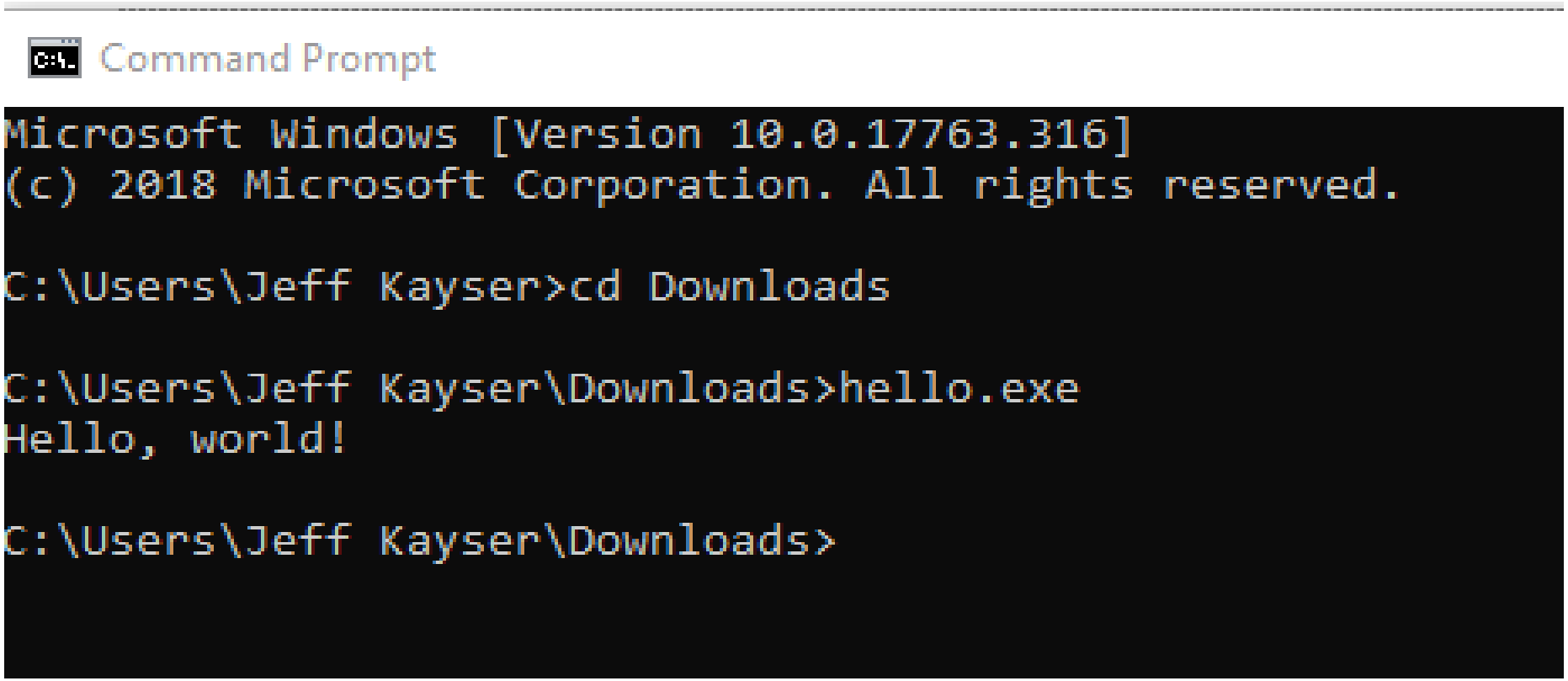
```
$ go build hello.go
$ ls -l
total 1880
-rwxrwxr-x 1 jkayser jkayser 1919441 Feb 18 14:24 hello
-rw-rw-r-- 1 jkayser jkayser      82 Feb 18 14:22 hello.go
$ file hello
hello: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
statically linked, not stripped
$ ./hello
Hello, world!
$
```

Have you ever made a Windows *.EXE file on Linux?

```
$ uname -a
Linux oel7latest.jkayser.pvt 4.14.35-1844.2.5.el7uek.x86_64 #2 SMP Mon
Feb 4 18:24:45 PST 2019 x86_64 x86_64 x86_64 GNU/Linux
$ GOOS=windows GOARCH=amd64 go build hello.go
$ ls -l
total 3808
-rwxrwxr-x 1 jkayser jkayser 1919441 Feb 18 14:24 hello
-rwxrwxr-x 1 jkayser jkayser 1972224 Feb 18 14:27 hello.exe
-rw-rw-r-- 1 jkayser jkayser      82 Feb 18 14:22 hello.go
$ file hello.exe
hello.exe: PE32+ executable (console) x86-64 (stripped to external PDB),
for MS Windows
$
```

Copy the binary to Windows, and run it!

- **Copy the Go binary to the server, and run it! NOTHING else needed.**
- The Go runtime environment is built into the Go binary.



No more Dependency or DLL Hell !!!

- No Java JDKs, JREs or JAR files.
- No C or C++ runtime libraries.
- No Python interpreter or libraries.
- No **LAMP** stack (Linux, Apache, MySQL, PHP) for PHP.
- No Windows DLLs.

- Go has its own HTML templating engine; no need for PHP
- Go can be its own web server; no need for Apache or NGINX
- Go can talk to lots of external databases (Oracle, SQL Server, MySQL, PostgreSQL, etc.)
- How about the **LOG** stack (Linux, Oracle, Go)?!? (You still need a database).

Platforms supported by Go out of the box

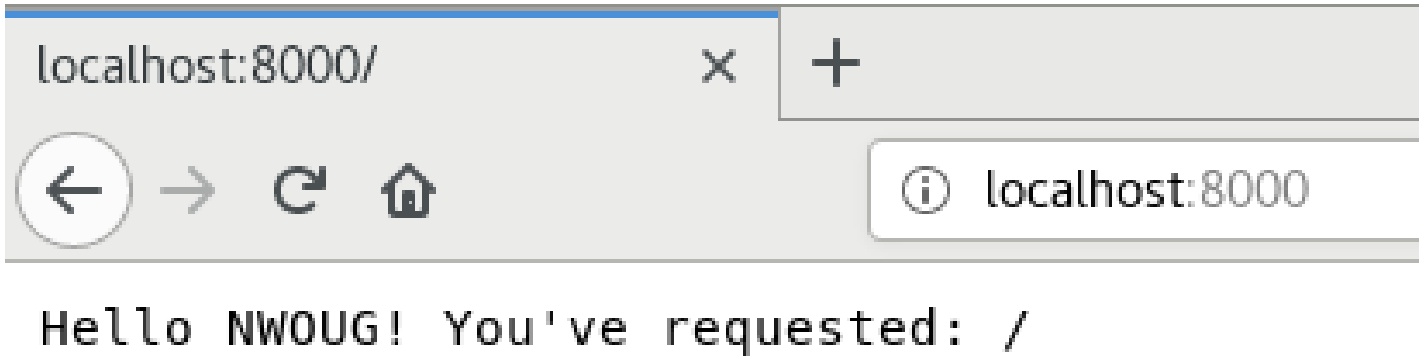
- android/386
- darwin/386
- darwin/amd64
- darwin/arm
- darwin/arm64
- dragonfly/amd64
- freebsd/386
- freebsd/amd64
- freebsd/arm
- js/wasm
- linux/386
- linux/amd64
- linux/arm
- linux/arm64
- linux/ppc64
- linux/ppc64le
- linux/mips
- linux/mipsle
- linux/mips64
- linux/mips64le
- linux/s390x
- nacl/386
- nacl/amd64p32
- nacl/arm
- netbsd/386
- netbsd/amd64
- netbsd/arm
- openbsd/386
- openbsd/amd64
- openbsd/arm
- plan9/386
- plan9/amd64
- plan9/arm
- solaris/amd64
- windows/386
- windows/amd64
- windows/arm

Make a simple web server in Go…

```
package main
import (
	"fmt"
	"net/http"
)
func main() {
	http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
		fmt.Fprintf(w, "Hello NWOUG! You've requested: %s\n",
			r.URL.Path)
	})
	http.ListenAndServe(":8000", nil)
}
```

Run our Go web server

```
$ go run server.go
```



- **So you want to expose Go on the Internet (Cloudflare's recommendations)**
  - https://blog.cloudflare.com/exposing-go-on-the-internet/

Note about statically linked binaries…
See: https://github.com/golang/go/issues/30419

```
$ go build server.go
$ file server
server:    ELF 64-bit LSB executable, x86-64, version 1
(SYSV), dynamically linked (uses shared libs), not stripped
$ rm server
$ go build -tags "netgo osusergo" server.go
$ file server
server:    ELF 64-bit LSB executable, x86-64, version 1
(SYSV), statically linked, not stripped
$
```

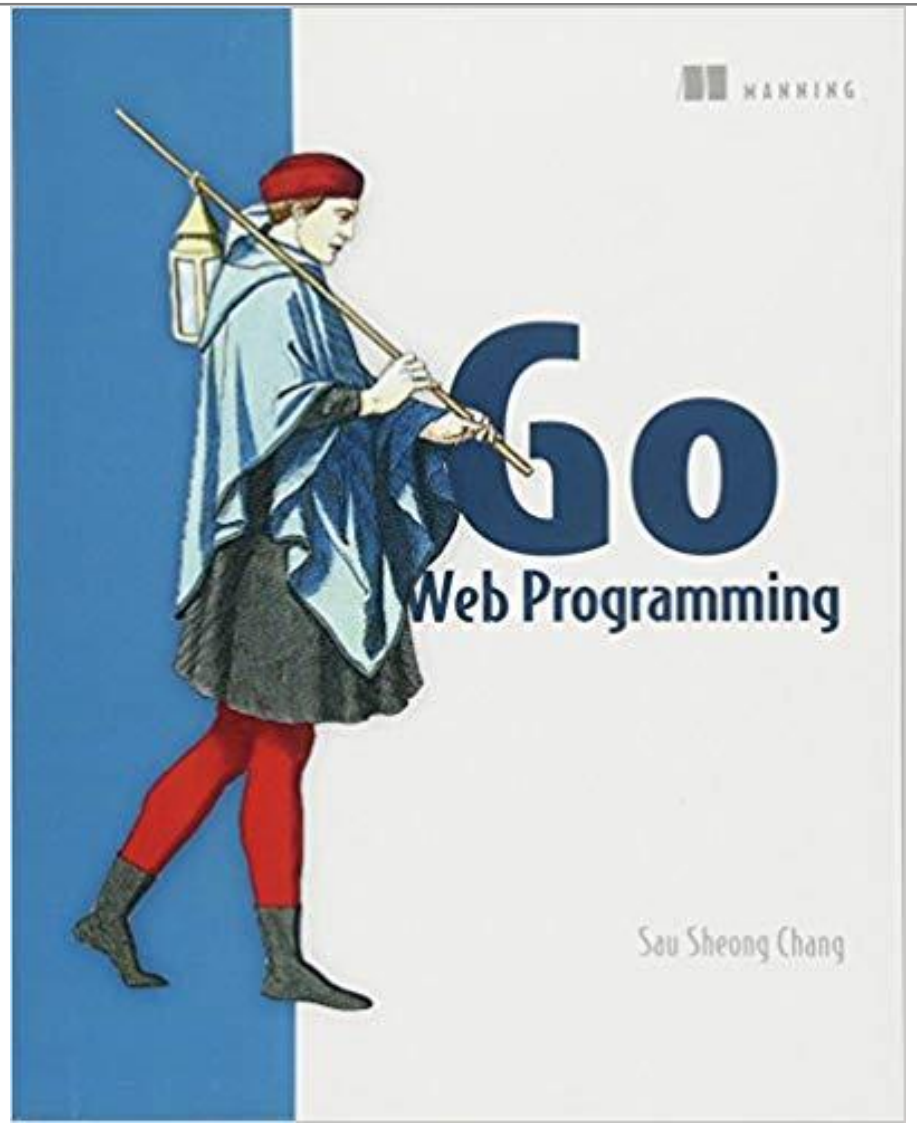
Make a simple web client in Go…

```
package main
import (
	"fmt"
	"io/ioutil"
	"log"
	"net/http"
)
func main() {
	var c http.Client
	resp, err := c.Get("http://localhost:8000"); if err != nil {
		log.Fatal(err.Error())
	}
	body, err := ioutil.ReadAll(resp.Body); if err != nil {
		log.Fatal(err.Error())
	}
	fmt.Printf("Response: %s\n", body)
}
```

Run our Go web client

```
$ go run client.go
2019/02/18 15:15:52 Get http://localhost:8000: dial tcp
[::1]:8000: connect: connection refused
exit status 1
$ go run server.go &
[1] 24136
$ go run client.go
Response: Hello NWOUG! You've requested: /


$
```

Lots more about Go Web Programming…



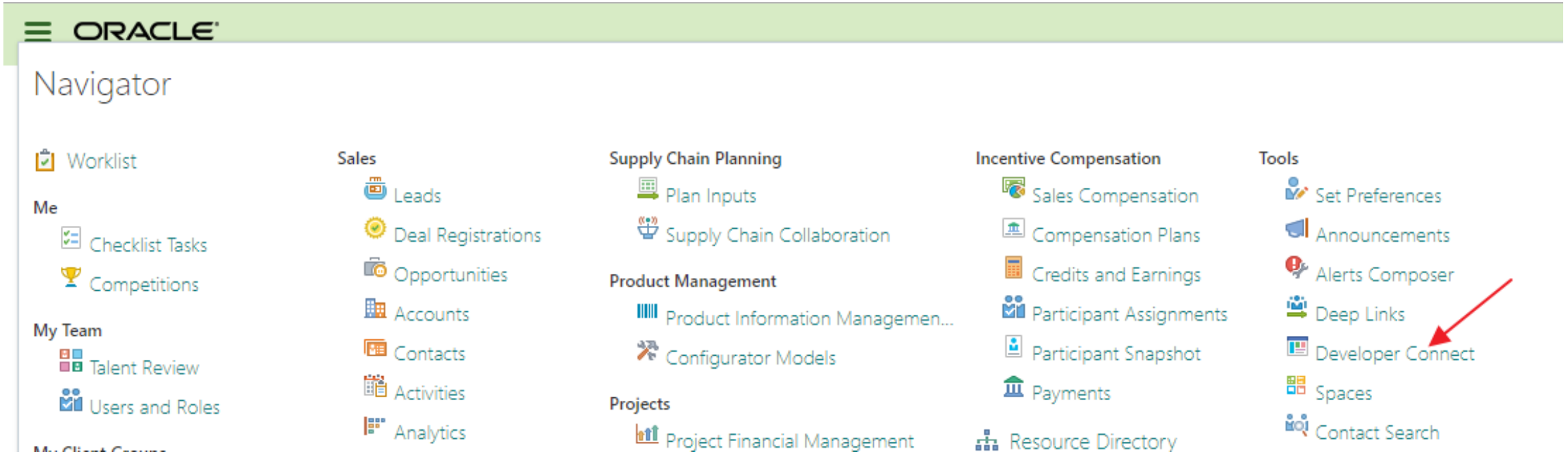
The Hackett Group

ORACLE Platinum Partner
Cloud Premier
North America

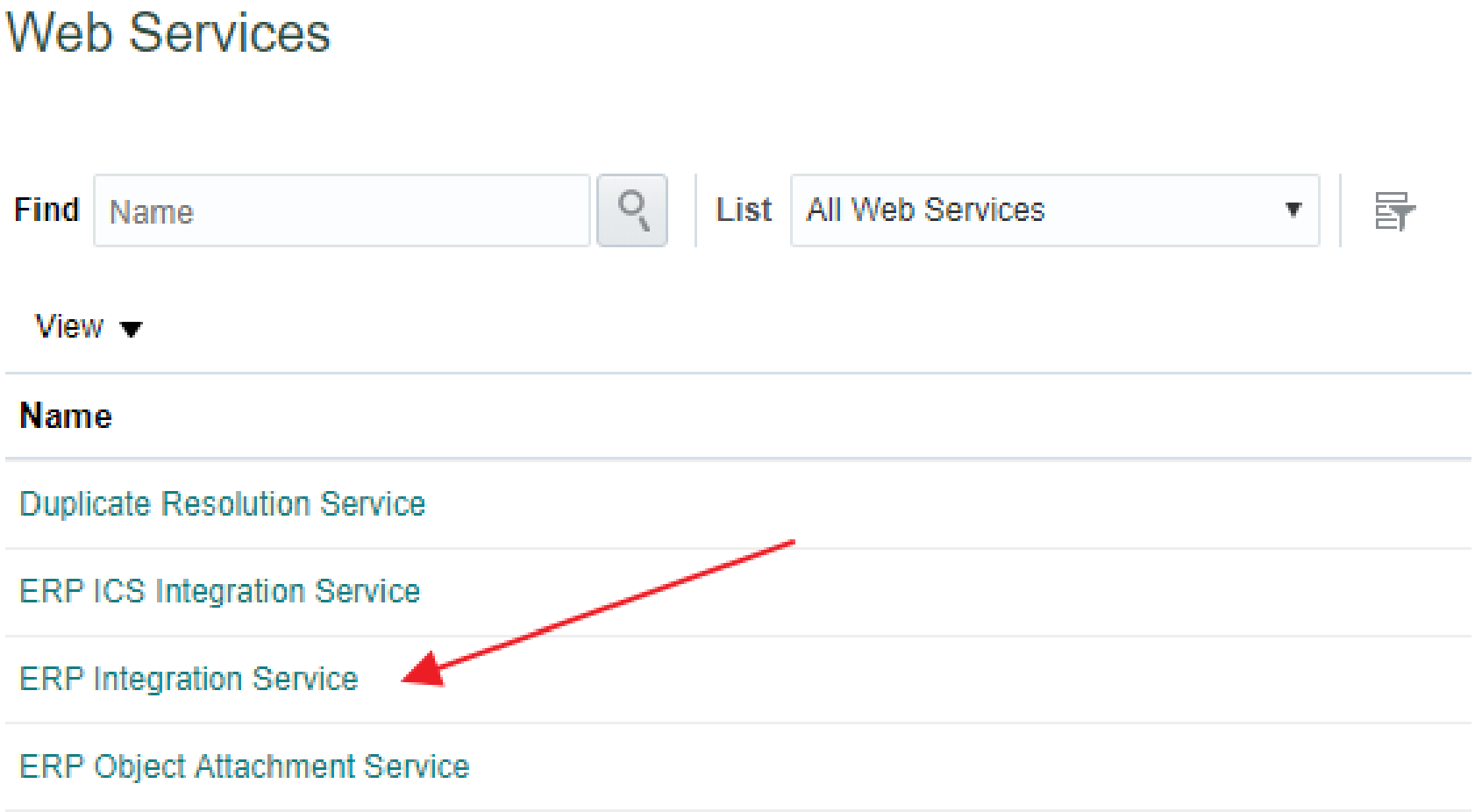
Web Service calls with Go

Most of the web service calls for Oracle Cloud Apps are SOAP. How do you do SOAP web service calls?

- First, get the WSDL:



Choose the web service you want to use…



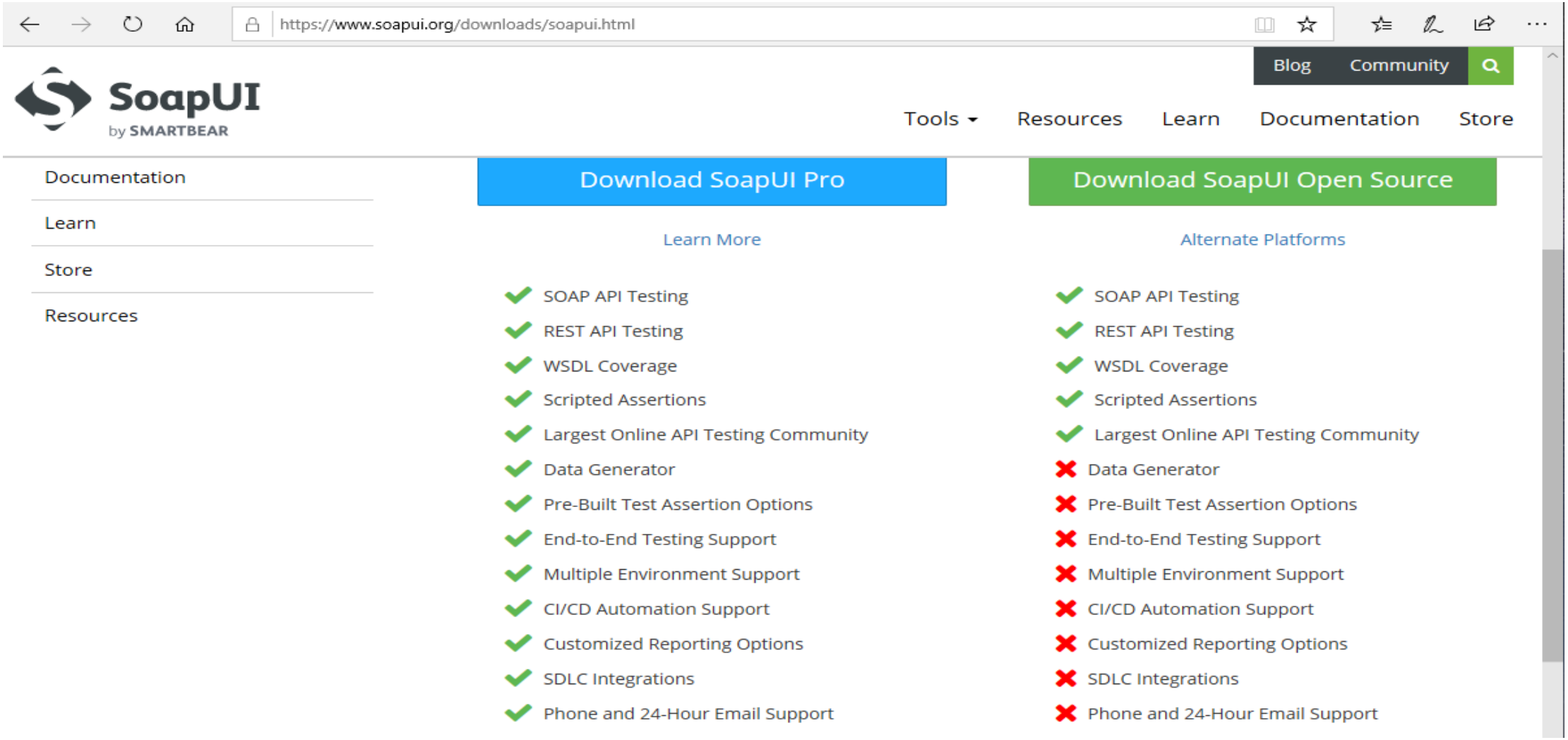
The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

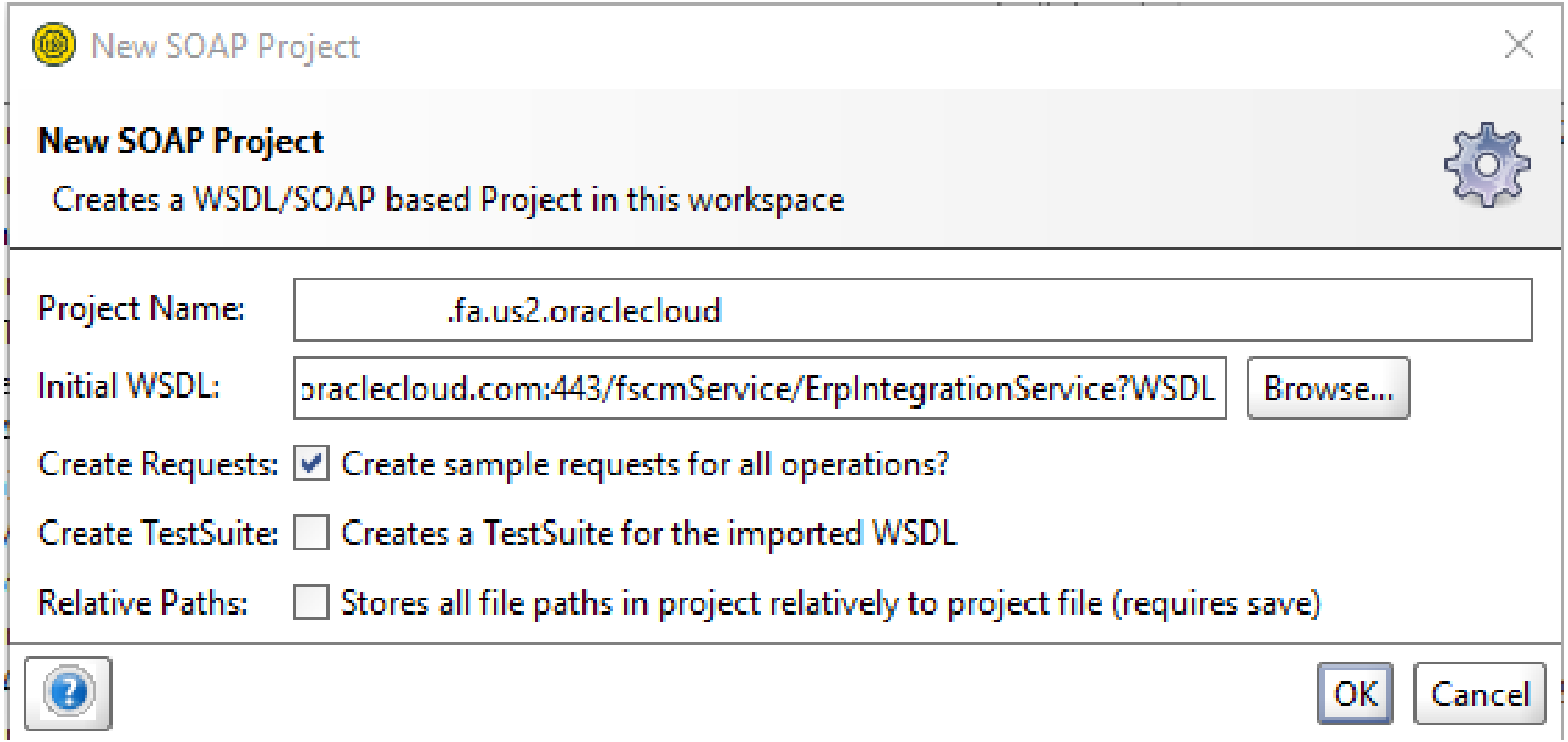
Then get the endpoint and the WSDL URLs…

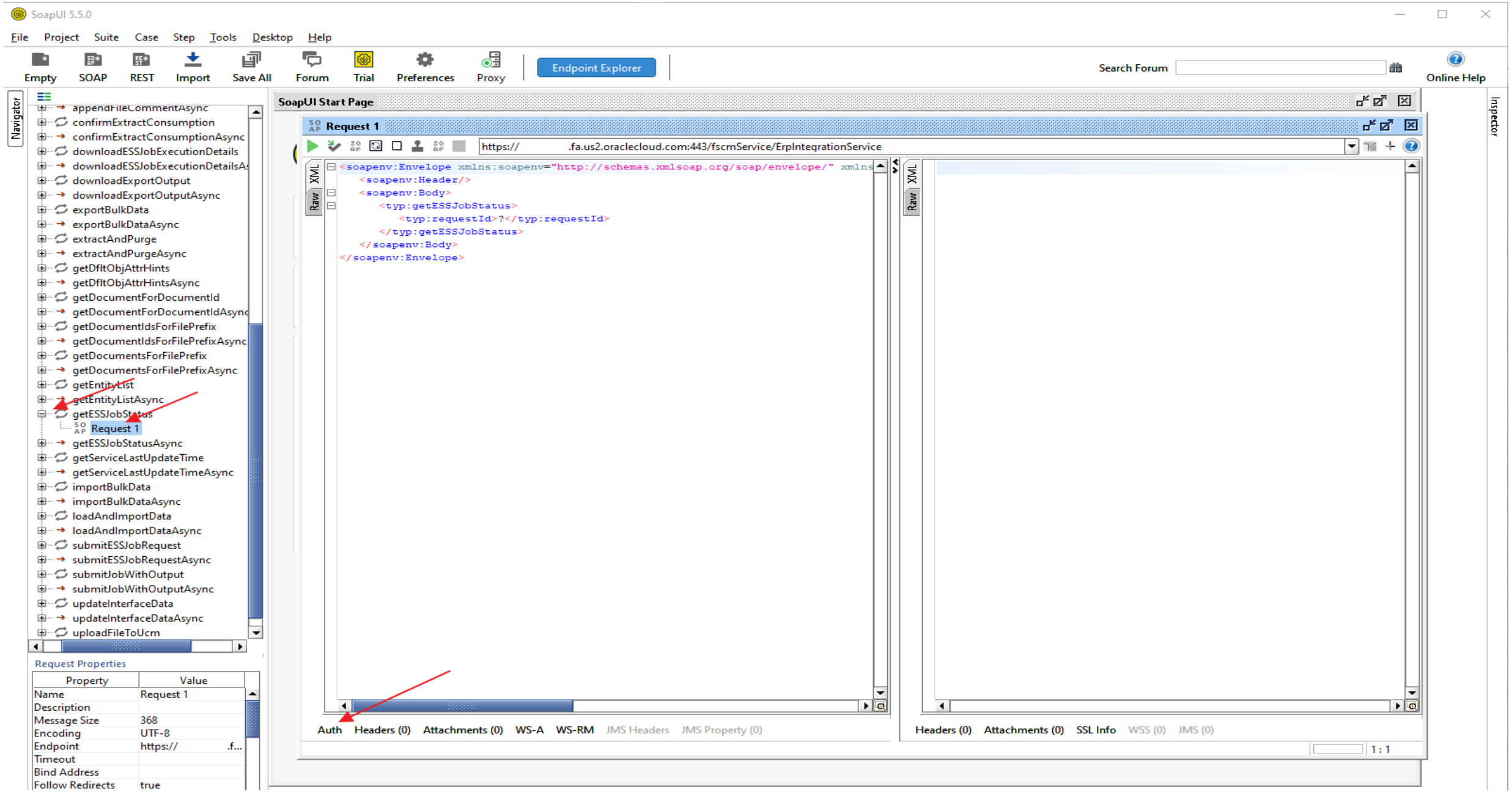
Name	ERP Integration Service
Business Object	
Life Cycle	Active
QName	{http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/}ErpIntegrationService
End Point URL	https:// .fa.us2.oraclecloud.com:443/fscmService/ErpIntegrationService
WSDL File	https:// .fa.us2.oraclecloud.com:443/fscmService/ErpIntegrationService?WSDL
Security Policy	oracle/wss11_saml_or_username_token_with_message_protection_service_policy
Description	provides external web service operations for ERP integration scenarios.

SoapUI is your friend.
https://www.soapui.org/downloads/soapui.html



Create new SOAP project in SoapUI

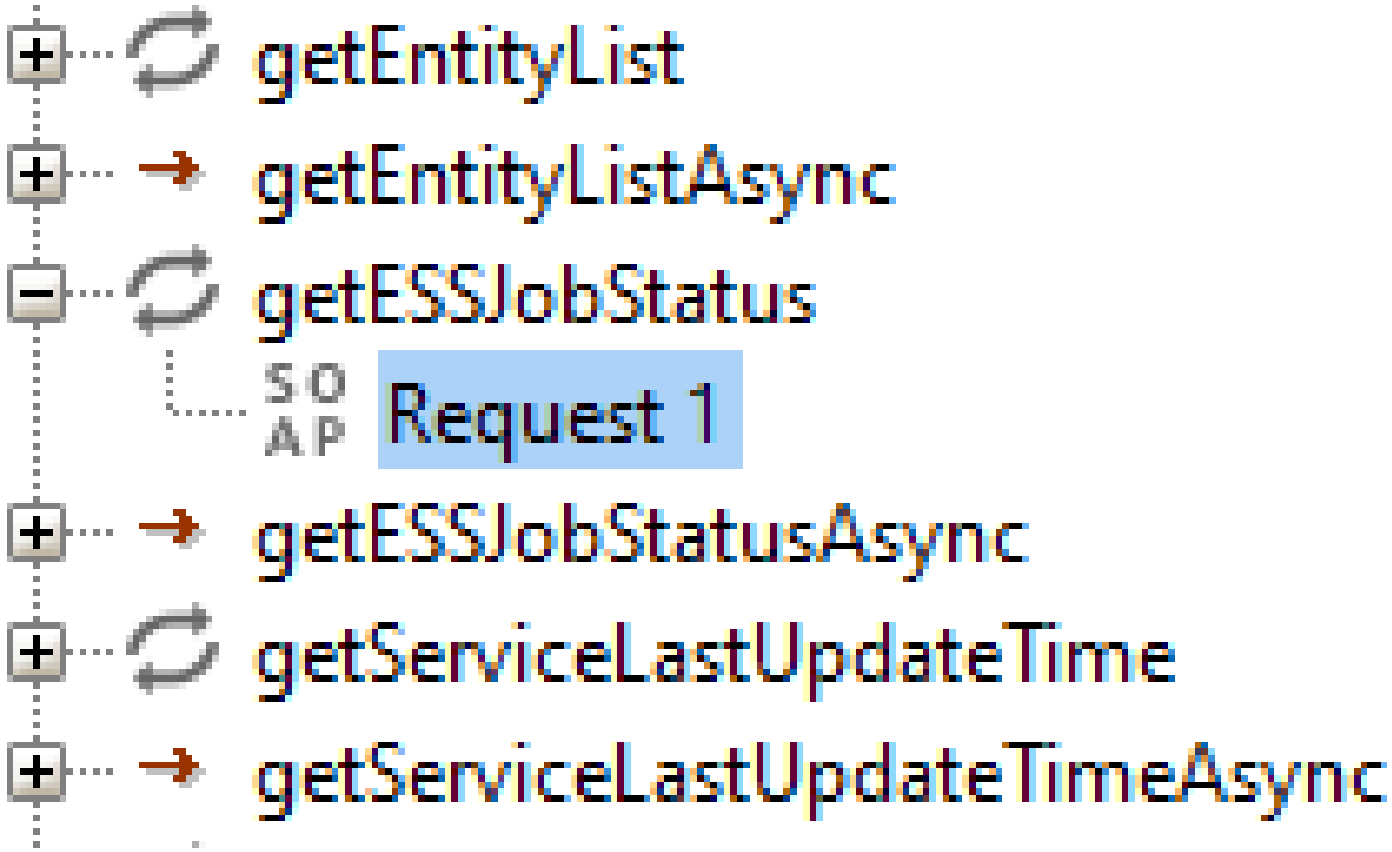




The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

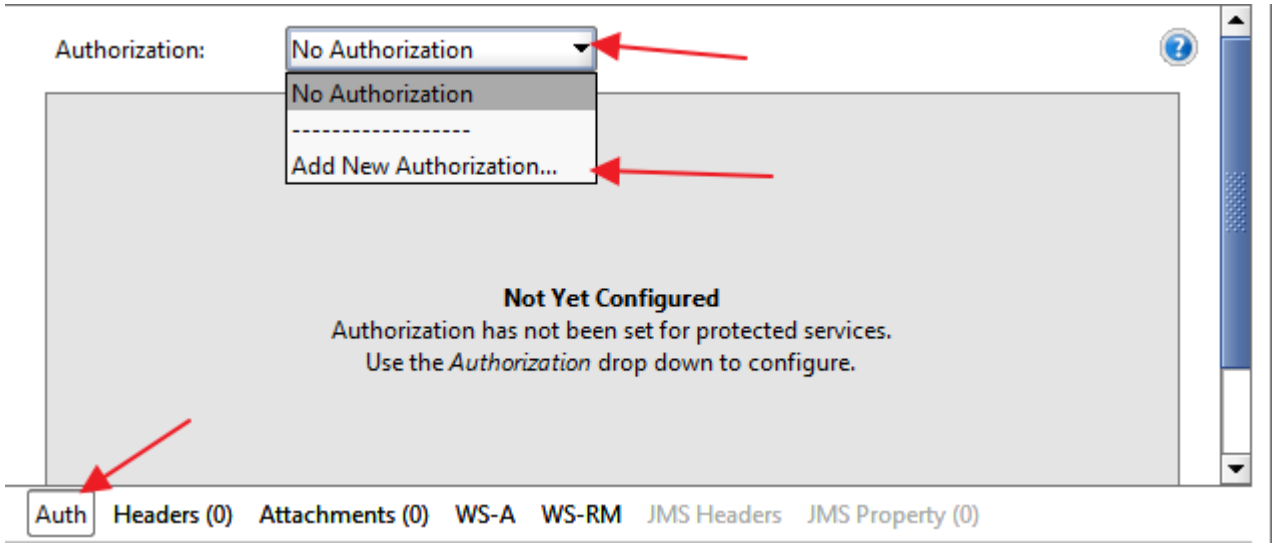
Select the SOAP operation you want

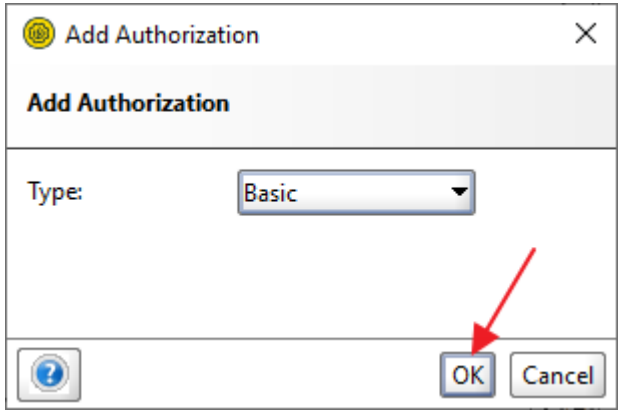


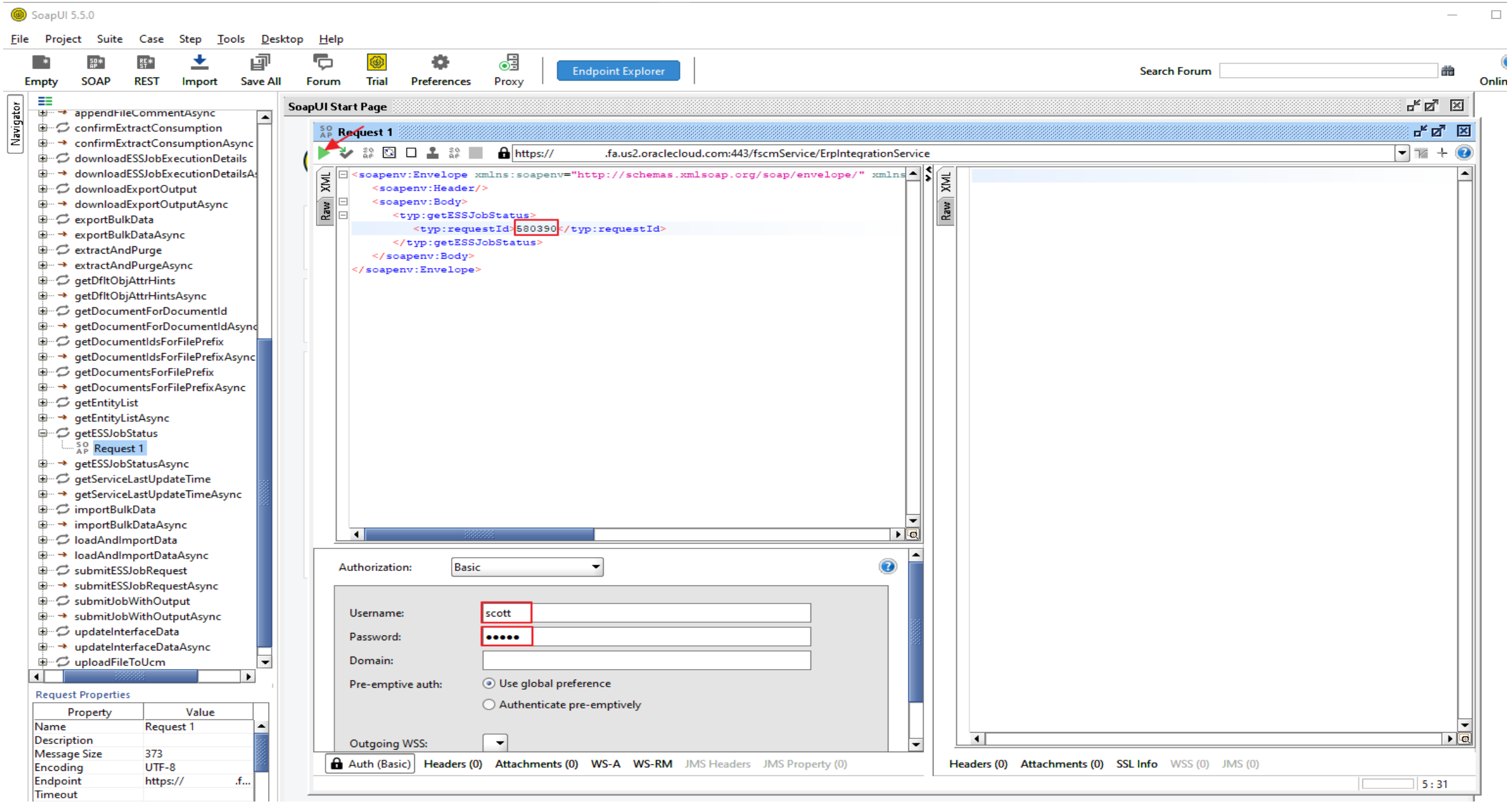
The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Add authentication…

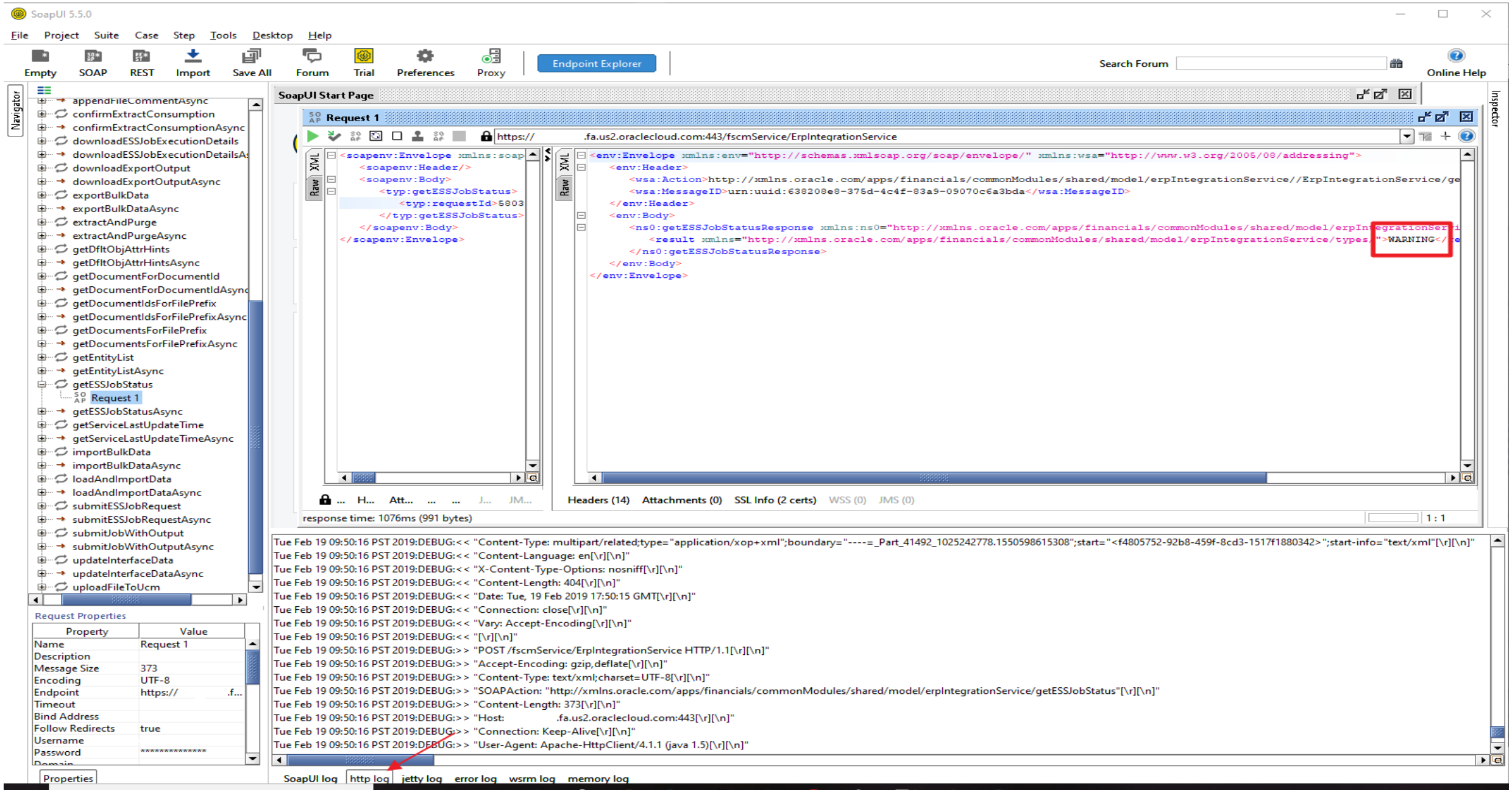






The Hackett Group

ORACLE Platinum Partner Cloud Premier North America



SOAP request (get status for ESS job 580390)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/types/">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getESSJobStatus>
         <typ:requestId>580390</typ:requestId>
      </typ:getESSJobStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

Request.go: A function to create that SOAP request

```
package main

func SoapRequest ( requestId string ) ( string ) {

      return `<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/financials/commonModules/shared/m
odel/erpIntegrationService/types/">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getESSJobStatus>
         <typ:requestId>` + requestId + `</typ:requestId>
      </typ:getESSJobStatus>
   </soapenv:Body>
</soapenv:Envelope>`

}
```

The SoapUI http log shows you what other stuff needs to be added to the HTTP request header

- "POST /fscmService/ErpIntegrationService HTTP/1.1[\r][\n]"
- "Accept-Encoding: gzip,deflate[\r][\n]"
- "Content-Type: text/xml;charset=UTF-8[\r][\n]"
- "SOAPAction: "http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpIntegrationService/getESSJobStatus"[\r][\n]"
- "Content-Length: 373[\r][\n]"
- "Host: something.fa.us2.oraclecloud.com:443[\r][\n]"
- "Connection: Keep-Alive[\r][\n]"
- "User-Agent: Apache-HttpClient/4.1.1 (java 1.5)[\r][\n]"
- "Authorization: Basic {gibberish}[\r][\n]"

SOAP response for success (request 580390 has WARNING status)

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://www.w3.org/2005/08/addressing">
   <env:Header>
<wsa:Action>http://xmlns.oracle.com/apps/financials/commonModules/shared/model/er
pIntegrationService//ErpIntegrationService/getESSJobStatusResponse</wsa:Action>
      <wsa:MessageID>urn:uuid:638208e8-375d-4c4f-83a9-
09070c6a3bda</wsa:MessageID>
   </env:Header>
   <env:Body>
      <ns0:getESSJobStatusResponse
xmlns:ns0="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erp
IntegrationService/types/">
         <result
xmlns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpInte
grationService/types/">WARNING</result>
      </ns0:getESSJobStatusResponse>
   </env:Body>
</env:Envelope>
```

SOAP response for failure (really helpful error message… ☺)

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <env:Fault>
         <faultcode>env:Server</faultcode>
         <faultstring><![CDATA[JBO-FND:::FND_CMN_SYS_ERR: <MESSAGE><NUMBER>FND-2</NUMBER><TEXT>An error
occurred. Contact your help
desk.</TEXT><CAUSE></CAUSE><ACTION></ACTION><DETAILS></DETAILS><INCIDENT></INCIDENT></MESSAGE>]]></faults
tring>
         <detail>
            <tns:ServiceErrorMessage xmlns:tns="http://xmlns.oracle.com/adf/svc/errors/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
               <tns:code>FND:::FND_CMN_SYS_ERR</tns:code>
               <tns:message><![CDATA[JBO-FND:::FND_CMN_SYS_ERR: <MESSAGE><NUMBER>FND-2</NUMBER><TEXT>An
error occurred. Contact your help
desk.</TEXT><CAUSE></CAUSE><ACTION></ACTION><DETAILS></DETAILS><INCIDENT></INCIDENT></MESSAGE>]]></tns:me
ssage>
               <tns:severity>SEVERITY_ERROR</tns:severity>

<tns:exceptionClassName>oracle.apps.fnd.applcore.messages.ApplcoreException</tns:exceptionClassName>
            </tns:ServiceErrorMessage>
         </detail>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

The Go standard library has support for XML. Hundreds of other packages to choose from…

pe	Package pe implements access to PE (Microsoft Windows Portable Executable) files.
plan9obj	Package plan9obj implements access to Plan 9 a.out object files.
encoding	Package encoding defines interfaces shared by other packages that convert data to and from byte-level and textual representations.
ascii85	Package ascii85 implements the ascii85 data encoding as used in the btoa tool and Adobe's PostScript and PDF document formats.
asn1	Package asn1 implements parsing of DER-encoded ASN.1 data structures, as defined in ITU-T Rec X.690.
base32	Package base32 implements base32 encoding as specified by RFC 4648.
base64	Package base64 implements base64 encoding as specified by RFC 4648.
binary	Package binary implements simple translation between numbers and byte sequences and encoding and decoding of varints.
csv	Package csv reads and writes comma-separated values (CSV) files.
gob	Package gob manages streams of gobs - binary values exchanged between an Encoder (transmitter) and a Decoder (receiver).
hex	Package hex implements hexadecimal encoding and decoding.
json	Package json implements encoding and decoding of JSON as defined in RFC 7159.
pem	Package pem implements the PEM data encoding, which originated in Privacy Enhanced Mail.
xml	Package xml implements a simple XML 1.0 parser that understands XML name spaces.
errors	Package errors implements functions to manipulate errors.
expvar	Package expvar provides a standardized interface to public variables, such as operation counters in servers.
flag	Package flag implements command-line flag parsing.
fmt	Package fmt implements formatted I/O with functions analogous to C's printf and scanf.
go	

Parsing the SOAP response

- One way to parse the XML response is to read it into a Go struct using the xml.Unmarshal function. To do that, you need a receiving Go struct.
- How to generate a Go struct from XML
  - https://github.com/gnewton/chidley
  - https://github.com/miku/zek

- $ **`zek Success.xml > Success.go`**
- $ **`vi Success.go`**

Success.go: The struct we need to parse a successful SOAP response.

```
package main
import (
        "encoding/xml"
)
// Envelope was generated 2019-02-21 21:32:35 by jkayser on
oel7latest.jkayser.pvt.
type Success struct {
. . .
        Body struct {
                Text                    string `xml:",chardata"`
                GetESSJobStatusResponse struct {
. . .
                        Result struct {
                                Text  string `xml:",chardata"`
                                Xmlns string `xml:"xmlns,attr"`
                        } `xml:"result"`
                } `xml:"getESSJobStatusResponse"`
        } `xml:"Body"`
}
```

Let’s write a Go program that does a SOAP call to get the status of an ESS job in Oracle Cloud Financials

- Warning: some Go code ahead

- But hey: It’s a working example of how to make SOAP calls with Go.

getESSJobStatus.go #1. Go libraries we use.

```
package main

import (
    "encoding/xml"
    "fmt"
    "io"
    "io/ioutil"
    "log"
    "mime"
    "mime/multipart"
    "net/http"
    "net/http/httputil"
    "net/url"
    "os"
    "strconv"
    "strings"
)
```

getESSJobStatus.go #2. Get some values from the environment

```
func main() {

    var debug = true

    // Get some values from environment
    endpoint := os.Getenv("CLOUD_ENDPOINT")
    username := os.Getenv("CLOUD_USERNAME")
    password := os.Getenv("CLOUD_PASSWORD")

    if endpoint == "" {
        log.Fatal("Environment variable CLOUD_ENDPOINT required")
    }
    if username == "" {
        log.Fatal("Environment variable CLOUD_USERNAME required")
    }
    if password == "" {
        log.Fatal("Environment variable CLOUD_PASSWORD required")
    }
```

getESSJobStatus.go #3. Verify the endpoint, and get request Id from command line.

```
// Make sure endpoint is valid
u, err := url.Parse(endpoint)
if err != nil {
    log.Fatal(err.Error())
}

// Get ESS Request Id from  the command line
var requestId string
if len(os.Args) > 1 {
    requestId = os.Args[1]
} else {
    log.Fatal("usage: getESSJobStatus {requestId}")
}
```

getESSJobStatus.go #4. Create the SOAP request, and a new HTTP POST request.

```
// Create the SOAP request
request := SoapRequest( requestId )

// Create new HTTP POST request
req, err := http.NewRequest("POST", endpoint,
    strings.NewReader(request))
req.Method = "POST"
if err != nil {
    log.Fatal(err.Error())
}
```

getESSJobStatus.go #5. Add request header information (specifics from SoapUI)

```
    // Add request header information from SoapUI http log
    req.Header.Add("Content-Type", "text/xml;charset=UTF-8")
    req.Header.Add("SOAPAction",
`"http://xmlns.oracle.com/apps/financials/commonModules/share
d/model/erpIntegrationService/getESSJobStatus"` )
    req.Header.Add("Content-Length",
strconv.Itoa(len(request)))
    req.Header.Add("Host", u.Host ) // from url.Parse
    req.Header.Add("Connection", "Keep-Alive")
    req.SetBasicAuth(username, password)
```

getESSJobStatus.go #6. Dump the request so we can look at it.

```
// Dump the request out
if debug {
    dump, err := httputil.DumpRequestOut(req, true)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Printf("\nRequest:\n%s\n", dump)
}
```

getESSJobStatus.go #7. Send the request to the cloud, and get a response back.

```
// Send request to cloud and get response back
client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    log.Fatal(err.Error())
}
```

getESSJobStatus.go #8. Dump the response so we can look at it.

```
// Dump the response out
if debug {
    dump, err := httputil.DumpResponse(resp, true)
    if err != nil {
        log.Fatal(err)
    }
    fmt.Printf("\nResponse:\n%s\n", dump)
}
```

getESSJobStatus.go #9. Response is in multipart format. Ugh.

```
    // Logic to process multipart response
    mediaType, params, err :=
mime.ParseMediaType(resp.Header.Get("Content-Type"))
    if err != nil {
            log.Fatal(err)
    }
    if strings.HasPrefix(mediaType, "multipart/") {
        mr := multipart.NewReader(resp.Body, params["boundary"])

. . .
```

getESSJobStatus.go #10. Read parts of multipart response.

```
for {
    p, err := mr.NextPart()
    if err == io.EOF {
        return
    }
    if err != nil {
        log.Fatal(err)
    }
    body, err := ioutil.ReadAll(p)
    if err != nil {
        log.Fatal(err)
    }
```

getESSJobStatus.go #11. Parse the success response to get the value we want.

```
if debug {
    fmt.Printf("\nMultipart Body:\n%s\n", body)
}

success := Success{}
err = xml.Unmarshal(body, &success)
if err != nil {
    log.Fatal(err.Error())
}
```

getESSJobStatus.go #12. Print the value we want.

```
            if debug {
                fmt.Printf("\nResult:\n")
            }
            fmt.Printf("%s\n",
success.Body.GetESSJobStatusResponse.Result.Text )
        }
    }
}
```

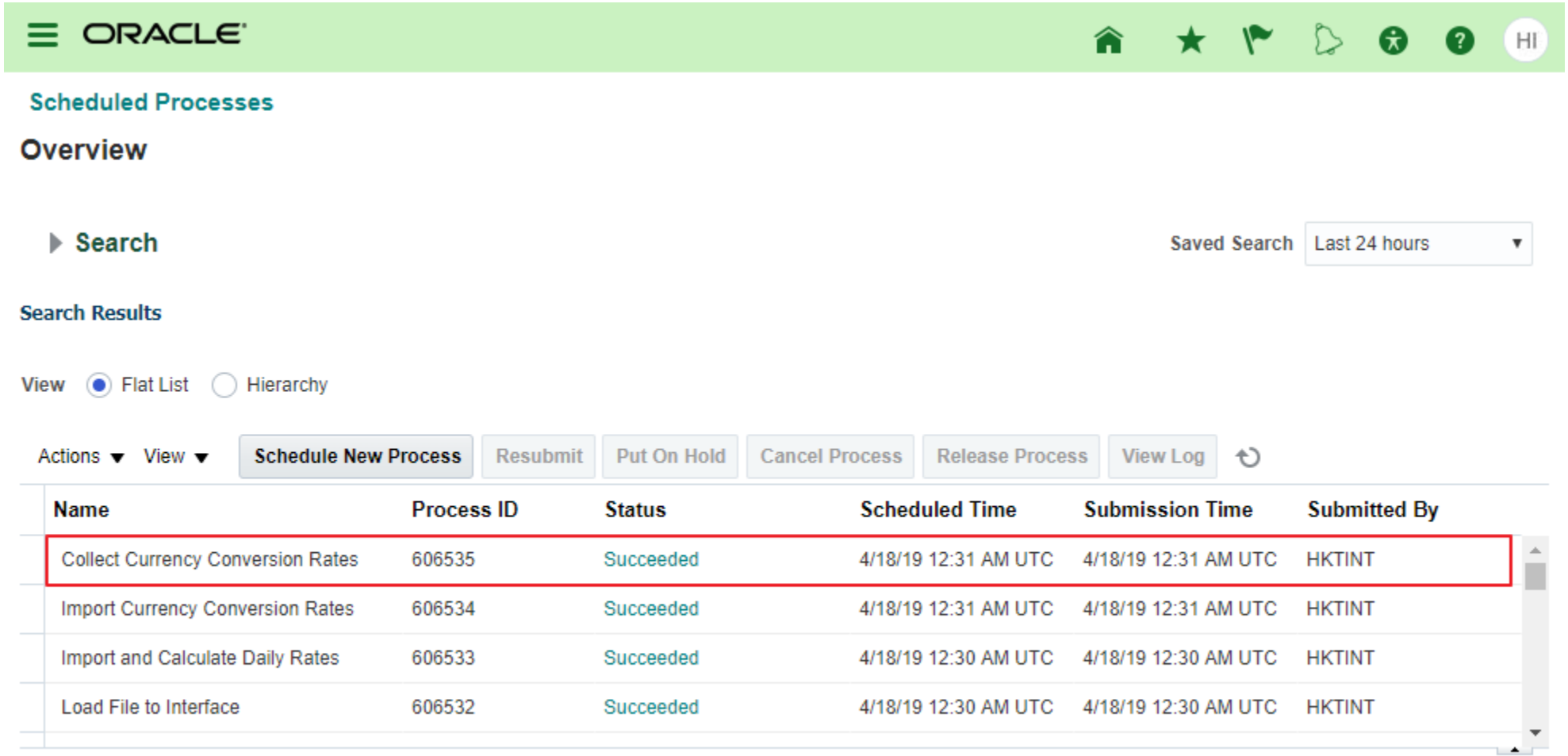
Create our getESSJobStatus program

```
$ cat setenv.sh
export
CLOUD_ENDPOINT=https://somewhere.fa.us2.oraclecloud.com:443/f
scmService/ErpIntegrationService
export CLOUD_USERNAME=scott
export CLOUD_PASSWORD=tiger
$ . ./setenv.sh
$ go build getESSJobStatus.go Request.go Success.go
$ ls -l getESSJobStatus
-rwxrwxr-x 1 jkayser jkayser 7292936 Apr 18 14:37
getESSJobStatus
$
```

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Get a sample ESS job request Id from Cloud Financials



Run it! Our HTTP request headers

```
$ ./getESSJobStatus 606535

Request:
POST /fscmService/ErpIntegrationService HTTP/1.1
Host: somewhere.fa.us2.oraclecloud.com:443
User-Agent: Go-http-client/1.1
Content-Length: 366
Authorization: Basic {gibberish}
Connection: Keep-Alive
Content-Type: text/xml;charset=UTF-8
Soapaction:
"http://xmlns.oracle.com/apps/financials/commonModules/shared
/model/erpIntegrationService/getESSJobStatus"
Accept-Encoding: gzip
```

Run it! Our SOAP request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://xmlns.oracle.com/apps/financials/commonModu
les/shared/model/erpIntegrationService/types/">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getESSJobStatus>
         <typ:requestId>606535</typ:requestId>
      </typ:getESSJobStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

Run it! Our response headers… Multipart. Ugh.

```
Response:
HTTP/1.1 200 OK
Connection: keep-alive
Content-Language: en
Content-Type: multipart/related;type="application/xop+xml";boundary="----=_Part_15643_406343183.1555623823435";start="<8761a2a8-fb88-4953-961b-2ace3a611945>";start-info="text/xml"
Date: Thu, 18 Apr 2019 21:43:43 GMT
Server: Oracle-HTTP-Server
Server-Timing: cdn-cache; desc=MISS
. . .
```

Run it! Our SOAP response is embedded within the multipart gobbledegook.

```
X-Oracle-Dms-Ecid: 005XhJ_OTqR4AxiLp6d9iY0002i20001^C


------=_Part_15643_406343183.1555623823435
Content-Type: application/xop+xml;charset=UTF-8;type="text/xml"
Content-Transfer-Encoding: 8bit
Content-ID: <8761a2a8-fb88-4953-961b-2ace3a611945>

<?xml version="1.0" encoding="UTF-8" ?>
. . .
------=_Part_15643_406343183.1555623823435--
```

Run it! Extract SOAP response from multipart response

```
Multipart Body:
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://www.w3.org/2005/08/addressing"><env:Header><wsa:Action>
http://xmlns.oracle.com/apps/financials/commonModules/shared/model/erpInt
egrationService//ErpIntegrationService/getESSJobStatusResponse</wsa:Actio
n><wsa:MessageID>urn:uuid:8e80e9c3-f14d-4258-8355-
6969525b36b7</wsa:MessageID></env:Header><env:Body><ns0:getESSJobStatusRe
sponse
xmlns:ns0="http://xmlns.oracle.com/apps/financials/commonModules/shared/m
odel/erpIntegrationService/types/"><result
xmlns="http://xmlns.oracle.com/apps/financials/commonModules/shared/model
/erpIntegrationService/types/">SUCCEEDED</result></ns0:getESSJobStatusRes
ponse></env:Body></env:Envelope>
```

Run it! Results!

Result:

SUCCEEDED

$

Doing REST web services with Go
(Similar to SOAP, but uses JSON instead of XML)

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

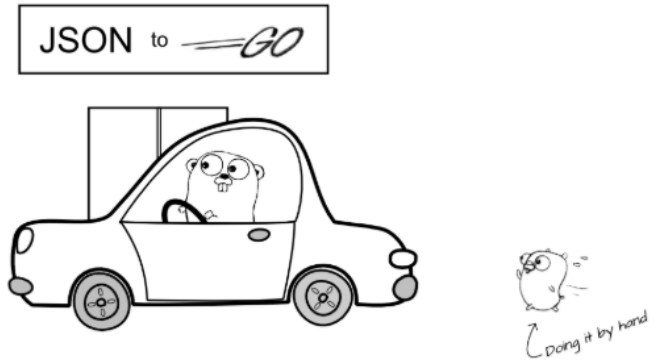
Converting a sample JSON doc to Go struct (https://mholt.github.io/json-to-go/)

JSON-to-Go

Convert JSON to Go struct

This tool instantly converts JSON into a Go type definition. Paste a JSON structure on the left and the equivalent Go type will be generated to the right, which you can paste into your program. The script has to make some assumptions, so double-check the output!

For an example, try converting JSON from the SmartyStreets API or the GitHub API.



JSON	→	Go

```
{
    "glossary": {
        "title": "example glossary",
        "GlossDiv": {
            "title": "S",
            "GlossList": {
                "GlossEntry": {
                    "ID": "SGML",
                    "SortAs": "SGML",
                    "GlossTerm": "Standard Generalized Markup Language",
                    "Acronym": "SGML",
                    "Abbrev": "ISO 8879:1986",
                    "GlossDef": {
                        "para": "A meta-markup language, used to create markup la
                        "GlossSeeAlso": ["GML", "XML"]
                    },
                    "GlossSee": "markup"
                }
            }
        }
    }
```

```
type AutoGenerated struct {
    Glossary struct {
        Title    string `json:"title"`
        GlossDiv struct {
            Title     string `json:"title"`
            GlossList struct {
                GlossEntry struct {
                    ID        string `json:"ID"`
                    SortAs    string `json:"SortAs"`
                    GlossTerm string `json:"GlossTerm"`
                    Acronym   string `json:"Acronym"`
                    Abbrev    string `json:"Abbrev"`
                    GlossDef  struct {
                        Para         string   `json:"para"`
                        GlossSeeAlso []string `json:"GlossSeeAlso"`
                    } `json:"GlossDef"`
                    GlossSee string `json:"GlossSee"`
                } `json:"GlossEntry"`
            } `json:"GlossList"`
        } `json:"GlossDiv"`
    } `json:"glossary"`
```

The Hackett Group

Real world examples using Go for Cloud Data Integration

So, what have we done with Go?

- SOAP: Cloud Revenue Management to EBS integration.
- SOAP: Make generic SOAP calls based on SOAP templates.
- SOAP: Upload files to UCM, submit ESS requests, check ESS request status, cancel ESS requests, download ESS request logfiles and XML data files, etc.
- SOAP: Update Fusion HCM employee information and service dates.
- SOAP: Import and load HCM Data Loader files, get Data Set status.
- SOAP: Release Payables invoices when they should be paid.
- SOAP: Create customer reference accounts for bill-to sites.
- SOAP: Update GL and OIC daily exchange rates.
- SOAP: Salesforce to OIC integration.
- REST: Push Fusion HCM employees to Oracle Identity Cloud Service.

ASC 606 / IFRS 15: New Revenue Recognition Rules

- Reference: E-Business Suite Release 12: New Revenue Recognition Accounting Standard, April 2014, effective January 1, 2018: International Financial Reporting Standard IFRS 15 and US GAAP Standard ASC 606 “Revenue from Contracts with Customers”, June 2017 (Doc ID 2293858.1)

- For EBS customers, the MOS says that to support the new Revenue Recognition rules, customers must integrate with Revenue Management Cloud Services (RMCS).
- That requires integrating EBS with RMCS.
- Oracle delivers part of the integration with “co-existence” patches for EBS. That integrates transaction data.
- RMCS also requires some Master Data (Customers and Items). That is not handled by the co-existence patches.
- We used Go to integrate Customers and Items.

Sample 1: Download BI Publisher report CSV output Use Go templating engine to build SOAP call.

```
$ cat Go_Xref_Rpt.tmpl
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://xmlns.oracle.com/oxp/service/v2">
   <soapenv:Header/>
   <soapenv:Body>
      <v2:runReport>
         <v2:reportRequest>
            <v2:attributeFormat>csv</v2:attributeFormat>
            <v2:reportAbsolutePath>/Custom/Human Capital
Management/Integrations/Mercer/Retiree Address
Inbound/GO_XREF_RPT.xdo</v2:reportAbsolutePath>
            <v2:sizeOfDataChunkDownload>-1</v2:sizeOfDataChunkDownload>
         </v2:reportRequest>
         <v2:userID>{{ getenv "CLOUD_USERNAME" }}</v2:userID>
         <v2:password>{{ getenv "CLOUD_PASSWORD" }}</v2:password>
      </v2:runReport>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample 1: Download BI Publisher report CSV output Run it.

```
$ time ./Go_Xref_Rpt.sh Test
JibeCallWS infile=Go_Xref_Rpt.tmpl infiletemplate=true
endpoint=https://something.fa.us6.oraclecloud.com/xmlpserver/services/v2/
ReportService base64=true outfile=FileSet_Test/xref.csv
200 OK
Exit status: 0
real  0m21.832s
$ ls -l FileSet_Test/xref.csv
-rw-r--r-- 1 jkayser jkayser 50672114 Feb 20 16:56 FileSet_Test/xref.csv
$
```

Not bad: 50M of data in 21.8 seconds.

Sample 2: Automating Lockbox processing
Upload lockbox file to UCM, and import it.

```
$ uploadLockboxFileToUCM.sh PROD
/u01/Integrations/ARLockbox/data/PROD/Incoming/LBX123456_20190220160400
2019-02-20 THG_US_Lockbox
httpStatus: 200 OK
ucmDocumentId: 967219
# Exit status: 0
# UCM Document Id: 967219

$ submitESSJobImportLockbox.sh 967219
httpStatus: 200 OK
requestId: 715089
# Exit status: 0
# Import Request Id: 715089
```

Sample 2: Automating Lockbox processing
Check status, and download results.

```
$ getESSJobStatusUntilDone.sh 715089
Request status: WAIT, will check again after 15 seconds...
Request status: RUNNING, will check again after 15 seconds...
httpStatus: 200 OK
requestStatus: SUCCEEDED
# Exit status: 0

$ downloadESSJobExecutionDetails.sh 715089 log
httpStatus: 200 OK
documentTitle: ESS_L_715089
documentAuthor: HKTINT
documentSecurityGroup: Attachments
documentName: 715089.zip
zipFile: /u01/Integrations/ARLockbox/temp/715089.zip
# Exit status: 0
```

Sample 2: Automating Lockbox processing
Submit lockbox processing job

```
$ submitESSJobProcessLockbox.sh PROD
/u01/Integrations/ARLockbox/data/PROD/Incoming/LBX123456_20190220160400 715089
LBX123456_20190220160400 2019-02-20 THG_US_Lockbox
Running with parameters:
01: New transmission (Y/N)          : Y
02: Import Process Id               : 715089
03: Transmission Name               : LBX123456_20190220160400
04: Submit Import (Y/N)             : N
. . .
16: Org Identifier                  : 300000001328317 (THG US)
17: Number of Instances to Process AutoApply : 1
httpStatus: 200 OK
requestId: 715099
# Exit status: 0
# Process Request Id: 715099
```

Sample 2: Automating Lockbox processing
Wait until done, download job detail

```
$ getESSJobStatusUntilDone.sh 715099
Request status: WAIT, will check again after 15 seconds...
Request status: RUNNING, will check again after 15 seconds...
Request status: PAUSED, will check again after 15 seconds...
Request status: PAUSED, will check again after 15 seconds...
Request status: RUNNING, will check again after 15 seconds...
Request status: PAUSED, will check again after 15 seconds...
Request status: RUNNING, will check again after 15 seconds...
httpStatus: 200 OK
requestStatus: SUCCEEDED
# Exit status: 0

$ downloadESSJobExecutionDetails.sh 715099 log
httpStatus: 200 OK
contentType: zip
documentTitle: ESS_L_715099
documentAuthor: HKTINT
documentSecurityGroup: Attachments
documentName: 715099.zip
zipFile: /u01/Integrations/ARLockbox/temp/715099.zip
# Exit status: 0
```

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Sample 2: Automating Lockbox processing
What, no PDF file?

- Unfortunately, in Fusion Apps, you can only download the XML data for the report, but not the PDF report itself. You have to create the PDF file.
- MOS Reference: PDF Output Is Not Returned By ERPIntegrationService For Standard Report ESS Job (Doc ID 2162075.1)
- Getting started with BIP APIs
  - https://blogs.oracle.com/xmlpublisher/getting-started-with-bip-apis
  - The web page has the Java code you need.

- The hardest part was collecting all the required JAR files.
  - Can't get them from Fusion Apps, since there is no OS access to Fusion.
  - I got them from a newer version of R12.2, and crossed my fingers…

It worked…

Sample 2: Automating Lockbox processing With Go, and a little Java…

ORACLE Receivables

Lockbox Execution Report

Report Date 2/21/19 12:04 AM

Page 1 of 18

Business Unit	THG US
Process Request ID	715099
Transmission Name	LBX[redacted]_20190220160400
Report Format	All

Record Count	Status
6	Record was validated.
16	The receipt was transferred.
22	

Deposit Date	2/20/19	Bank Origination Number	[redacted]
Deposit Time	1602	Destination Account	
Transmission Record Count	22	Transmission Amount	256438.66
Records Transferred To Date	22	Amount Transferred To Date	256438.66
Records Transferred This Run		Amount Transferred This Run	25643866

Lockbox Number	[redacted]	Record Identifier	5
Receipt Batch Source	THG US Lockbox		
Receipts with Valid Customer Account Numbers		Receipts with Invalid Customer Account Numbers	
Receipts with Customers		Receipts with No Customers	
Receipts with Valid Customer Billing Sites		Receipts with Invalid Customer Billing Sites	
Receipts with No Duplicates		Receipts with Duplicates	
Receipts with Valid Receipt Methods		Receipts with Invalid Receipt Methods	
Receipts with Valid Currency		Receipts with Invalid Currency	
Receipts with Valid Applications		Receipts with Invalid Applications	

Error Number	Lockbox Batch Name	Record Identifier	Batch Name	Count	Amount	Currency	Deposit Date	Accounting Date
	001	7	1089	5	256438.66	USD	2/20/19	2/20/19

Record was validated.

Record Identifier	7	Deposit Date	2/20/19	Accounting Date	
Bank Origination Number				Destination Account	
Receipt Currency	USD				
Conversion Type				Conversion Rate	
Transmission Record Count				Transmission Amount	
Lockbox Number	[redacted]			Lockbox Amount	

The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Sample 3: Update benefit dates, service dates in HCM (running in AWS) Generate input data

```
$ ./service_date/bin/service_date.run
File Set: 20190221105441

$ Seniority_Dates_Data_File.sh 20190221105441

$ JibeCallWS infile=service_date/bin/Seniority_Dates_Data_File.tmpl infiletemplate=true
endpoint=https://something.fa.us6.oraclecloud.com/xmlpserver/services/v2/ReportService
base64=true outfile=service_date/env/DEV1/data/FileSet_20190221105441/dates.csv
200 OK
Exit status: 0

$ ACAHours_Data_File.sh 20190221105441

$ JibeCallWS infile=service_date/bin/ACAHours_Data_File.tmpl infiletemplate=true
endpoint=https://something.fa.us6.oraclecloud.com/xmlpserver/services/v2/ReportService
base64=true outfile=service_date/env/DEV1/data/FileSet_20190221105441/hours.csv
200 OK
Exit status: 0
Run timestamp: 20190221105441
```

Sample 3: Update benefit dates, service dates in HCM
Process input data, do date calculations and upload file to UCM

```
$ service_date infiledates=service_date/env/DEV1/data/FileSet_20190221105441/dates.csv
infilehours=service_date/env/DEV1/data/FileSet_20190221105441/hours.csv
outfile=service_date/env/DEV1/data/FileSet_20190221105441/Run_20190221105441/run.out
logfile=service_date/env/DEV1/data/FileSet_20190221105441/Run_20190221105441/run.log
trcfile=service_date/env/DEV1/data/FileSet_20190221105441/Run_20190221105441/run.trc
debug=true


$ upload.run service_date/env/DEV1/data/FileSet_20190221105441/Run_20190221105441/run.out
Worker
  adding: Worker.dat (deflated 94%)
Oracle WebCenter Content Document Transfer Utility
Oracle Fusion Applications
Copyright (c) 2013-2014, Oracle.  All rights reserved.
Performing upload (CHECKIN_UNIVERSAL) ...
Upload successful.
[dID=29369 | dDocName=UCMFA00029487]
Exit status: 0
```

Sample 3: Update benefit dates, service dates in HCM Import and Load HCM data

```
$ importAndLoadData.sh UCMFA00029487 DeleteSourceFile=Y
httpStatus: 200 OK
wsaAction:
http://xmlns.oracle.com/apps/hcm/common/dataLoader/core/dataLoader
IntegrationService//HCMDataLoader/importAndLoadDataResponse
wsaMessageId: urn:uuid:a56c680e-7f04-41f6-a623-0564bc47f40f
requestId: 32489
Exit status: 0
```

Sample 3: Update benefit dates, service dates in HCM
Get HCM data set status

```
$ getDataSetStatusUntilDone.sh 32489
Request status: NOT_STARTED, will check again after 15 seconds...
Request status: IN_PROGRESS, will check again after 15 seconds...
. . .
httpStatus: 200 OK
wsaAction:
http://xmlns.oracle.com/apps/hcm/common/dataLoader/core/dataLoaderIntegrationService//HCMDataLoad
er/getDataSetStatusResponse
wsaMessageId: urn:uuid:1d2218d0-384d-466a-aa4b-6e48d20610a2
/DATA_SET_STATUS/RESULT_COUNT 1
/DATA_SET_STATUS/DATA_SET/DATA_SET_NAME Worker.zip
/DATA_SET_STATUS/DATA_SET/CONTENT_ID UCMFA00029487
/DATA_SET_STATUS/DATA_SET/CREATION_DATE 2019/02/21 18:55:58
/DATA_SET_STATUS/DATA_SET/STATUS COMPLETED
/DATA_SET_STATUS/DATA_SET/PROCESS/PROCESS_ID 32489
/DATA_SET_STATUS/DATA_SET/PROCESS/SUBMISSION_DATE 2019/02/21 18:55:42
. . .

Exit status: 0
```

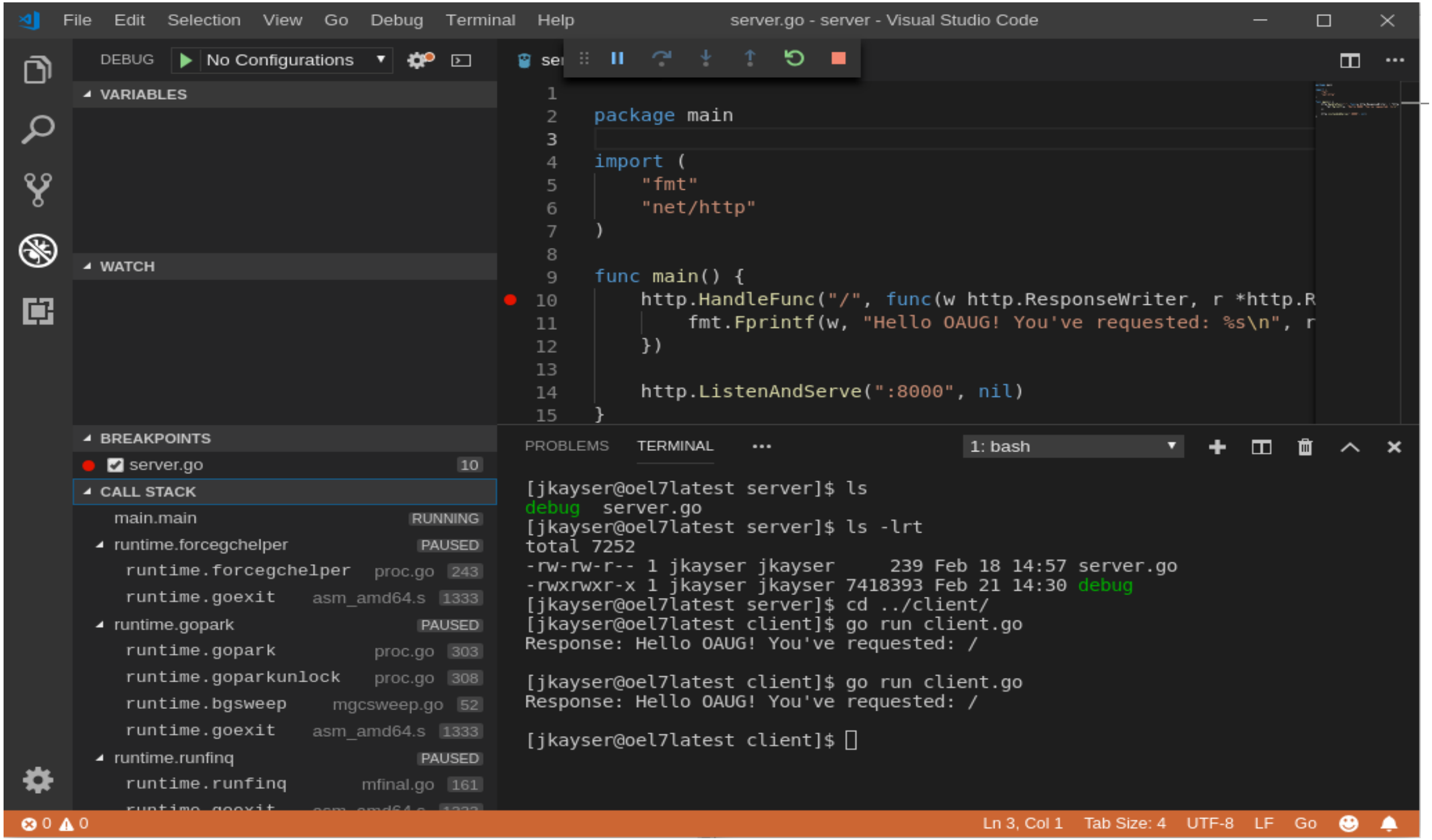
A few Go odds and ends

Testing / Debugging Go

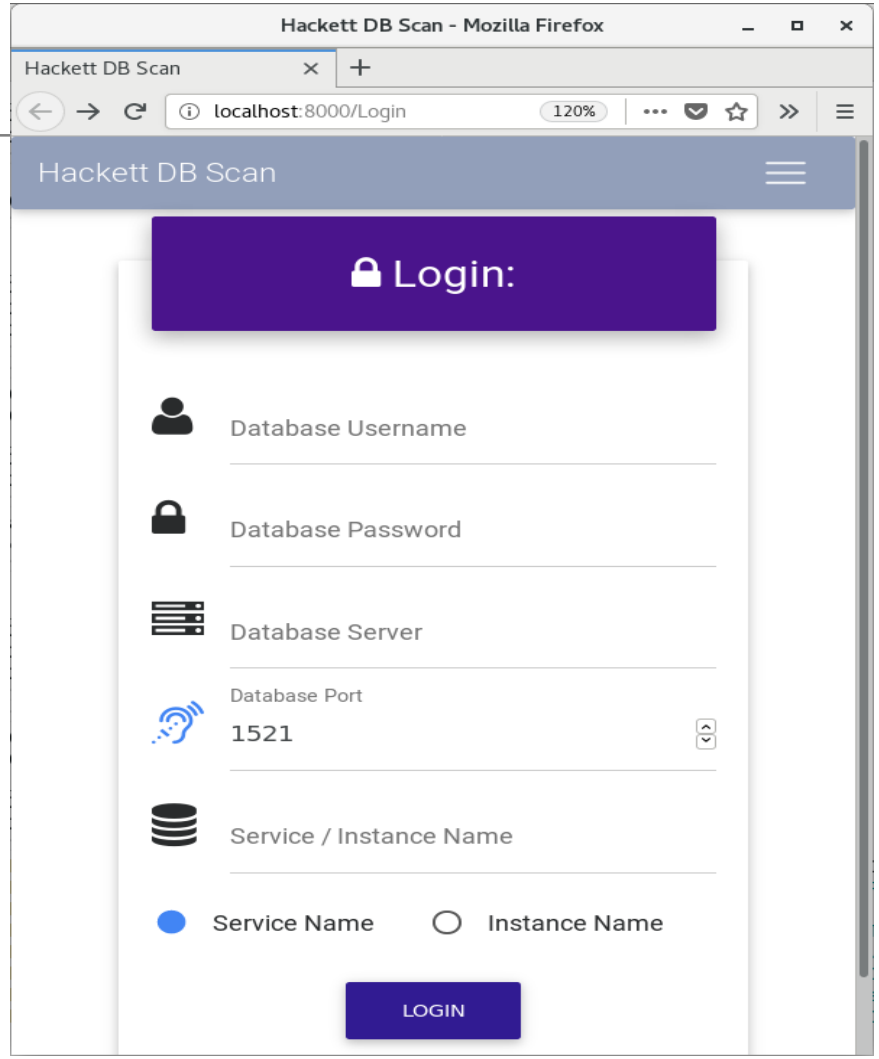
- **Testing:**
- **Chapter 11 in Go book describes testing facilities built into Go**
- **(And, of course, there is “Old School”: lots of print lines. ☺)**

- **Debugging:**
- **Microsoft Visual Studio Code (free)**
  - **https://code.visualstudio.com/**
- **Go Extension for Visual Studio Code**
  - **https://code.visualstudio.com/docs/languages/go**
  - If you set go.autocompleteUnimportedPackages to true, VS Code will automatically update your import section with the packages you are referencing. Nice!
- **Go debugger (Delve)**
  - **https://github.com/go-delve/delve**

This allows you to step through a Go program while it is executing, examine variable values, etc. Invaluable, if you need it.


File Edit Selection View Go Debug Terminal Help
server.go - server - Visual Studio Code
DEBUG
No Configurations
VARIABLES
WATCH
BREAKPOINTS
server.go 10
CALL STACK
main.main RUNNING
runtime.forcegchelper PAUSED
runtime.forcegchelper proc.go 243
runtime.goexit asm_amd64.s 1333
runtime.gopark PAUSED
runtime.gopark proc.go 303
runtime.goparkunlock proc.go 308
runtime.bgsweep mgcsweep.go 52
runtime.goexit asm_amd64.s 1333
runtime.runfinq PAUSED
runtime.runfinq mfinal.go 161
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
package main
import (
"fmt"
"net/http"
)
func main() {
http.HandleFunc("/", func(w http.ResponseWriter, r *http.R
fmt.Fprintf(w, "Hello OAUG! You've requested: %s\n", r
})
http.ListenAndServe(":8000", nil)
}
PROBLEMS TERMINAL
1: bash
[jkayser@oel7latest server]$ ls
debug server.go
[jkayser@oel7latest server]$ ls -lrt
total 7252
-rw-rw-r-- 1 jkayser jkayser 239 Feb 18 14:57 server.go
-rwxrwxr-x 1 jkayser jkayser 7418393 Feb 21 14:30 debug
[jkayser@oel7latest server]$ cd ../client/
[jkayser@oel7latest client]$ go run client.go
Response: Hello OAUG! You've requested: /
[jkayser@oel7latest client]$ go run client.go
Response: Hello OAUG! You've requested: /
[jkayser@oel7latest client]$
0 0
Ln 3, Col 1 Tab Size: 4 UTF-8 LF Go

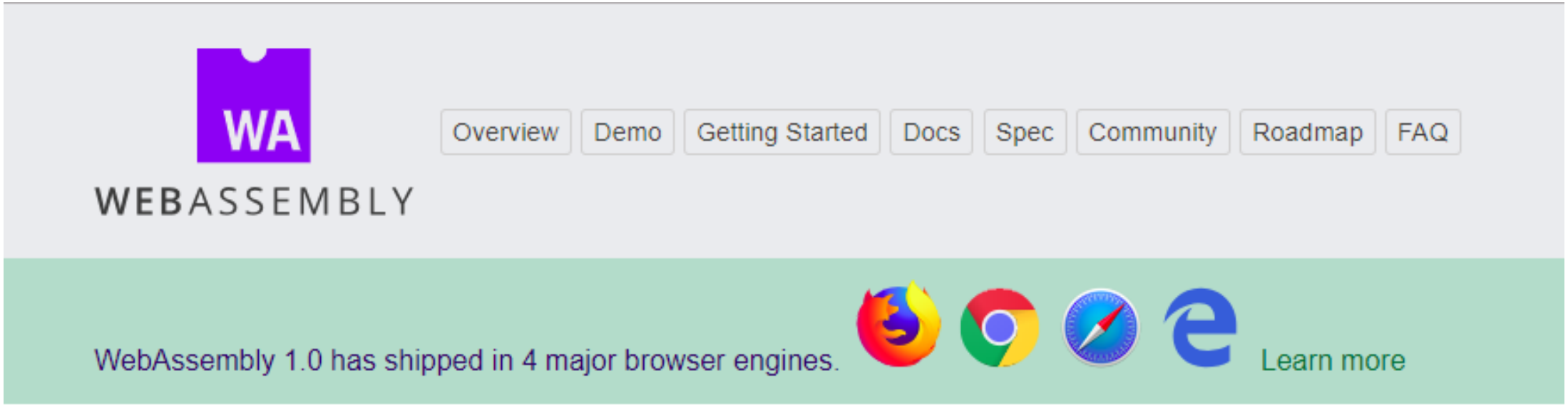
So what is Go not good at? GUI.



- No native Go GUI library; use HTML instead
- When I've wanted a GUI, I do the following:
  - Write a Go program that provides a web server on localhost:8000
  - Write your GUI as a website.
  - Use the Material Design for Bootstrap library – its beautiful =>
    - https://mdbootstrap.com/
  - Use any browser to browse to localhost:8000
  - If you want to make it accessible over the network, add security
    - Go aggressively supports emerging web standards (HTTP/2, TLS 1.3. etc.)

- **This is changing.**
- Go 1.11 added an experimental port to WebAssembly.
  - https://github.com/golang/go/wiki/WebAssembly
  - https://webassembly.org/
- Why WebAssembly is a game changer for the web—
- and a source of pride for Mozilla and Firefox
  - https://medium.com/mozilla-tech/why-webassembly-is-a-game-changer-for-the-web-and-a-source-of-pride-for-mozilla-and-firefox-dda80e4c43cb

Web Assembly: https://webassembly.org/



WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust, enabling deployment on the web for client and server applications.

Cgo: the Go escape hatch to C

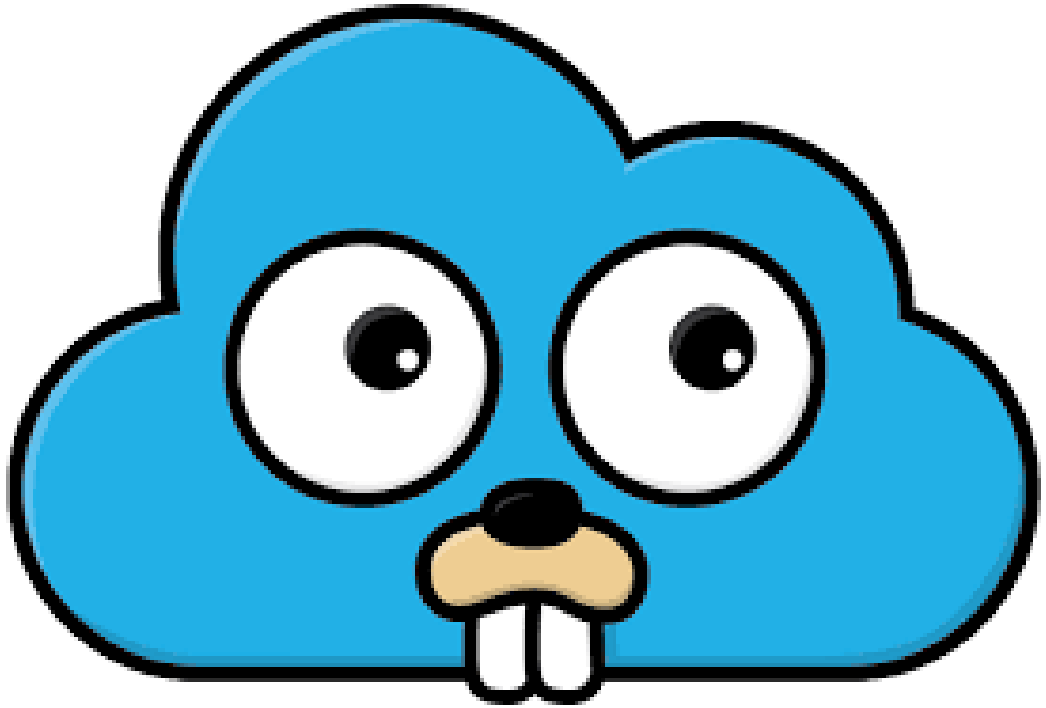
- Native Go (what you have seen so far) requires Go source code to build.
- Not everything has been rewritten in Go (yet ☺). Examples:
  - Oracle Database Drivers for Go (we all need to pester Oracle about this).
    - See: https://github.com/jeffkayser2/learningGo/wiki, item #21
  - OpenCV for computer vision (see: https://gocv.io/)
  - Etc.

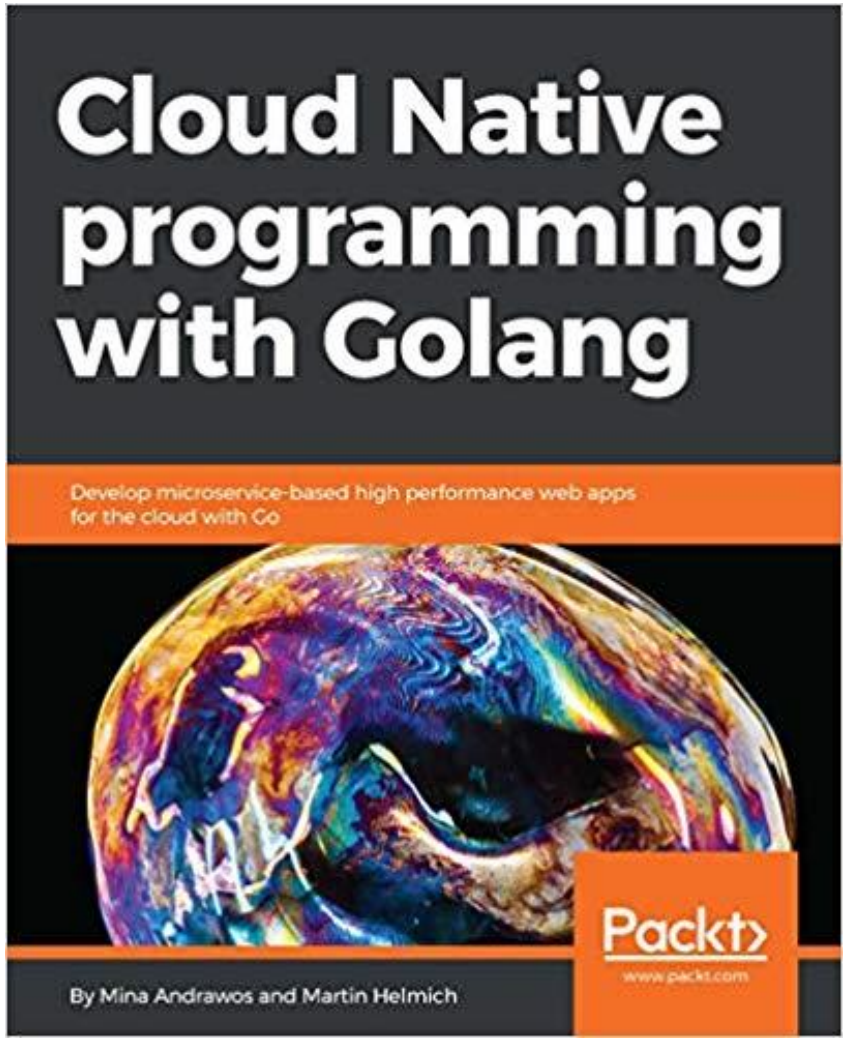
- Go includes another build process to allow Go to call C libraries.
  - Cgo: https://golang.org/cmd/cgo/
    - Pros: You can call C from Go.
    - Cons: Many. See: https://dave.cheney.net/2016/01/18/cgo-is-not-go

The Hackett Group

Go and AWS

Go and the Cloud
(Watch for Packt's sales: I got 20 ebooks for $5 apiece!)





Google's open Go SDK for various Cloud providers: https://github.com/google/go-cloud

README.md

The Go Cloud Development Kit (Go CDK)

Write once, run on any cloud

build passing | godoc reference | codecov 76%

The Go Cloud Development Kit (Go CDK) allows Go application developers to seamlessly deploy cloud applications on any combination of cloud providers. It does this by providing stable, idiomatic interfaces for common uses like storage and databases. Think `database/sql` for cloud products.

Imagine writing this to read from blob storage (like Google Cloud Storage or S3):

```
blobReader, err := bucket.NewReader(context.Background(), "my-blob", nil)
```

and being able to run that code on any cloud you want, avoiding all the ceremony of cloud-specific authorization, tracing, SDKs and all the other code required to make an application portable across cloud platforms.

The project works well with a code generator called Wire. It creates human-readable code that only imports the cloud SDKs for providers you use. This allows the Go CDK to grow to support any number of cloud providers, without increasing compile times or binary sizes, and avoiding any side effects from `init()` functions.

You can learn more about the project from our announcement blog post, or our talk at Next 2018:

AWS and Go (Amazon AWS flavored): https://aws.amazon.com/sdk-for-go/

AWS SDK for Go API Reference | Developer Guide | Blog | Feedback

- aws
  - arn
  - awserr
  - awsutil
  - client
    - metadata
  - corehandlers
  - credentials
    - ec2rolecreds
    - endpointcreds
    - plugincreds
    - processcreds
    - stscreds
  - crr
  - csm
  - defaults
  - ec2metadata
  - endpoints
  - request
  - session
  - signer
    - v4
- service
  - acm
  - acmpca
  - alexaforbusiness
  - amplify
  - apigateway
  - apigatewaymanagementapi
  - apigatewayv2
  - applicationautoscaling
  - applicationdiscoveryservice
  - appmesh
  - appstream
  - appsync

api reference | gitter join chat | build passing | license Apache V2

AWS SDK for Go

aws-sdk-go is the official AWS SDK for the Go programming language.

Checkout our release notes for information about the latest bug fixes, updates, and features added to the SDK.

We announced the Developer Preview for the v2 AWS SDK for Go. The v2 SDK source is available at https://github.com/aws/aws-sdk-go-v2, and add it to your project with `go get github.com/aws/aws-sdk-go-v2`. Check out the v2 SDK's changes and updates, and let us know what you think. We want your feedback.

Installing

Use `go get` to retrieve the SDK to add it to your `GOPATH` workspace, or project's Go module dependencies.

```
go get github.com/aws/aws-sdk-go
```

To update the SDK use `go get -u` to retrieve the latest version of the SDK.

```
go get -u github.com/aws/aws-sdk-go
```

Dependencies

The SDK includes a `vendor` folder containing the runtime dependencies of the SDK. The metadata of the SDK's dependencies can be found in the Go module file `go.mod` or Dep file `Gopkg.toml`.

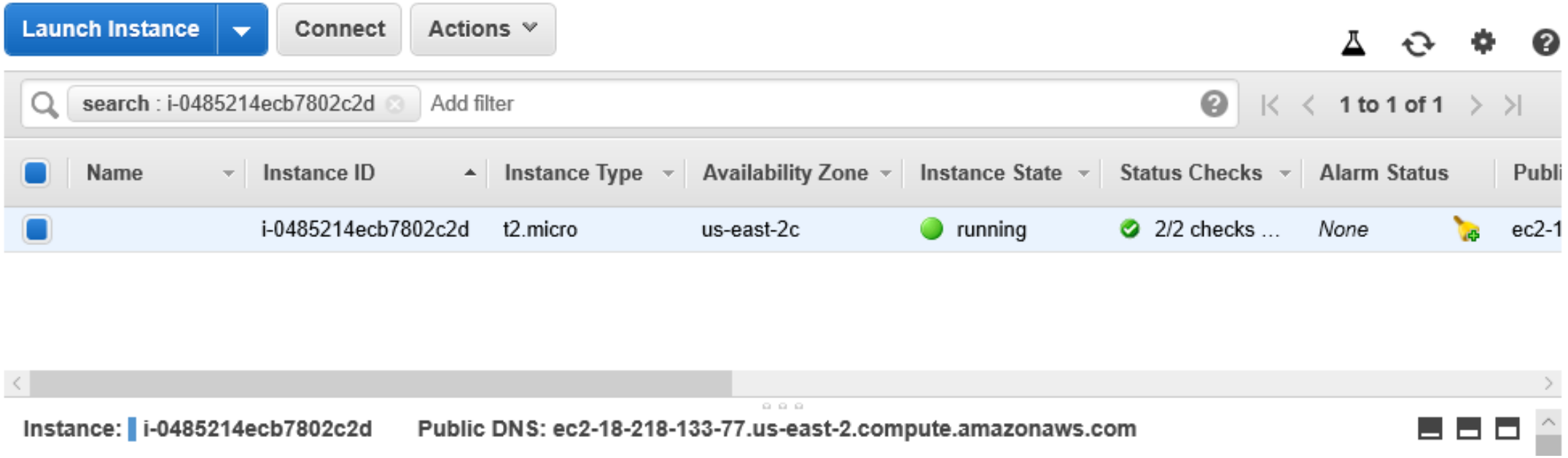
Go Modules

If you are using Go modules, your `go get` will default to the latest tagged release version of the SDK. To get a specific release version of the SDK use `@<tag>` in your `go get` command.

```
go get github.com/aws/aws-sdk-go@v1.15.77
```

OPEN CHAT

Launch an AWS Linux server



Login to AWS Linux server (with PuTTY)

```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key": {passphrase}
Last login: Fri Feb 22 22:08:57 2019 from c-73-157-136-
90.hsd1.or.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-41-34 ~]$
```

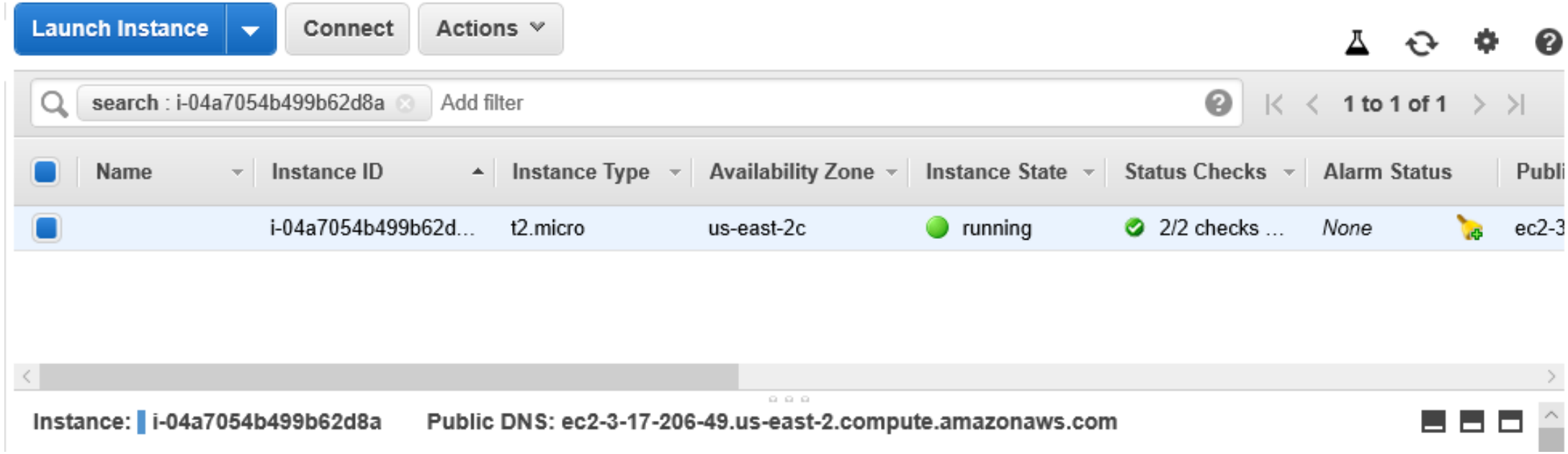
Build Linux binaries on your development server, and copy them to the AWS Linux server

```
$ uname -a
Linux oel7latest.jkayser.pvt 4.14.35-1844.2.5.el7uek.x86_64 #2 SMP
Mon Feb 4 18:24:45 PST 2019 x86_64 x86_64 x86_64 GNU/Linux
$ go build -tags "netgo osusergo" server.go
$ scp -i ~/Amazon_AWS/AmazonAWSEC2key.pem server ec2-user@ec2-18-
218-133-77.us-east-2.compute.amazonaws.com:
server                                100% 7141KB 713.6KB/s   00:10
$ go build -tags "netgo osusergo" client.go
$ scp -i ~/Amazon_AWS/AmazonAWSEC2key.pem client ec2-user@ec2-18-
218-133-77.us-east-2.compute.amazonaws.com:
client                                100% 6689KB 742.9KB/s   00:09
$
```

Now, on the AWS Linux server, run the binaries. No dependencies. Nice.

```
$ uname -a
Linux ip-172-31-41-34 4.14.97-74.72.amzn1.x86_64 #1 SMP Tue Feb 5
20:59:30 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
$ ls -l
total 13836
-rwxrwxr-x 1 ec2-user ec2-user 6849693 Feb 26 23:34 client
-rwxrwxr-x 1 ec2-user ec2-user 7312324 Feb 26 23:32 server
$ ./server &
[1] 2929
$ ./client
Response: Hello NWOUG! You've requested: /

$
```

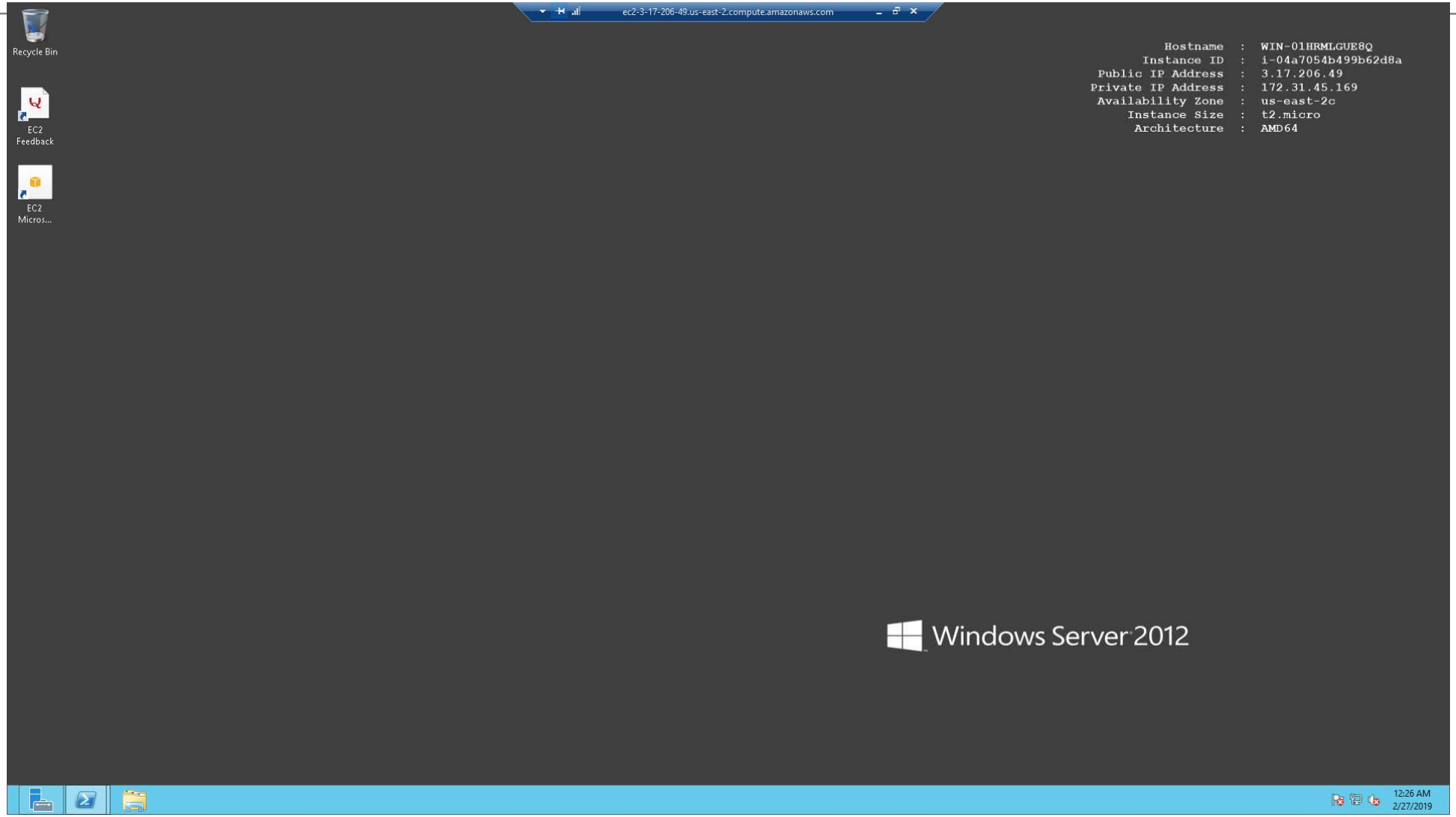
Launch an AWS Windows server



The Hackett Group

ORACLE Platinum Partner Cloud Premier North America

Login to AWS Windows server



Build Windows *.EXE binaries on your (Linux) development server…

```
$ GOOS=windows GOARCH=amd64 go build -tags "netgo osusergo"
server.go
$ GOOS=windows GOARCH=amd64 go build -tags "netgo osusergo"
client.go
$ file *exe
client.exe: PE32+ executable (console) x86-64 (stripped to
external PDB), for MS Windows
server.exe: PE32+ executable (console) x86-64 (stripped to
external PDB), for MS Windows
$
```

Copy the Windows *.exe binaries to the AWS Windows server

- https://serverfault.com/questions/150369/how-to-copy-files-to-amazon-ec2-windows-instance-from-my-local-machine

- Actually we can take benefit from the disk sharing feature of mstsc. Here is how it works:
  - 1. Go to find the RDP file amazon asked you to downloaded onto your local PC. right-click --> Edit
  - 2. Go to "Local Resources" tab --> "Local devices and resources" --> "More" button
  - 3. Expand the "Drives" and check the disks you want to share when you TS to the remote box.
  - 4. after connect, you will see the new drives in My Computer already mounted for you.

Public IP Address : 3.17.206.49

Administrator: Windows PowerShell (8)

```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> cd .\Desktop
PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         2/27/2019  12:01 AM    6574592 client.exe
-a---         9/24/2012   5:14 PM        200 EC2 Feedback.url
-a---         9/24/2012   6:44 PM        594 EC2 Microsoft Windows Guide.website
-a---         2/26/2019  11:59 PM    7072768 server.exe


PS C:\Users\Administrator\Desktop> .\server.exe
```

server.exe

Administrator: Windows PowerShell (8)

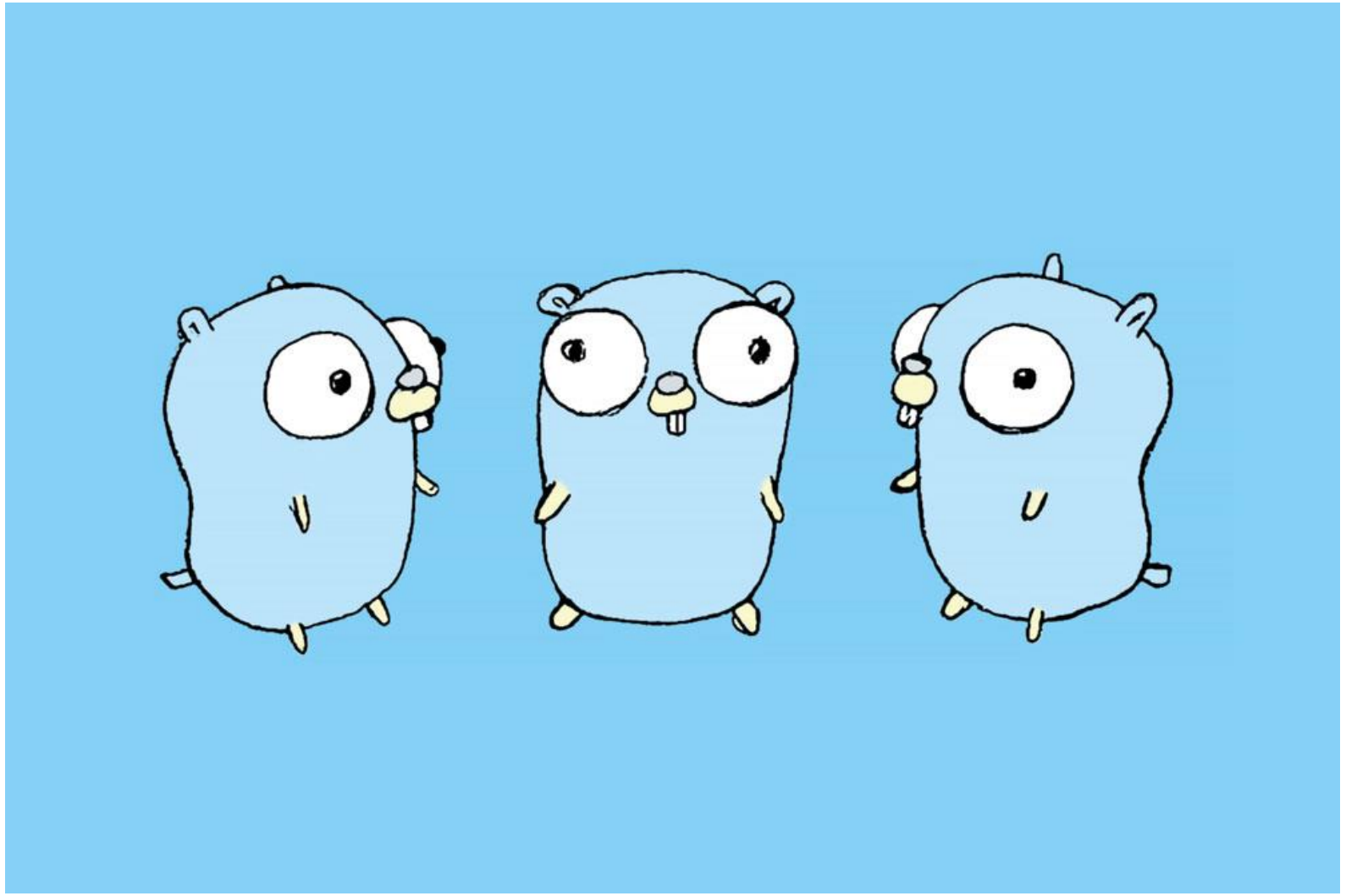
```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> cd .\Desktop
PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         2/27/2019  12:01 AM    6574592 client.exe
-a---         9/24/2012   5:14 PM        200 EC2 Feedback.url
-a---         9/24/2012   6:44 PM        594 EC2 Microsoft Windows Guide.website
-a---         2/26/2019  11:59 PM    7072768 server.exe


PS C:\Users\Administrator\Desktop> .\client.exe
Response: Hello OAUG! You've requested: /

PS C:\Users\Administrator\Desktop> _
```

client.exe

Lots more platforms to choose from…

- android/386
- darwin/386
- darwin/amd64
- darwin/arm
- darwin/arm64
- dragonfly/amd64
- freebsd/386
- freebsd/amd64
- freebsd/arm
- js/wasm
- linux/386
- linux/amd64
- linux/arm
- linux/arm64
- linux/ppc64
- linux/ppc64le
- linux/mips
- linux/mipsle
- linux/mips64
- linux/mips64le
- linux/s390x
- nacl/386
- nacl/amd64p32
- nacl/arm
- netbsd/386
- netbsd/amd64
- netbsd/arm
- openbsd/386
- openbsd/amd64
- openbsd/arm
- plan9/386
- plan9/amd64
- plan9/arm
- solaris/amd64
- windows/386
- windows/amd64
- windows/arm

Enhance your Cloud career: Become a Gopher!



The Hackett Group

World Class Defined and Enabled

www.thehackettgroup.com