

API Gateway & Assembly

Hybrid Integration Enablement

October 2017



API Gateway

DataPower Gateways

Over 14 years of innovation & over 2,000 global installations



Government

- Agencies and ministries
- Defense and security organizations
- Crown corporations



Banking

- Majority of the big US and European banks
- All of the big 5 Canadian banks
- Numerous regional banks and credit unions



Insurance

- Used by 95% of top global insurance firms
- SaaS providers, ASPs, regulators, etc.



Many, many, more

- Healthcare
- Retailers
- Utilities, Power, Oil and Gas
- Telecom
- Airlines
- Others



Enterprise grade security requires a secure platform

- **Purpose-built hardware** provides **physical security**

- Sealed, tamper-evident case
- No usable USB, VGA, other ports
- Intrusion detection switch
- Trusted Platform Module
- Encrypted flash drive
- FIPS 140-2 level 3 Hardware Security Module (option) for secure storage of private keys

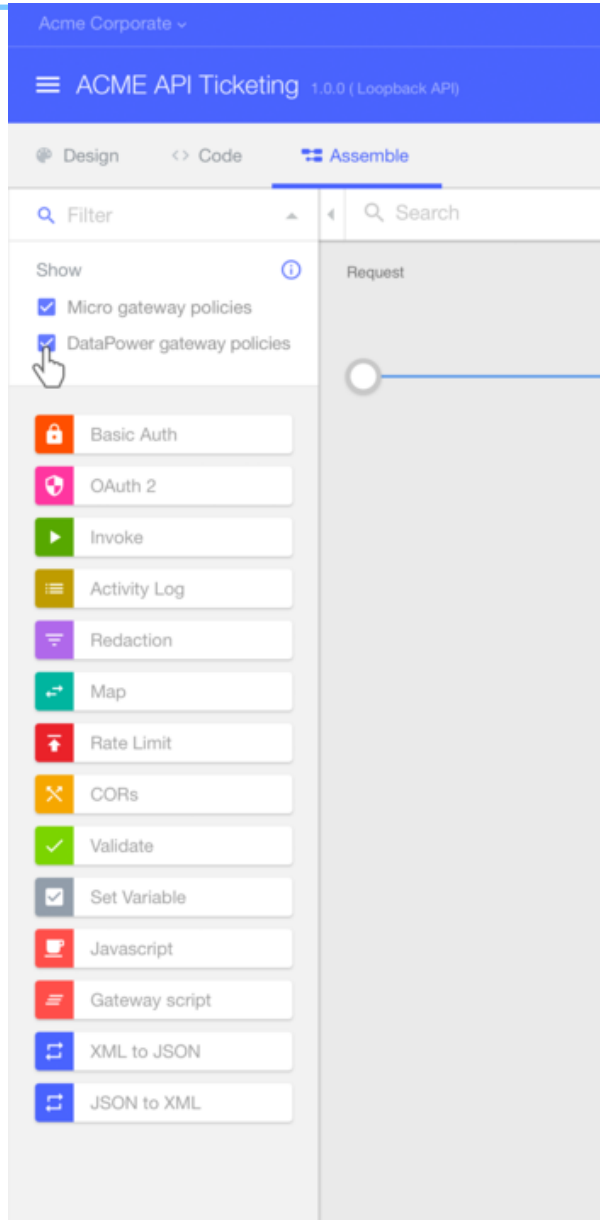


- **Hardened firmware** provides platform security for **physical & virtual gateways**

- Single signed and encrypted firmware by IBM
- No arbitrary software
- Optimized, embedded operating system
- High assurance, “locked-down” configuration
- Key materials are not exportable from the appliance *



Built-in Policies



Rate Limiting

Rate Limit
Message Limit
Circuit Breaker
Concurrent Requests

Caching

Response
Fragment

Traffic Management

Invoke
Routing
Load Balancing
Compression

Validation

REST Payload
Web Services Payload

Logging & Analytics

Activity

Security Definitions

Client ID Validation
Client ID/Secret Validation
OAuth2 Validation
Basic Auth Validation
JWT Validation
OpenID Connect Validation
WS-Security UNT/BST Validation
SAML Validation
Kerberos Validation
LTPA Validation

Security

Authentication
Redaction
CORs
JWT Generation
LTPA Generation
SAML Generation
Kerberos Generation
XML Threat Protection
JSON Threat Protection
WS-Security BST Generation
JSON Sign/Verify
JSON Encrypt/Decrypt
XML Sign/Verify
XML Encrypt/Decrypt
Anti-Virus Invoke

Transformation

Map
Set Variable
JSON to XML
XML to JSON
Any to Any

Extensions

REST Service Invoke
Web Service Invoke
GatewayScript
JavaScript
XSLT

Pluggable Authorization

IBM MobileFirst
IBM ISAM
Custom

IBM API Connect: Which Gateway is for me?

➤ **DataPower Gateway** (Enterprise API Gateway)

- ✓ Built for Departments & Cross Enterprise usage
- ✓ Enterprise-grade security, performance & stability
- ✓ Low touch gateway without external dependencies
(Physical, Virtual, Cloud, Docker form factors)
- ✓ Comprehensive set of security, traffic management, mediation & acceleration functionality
- ✓ Supports multiple catalogs per instance/cluster

➤ **Micro Gateway** (Basic API Gateway)





- ✓ Built for Developers & Single Projects
- ✓ Programmable using JavaScript (built on Node.js)
- ✓ Embedded into native developer experience
- ✓ Basic set of security & traffic management functionality
- ✓ Supports single catalog per instance/cluster

General Conversation with Customers

- We offer different form factors
 - Physical, virtual (OVA), and docker (now production).
- Competitive solutions mention the keyword “API Gateway”, use this term as well
 - Underneath the covers, you can say it’s DataPower driving this functionality
- With version 5.x, we do not require you to be a DataPower expert
 - Existing v4 customers may want to know about DP Access, assembly is their gateway
 - Highlight that custom policies can be imported
 - For new customers, we offer user friendly way to configure gateway

DataPower Playground

- <https://developer.ibm.com/datapower/datapower-playground/>

 developerWorks Marketplace   

IBM DataPower Gateways Home Getting Started DataPower Playground Forum Blog GitHub Videos YouTube Docs

DataPower Playground Samples Code Request Response Log API Reference

Sample 1: Hello GatewayScript

This simple sample will help you get familiar with the environment and log the customary "Hello World" with a twist to both the DataPower system log and write the same message string as transactional data.

Things to notice after you send the request via the Request tab:

1. System log message will contain "Hello GatawayScript!!" at info level
2. Response data sent back to this web application will contain the string "Hello GatewayScript!!"
3. Request tab in this sample will not contain any data since the request is a GET request

The **console** API is mechanism for logging custom entries for reporting, auditing or serviceability purposes to the DataPower system log, which using Log Target configuration allows for very simple on box logging to complex multi-target off-box logging using different protocols. The **session** object refers to the transaction (request/response) as it flows through the DataPower Processing Policy. Take a look at the API reference now to get a better understanding for both the console and session API before continuing.

```
// Log to the DataPower logging system the following message
console.info("Hello GatewayScript!!");

// Write the same message as transactional data
session.output.write("Hello GatewayScript!!");
```

Edit and Test

Sample 2: Logging to the DataPower logging system

Writing to the system log has never been easier. Simply pass an object to the console object using the method that describes the severity of the message. Whether the log message is logged and where this log message is logged to depends on the DataPower log target configuration. For example each of the log messages below could be transmitted by syslog to a centralized off-box log management solution.

Notice:

1. System log messages - different priorities, syntax
2. Message returned - Console - Successful Completion

```
console.info("Starting Console Demo");

// Print a json variable directly to the log at info level
console.info({'myLabel' : 'myValue'});
```


API Designer Tips

Use Case: How do I apply gateway policies to APIs?

Api Connect Toolkit CLI

Commands (type `apic COMMAND -h` for additional help):

Creating and validating artifacts

<code>config</code>	manage configuration variables
<code>create</code>	create development artifacts
<code>edit</code>	run the API Designer
<code>explore</code>	run the API Explorer
<code>validate</code>	validate development artifacts

Creating and testing applications

<code>logs</code>	display service logs
<code>loopback</code>	create and manage LoopBack applications
<code>swiftserver</code>	create and manage Swift Server Generator applications
<code>microgateway</code>	create Micro Gateway applications
<code>props</code>	service properties
<code>services</code>	service management
<code>start</code>	start services
<code>stop</code>	stop services

Publishing to the cloud

<code>apis</code>	manage APIs in a catalog
<code>apps</code>	manage provider applications
<code>catalogs</code>	manage catalogs in an organization
<code>devapps</code>	manage consumer applications
<code>drafts</code>	manage APIs and products in drafts
<code>extensions</code>	manage vendor extensions in a catalog
<code>login</code>	log in to an IBM API Connect cloud
<code>logout</code>	log out of an IBM API Connect cloud
<code>members</code>	manage members
<code>orgs</code>	manage organizations
<code>policies</code>	manage policies in a catalog
<code>products</code>	manage products in a catalog
<code>publish</code>	publish products and APIs to a catalog
<code>securegateways</code>	manage secure gateways
<code>spaces</code>	manage spaces in a catalog
<code>subscriptions</code>	manage subscriptions

- API Designer is powered by the CLI commands in the background
- Access it by typing `apic`
 - Full options for creating, testing and publishing APIs (will be visible when typing ‘`apic`’)

Starting the API Designer

```
apic edit
```

- Starts API Designer
 - Will ask for Bluemix Login to connect prep for publishing later

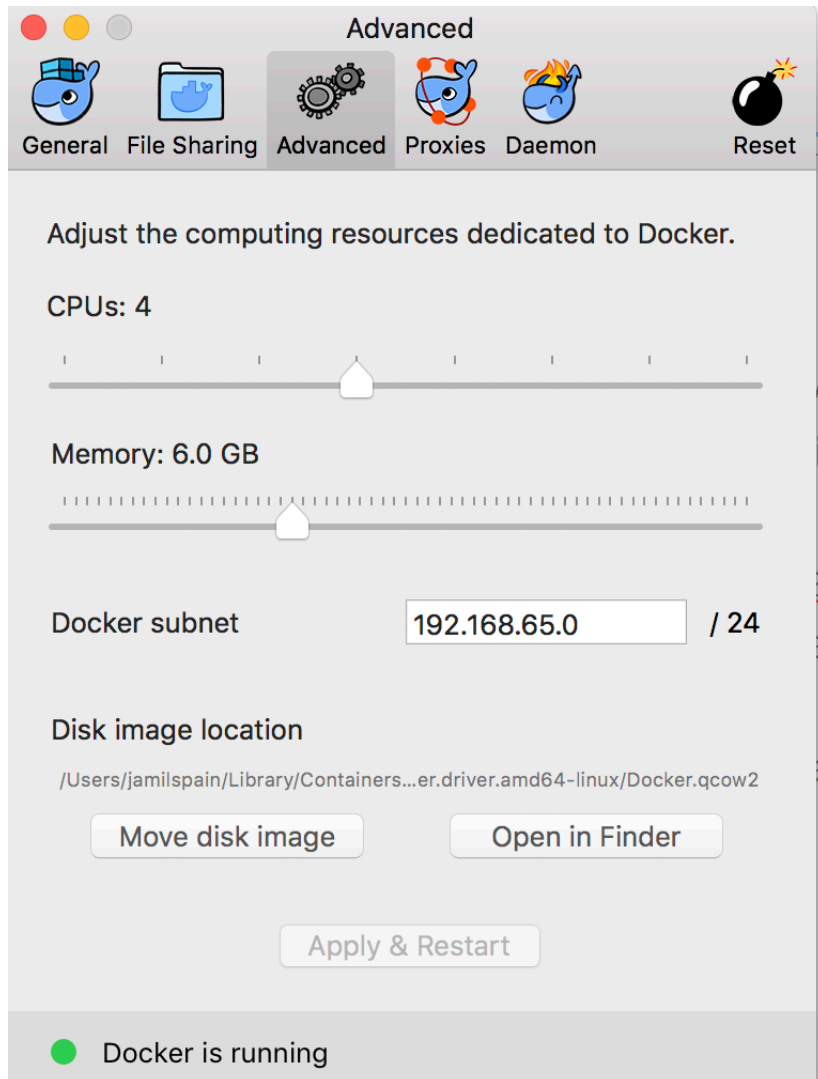
```
SKIP_LOGIN=true apic edit
```

- Starts API Designer with no Bluemix Login Screen
 - Will start API Designer and allow working on APIs without Login Screen (you can authenticate later)

```
PORT=9001 apic edit
```

- Default port is 9000
 - Using this prefix, you can start additional API Designer sessions on other ports.

Docker Tips



- Docker installs a management panel
- By Default, it starts with 2GB of memory allocated. On the first time you start the gateway in the toolkit, make sure you upgrade this to at least 4GB.
- Of course, depending on how much memory you have available. The command are handy below

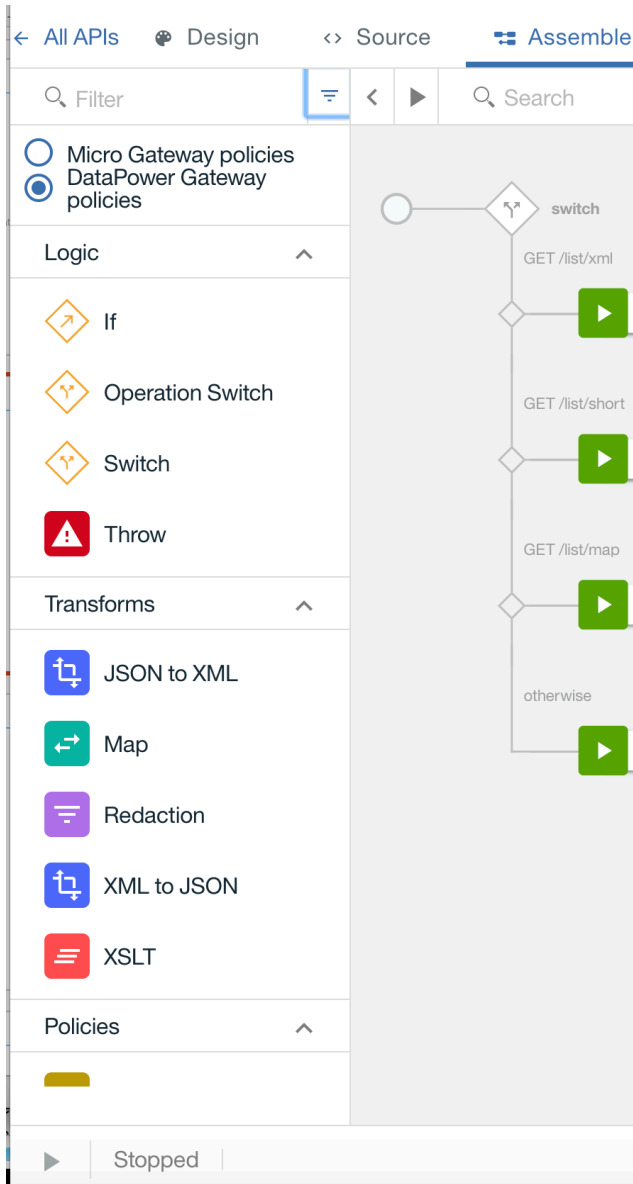
List all containers (only IDs)

```
docker ps -aq
```

Stop all running containers

```
docker stop $(docker ps -aq)
```

Starting the Gateway

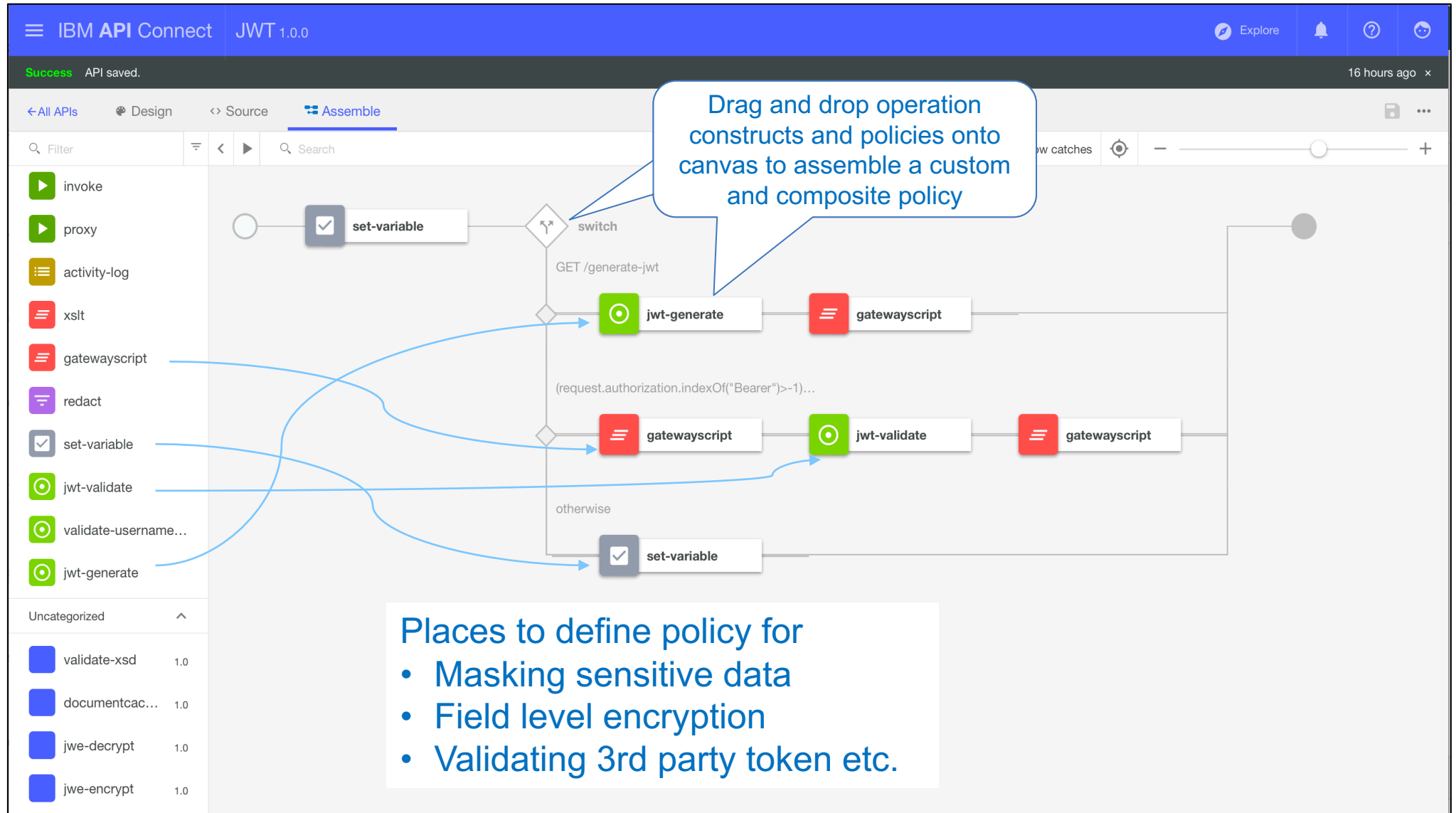


- Bottom Bar is used to stop and start the Gateway.
 - What it starts depends on whether you select the Micro Gateway or DataPower
 - Generally this process of stopping and starting between both works great

ASSEMBLY Programming

Use Case: How do I apply gateway policies to APIs?


API Orchestration with the API Assembler





API Designer: Policies Panel


- With version 5.x, we do not require you to be a DataPower expert
 - Existing v4 customers may want to know about DP Access
 - Highlight that custom policies can be imported
 - For new customers, we offer user friendly way to configure gateway

Logic ^


 If


 Operation Switch


 Switch


 Throw


Transforms ^

 JSON to XML

 Map

 Redaction

 XML to JSON

 XSLT

Policies ^

 Activity Log

 GatewayScript

 Invoke

 Proxy

 Set Variable

 Validate

Security ^

 Generate JWT

 Generate LTPA Tok...

 Validate JWT

 Validate Username...

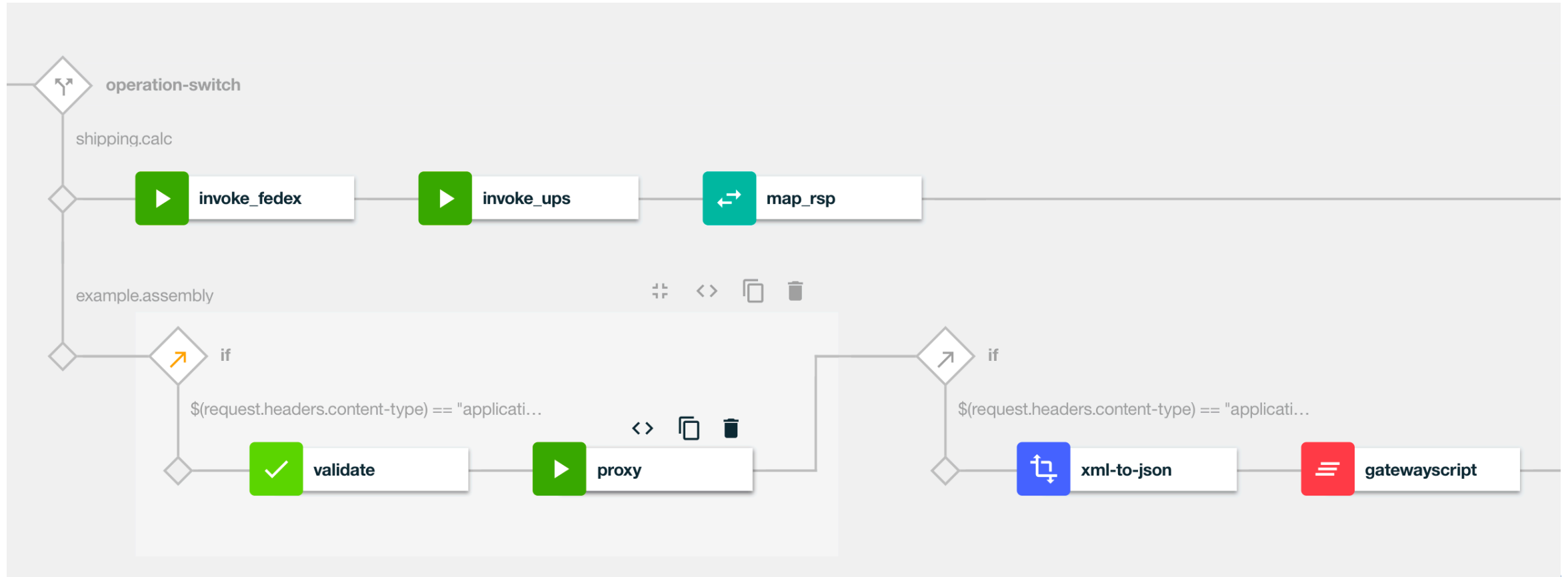
Invoke Panel

- Know what it controls
 - Substitute URL for API catalog environments
- Caching is possible from here
 - Set TTLs for common responses
- Assigning responses to variables
 - Great setup from Map Policy
- Multiple Invokes allowed, only 1 Proxy
 - Proxies for large responses
 - Invokes can be used as needed.

<p>The password to use for HTTP Basic authentication.</p> <p>HTTP Method *</p> <p>Keep</p> <p>The HTTP method to use for the invocation. If omitted or set to 'Keep', then the method from the incoming request will be used.</p> <p><input type="checkbox"/> Compression</p> <p>Used to enable/disable Content-Encoding compression on upload.</p> <p>Cache Type *</p> <p>Protocol</p> <p>The cache type determines whether to cache or overriding the HTTP Cache Control directive response of the Server. This property takes effect only if the Cache Type is set to 'Time to Live', then the method from the incoming request, otherwise returns the non-expired response that was previously cached.</p> <p>Time to Live</p> <p>Sets the validity period in seconds for document. Applies only if the Cache Type is set to 'Time to Live'. The range is 5 - 31708800. The default value is 300.</p> <p>Cache Key</p> <p>The string to use as a key for the cache. If omitted, the request URL string is used as the key.</p> <p><input type="checkbox"/> Stop on error</p> <p>Defines whether the flow stops when a particular error occurs during the policy execution. Errors not specified here will not stop the flow.</p> <p>Response object variable</p> <p>invoke_response</p> <p>The name of a variable that will be used to store the response from the request. This can then be referenced in other policies as 'Map'.</p>	<p>Title</p> <p>Country API Invoke</p> <p>Description</p> <p>URL *</p> <p>\$(api.properties.endpoint-server)</p> <p>The URL to be invoked.</p> <p><input type="checkbox"/> Access URL through Secure Gateway</p> <p>Access the URL through a Secure Gateway. The gateway is set when the API is moved from staged state to published state.</p> <p>TLS Profile</p> <p>The TLS Profile to use for the secure transmission of data.</p> <p>Timeout</p> <p>Time to wait before a reply back from the endpoint. Default is 60 seconds.</p> <p>Username</p> <p>The username to use for HTTP Basic authentication.</p> <p>Password</p> <p>The password to use for HTTP Basic authentication.</p> <p>HTTP Method *</p> <p>The HTTP method to use for the invocation. If omitted or set to 'Keep', then the method from the incoming request will be used.</p> <p><input type="checkbox"/> Compression</p> <p>Used to enable/disable Content-Encoding compression on upload.</p> <p>Cache Type *</p>
--	--

API Designer: Drag and Drop Palette

- Policies apply to all API Endpoints
- We have ability to isolate many different endpoints as well
- All these pre configured policies will configure the API Gateway, making it easier.



API Designer: Scripting through Gateway Script

- Very powerful scripting is available with Gateway script
- Get to know the major endpoints available
- Great link below to variables available
 - https://www.ibm.com/support/knowledgecenter/en/SSMNED_5.0.0/com.ibm.apic.toolkit.doc/rapim_gwscript_codesnip.html

```
// Save the Google Geocode response body to variable
var mapsApiRsp = apim.getvariable('google_geocode_response.body');

// Get location attributes from geocode response body
var location = mapsApiRsp.results[0].geometry.location;

// Set up the response data object, concat the latitude and longitude
var rspObj = {
  "google_maps_link": "https://www.google.com/maps?q=" + location.lat + "," + location.lng
};

// Save the output
apim.setvariable('message.body', rspObj);
```

Lab

