

API Persona Validation

JUNE - AUGUST 2016

Lowell Staveland, lead UX Researcher

Executive Summary

In order to learn more about the prevalent roles within the API development ecosystem we conducted 3 card sorts. Each card sort built upon the findings of the previous, with some additional modifications to drive insightful findings (e.g. having participants self-describe job titles rather than having them select from options).

PARTICIPANTS

Altogether we tested 68 participants

METHOD

The card sorts provided some of the categories, but participants were also able and encouraged to name their own categories.

Two of the card sorts had cards with different tasks involved in the API development cycle, whereas in one of the card sorts we focused solely on tasks related to building and deploying serverless APIs. In each case we were looking to answer the research question: 'How are tasks divided amongst various teams?'.

RESULTS

While initially we had expected to find a consensus within the developer community, what we actually found is the opposite: there is no consensus. Participants were not aligned in how they sorted the cards in any of these studies.

Additionally we couldn't find any participant who described themselves as an API developer, instead participants who are responsible for tasks associated with API developers predominantly described themselves as 'Software Engineers,' and 'Software/Web Developers'.

Table of Contents

Executive Summary	Pg 2
Task-Based Card Sort	Pg 5
Serverless API Card Sort	Pg 12
Task to Role Card Sort	Pg 16
Conclusion	Pg 21
Summary Table	Pg 22

Task-based Card Sort Results

Research Goals

- Explore how tasks during API development are sequenced and categorized.
- Identify if API Connect currently supports how users commonly sequence these tasks.

Participants

25 participants completed the card sorting activity

Method

Prior to completing the card sort we had participants answer several questions about their job title, the type of APIs they create, and the tools they use.

We used a usability testing tool (Optimal Workshop) to generate the card sort with 57 cards. It was a hybrid card sort; the participants were given 7 categories of types of tasks (E.g. Deploy Apps, Test APIs, Monitor API usage, etc.) to sort the cards under, but they could also create their own categories. In case there were tasks they were unfamiliar with, participants were not required to sort all the cards.

Pre-Card Sort Survey

1. What is your job title or description?

Multiple choice, single answer (radio select) - required

API developer		52.6%	10
Application developer		21.1%	4
API architect		5.3%	1
Application architect		10.5%	2
API product manager (api packaging, pricing, manage...		10.5%	2
Security architect		0%	0
Application infrastructure architect		0%	0
Application infrastructure admin		0%	0
Other		0%	0

We need to continue to explore if 'API Developer' is a title that is actually used.

2. Which types of API's have you created?

Multiple choice, multiple answer (checkbox select) - required

Remote API's for use within my company		42.1%	8	▼
Remote API's for use outside my company		31.6%	6	▼
REST API's		57.9%	11	▼
Loopback API's		15.8%	3	▼
Javascript API's		57.9%	11	▼
OATH API's		31.6%	6	▼
XML-RPC		36.8%	7	▼
JSON-RPC		15.8%	3	▼
Legacy API's		10.5%	2	▼

Pre-Card Sort Survey

3. How many APIs and/or APPs have you built?

Multiple choice, single answer (radio select) - required

None		0%	0	▼
1-10		36.8%	7	▼
11-50		52.6%	10	▼
51-100		10.5%	2	▼
>100		0%	0	▼

Most of the participants said they would create between 11-50 APIs.

Next, we need to provide slimmer ranges, or open answer (not multiple choice) to pinpoint more specifically what the average is.

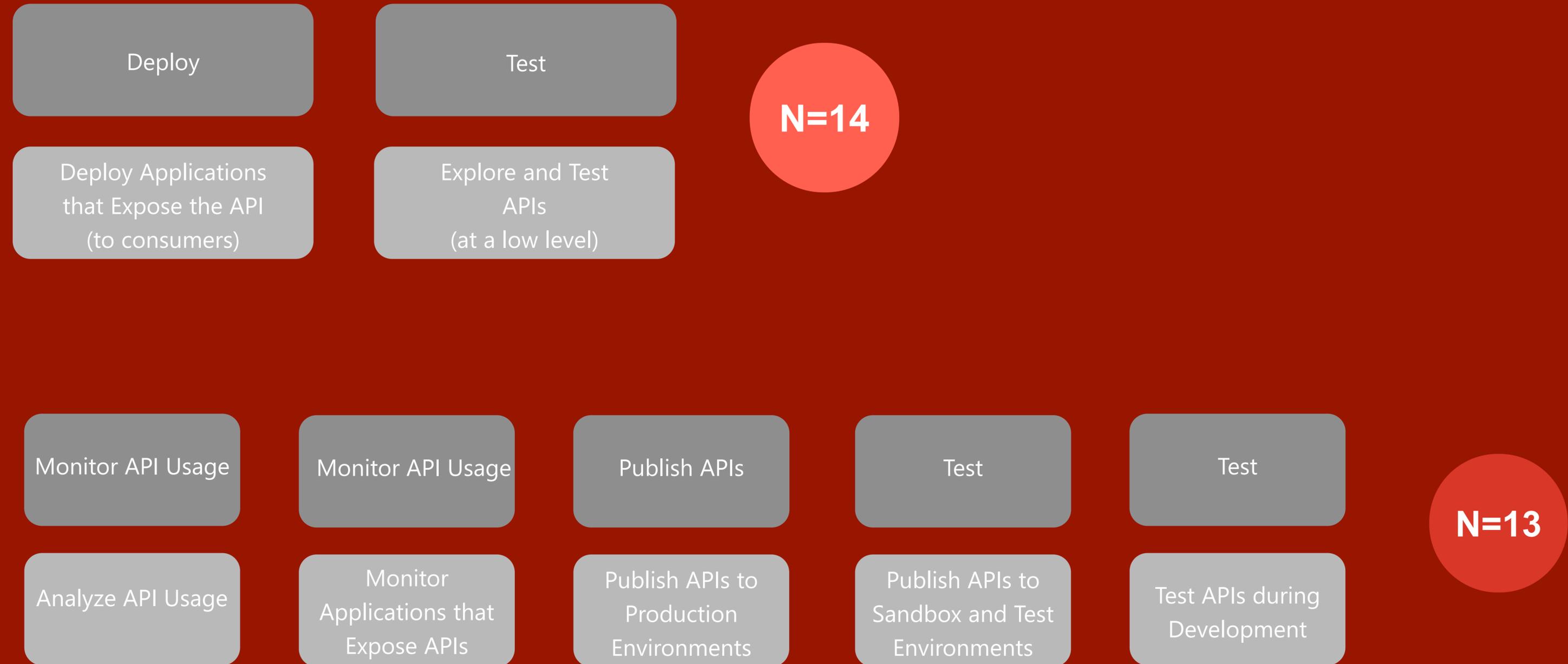
4. Which languages do you develop in?

Multiple choice, multiple answer (checkbox select)

.net		36.8%	7	▼
node.js		47.4%	9	▼
Java		73.7%	14	▼
Perl		5.3%	1	▼
PHP		31.6%	6	▼
Python		42.1%	8	▼
Ruby		15.8%	3	▼
Other		21.1%	4	▼
Unanswered		0%	0	▼

Card Sort Results

Out of the 57 cards, only these 7 tasks cards were commonly paired with the categories we provided. The darker gray are the categories and the lighter gray are the cards that were sorted. Fourteen out of nineteen paired the top two sets together, and thirteen out of nineteen paired the bottom five sets.



Card Sort Results

As you can see, we received a large variety of groupings from our participants. This could be because different organizations categorize and sequence tasks differently or that the API management space is fairly new. It could also be that some of the participants weren't that familiar with the tasks and weren't confident where to put them.

	Build Apps	Create API's	Deploy APPs	Deploy Gateways	Monitor API Usage	Publish APIs	Test APIs
Analyze API up-times, response ti...		1		1	11		2
Analyze API usage (as an API cons...				1	13		1
Analyze revenue generated by API's				1	11		
Apply and configure API policies	1	2	1	2		6	1
Author API's that utilize existing A...	3	7			1	1	
Author applications that expose A...	9	5				2	
Author applications that utilize for...	5	3	3			3	
Configure API authentication (secu...		5	3	1		1	3
Configure API metric thresholds a...		4	2	3	4		2
Configure connections to LDAP us...		4	3	2			2
Create and configure API portals (f...	1	7		4		1	1
Create and share re-usable API po...	2	5			1	5	
Create API catalogs and portals (to...	2	6	1	4		2	
Create API portal accounts	2	3	3	1		2	
Define API security standards and ...	3	2		2	1	1	2
Define API subscription plans (cost...	2	3		1	1	2	2
Define deployment topology for A...	1	3	3	3		1	
Define requirements for API mana...	1	3	1	1		1	2
Deploy applications that expose A...	2		11	3			
Discover and evaluate API manag...	1	5		3			2
Discover API's formally published ...	1	5		1		3	
Evaluate API development and tes...	1	6					5
Evaluate the license terms of form...	1	1			1		6
Explore and test API's (at a low lev...		1			1		14
Explore formally published API's b...		5		1	1		3
Import API's exposed by applicatio...	9	2		2			1
Install and configure application s...	5	2	4	2		1	1

Install and configure application s...	5	2	4	2			1	1	
Install API development and testin...	4	9						4	
Integrate API management with en...	3	1	1	3	3		1		
Integrate API management with lo...	3	2	1		2		1	2	
Manage accounting and billing ass...	1			1	3		1		
Manage API developers and roles ...	1	2	2	1	1		3	1	
Manage API management adminis...		2		2	1		1	2	
Manage API portal administrators ...	1	1		3	2		1		
Manage defects submitted by dev...		1	1		2		1	5	
Manage relationships with API con...	1			1	7		2		
Market API's	2		4				7		
Monitor applications that expose ...	1	1		1	13			1	
Monitor cloud-based API manage...				2	9			1	
Monitor on-premise API managem...				1	11			1	
Notify API consumers regarding o...				1	5		3	2	
Perform detailed, end-to-end trace...	1	1		1	2			7	
Provision cloud-based API manag...	2	1	1	3	3				
Publish API's to production enviro...	1		3	1			13		
Publish API's to sandbox and test ...							5	13	
Respond to API portal account cre...			2		3		3	1	
Respond to notifications when APL...	1	1		2	7			1	
Review (enforce) API security com...			1	2	3		1	4	
Scale in/out application serving en...	2	1	2	1	1		1	1	
Search for formally published API's...	1	5	2				1	1	
Secure API management middlew...	1	1		2	3		3	1	
Sell API's			1		1		11		
Specify/configure API lifecycle mo...	4	6						2	
Submit a request for API usage hel...		3					2	2	3
Subscribe to API usage plans (that...	2	4	1				2	1	
Test API's during development		1	1						13
Troubleshoot application serving e...	2		2	2	1				6

Task-Based Card Sort Categories Provided

- Create APIs
- Test APIs
- Publish APIs
- Monitor API Usage
- Build Apps
- Deploy Apps
- Deploy Gateways

Task-Based Card Sort Cards

- Apply and configure API policies
- Analyze API up-times, response times, volumes, users, etc
- Analyze API usage (as an API consumer)
- Analyze revenue generated by API's
- Author API's that utilize existing API's and services
- Author applications that expose API's (using Java, Node, etc)
- Author applications that utilize formally published API's
- Configure API metric thresholds and responses (alerts)
- Configure connections to LDAP user registries
- Configure API authentication (security settings)
- Create and share re-usable API policies
- Create API catalogs and portals (to organize API's exposed to others)
- Create and configure API portals (for API consumers)
- Create API portal accounts
- Define requirements for API management middleware
- Define deployment topology for API management middleware
- Define API security standards and best practices
- Define API subscription plans (costs, quota's, rates, etc)
- Deploy applications that expose API's (to consumers)
- Discover and evaluate API management middleware
- Discover API's formally published by an organization
- Evaluate API development and testing tools
- Evaluate the license terms of formally published API's prior to consumption
- Explore formally published API's by another organization (at a high level)
- Explore and test API's (at a low level)
- Import API's exposed by applications (Java, Node, etc)
- Install and configure application serving environments
- Install API development and testing tools Integrate API management with enterprise monitoring systems

Task-Based Card Sort Cards Continued...

- Integrate API management with logging systems
- Market API's
- Manage API management administrators
- Manage API developers and roles (publishing and management rights)
- Manage API portal administrators and roles
- Monitor cloud-based API management middleware
- Monitor on-premise API management middleware
- Monitor applications that expose API's
- Manage accounting and billing associated with API subscriptions
- Manage defects submitted by developers who consume formally published API's
- Manage relationships with API consumers via social forums, blogs, etc
- Notify API consumers regarding outages, updates, deprecations, etc
- Provision cloud-based API management middleware
- Publish API's to sandbox and test environments
- Publish API's to production environments
- Perform detailed, end-to-end traces on API operations
- Respond to notifications when API metric thresholds are triggered
- Respond to API portal account creation requests
- Review (enforce) API security compliance
- Scale in/out application serving environments that support API's
- Secure API management middleware
- Sell API's
- Search for formally published API's that perform specific operations
- Specify/configure API lifecycle models
- Subscribe to API usage plans (that may or may not involve charges)
- Submit a request for API usage help to an API provider
- Test API's during development
- Troubleshoot application serving environments that support API's

Serverless APIs Card Sort Results

Research Goals

- Explore how Serverless API development tasks are divided amongst the development team.
- Validate assumed task division for APIC personas.

Participants

We had 16 participants take the first survey, and 4 who successfully completed the survey according to the eligibility criteria.

11 participants responded to the card sort and 3 were included in the final results.

Method

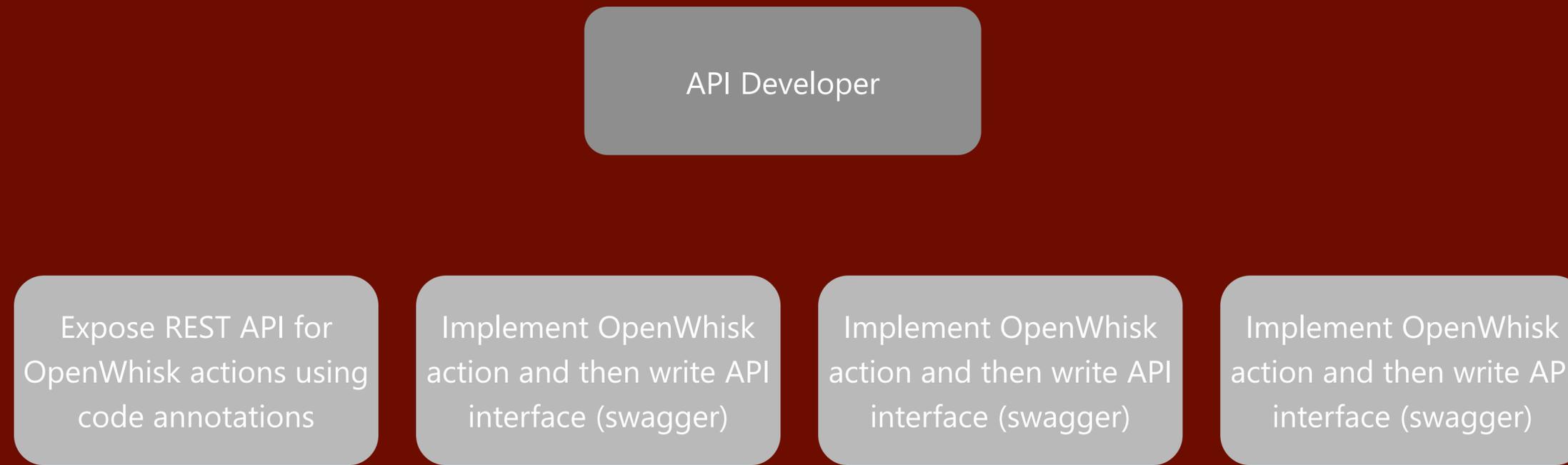
We conducted a card sort with a 3 question survey to find out participant familiarity with server less platforms. The card sort included:

- 14 cards focused on tasks involving OpenWhisk
- 5 categories of roles: API developer, App developer, DevOps, Cloud/Infrastructure admin, and Not sure.

The card sort was a hybrid, meaning that participants could write in another category if they choose to. All of the cards were required to be sorted.

Unique Categorizations

There were 4 tasks that were sorted under the same job title by all three participants included in the results: The API developer not the Open Whisk microservice developer creates the REST API's to expose the services to the end user.



API Developer was also the title (or category) with the highest level of agreement: 66%. This means that 66% of the time the 3 participants agreed (i.e. overlapped) on which tasks were categorized under API developer.

Card Sort Results

Most respondents said that API developers do these tasks. This is where the participants overlapped the most frequently.

This grid shows the frequency with which the cards on the y axis were matched with the categories on the x axis.

These tasks also included the term 'API' and it is possible that the participants were phrase matching. To validate this data, we need to test this with a larger sample size, and by having participants generate the titles on their own.

Our card sorting process continuously evolved so for the next card sort we had participants self-describe their titles instead of having multiple choice.

Standardization Grid ?

	API Developer	App Developer	Cloud/Infrastruct...	DevOps	Not sure
Need recipes with common use c...		1	1		
Troubleshoot the solution so you ...		1	1	1	
Ability to manage traffic including ...			2	1	
Authenticate consumers using ind...	1	1	1		
Expose as a well defined REST AP...	2		1		
Expose REST API for OpenWhisk ...	2		1		
Expose REST API for OpenWhisk ...	1	1	1		
Expose REST API for OpenWhisk ...	3				
Expose REST API for OpenWhisk ...	1	1		1	
Implement OpenWhisk action and...	3				
Implement OpenWhisk action and...	3				
Quickly diagnose and troublesho...	1			1	1
Translate incoming data from inco...		1	2		
Write API interface (swagger) first ...	3				

Card Sort Results

Out of the three participants who were included in the initial card sort screener results: 2 uses AWS Lambda and 1 uses Google Cloud Function. Below are some of the reasons for using Amazon Web Services (AWS) as a “serverless cloud platform”:



“Our company uses Amazon for our cloud environment. We have cloud based server instances and have recently started using Lambda as a way to increase computing power without adding additional instances.”

“I have only used the one from Amazon's Web Services. I liked it. I thought it was reliable and I think Amazon has a great track record.”

Serverless APIs Categories Provided

- API Developer
- App Developer
- Cloud/Infrastructure Admin
- DevOps
- Not sure

Serverless API cards

- Authenticate consumers using industry standards (API Keys, basic authentication OAuth) and support custom domain and associated certificates
- Translate incoming data from incompatible to compatible formats without building your own web server so that your OpenWhisk action is scalable and portable
- Quickly diagnose and troubleshoot issues with the API solution pipeline
- Troubleshoot the solution so you can seamlessly integrate with your existing pipeline
- Need recipes with common use cases for OpenWhisk to expose actions so you can be productive in a short time without using docs.
- Expose as a well defined REST API with support for all HTTP verbs quickly and easily, within a few minutes
- Ability to manage traffic including rate limit, caching, and A/B testing
- Implement OpenWhisk action and then write API interface (swagger)
- Write API interface (swagger) first and then implement OpenWhisk action
- Implement OpenWhisk action and write API interface at the same time
- Expose REST API for OpenWhisk actions using Browser based UI
- Expose REST API for OpenWhisk actions using CLI
- Expose REST API for OpenWhisk actions using Metadata (Swagger and YAML)
- Expose REST API for OpenWhisk actions using code annotations

Task to Role Card Sort Results

Research Goals

- To further identify which roles perform what API development tasks within different organizations
- To assess what roles are unfamiliar or not performed at all within certain organizations

Participants

27 participants participated in the card sort, and 12 were included in the results.

Method

We launched the card sort to 67 participants, of which 27 completed the activity.

Participants were provided 57 cards with various tasks relating to building, testing, and selling APIs. This card sort was hybrid, meaning that they were given two categories, but additionally were able to create their own categories. Participants were instructed to generate categories for the roles or titles that completed each of the tasks written on the cards. They were required to sort all of the cards.

For data analysis, synonymous categories were consolidated (or standardized). E.g. Dev team and developer team.

Unique Categorizations

The number of unique categorizations were fairly high. In other words, they couldn't agree on how the tasks should be categorized.

One potential reason is that the API developer space is new and currently includes a wide range of developers that conduct a wide range of tasks.

The cards with the lowest number of unique categorizations were: 1) Market APIs, and 2) Publish APIs to sandbox and test environment. Again, it's important to note that the lowest number of unique categorizations was 7, meaning that 5 maximum out of 12 agreed on how that card/or task matches a particular role.

Even though the number of unique categorizations is lower, still the roles they stated would complete these tasks were all over the map (from Marketing to Sales to Engineering).

CARD NAME	UNIQUE CATEGORIZATIONS	CATEGORIES	AVG POS	FREQ
Market API's	7	Marketing	3.0	3
		Marketing Manager	2.3	3
		This is not a task that is done at my organization	2.0	2
		Sales	2.0	1
		I don't know who does this task	2.0	1
		Software Engineer	1.0	1
		Sales and Marketing	1.0	1
		Show less		
Publish API's to sandbox and test environments	7	Developer Teams	10.0	3
		Programmers	12.5	2
		Software Engineer	4.5	2
		Testing Team	3.0	2
		Team Leader	5.0	1
		Senior Software Engineer	4.0	1
		Initiation	4.0	1
		Show less		
Explore and test API's (at a low level)	8	Testing Team	3.8	4
		Software Engineer	5.0	2
		Programmers	15.0	1
		API Developer	9.0	1
		Web Developer	8.0	1
		Software Testers	6.0	1
		Tech Analyst	4.0	1
		Developer Teams	2.0	1
		Show less		

Card Sort Results

This grid shows the frequency with which the cards on the y axis were matched with the categories on the x axis.

Overall it shows a lack in consistency in the way the cards were sorted.

The four titles/roles (outlined in black) where most of the tasks were grouped were:

- Developer Teams
- Programmers
- Software Engineer
- Testing team

The most frequent pairing:

Testing team

Explore and test APIs (at a low level)

Standardization Grid

	Accounts Manager	Admin	CTO	Developer Teams	I don't know who...	Marketing	Marketing Mana...	Product Manager	Programmers	Project Manager	Quality Assuranc...	Sales	Software Engineer	Tech Analyst	Testing Team	This is not a task ...
Analyze API up-times, response ti...						1				1				1	1	
Analyze API usage (as an API cons...			1		1	2			1		1			1	1	
Analyze revenue generated by AP...	1		1		1	1				1		1	1	1		1
Apply and configure API policies				2									1	1	2	
Author API's that utilize existing A...				3									1	1	1	1
Author applications that expose A...				3					1				2		1	
Author applications that utilize for...				3				1					2			
Configure API authentication (sec...				2				1	1				1		2	
Configure API metric thresholds a...				2	1			1	1							
Configure connections to LDAP us...					1				1				1	1	1	2
Create and configure API portals (f...				2					2				2			
Create and share re-usable API po...				1					1				1	2		
Create API catalogs and portals (t...									1	2			2	1		
Create API portal accounts		1		1	1				1						2	
Define API security standards and ...			1					1	1				1	1		
Define API subscription plans (cos...	1		1		1								1			2
Define deployment topology for A...			1					1	1	1			1	1	1	
Define requirements for API mana...			2											1		
Deploy applications that expose A...				2					1				1			
Discover and evaluate API manag...			1					1					1	1	1	1
Discover API's formally published ...								1		1			1	1	2	1
Evaluate API development and tes...			1						1		1		1	1	1	
Evaluate the license terms of form...								1		1	1	1	1	1	2	2
Explore and test API's (at a low lev...				1					1				2	1	4	
Explore formally published API's b...				1	1				1	1			2	2	1	
Import API's exposed by applicatio...				2					2		1		2		1	1
Install and configure application s...									1				1		2	
Install API development and testin...									1		1		1		3	
Integrate API management with e...				2				1	1				2		1	1
Integrate API management with lo...				3					1				2		1	
Manage accounting and billing as...	3						2			1			1		1	
Manage API developers and roles ...								2		3			1	1		
Manage API management adminis...		1						1		1					1	2
Manage API portal administrators ...		1												1	2	
Manage defects submitted by dev...					1	1			1	1			1		2	1
Manage relationships with API con...		1				1	1					1	1			2
Market API's					1	3	3					1	1			2
Monitor applications that expose ...									1				1		1	
Monitor cloud-based API manage...			1		1									1		
Monitor on-premise API managem...									1				1		1	1
Notify API consumers regarding o...			1							2					1	
Perform detailed, end-to-end trace...				1					1				1			
Provision cloud-based API manag...			1		1								2	1	1	
Publish API's to production enviro...				1					1				1			1
Publish API's to sandbox and test ...				3					2				2		2	
Respond to API portal account cre...					1			1							1	
Respond to notifications when API...				1	2				1	1						
Review (enforce) API security com...								1	2		1				2	
Scale in/out application serving en...				2				1					1			
Search for formally published API'...					1					1			1	1		2
Secure API management middlew...				1					1				1	1	1	
Sell API's						2	2	1				2				2
Specify/configure API lifecycle mo...				1					2				1	1		
Submit a request for API usage he...		1		1	1					3					1	
Subscribe to API usage plans (that...					1		1		1	2		1				1
Test API's during development									1		2		2		3	
Troubleshoot application serving e...									1		1		1		1	

Task to Role Card sort Categories Provided

- Create APIs
- Test APIs
- Publish APIs
- Monitor API Usage
- Build Apps
- Deploy Apps
- Deploy Gateways

Task to Role Card Sort Cards

- Apply and configure API policies
- Analyze API up-times, response times, volumes, users, etc
- Analyze API usage (as an API consumer)
- Analyze revenue generated by API's
- Author API's that utilize existing API's and services
- Author applications that expose API's (using Java, Node, etc)
- Author applications that utilize formally published API's
- Configure API metric thresholds and responses (alerts)
- Configure connections to LDAP user registries
- Configure API authentication (security settings)
- Create and share re-usable API policies
- Create API catalogs and portals (to organize API's exposed to others)
- Create and configure API portals (for API consumers)
- Create API portal accounts
- Define requirements for API management middleware
- Define deployment topology for API management middleware
- Define API security standards and best practices
- Define API subscription plans (costs, quota's, rates, etc)
- Deploy applications that expose API's (to consumers)
- Discover and evaluate API management middleware
- Discover API's formally published by an organization
- Evaluate API development and testing tools
- Evaluate the license terms of formally published API's prior to consumption
- Explore formally published API's by another organization (at a high level)
- Explore and test API's (at a low level)
- Import API's exposed by applications (Java, Node, etc)
- Install and configure application serving environments
- Install API development and testing tools Integrate API management with enterprise monitoring systems

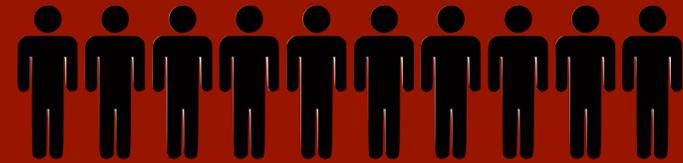
Task to Role Card Sort Cards Continued...

- Integrate API management with logging systems
- Market API's
- Manage API management administrators
- Manage API developers and roles (publishing and management rights)
- Manage API portal administrators and roles
- Monitor cloud-based API management middleware
- Monitor on-premise API management middleware
- Monitor applications that expose API's
- Manage accounting and billing associated with API subscriptions
- Manage defects submitted by developers who consume formally published API's
- Manage relationships with API consumers via social forums, blogs, etc
- Notify API consumers regarding outages, updates, deprecations, etc
- Provision cloud-based API management middleware
- Publish API's to sandbox and test environments
- Publish API's to production environments
- Perform detailed, end-to-end traces on API operations
- Respond to notifications when API metric thresholds are triggered
- Respond to API portal account creation requests
- Review (enforce) API security compliance
- Scale in/out application serving environments that support API's
- Secure API management middleware
- Sell API's
- Search for formally published API's that perform specific operations
- Specify/configure API lifecycle models
- Subscribe to API usage plans (that may or may not involve charges)
- Submit a request for API usage help to an API provider
- Test API's during development
- Troubleshoot application serving environments that support API's

Participant Demographics

While participants identified mostly with tasks represented by the title 'API developer', none of the participants self-described as an API developer. This suggests that the titles currently used for APIC personas and the titles for target APIC users do not align with titles being used at least at this time.

When forced to choose - titles provided



API Developer



Application Developer



API Product Manager
Application Architect



API Architect

When they self-describe



Software Engineer



Software Developer



Web Developer



Manager Of Information Technology Developer
Senior Software Engineer Senior Web Developer
Senior Software Developer

Next Steps

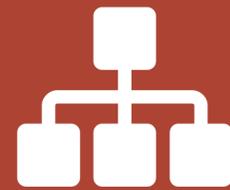
Building on what we learned

The next steps include further investigating the relationship between titles and tasks, cross-validating the data we gathered, and refining the card sort itself for future studies.



INVESTIGATE ROLES

Continue investigating how different companies divide up their labor and what titles are used. This way we can continue validating our personas and their roles.



CROSS VALIDATE

Cross-validate the key findings from this study such as the most common types of APIs, the tools they are created with, and how often they are generated.



REFINE THE CARD SORT

Make some modifications for the next card sort. Run the card sort with a reduced set of cards, simplify the instructions, and ensure titles are self-described.



Conclusion

There reoccurring theme

The results of the three card sorts we conducted indicate that there is a lack of consistency in who performs what tasks during API development in different organizations.

Based on these findings we currently cannot support the underlying base assumption that we had when we initiated these studies, that there are API developers, titled as such. In other words, at this time we could not find anyone who describes themselves as an API developers.

