SYMPRAXIS CONSULTING

Azure Functions: Real world scenarios and NodeJS implementation overview

Julie Turner

PHOTOGRAPHY DURING THE SESSION

Feel free to capture the moment! Taking pictures of the presentation, during the session is perfectly fine.

SELFIES WITH THE SPEAKER

We encourage interaction! If you'd like to take a selfie with the speaker, don't hesitate to ask. Most speakers will be happy to oblige during appropriate breaks.

RECORDING AND LIVE STREAMING

Recording or live streaming the session, in part or in full, is strictly prohibited. Thank you for respecting our content and Speakers.

Julie Turner

Partner/CTO

Working with SharePoint since 2007

Microsoft MVP, Office Apps and Services since 2017

Microsoft 365 & Power Platform Community Team since 2019

Open-source project co-maintainer: PnPjs & hTWOo

Co-host: Code. Deploy. Go Live show

https://julieturner.net/me

SYMPRAXIS CONSULTING

Azure Functions Agenda

What are they?

What are they good for?

Common Architecture

Security

Configuration & Deployment

Q&A

What are they?

- Azure serverless compute
- Run small pieces of code, or "functions," in the cloud without being responsible for the infrastructure.
  - Serverless
  - Event-Driven
  - Scalable
  - Multi-language support

What are they?

- Consumption/Flex Consumption:
  Scales dynamically based on demand. You pay only for the time your functions are running.
- Premium:
  EP1, EP2, EP3: More powerful instances with pre-warmed workers to reduce cold starts. Support virtual network connectivity.
- Dedicated (App Service) Plan:
  Runs on regular App Service plan rates. Suitable for long-running functions and scenarios where you need more control over the infrastructure.

What are they good for?

- Triggered when items in Microsoft 365 change
  - Document drop library
  - Provisioning sites
  - Approval workflows
- Timer Jobs
  - Updating location fields using maps api
  - Syncronizing data (could also be real time)
  - Sending notification emails
- SPFx APIs with OBO flow
  - Elevating permissions through controls code flow

What are they good for?

- **HTTP Trigger**: Executes in response to HTTP requests. This is useful for creating APIs and webhooks.
- **Timer Trigger**: Runs on a specified schedule, similar to a cron job. Ideal for periodic tasks like data cleanup.
- **Queue Trigger**: Activated by new messages in an Azure Storage Queue. Useful for processing background tasks.
- **Event Grid Trigger**: Responds to events from Azure Event Grid, such as resource changes or custom events.
- **Blob Trigger**: Fires when a new or updated blob is detected in Azure Storage. Great for processing files.
- **Service Bus Trigger**: Triggered by messages in an Azure Service Bus queue or topic. Suitable for enterprise messaging scenarios.
- **Cosmos DB Trigger**: Executes when there are changes in an Azure Cosmos DB collection. Useful for real-time data processing

Common Architectures

HTTP + Queue

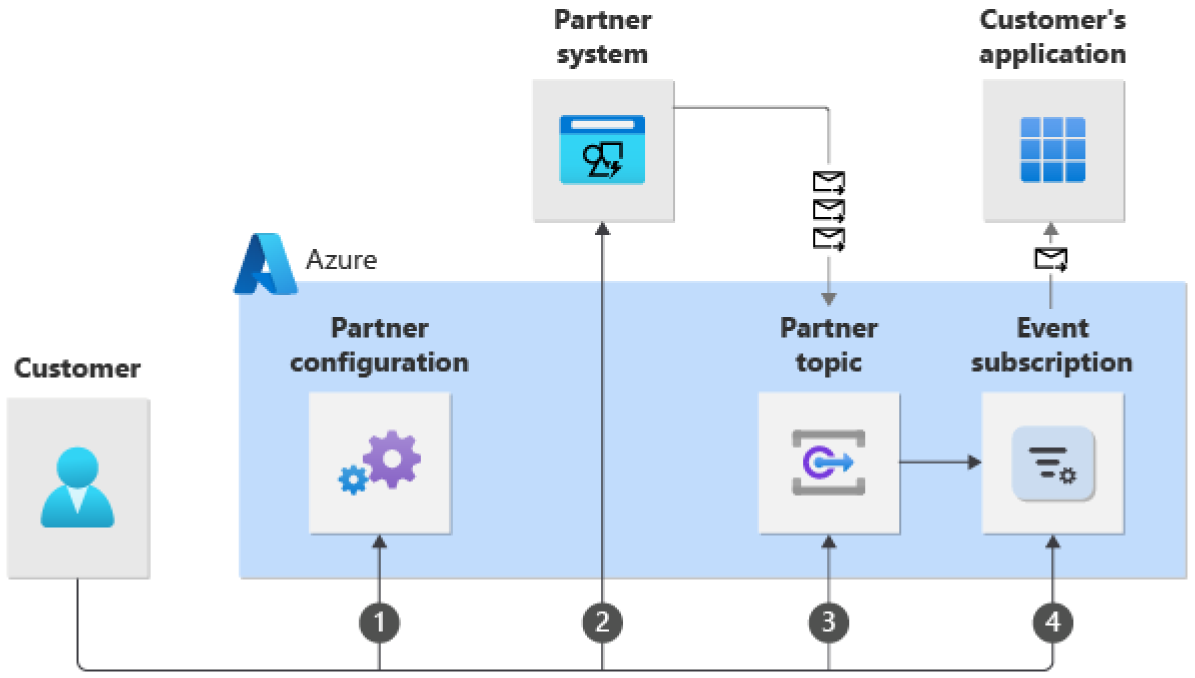
Event Grid + Queue

“Event” plus Queue

- Keeps the code that receives the event as simple as possible to avoid error conditions losing data integrity
- Failed queue items can be configured to reprocess automatically to n retries on t interval.
- Poison queue allows reprocessing of failed queue items.
- Allows you to manually processes events.

Event Grid

1. Authorize partner to create a partner topic in a resource group you designate.
2. Request partner to forward your events from its service to your partner topic. Partner provisions a partner topic in the specified resource group of your Azure subscription.
3. After the partner creates a partner topic in your Azure subscription and resource group, activate your partner topic.
4. Subscribe to events by creating one or more event subscriptions for the partner topic.



Security

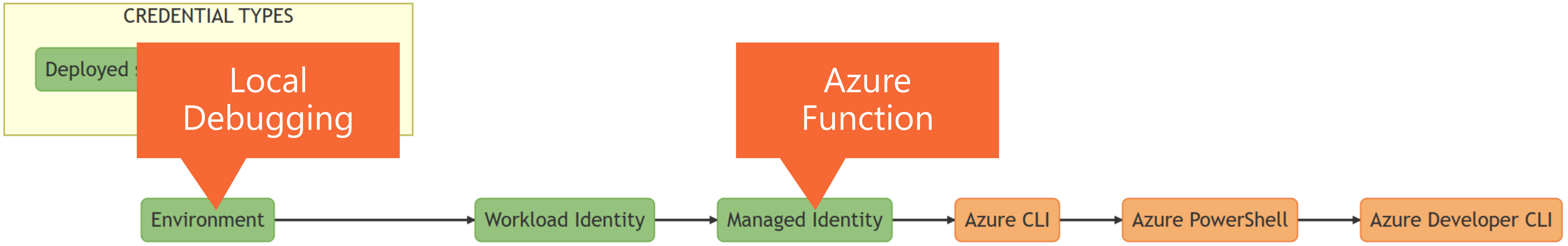
- EntraId App Registrations/ MSAL
- Managed Identity/ Azure Identity
- OBO Flow

SYMPRAXIS CONSULTING

@Azure/Identity

DefaultAzureCredential

- Environment - account information specified via environment variables and use it to authenticate.
- Workload Identity - deployed to Azure Kubernetes Service with Managed Identity enabled
- Managed Identity - deployed to an Azure host with Managed Identity enabled
- Azure CLI - developer has authenticated an account via the Azure CLI az login command
- Azure PowerShell - developer has authenticated using the Azure PowerShell module Connect-AzAccount command
- Azure Developer CLI - developer has authenticated an account via the Azure Developer CLI azd auth login comman

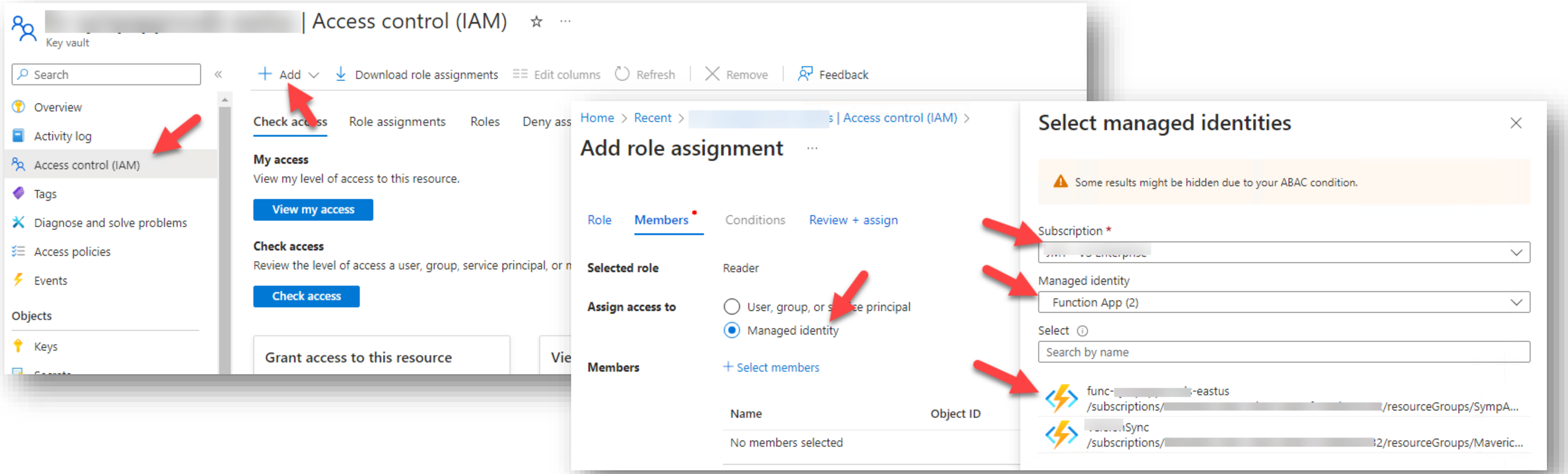


Managed Identity

Managed identities in Azure are a service that allows Azure resources to authenticate cloud services without the need for storing credentials in code or configuration files.

- You don't need to manage credentials.
  - Credentials aren’t even accessible to you.
- You can use managed identities to authenticate to any resource that supports Microsoft Entra authentication, including your own applications.
- Managed identities can be used at no extra cost.
- System-assigned
  - 1:1 relationship with the azure resource and its lifecycle is tied to the resource
- User-assigned
  - 1:many relationship to azure resources
- You authorize a managed identity to have access to one or more services

Authorizing a managed identity



My access
View my level of access to this resource.

Check access
Review the level of access a user, group, service principal, or m

Selected role: Reader

Assign access to: User, group, or service principal [✓ Managed identity]

Members: No members selected

Some results might be hidden due to your ABAC condition.

Subscription *: [redacted]

Managed identity: Function App (2)

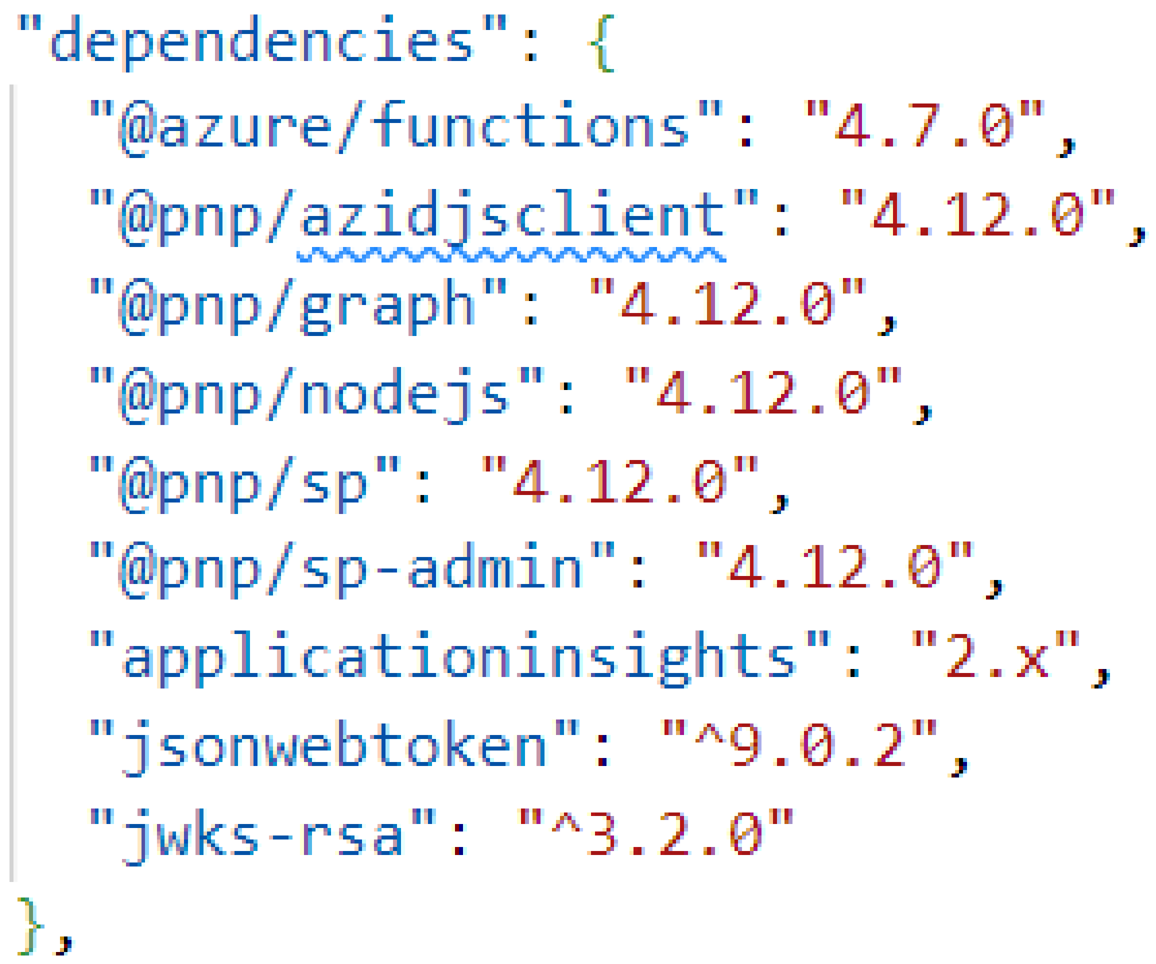
func-[redacted]-eastus
/subscriptions/[redacted]/resourceGroups/SympA...

[redacted]Sync
/subscriptions/[redacted]2/resourceGroups/Maveric...

```
m365 login --authType browser

m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "Microsoft Graph" --scopes "Files.ReadWrite.All,Group.
Read.All,Mail.Send,User.Read.All"

m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "SharePoint" --scopes "Sites.FullControl.All"

m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "SharePoint" --scopes "TermStore.ReadWrite.All"
```

OBO (On Behalf Of) Flow

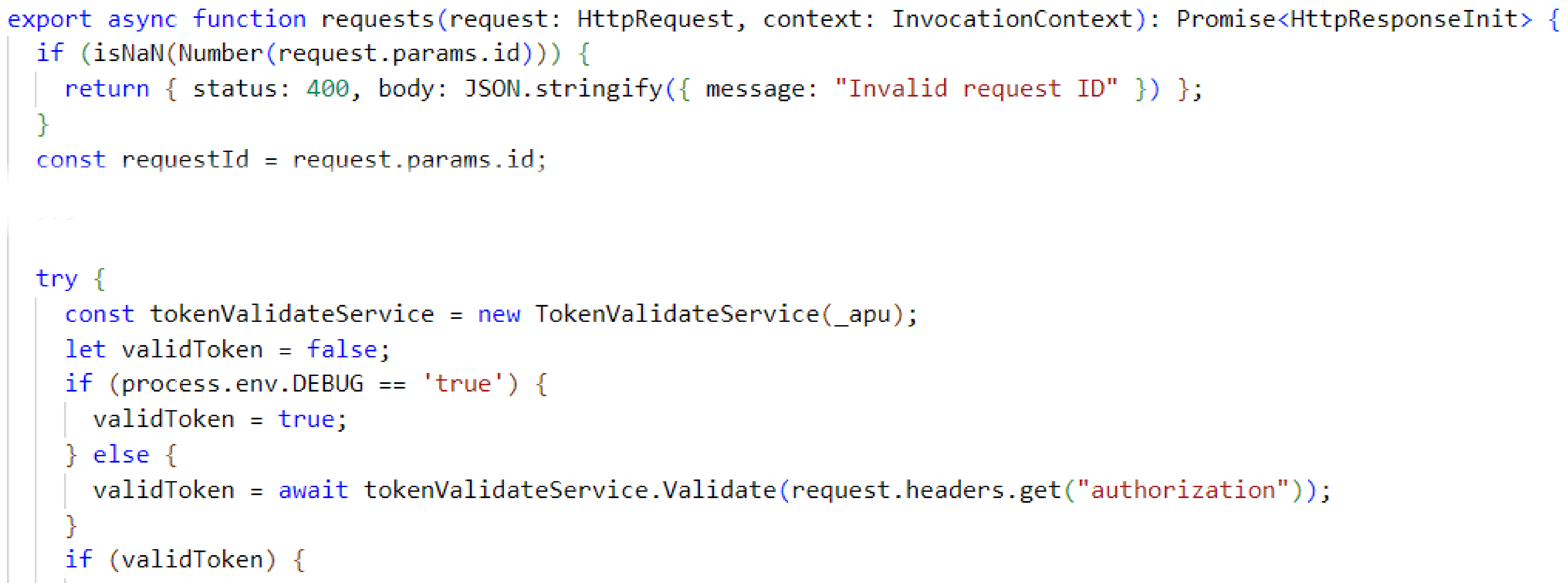
- Request header will have authorization token
- Validate the token
- Optionally, get the user that made the request from the token



```
"dependencies": {
  "@azure/functions": "4.7.0",
  "@pnp/azidjsclient": "4.12.0",
  "@pnp/graph": "4.12.0",
  "@pnp/nodejs": "4.12.0",
  "@pnp/sp": "4.12.0",
  "@pnp/sp-admin": "4.12.0",
  "applicationinsights": "2.x",
  "jsonwebtoken": "^9.0.2",
  "jwks-rsa": "^3.2.0"
},
```

SYMPRAXIS CONSULTING

Validate

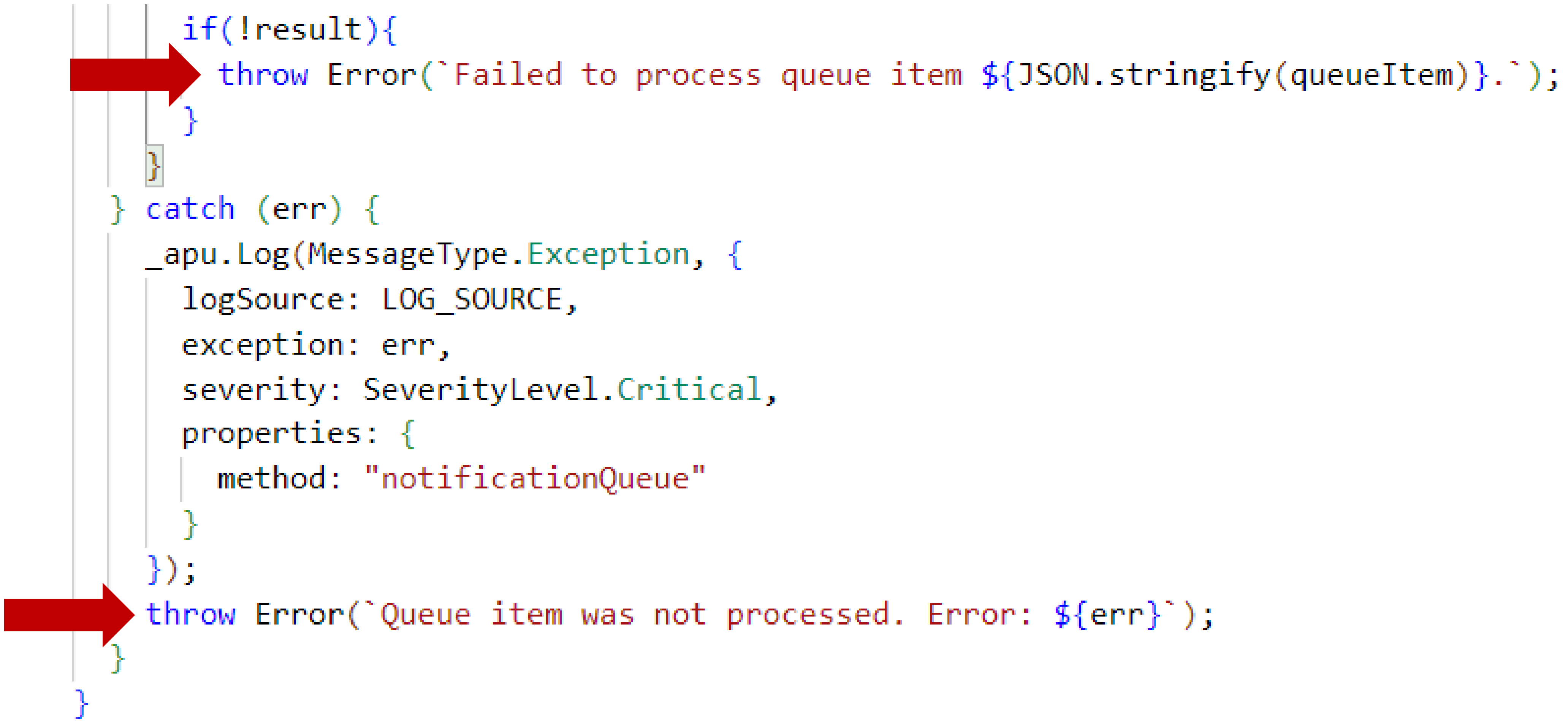


```
export async function requests(request: HttpRequest, context: InvocationContext): Promise<HttpResponseInit> {
  if (isNaN(Number(request.params.id))) {
    return { status: 400, body: JSON.stringify({ message: "Invalid request ID" }) };
  }
  const requestId = request.params.id;

  ...

  try {
    const tokenValidateService = new TokenValidateService(_apu);
    let validToken = false;
    if (process.env.DEBUG == 'true') {
      validToken = true;
    } else {
      validToken = await tokenValidateService.Validate(request.headers.get("authorization"));
    }
    if (validToken) {
```

Configuration & Deployment Demo

- New Project
- NodeJS – CommonJS vs ESModules setup
- Authentication
- Application Insights – logging
- The three most common triggers
  - HTTP
  - Queue
  - Timer
- Deploying & a word on BICEP

Using Queues - Requeuing

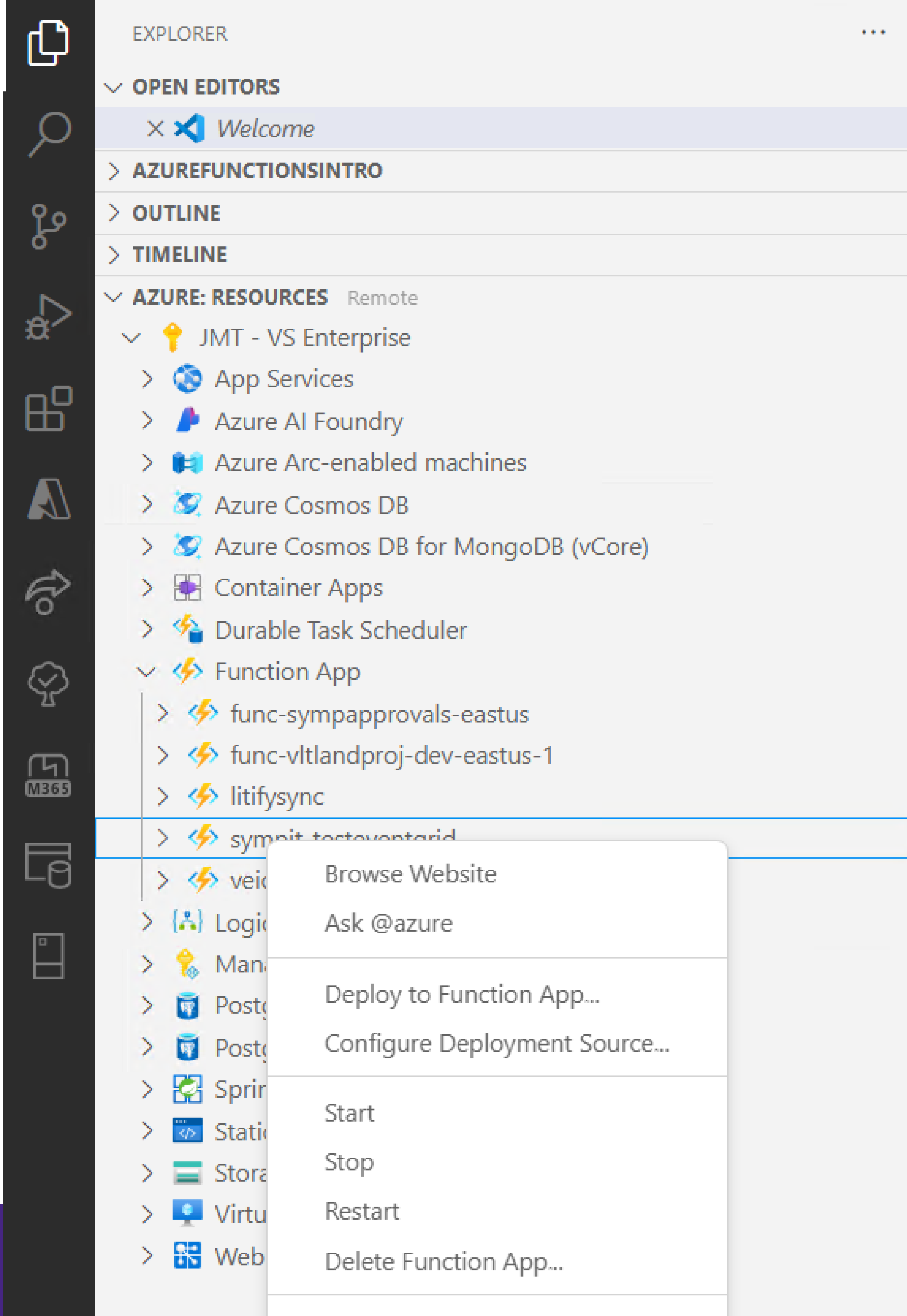


```
        if(!result){
          throw Error(`Failed to process queue item ${JSON.stringify(queueItem)}.`);
        }
      }
    } catch (err) {
      _apu.Log(MessageType.Exception, {
        logSource: LOG_SOURCE,
        exception: err,
        severity: SeverityLevel.Critical,
        properties: {
          method: "notificationQueue"
        }
      });
      throw Error(`Queue item was not processed. Error: ${err}`);
    }
  }
```

SYMPRAXIS CONSULTING

Deployment

- Right-Click deploy with Azure Tools extension.
- Create CI/CD pipeline with Azure DevOps or GitHub Actions
- Add IaC with BICEP files defining your Azure resources.



EXPLORER ...

OPEN EDITORS
- Welcome

AZUREFUNCTIONSINTRO

OUTLINE

TIMELINE

AZURE: RESOURCES Remote
- JMT - VS Enterprise
  - App Services
  - Azure AI Foundry
  - Azure Arc-enabled machines
  - Azure Cosmos DB
  - Azure Cosmos DB for MongoDB (vCore)
  - Container Apps
  - Durable Task Scheduler
  - Function App
    - func-sympapprovals-eastus
    - func-vltlandproj-dev-eastus-1
    - litifysync
    - sympnit-testeventgrid
    - veic
  - Logic
  - Mana
  - Postg
  - Postg
  - Sprir
  - Static
  - Stora
  - Virtu
  - Web

Browse Website
Ask @azure

Deploy to Function App...
Configure Deployment Source...

Start
Stop
Restart
Delete Function App...

SYMPRAXIS CONSULTING

Summary

- Discussed common use cases and architectural patterns
- Discussed security, including MSAL, Azure Identity, and OBO Flows
- Created a NodeJS project and configured it
- Added helper functions for logging and authentication
- Reviewed some of the most common trigger types
- Discussed separation of business logic
- Quick review of deployment and automation

Resources

VSC Extension: Azure Tools

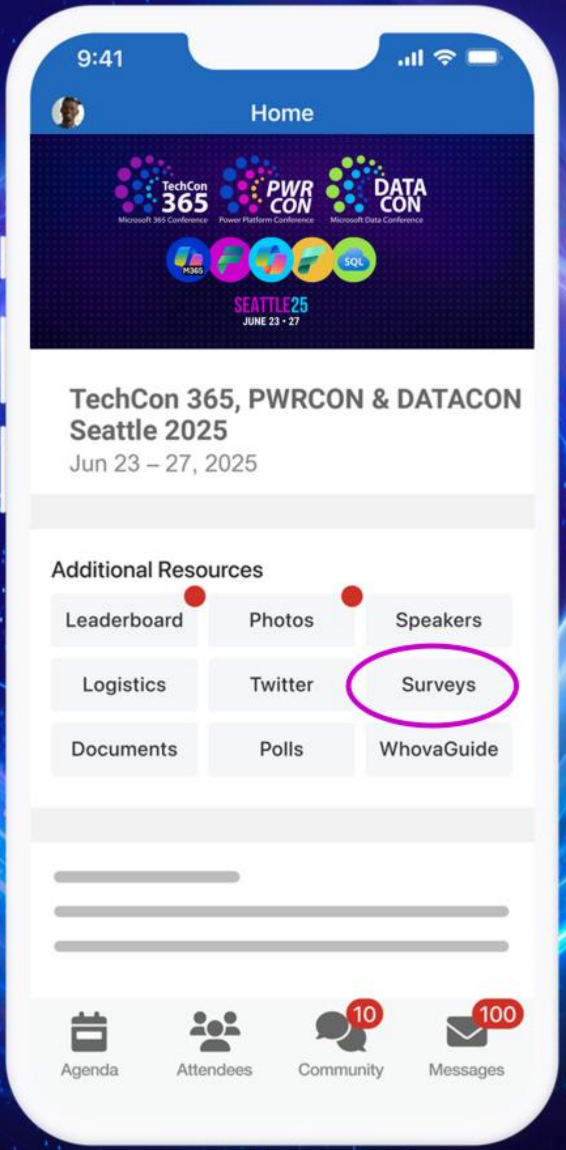
VSC Extension: Azure Functions

Partner Events overview for customers - Azure Event Grid

@azure/identity NPM Package

What are managed identities for Azure resources?

How to: CI/CD/IaC for Azure Function Apps and GitHub Actions


9:41

Home

TechCon 365, PWRCON & DATACON Seattle 2025
Jun 23 – 27, 2025

Additional Resources

Leaderboard	Photos	Speakers
Logistics	Twitter	Surveys
Documents	Polls	WhovaGuide

How was the session?

Search for Whova in the App Store or Google Play

Fill out the Session Surveys in the TechCon 365, PWRCON & DATACON Event App and be eligible to WIN PRIZES!