

# Data API Builder: scenari di utilizzo



Sabato 30 settembre 2023



Danilo Dominici



# SPONSORS



# Danilo Dominici



Consulente, trainer, speaker ed autore specializzato su SQL Server sin dalla versione 6.5  
Da luglio nel team di **Altitude!**



Che cosa faccio: progettazione, implementazione ed ottimizzazione architetturale, monitoring e performance tuning di soluzioni basate su SQL Server, on-premise e in cloud (Azure/AWS), PostgreSQL e Redis



Microsoft Certified Trainer dal 2000, Data Platform MVP (2014 → 2020)



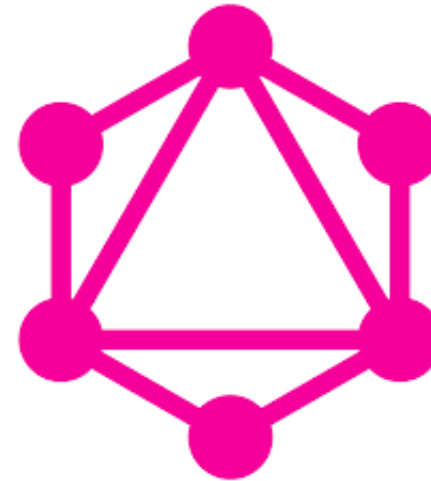
Autore di alcuni capitoli delle guide Microsoft sull'aggiornamento di SQL Server 2012 e 2014  
Co-autore del libro "Azure Data Solutions: an overview"

# REST vs GraphQL



**Representational state transfer (REST)** è uno stile architetturale per sistemi distribuiti. Introdotto nel 2000, rappresenta un sistema di trasmissione dati su HTTP, senza altri livelli (es. SOAP)

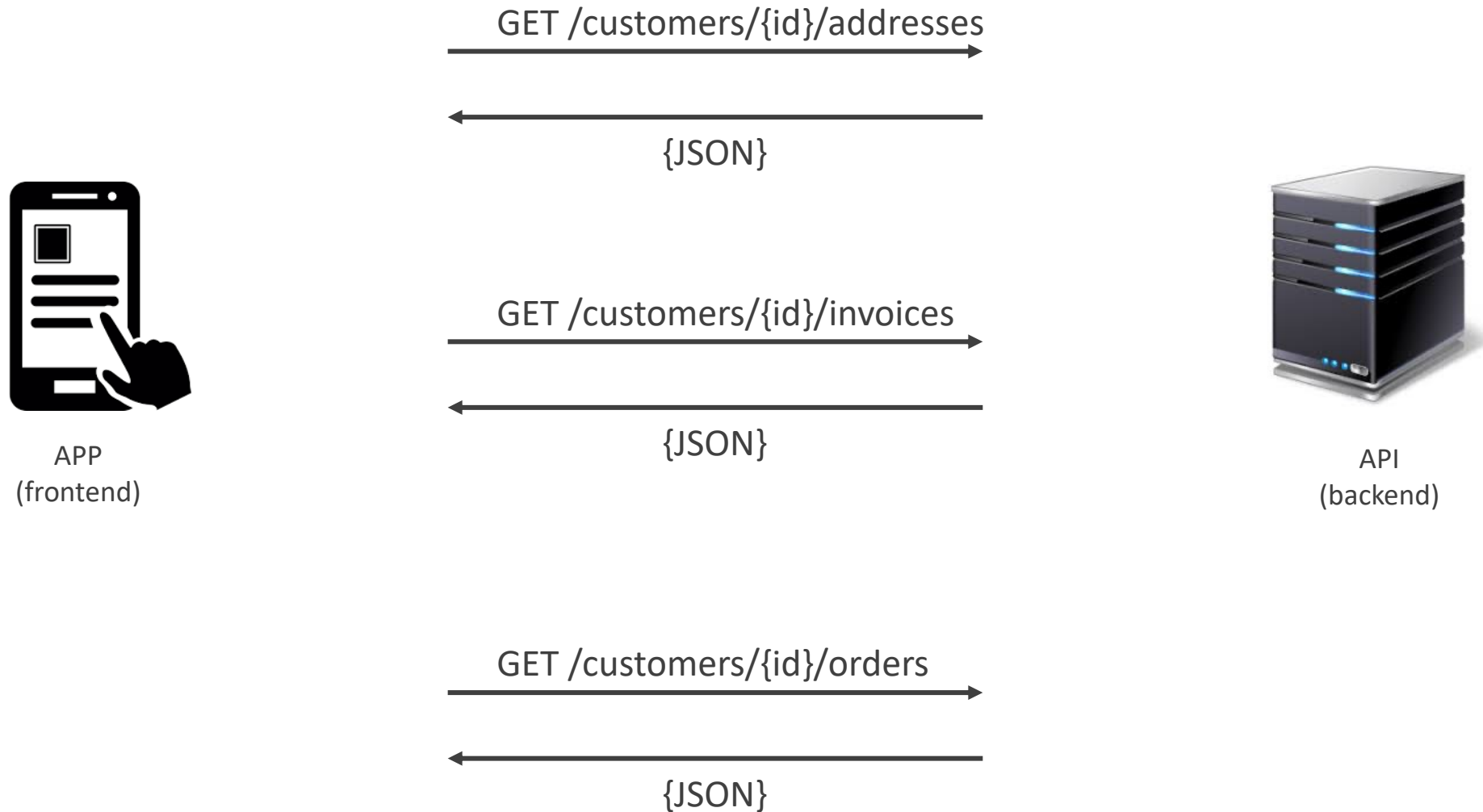
**GraphQL** è un linguaggio di interrogazione e manipolazione dei dati open-source per API e un runtime per eseguire interrogazioni sui dati. Nato internamente a Facebook nel 2012 è stato reso pubblico nel 2015.



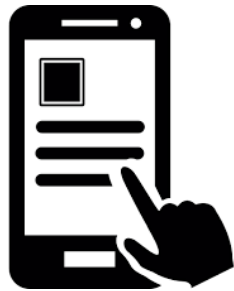


Graphql

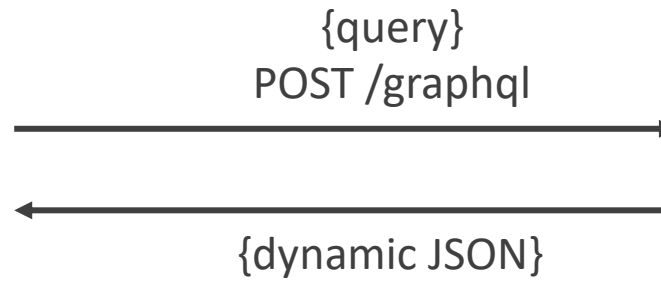
# Client-server communication (REST)



# Client-server communication (GraphQL)



APP  
(frontend)



API  
(backend)

# Concetti chiave in GraphQL



*SCHEMA*



QUERIES



MUTATIONS



SUBSCRIPTIONS



# Terminologia

- Fields
- Arguments
- Aliases
- Fragments
- Variables
- Directives
- Mutations

```
{
  hero {
    name
  }
  human(id: "1000") {
    empireHero: hero(episode: EMPIRE) {
      leftComparison: hero(episode: EMPIRE) {
        name
      }
    }
  }
  query HeroNameAndFriends($episode: Episode) {
    hero(episode: $episode) {
      name
    }
    query Hero($episode: Episode, $withFriends: Boolean!) {
      hero(episode: $episode) {
        name
      }
    }
  }
  mutation CreateReviewForEpisode($ep: Episode!) {
    createReview(episode: $ep, review: $review) {
      stars
      commentary
    }
  }
}
```

## VARIABLES

```
{
  "ep": "JEDI",
  "review": {
    "stars": 5,
    "commentary": "This is a great movie!"
  }
}
```

```
{
  "data": {
    "hero": {
      name: "Luke Skywalker"
    }
  }
}

{
  "data": {
    "hero": {
      leftComparison: {
        name: "Luke Skywalker"
      }
    }
  }
}

{
  "data": {
    "hero": {
      name: "Luke Skywalker"
    }
    query: {
      hero: {
        name: "Luke Skywalker"
      }
    }
  }
}

{
  "data": {
    "createReview": {
      stars: 5,
      commentary: "This is a great movie!"
    }
  }
}
```

data api builder

# Che cos'è Data API Builder ?

Esattamente  
quello che sembra!

Esponde le  
operazioni CRUD  
sui dati di un  
database

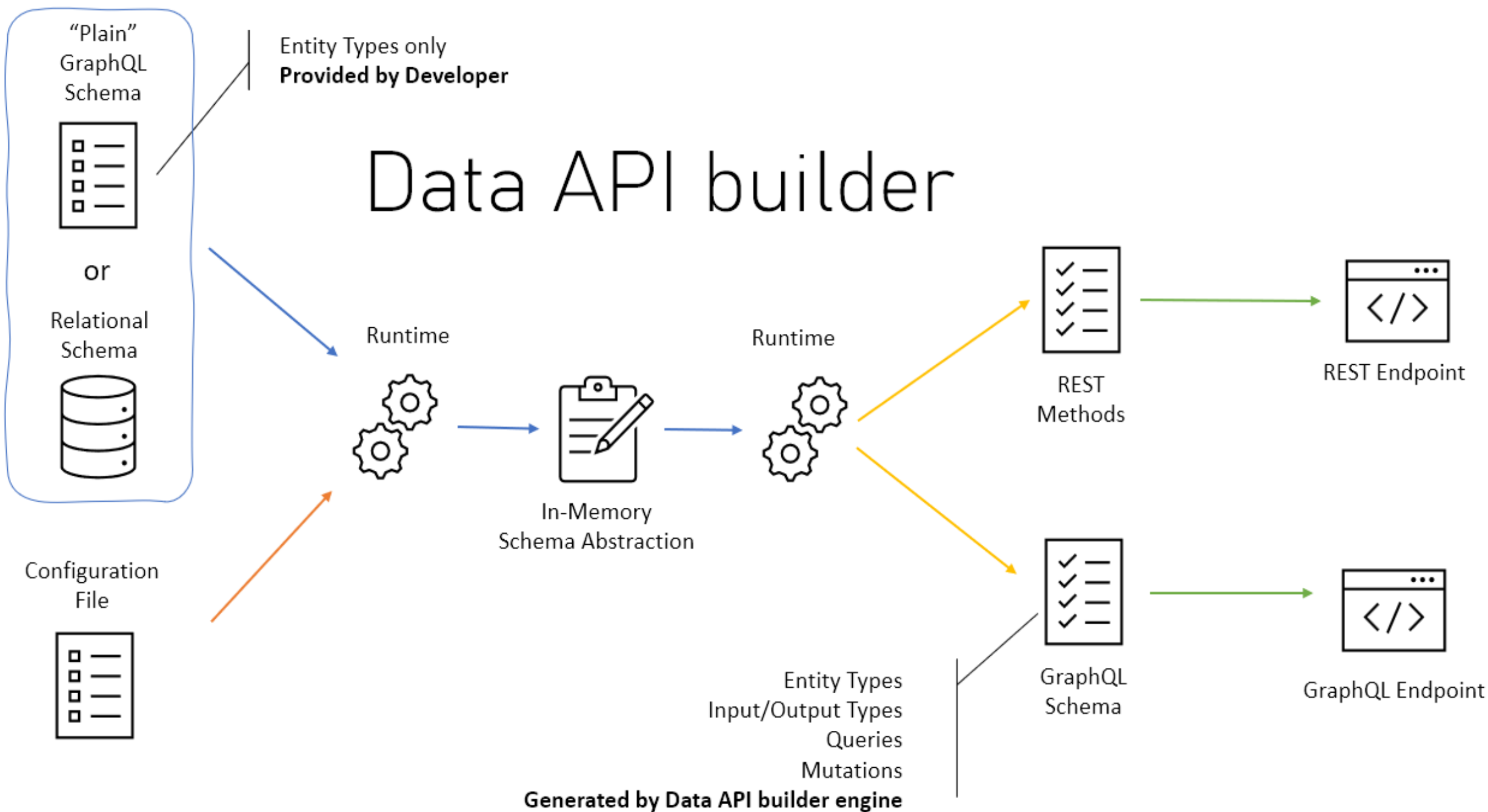
Cross-platform e  
open source

Sicuro

RDBMS (SQL  
Server,  
PostgreSQL,  
MySQL)

NoSQL (Cosmos  
DB)





# Le caratteristiche principali (versione attuale 0.8.52)

- Consente di esporre collezioni, tabelle, viste e stored procedures come API REST e GraphQL
- Supporta l'autenticazione via OAuth2/JWT
- Supporta EasyAuth se eseguito in Azure
- Role-based authorization utilizzando i claims ricevuti
- Item-level security via policy expressions
- REST
  - Operazioni CRUD via POST, GET, PUT, PATCH, DELETE
  - filtering, sorting e pagination
- GraphQL
  - queries e mutations
  - filtering, sorting e pagination
  - relationship navigation

# Le caratteristiche principali (versione attuale 0.8.52)

- Sviluppo semplificato via CLI
- Integrazione completa con le Static Web Apps via Database Connection (se utilizzato in Azure)
- Open Source

# Installare Data API Builder



Verificare che sia installato il .NET Core 6.0  
*dotnet --list-sdks*



Se necessario, installare il .NET Core SDK 6.0



Installare Data Api Builder  
*dotnet tool install -g Microsoft.DataApiBuilder*

In caso di problemi...

<https://learn.microsoft.com/en-us/azure/data-api-builder/troubleshoot-installation>

<https://learn.microsoft.com/en-us/dotnet/core/tools/troubleshoot-usage-issues#nuget-feed-cant-be-accessed>

*dotnet tool install -g --add-source 'https://api.nuget.org/v3/index.json' --ignore-failed-sources Microsoft.DataApiBuilder*

# Usare DAB CLI



Inizializzare il file di configurazione  
*dab init [opzioni]*



Aggiungere le entità del database  
*dab add [entità] [opzioni]*



Aggiornare le proprietà delle entità importate  
*dab update [entità] [opzioni]*



Avviare il web server DAB  
*dab start*



# Il file di configurazione

## Data source

Database type and connection string

```
{
  "$schema": "https://github.com/Azure/data-api-builder/releases/download/v0.8.51/dab.draft.schema.json",
  "data-source": {
    "database-type": "mssql",
    "connection-string": "Server=localhost;Database=Northwind;User ID=demouser;Password=D3moP@ssw0rd!;TrustServerCertificate=true",
    "options": {
      "set-session-context": false
    }
  }
}
```

## Runtime settings

Runtime behavior  
(REST/GraphQL configuration)

```
,
  "runtime": {
    "rest": {
      "enabled": true,
      "path": "/api"
    },
    "graphql": {
      "enabled": true,
      "path": "/graphql",
      "allow-introspection": true
    },
    "host": {
      "cors": {
        "origins": [],
        "allow-credentials": false
      },
      "authentication": {
        "provider": "StaticWebApps"
      },
      "mode": "development"
    }
  },
  "entities": {}
}
```

## Entity definitions

Database object mappings

```
"entities": {}
}
```

# I data source disponibili



*Azure SQL / Azure SQL MI / SQL Server*



PostgreSQL



MySQL



Cosmos DB NoSQL API (JSON documents)



Cosmos DB for PostgreSQL

# Le altre opzioni di configurazione



REST



GraphQL



Autenticazione, CORS, modalità di utilizzo (dev/prod)

# Le entità



Tabelle



Viste



Stored Procedures



Collezioni

# Best practices



*Utilizzare PascalCase*  
(tutte le parole iniziano con una lettera maiuscola)



Utilizzare il nome delle entità al singolare  
(Es. Cliente, non Clienti)

# DEMO – Data Api Builder

Powershell

# Perché Powershell ?



Esponendo le API CRUD del mio database, posso creare uno strumento veloce di manipolazione dei dati senza avere nessun ambiente di sviluppo!



Sapevate che da Powershell possiamo creare Windows Forms ?

Es. [https://www.sapien.com/software/powershell\\_studio](https://www.sapien.com/software/powershell_studio)





Azure

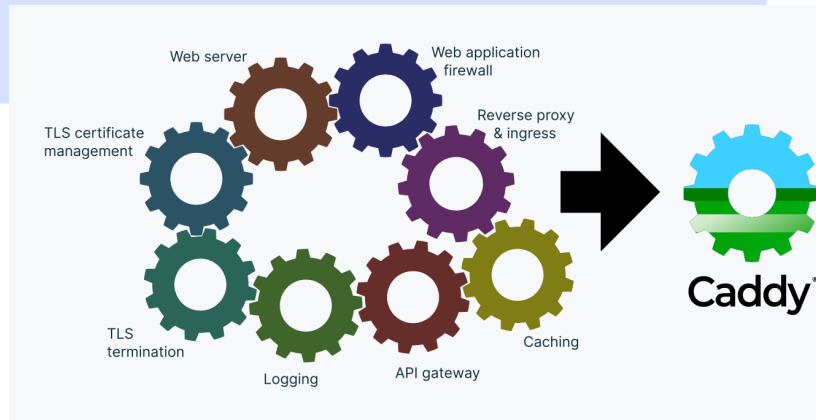
# Data API Builder in un Azure Container Instance (sicuro)



Microsoft ha pubblicato DAB anche come container nel registry ([mcr.microsoft.com](https://mcr.microsoft.com))



Si può esporre in maniera sicura attraverso un proxy (es. Caddy) che implementa in maniera automatica TLS 1.3



# Data API Builder in Azure Container step-by-step

```
az group create --location westeurope --name 1nn0va
```

```
az sql server create --name sql-1nn0va --resource-group 1nn0va --admin-password "xxxxxx" --admin-user "sqladmin"
```

```
az sql db create --name sqldb-adventureworks --resource-group 1nn0va --server sql-1nn0va --backup-storage-redundancy Local --edition Basic --capacity 5 --max-size 2GB --sample-name AdventureWorksLT
```

```
az sql server firewall-rule create --server sql-1nn0va --resource-group 1nn0va --name AllowAzureServices --start-ip-address 0.0.0.0 --end-ip-address 0.0.0.0
```

```
az storage account create --name 1nn0vadabdemo --resource-group 1nn0va --location westeurope
```

```
az storage share create --name dab-config --account-name 1nn0vadabdemo
```

```
az storage share create --name proxy-caddyfile --account-name 1nn0vadabdemo
```

```
az storage share create --name proxy-config --account-name 1nn0vadabdemo
```

```
az storage share create --name proxy-data --account-name 1nn0vadabdemo
```

```
az container create --resource-group 1nn0va --file ci-adventureworkslt-api.yaml
```

# Data API Builder in Azure Container step-by-step

## File di configurazione Caddy

```
1nn0va-dab-api-demo.westeurope.azurecontainer.io {  
    reverse_proxy http://localhost:5000  
}
```

... il file .yaml per la configurazione del Azure Container Instance lo trovate su GitHub



Docker



# Data API Builder in un container Docker per lo sviluppo



Esposizione rapida di tutti o parte dei dati di un database di sviluppo



Zero (o quasi) tempo di setup

```
docker run -it -v "C:\dab\config:/App/configs" -p 5000:5000 mcr.microsoft.com/azure-databases/data-api-builder:latest  
--ConfigFileName ./configs/dab-config.json
```



Power BI



# Data API Builder come sorgente per Power BI



Si può utilizzare sia tramite REST che GraphQL (demo tra un attimo)



Utile per includere «al volo» dati per i quali non abbiamo ancora API disponibili e dobbiamo testare il modello





Excel

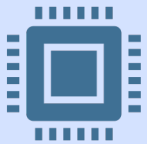


Oracle APEX

# Che cos'è Oracle APEX ?



Tool di progettazione rapida di applicazioni gestionali e di reportistica



Può essere installato su un Oracle esistente o si può utilizzare nello stack cloud gratuito di Oracle



Tra le altre cose, consente di interagire con API esterne 😊

Thanks!



Sabato 30 settembre 2023

Marco Parenzan

