Retirement of SQL 2014-04-01 API

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:25 AM PST

Contents

We are retiring 2014-04-01 Azure SQL DB APIs on **31 October 2025**. To avoid system disruptions, customer should follow the migration guidance to migrate before the final deprecation date. **The retirement communication is planned to send to subscription owners on October 31, 2022.**

Public Documentation

Migration guide

Escalate to PG

Before escalation, you can send email to azuresqldbapiretirement@service.microsoft.com for help. If there is no response, check with EEE Charlene Wang or Soma Jagadeesh for help.

- If the support case is using correct problem type "Portal and Client Tools" and problem subtype "Problem Retire Older Version Azure SQL APIs", then escalate through ASC will choose correct ICM template by default.
- If the support case is not using correct problem type and subtype, make sure change them before escalation
- If no support case is linked, please use ICM template https://portal.microsofticm.com/imp/v3/incidents/create?tmpl=52TF2s to raise Sev 3 with tag [SQL API retirement] to "Azure SQL DB\Provisioning and Elastic Jobs" through ASC.

Schedule

1. ~~Release 2021-11-01 stable API version~~
2. ~~Send out retirement notice~~
3. Bump up the SDK versions and release downstream clients with 2021-11-01 [in progress]
4. Update the API usage in Azure Portal.

5. Update ARM templates in Azure Portal.

FAQ

1. Is customer still able to use it until October 31, 2025?

Yes, no error will show up. But they will start to receive errors after October 31, 2025. It is suggested to evaluate and migrate to latest stable version asap.

2. Is there any replacement for all APIs?

This depends on APIs. Some of them are complete retirement without any newer version. Refer to Migration guide for more details.

3. Why customers who don't have any SQL resource created but receive this notification?

- From customer side, subscription owner can open Azure portal -> Subscriptions -> Resource Providers, search for **"Microsoft.Sql"**, if this is registered, they will receive the notification. If there is no SQL resource created in the subscription, they can ignore the notification email or unregister the resource provider to avoid confusion. Customer will not be charged.
- CSS could also use ASC to confirm whether one customer is using Microsoft.SQL Database resource(s).
- From Kusto, you can run below query to verify the registration status and date.

```
Execute: [Web] [Desktop] [Web (Lens)] [Desktop (SAW)] https://armprod.kusto.windows.net/CosmosToKusto
SubscriptionRegistrations
| where RegistrationState =~ "registered"
| where ResourceProviderNamespace =~ "Microsoft.SQL"
| where SubscriptionId == '<customer's subscription id>' or SubscriptionId == '<customer's subscription id>'
```

4. How to monitor 2014-04-01 SQL API usage internally?

We have an internal way for the API version usage from customers' sub id. It is based on customers' support ticket requests.

```
Execute in [Web] [Desktop] [cluster('armprod.kusto.windows.net').database('ARMProd’)]
HttpIncomingRequests
| where TIMESTAMP > ago(14d)
| where subscriptionId =~ "subscriptionId"
| where targetResourceProvider == 'MICROSOFT.SQL'
| where apiVersion == "2014-04-01" or apiVersion == "2014-01-01" or apiVersion == "2014-04-01-preview"
| summarize CallsCount = count() by operationName, apiVersion = tolower(apiVersion)
| project OperationName = operationName, CallsCount, apiVersion
| order by OperationName desc
```

Above query could return the internal usage for old APIs. To verify old API calls from customer, you can use below query:

```
HttpIncomingRequests
| where TIMESTAMP > ago(14d)
| where subscriptionId =~ 'subscription_id'
| where clientApplicationId != "c44b4083-3bb0-49c1-b47d-974e53cbdf3c" // this appId is portal
| where isnotempty(userAgent) //userAgent should not be empty
| where userAgent != "Microsoft-ChangeAnalysis" // this seems like internal microsoft (ARG?)
| where userAgent != "azure-resource-manager/2.0" // arm template ? Linked notifications?
| where userAgent != "PolicyScan" // Azure policy
| where targetResourceProvider == 'MICROSOFT.SQL'
| where apiVersion == "2014-04-01" or apiVersion == "2014-01-01" or apiVersion == "2014-04-01-preview"
| summarize CallsCount = count() by operationName, apiVersion = tolower(apiVersion), userAgent
| project OperationName = operationName, CallsCount, apiVersion, userAgent
| order by OperationName desc
```

4. Why doesn't my customer receive the notification even they have sql database resource registered?

Here is an ICM track those missing notifications: https://portal.microsofticm.com/imp/v3/incidents/details/350396909/home Due to excel rows limitation, there are some subscriptions missing when CXP team send the notification email.

Internal reference:

- ICM 346078480
- CSS Mentoring Sessions: Recording PPT

5. Why did my customer did not received a retirement notification on Oct 31, 2022? But they received an additional retirement email in early December.

The subscription data provided for customers targeted by this subscription was truncated when viewed by my team due to row limitations within Excel due to it being reached to the limit. The missing subscriptions were identified, and the communication has been published to those subscriptions as well. Processes have been stood up to ensure that subscription data for targeted customer communications is provided in batches of fewer than the limit to ensure we're capturing all intended recipients.

Internal Info: The **limit number** in the RCA above is 1 Million rows. We could not disclose this business info in public.

How good have you found this content?