Protect APIs and Improve Their Performance with API Management



Daniel KrzyczkowskiMICROSOFT MVP & SOFTWARE DEVELOPER

@DKrzyczkowski www.techmindfactory.com



Module Overview



Protect APIs from unauthorized with API keys and client certificate

Use policies to change the behavior of the API through configuration

Implement throttling to prevent resource exhaustion

Improve performance using caching policy



Concepts of Azure API Management Security and Policies



Policies

Policies are a powerful capability of Azure API Management that allow changing the behavior of the API through configuration.

Policies are a collection of statements that are executed sequentially on the request or response of an API.



Azure API Management Policies



Format conversion from XML to JSON



Restrict the amount of incoming calls



Enforces existence and/or value of a HTTP Header



Caches response according to the specified cache control configuration



Access Restriction Policies

Limit call rate by key

Prevents API usage by limiting call rate, on a per key basis

Validate JWT tokens

Enforces existence and validity of a JWT token in header or query parameter

Set usage quota by key

Enforces a renewable or lifetime call volume and/or bandwidth quota

Check HTTP header presence

Enforces existence and/or value of a HTTP Header

Limit call rate by subscription

Prevents API usage by limiting call rate, on a per subscription basis



Advanced Policies

Mock response

Returns a mocked response directly to the caller

Forward request

Forwards the request to the backend service

Retry

Retries execution of a request at the specified time intervals

Set request method

Allows changing the HTTP method for a request

Trace

Adds custom traces into the API Inspector output or Application Insights



Transformation Policies

Convert XML to JSON

Converts request or response body from XML to JSON

Convert JSON to XML

Converts request or response body from JSON to XML

Find and replace string in body

Finds a request or response substring and replaces it with a different substring

Set backend service

Changes the backend service for an incoming request

Set query string parameter

Adds, replaces value of, or deletes request query string parameter



Caching Policies

Store to cache

Caches response according to the specified cache control configuration

Get from cache

Perform cache look up and return a valid cached response when available

Remove value from cache

Remove an item in the cache by key



There are many more policies available in the Azure API Management



Policy Scope



Global scope - affects all APIs within the instance of API Management



Product scope - manages access to the product as a single entity



API scope - affects only a single API



Operation scope - affects only one operation within the API



When Do Policies Execute?



Inbound policies execute when a request is received from a client



Backend policies execute before a request is forwarded to a managed API



Outbound policies execute before a response is sent to a client



On-Error policies execute when an exception is raised



Policy Structure Example

```
<policies>
  <inbound>
    <rate-limit calls="5" renewal-period="10" />
    <cache-lookup vary-by-developer="false" vary-by-developer-groups="false" must-revalidate="true" downstream-caching-type="none" caching-type="internal" />
    <base />
  </inbound>
  <backend>
    <base />
  </backend>
  <outbound>
    <cache-store duration="60" />
    <base />
  </outbound>
  <on-error>
    <base />
  </on-error>
</policies>
```



Demo

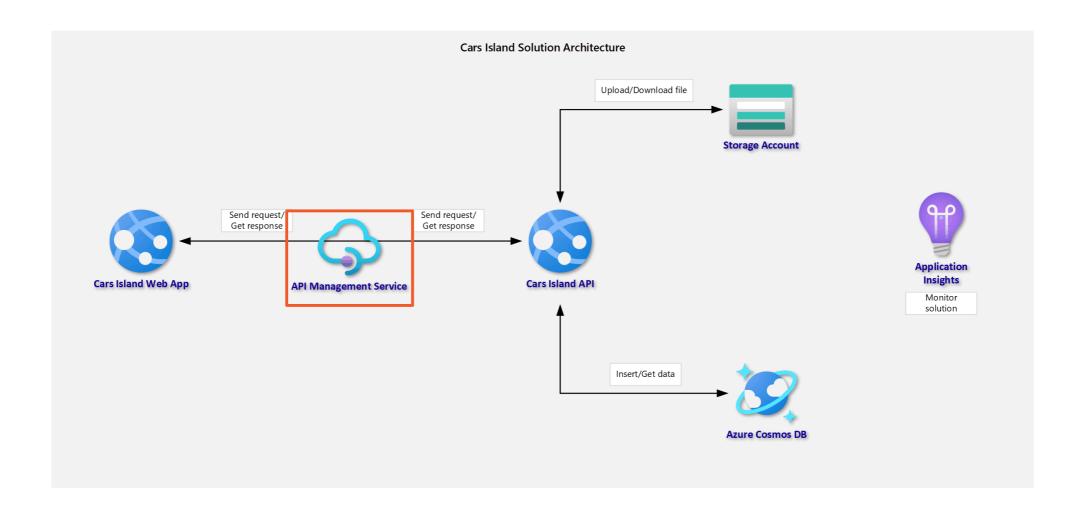


Protect APIs from unauthorized access

Access API with API key and client certificate



Solution Architecture





Before We Begin





Demo

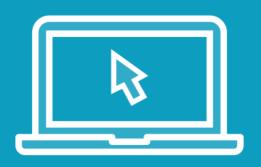


Implement throttling to prevent resource exhaustion

- Implement throttling in the Azure API Management



Demo



Improve API performance

- Implement caching policy in the Azure API Management



Summary



Access restriction, transformation, caching and advances policies in the Azure API Management

Prevent unauthorized access by using API keys and client certificates

Using throttling to limit access to API endpoints by putting limits on the number of times an API can be called

Improve performance using caching policy



Thank you!



Daniel Krzyczkowski
MICROSOFT MVP & SOFTWARE DEVELOPER

@DKrzyczkowski www.techmindfactory.com

