BOOMI REST API Managed through Azure API Management

Azure API Management

Today's innovative enterprises are adopting API architectures to accelerate growth. Streamline your work across hybrid and multi cloud environments with a single place to manage all your APIs.

Deploy API gateways side-by-side with the APIs hosted in Azure, other clouds, and on-premises, optimizing API traffic flow. Meet security and compliance requirements while enjoying a unified management experience and full observability across all internal and external APIs.

Azure APIM can with rich configuration and zero-code features can help:

- # Protect organization resources

  Control how data and services are exposed to employees, partners, and customers by applying authentication, authorization, and usage limits.

- # Accelerate your business:

  Build apps faster and deliver immediate value to your customers through API-first approaches. Decouple front- and back-end teams through API mocking, API revisions and versioning, and automated API documentation. See how supermarket chain Wegmans created a new mobile application in less than eight weeks.

- # Improve API discoverability

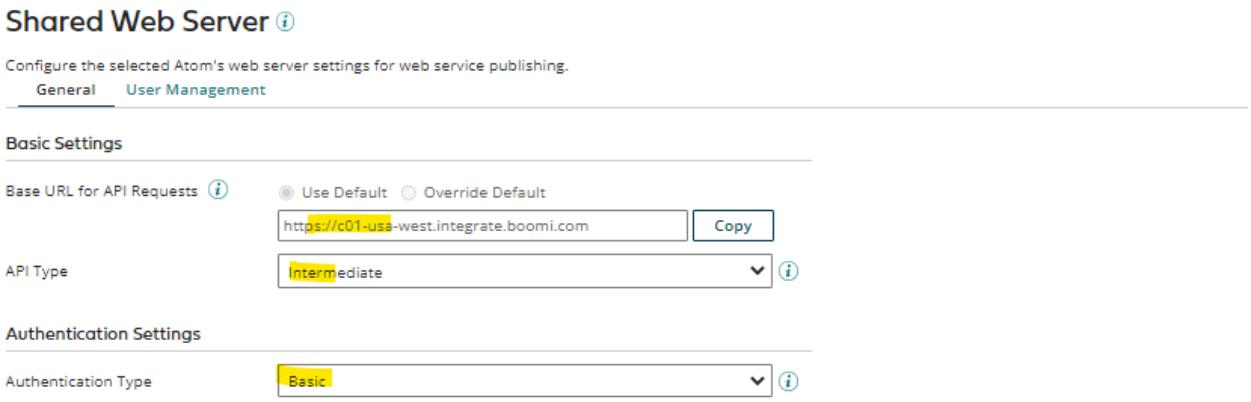
  Create a customizable developer portal for all your APIs. Easily manage and share APIs with internal teams, external partners, and customers.

- # Transform your existing services

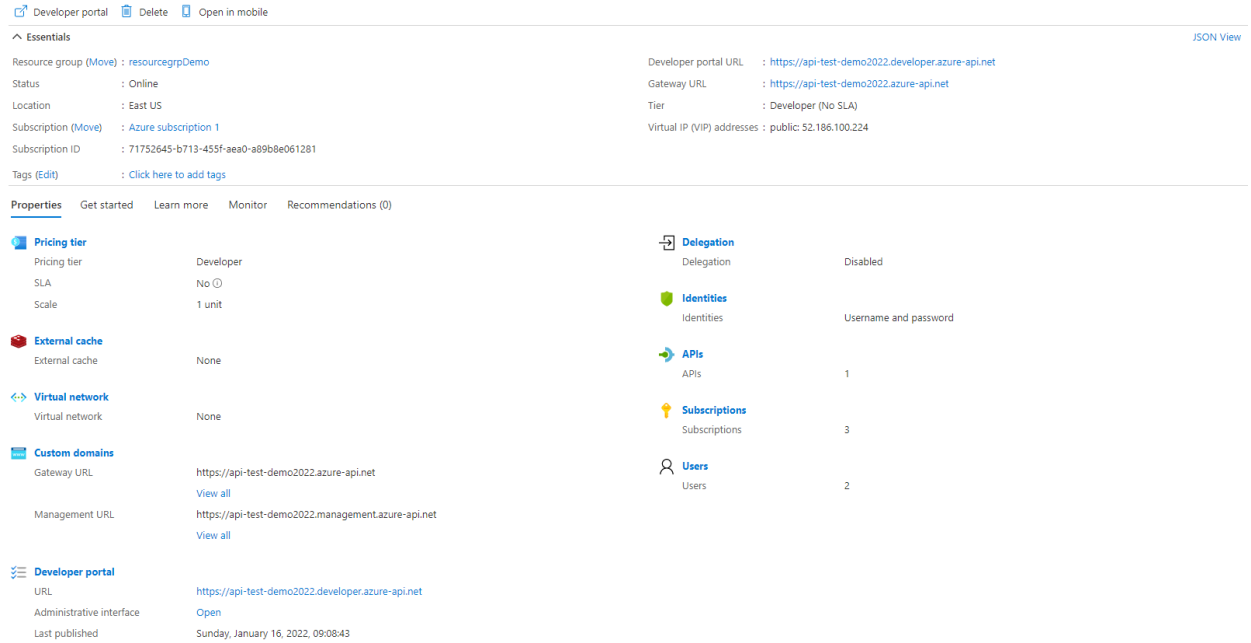
  Automatically turn legacy web services into modern REST-based APIs by creating façades for your back-end services

1. Create REST service on BOOMI.

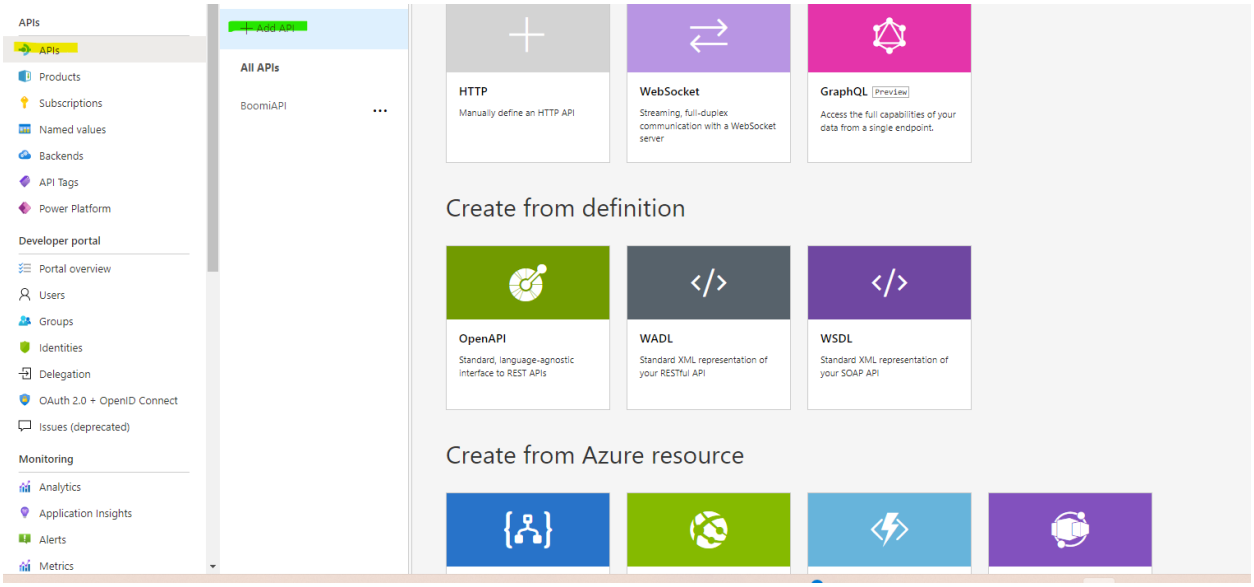
2. We are managing the API on Azure APIM. We are not using Boomi API management feature here. So set below on BOOMI Shared web service setting.

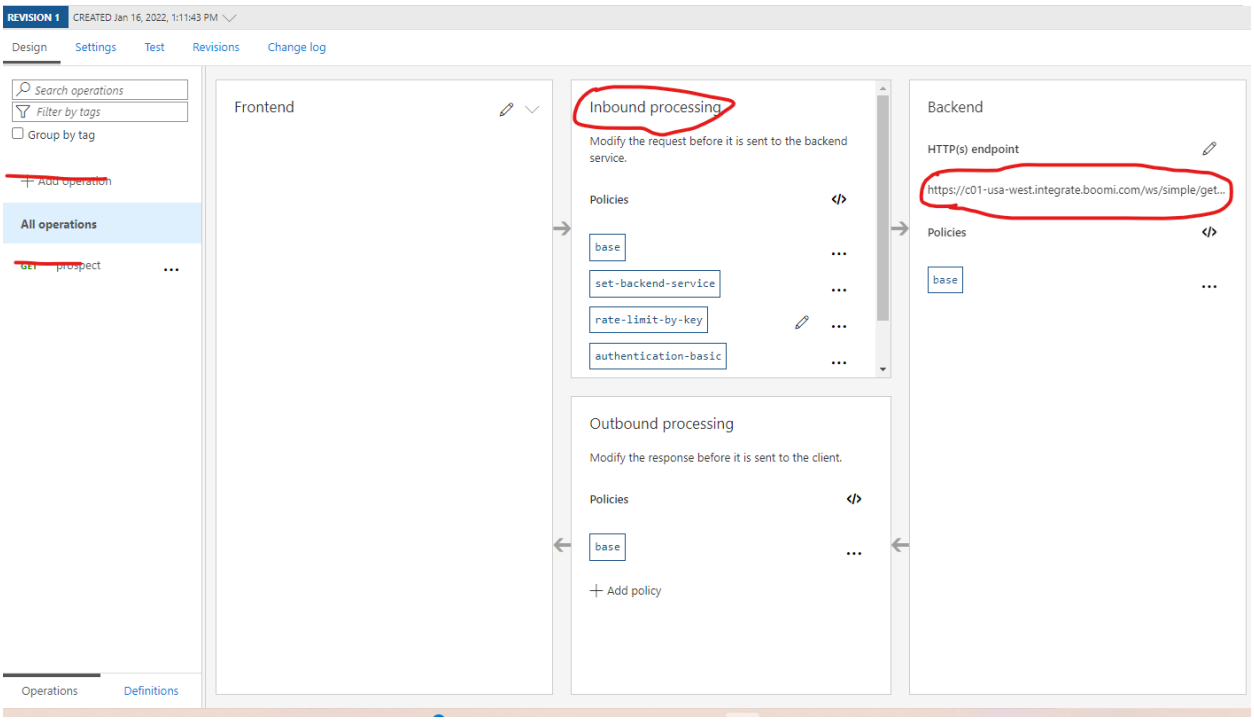


3. Create APIM resource on Azure portal

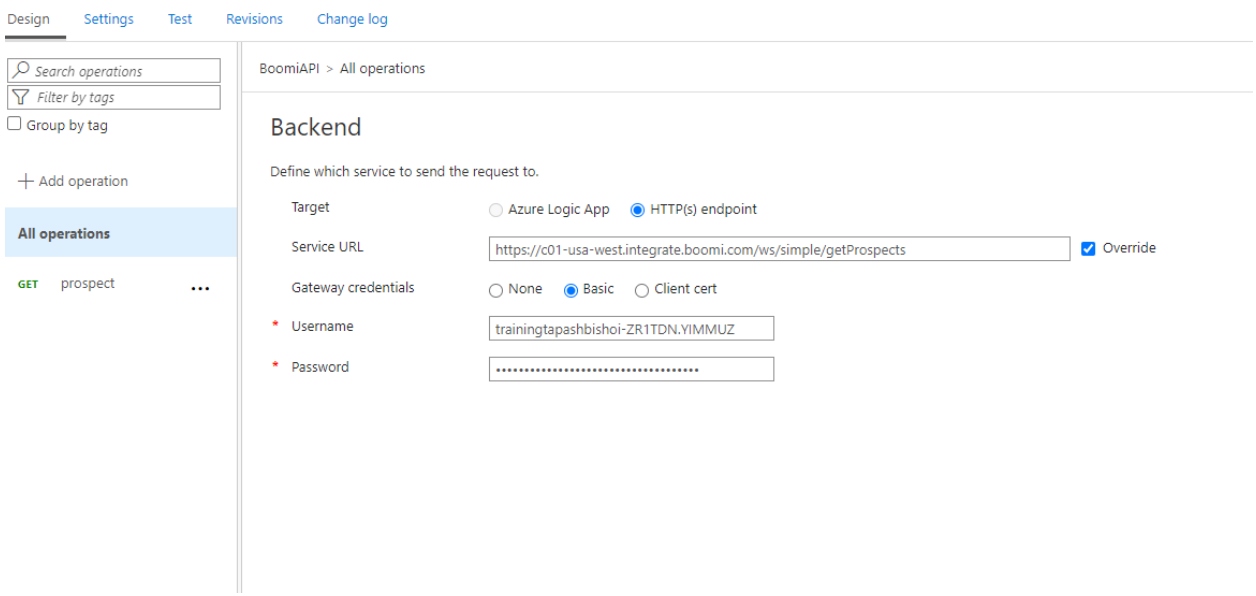


4. Once resource is created, create an APIM.





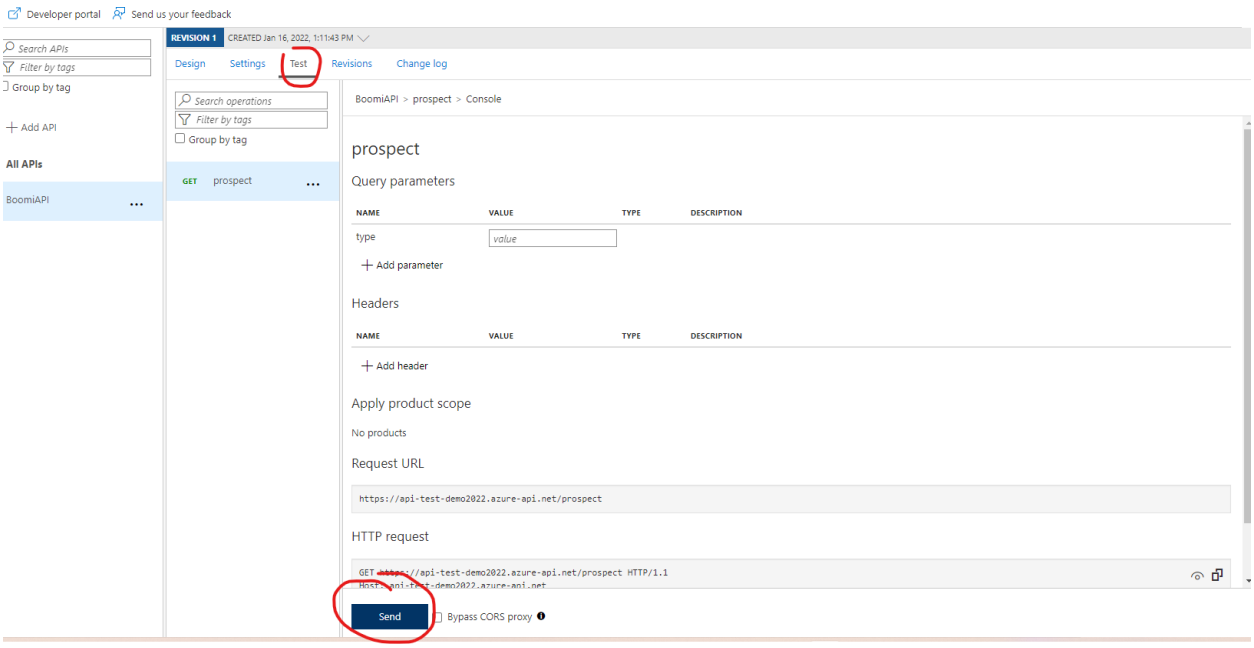
You can set APIM policies on Imbound Processing steps.
Configure Backend , to call HTTP end point.



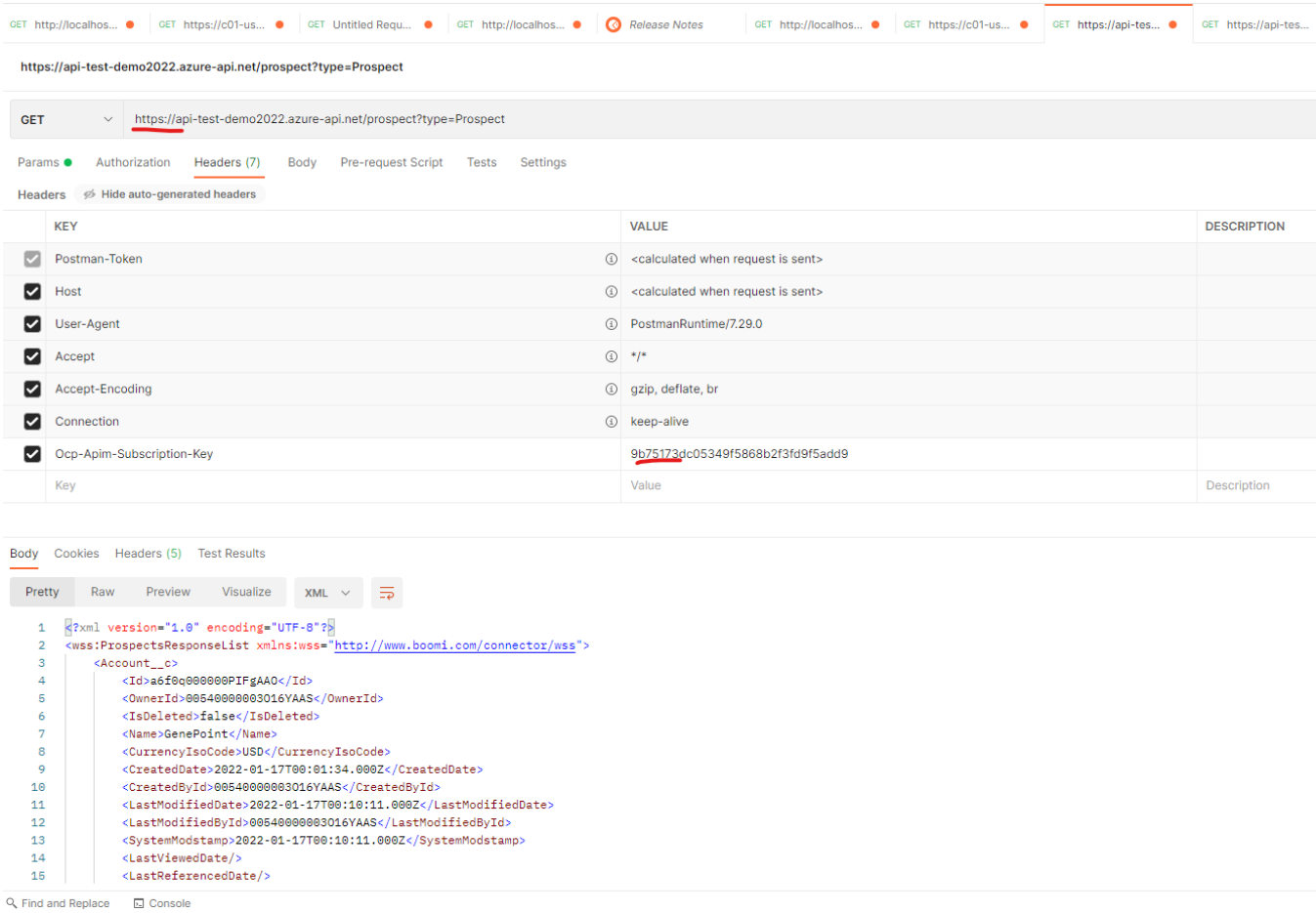
Below is the sample policy that I have applied to API.

```
<!--
    IMPORTANT:
    - Policy elements can appear only within the <inbound>, <outbound>, <backend> section elements.
    - To apply a policy to the incoming request (before it is forwarded to the backend service), place a corresponding policy element within the <inbound> section element.
    - To apply a policy to the outgoing response (before it is sent back to the caller), place a corresponding policy element within the <outbound> section element.
    - To add a policy, place the cursor at the desired insertion point and select a policy from the sidebar.
    - To remove a policy, delete the corresponding policy statement from the policy document.
    - Position the <base> element within a section element to inherit all policies from the corresponding section element in the enclosing scope.
    - Remove the <base> element to prevent inheriting policies from the corresponding section element in the enclosing scope.
    - Policies are applied in the order of their appearance, from the top down.
    - Comments within policy elements are not supported and may disappear. Place your comments between policy elements or at a higher level scope.
-->
<policies>
    <inbound>
        <base />
        <set-backend-service base-url="https://c01-usa-west.integrate.boomi.com/ws/simple/getProspects" />
        <rate-limit-by-key calls="5" renewal-period="60" counter-key="@(context.Subscription?.Key ?? "anonymous")" increment-condition="@(context.Response.StatusCode >= 200 && context.Response.StatusCode < 300)" />
        <authentication-basic username="trainingtapashbishoi-ZR1TDN.YIMMUZ" password="XXXXXXXXXXXXXXXX" />
    </inbound>
    <backend>
        <base />
    </backend>
    <outbound>
        <base />
    </outbound>
    <on-error>
        <base />
    </on-error>
</policies>
```

You can test the API on Azure APIM. Below is the steps.



Below is the steps to test from Postman



Please note , I am using Azure APIM URL and authentication method. The service is set on Boomi for Intermediate But We have added API management for the service on Azure APIM.