

WSO2 API Manager

Jayanth R

ACE11696

Here are the steps to create an API in WSO2 API Manager.

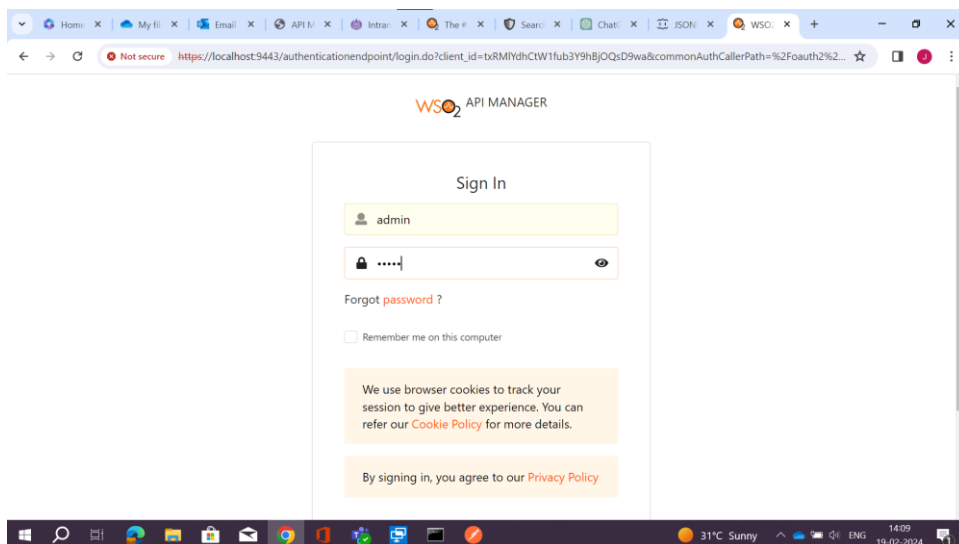
Step 1: Run the .bat file for installing WSO2 in windows.

```

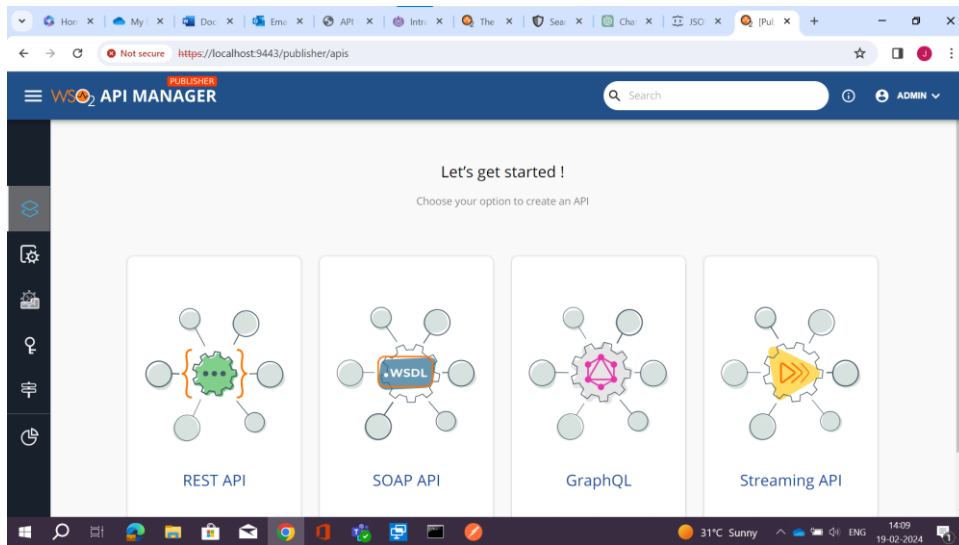
C:\Windows\System32\cmd.exe - api-manager bat -n
2024-02-19 14:07:46,297 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,297 INFO - JMSListener Connection attempt: 1 for JMS Provider for listener: Siddhi-JMS-Consumer#cacheInvalidation was successful
2024-02-19 14:07:46,297 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,297 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,298 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,298 INFO - JMSListener Connection attempt: 1 for JMS Provider for listener: Siddhi-JMS-Consumer#notification was successful
2024-02-19 14:07:46,298 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,298 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,298 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,299 INFO - JMSListenerManager Task manager for Siddhi-JMS-Consumer [re-]initialized
2024-02-19 14:07:46,313 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan: carbon.super.sub_AsyncBronze saved in the filesystem
2024-02-19 14:07:46,313 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super.sub_AsyncBronze
2024-02-19 14:07:46,349 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,349 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan: carbon.super.sub_50PPerMin saved in the filesystem
2024-02-19 14:07:46,349 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super.sub_50PPerMin
2024-02-19 14:07:46,382 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,382 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan: carbon.super.sub_20PPerMin saved in the filesystem
2024-02-19 14:07:46,382 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super.sub_20PPerMin
2024-02-19 14:07:46,391 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,391 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan: carbon.super.sub_10PPerMin saved in the filesystem
2024-02-19 14:07:46,391 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super.sub_10PPerMin
2024-02-19 14:07:46,511 INFO - CarbonUserServiceComponent Mgt Console URL : https://localhost:9443/carbon/
2024-02-19 14:07:46,543 INFO - CarbonUserServiceComponent API Developer Portal Default context : https://localhost:9443/default
2024-02-19 14:07:46,543 INFO - CarbonUserServiceComponent API Publisher Default context : https://localhost:9443/publisher
2024-02-19 14:07:46,545 INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.global.throttle.stream:1.0.0
2024-02-19 14:07:46,545 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,578 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_10PPerMin
2024-02-19 14:07:46,578 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_50KPerMin
2024-02-19 14:07:46,591 INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.global.throttle.stream:1.1.0
2024-02-19 14:07:46,591 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,609 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_10PPerMin
2024-02-19 14:07:46,609 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_20KPerMin
2024-02-19 14:07:46,644 INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.global.throttle.stream:1.1.0
2024-02-19 14:07:46,683 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,683 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_20KPerMin
2024-02-19 14:07:46,756 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan: carbon.super_resource_10KPerMin.default saved in the filesystem
2024-02-19 14:07:46,756 INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.global.throttle.stream:1.1.0
2024-02-19 14:07:46,780 INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
2024-02-19 14:07:46,780 INFO - EventProcessorConfigurationJmsEventInvoker Execution plan is deployed successfully and in active state : carbon.super_resource_10KPerMin.default

```

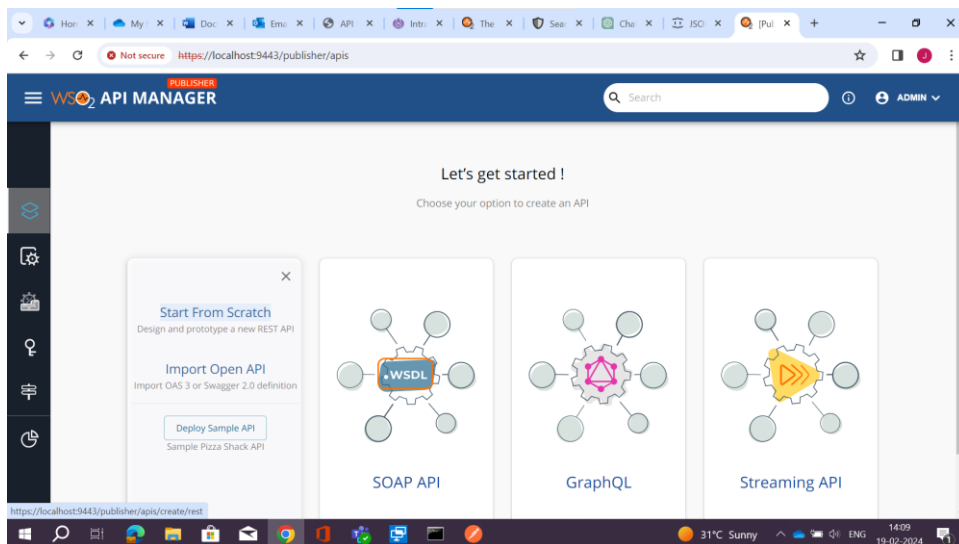
Step 2: Navigate to the Publisher server portal in localhost and login using credentials.



Step 3: This is the Publisher portal of WSO2.



Step 4: Create a Rest API give "Start from Scratch" there.



Step 5: Fill the details for creating API (I'm using a demo Endpoints already available in json place holder)

The screenshot shows the 'Create an API' form in the WSO2 API Manager. The form has the following fields and values:

- Name:** DemoAPI
- Context:** /Demo
- Version:** 1.0.0
- Endpoint:** <https://jsonplaceholder.typicode.com/>

Below the fields are three buttons: 'Create', 'Create & Publish', and 'Cancel'. A note states: 'API will be exposed in /Demo/1.0.0 context at the gateway.' A footer note says 'WSO2 API-M v4.2.0 | © 2023 WSO2 LLC'.

Step 6: Now the API is in developed Stage after creating.

The screenshot shows the 'Overview' page of the WSO2 API Manager for a newly created API. The page displays the API's lifecycle stages and configuration details.

Overview

Developed Stage: Develop (green circle), Deploy (blue circle), Test (grey circle), Publish (grey circle).

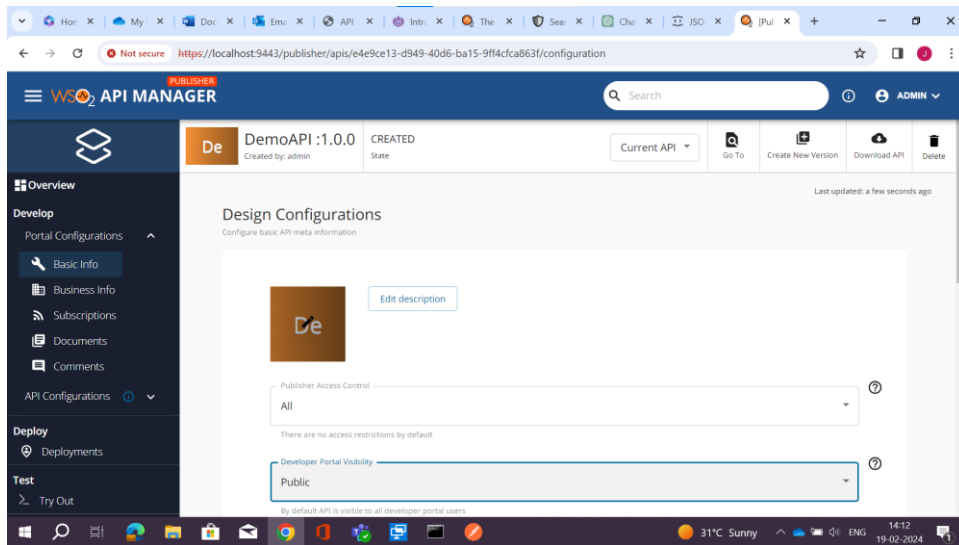
Context:

Property	Value
Description	-
Provider	admin
Context	/todos
Version	1.0.0
Type	HTTP
Created Time	A few seconds ago

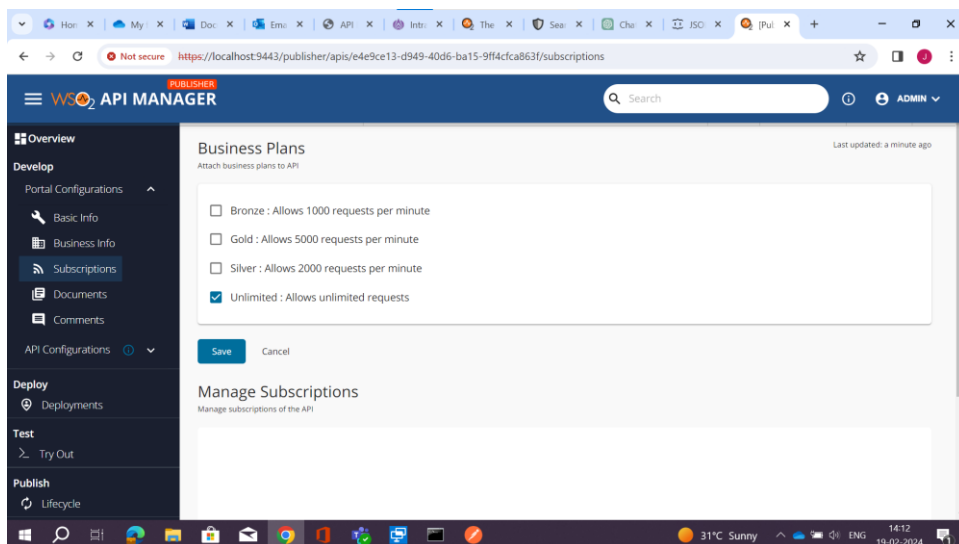
Configuration

Property	Value
Transports	HTTP, HTTPS
API Security	OAuth2
Access Control	None
Workflow Status	-
Visibility on	Public

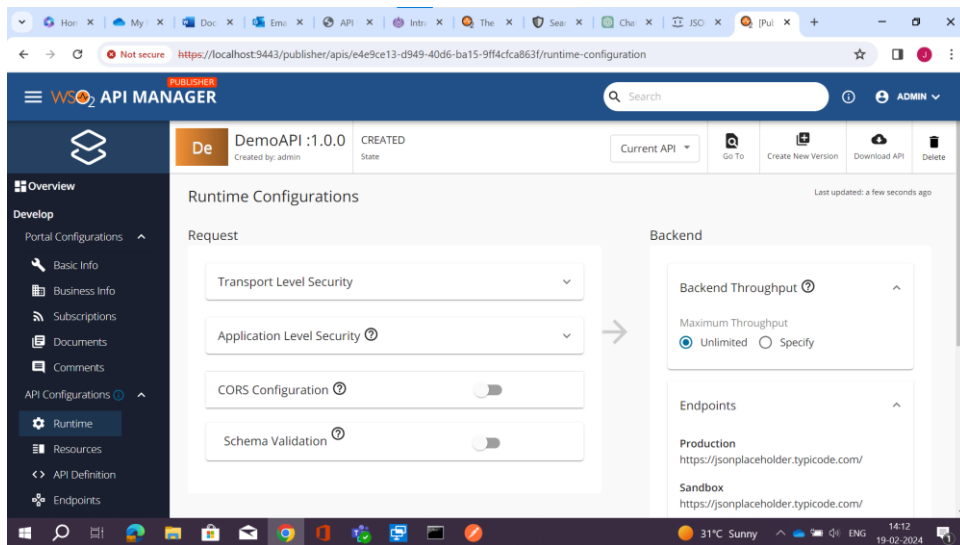
Step 7: Check the API Design and edit if needed in Portal Configurations.



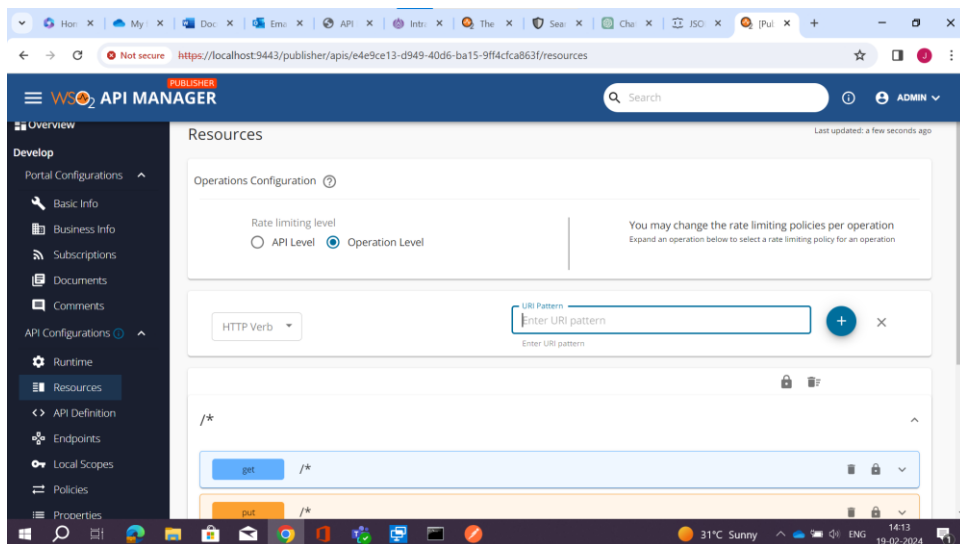
Step 8: Check the Subscription plan select based on needed.



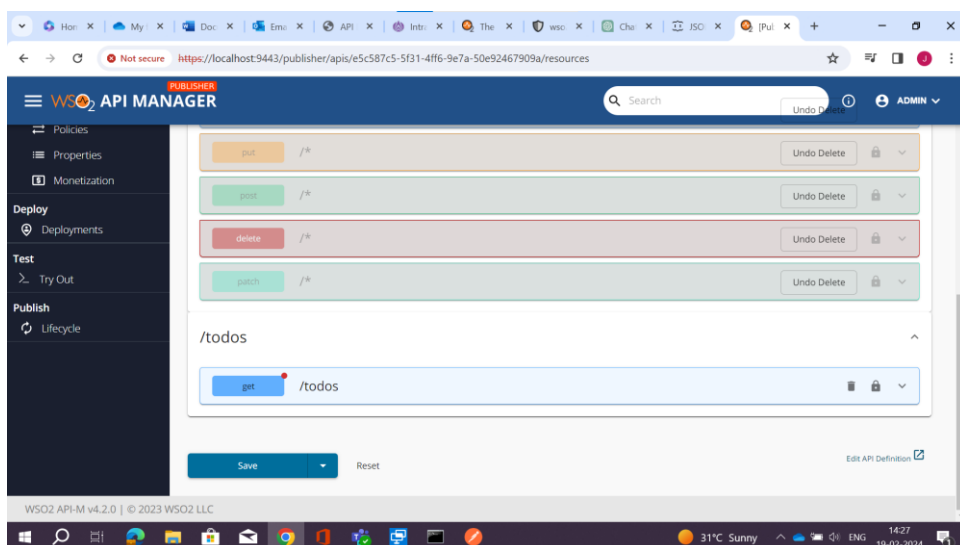
Step 9: Now start with API Configurations.



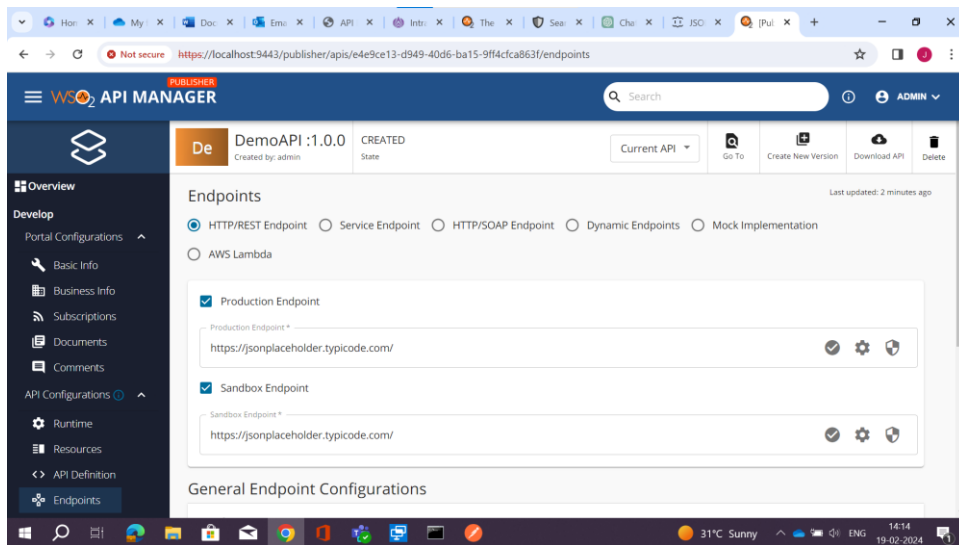
Step 10: Give the URL pattern where you want to access the data and select the HTTP Verb needed.



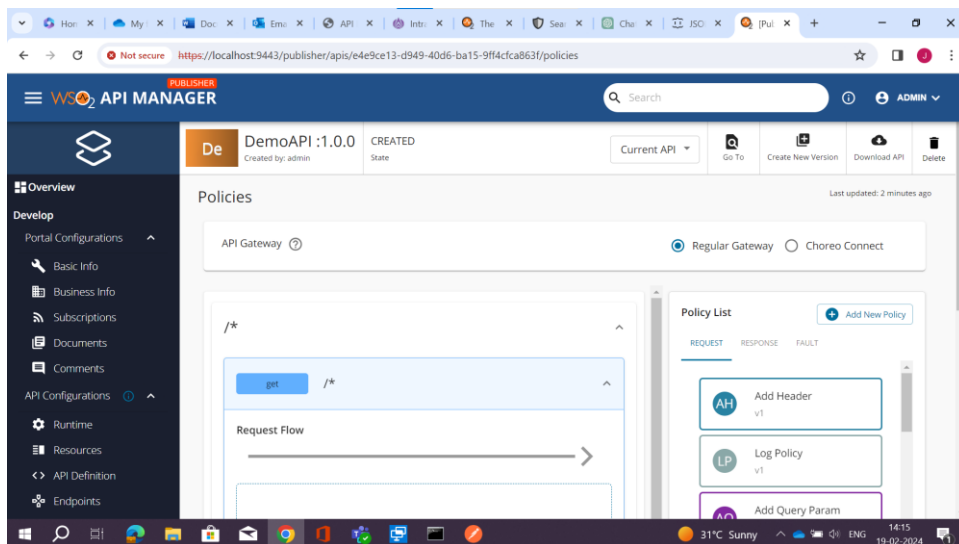
Step 11: I have given todos and created a GET method (Verb) for it.



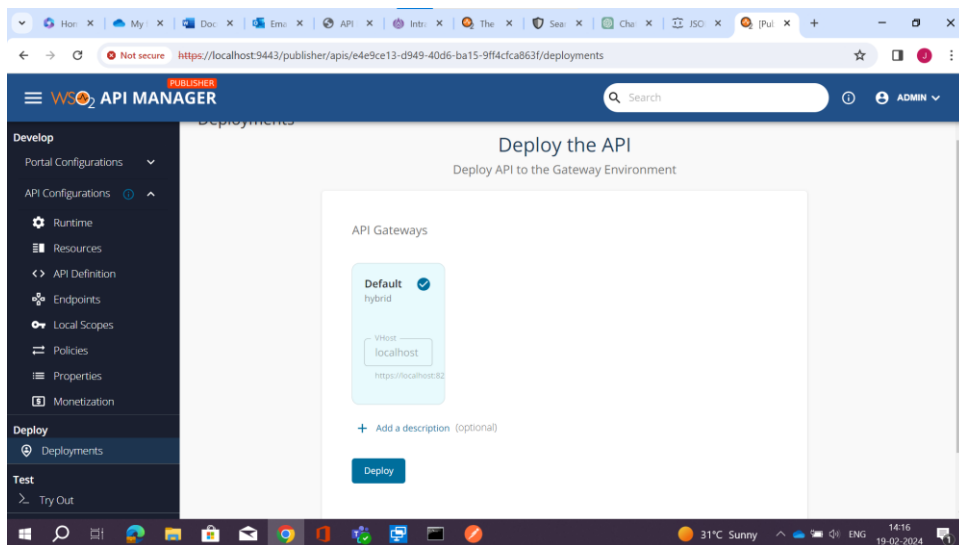
Step 12: Check the EndPoints What type of endpoint needed based on your requirement.



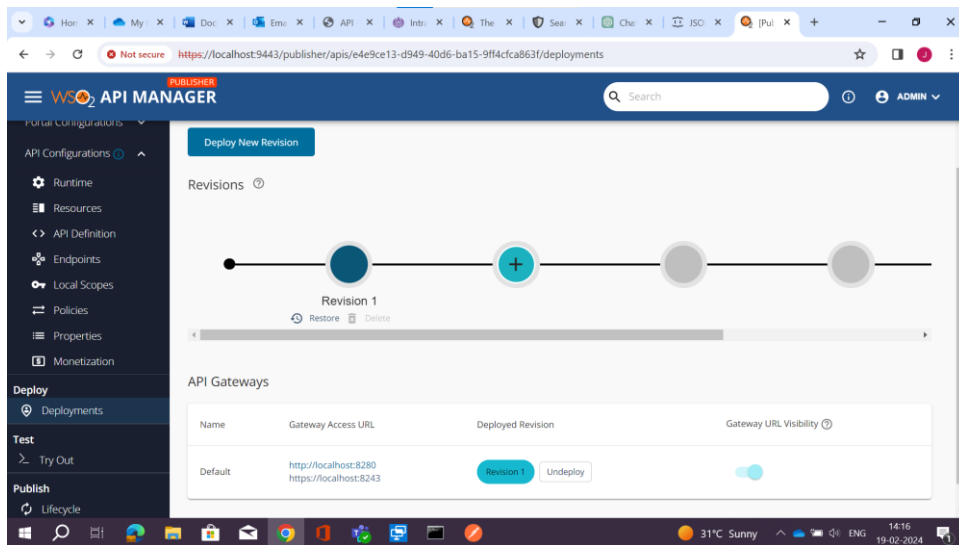
Step 13: Check the policies.



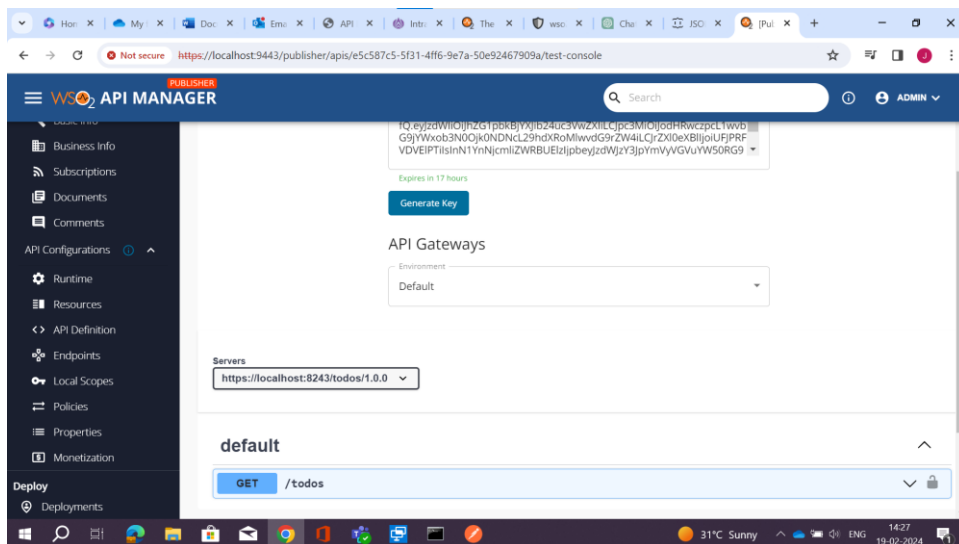
Step 14: Now you can deploy the API that you have designed and developed.



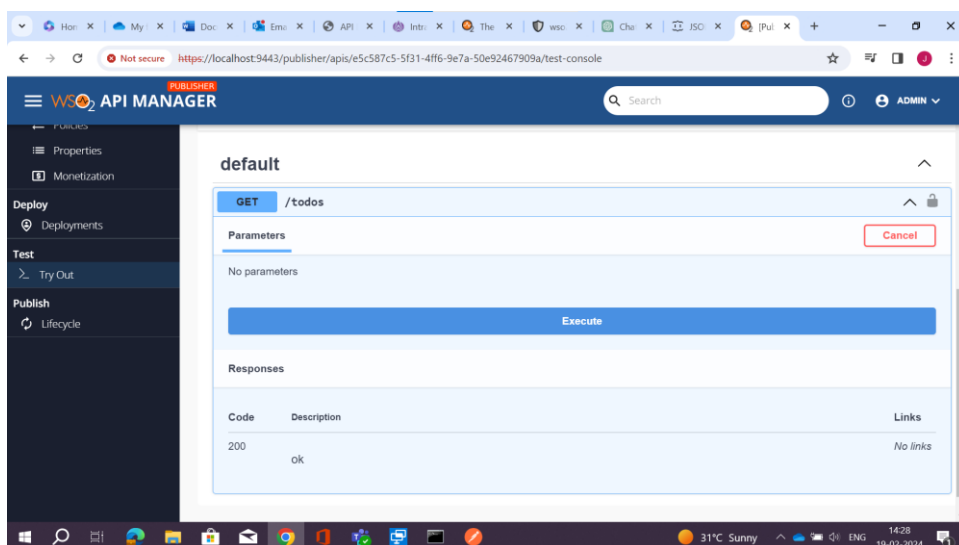
Step 15: You can deploy the API in revisions. (like after updated if u deploy u can give it revision 2).



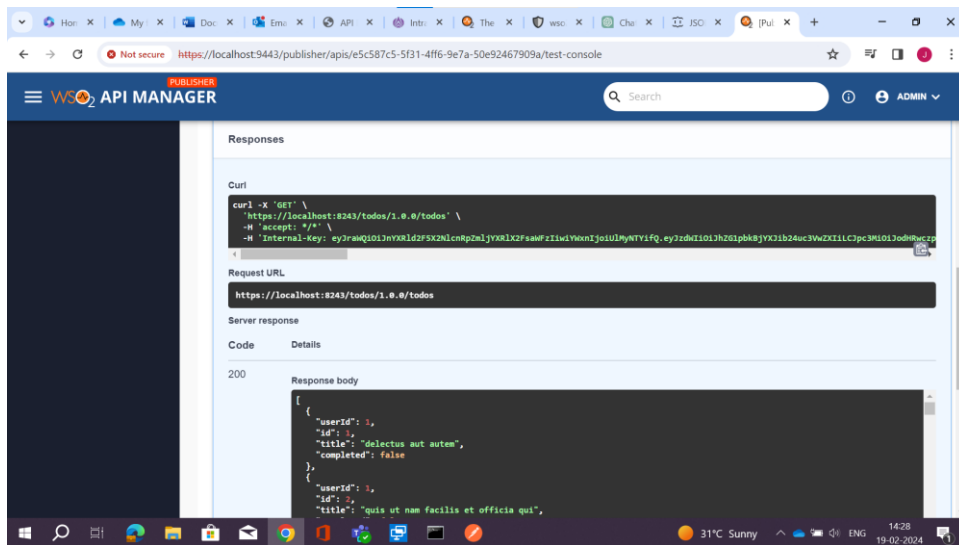
Step 16: Now you can Try out the API if you want after the deployment.



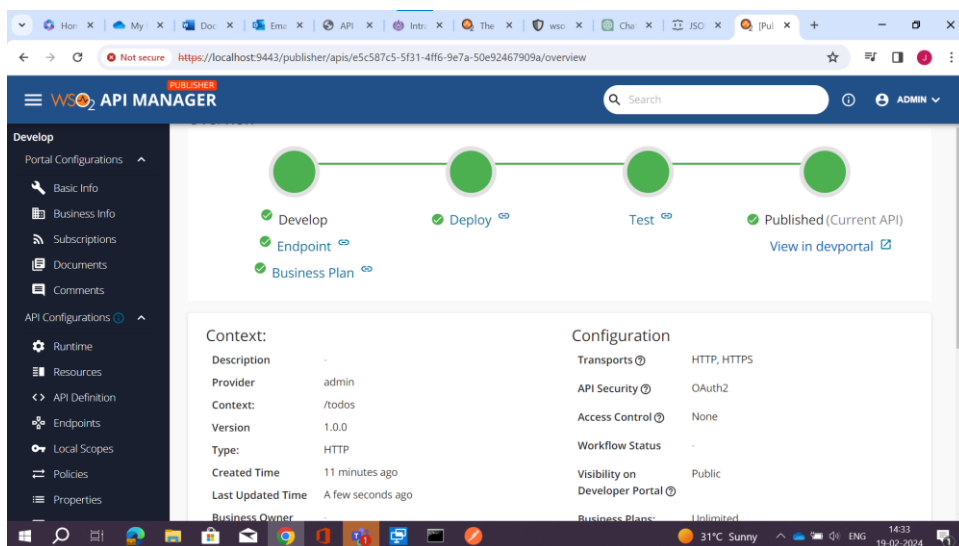
Step 17: Execute the GET request.



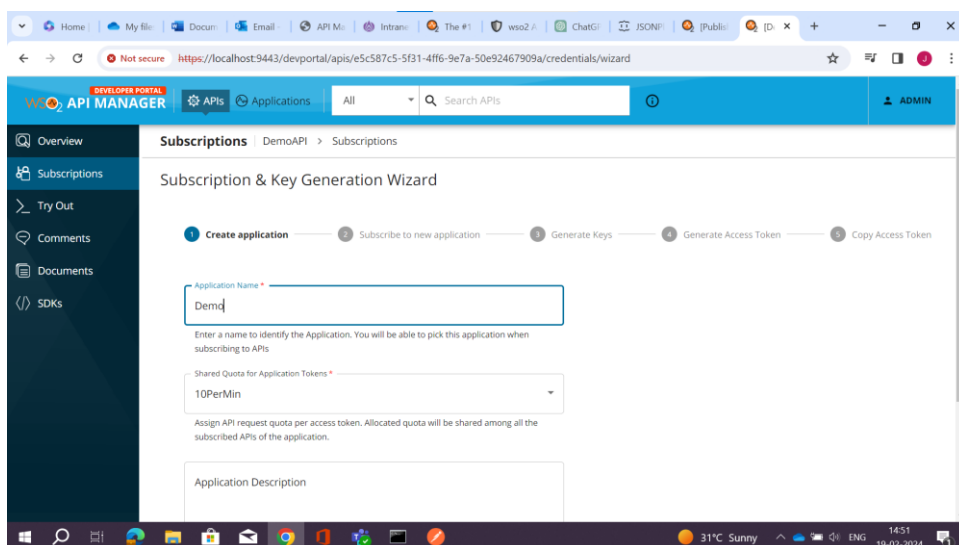
Step 18: See you will get response as status code 200 and OK.



Step 19: Now publish your API.

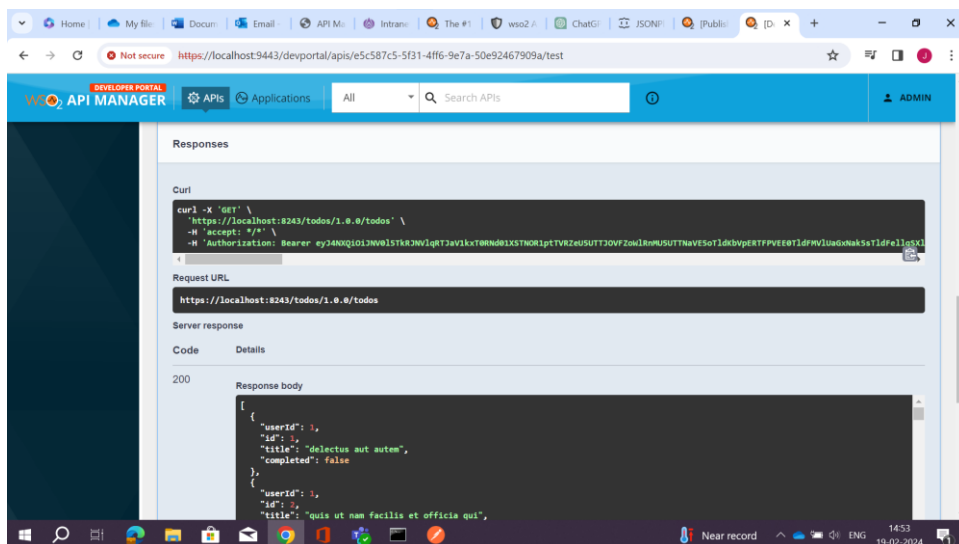


Step 20: Now navigate to developer console and do subscription before using it.

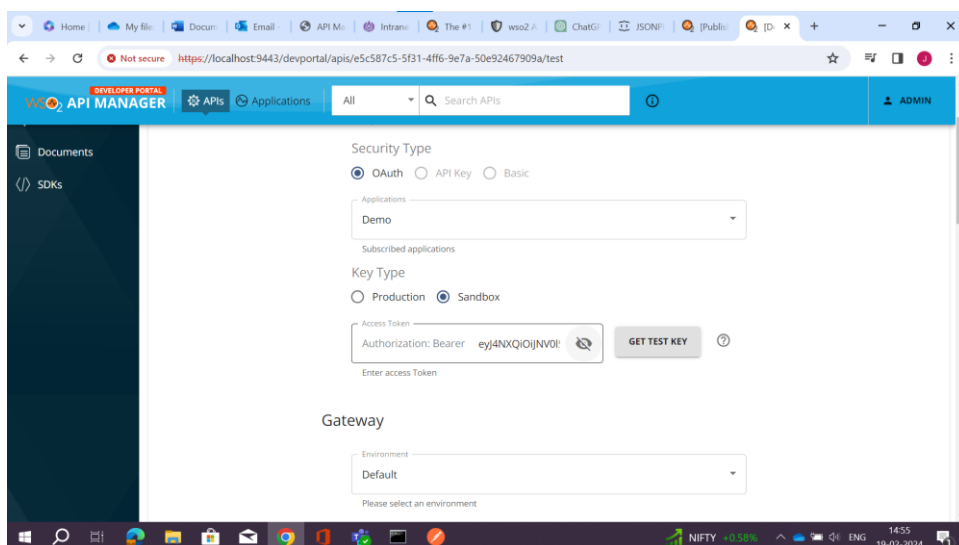


Step 21: Copy the Auto generated Access Token.

Step 24: Now you will get the response and status code as 200 OK.



Step 25: Now Go to Postman and try giving the url of your API and check it.



Step 25: Make sure you have provided the access token and send the Request and you should get response

