

# WSO2 API Manager

Naveen G

ACE11699

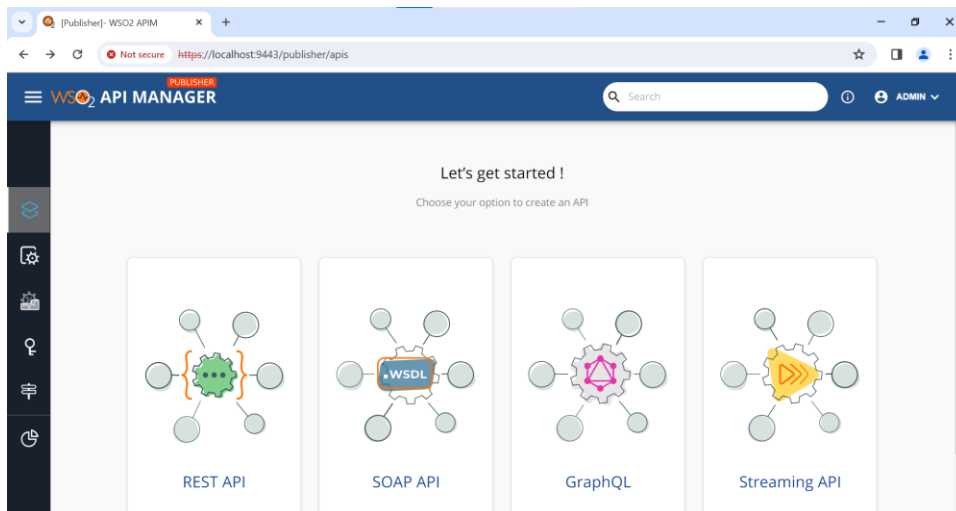
Here are the steps to create an API in WSO2 API Manager.

- Run the .bat file for installing WSO2 in windows.

```
C:\Windows\System32\cmd.exe - api-gate-manager > -run
```

```
[2024-02-19 14:50:47,524] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#notification was successful!
[2024-02-19 14:50:47,525] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#notification was successful!
[2024-02-19 14:50:47,525] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#notificationData was successful!
[2024-02-19 14:50:47,526] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#notificationData was successful!
[2024-02-19 14:50:47,526] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#tokenRevocation was successful!
[2024-02-19 14:50:47,526] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,526] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,526] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,527] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,527] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,527] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,527] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,527] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,541] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:47,541] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:47,541] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_app_20PerMin saved in the filesystem
[2024-02-19 14:50:47,559] INFO - JMSListener Connection attempt: 1 for JMS Provider For listener: Siddhi-JMS-Consumer#keyManager was successful!
[2024-02-19 14:50:47,566] INFO - JMSTaskManager Task manager For Siddhi-JMS-Consumer [re]-initialized
[2024-02-19 14:50:47,592] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:47,592] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:47,592] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_app_20PerMin saved in the filesystem
[2024-02-19 14:50:47,637] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_50PerMin saved in the filesystem
[2024-02-19 14:50:47,662] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:47,662] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:47,662] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_app_10PerMin saved in the filesystem
[2024-02-19 14:50:47,800] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_50PerMin default saved in the filesystem
[2024-02-19 14:50:47,800] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:47,835] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:47,835] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_50PerMin default saved in the filesystem
[2024-02-19 14:50:47,888] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_20PerMin default saved in the filesystem
[2024-02-19 14:50:47,888] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:47,938] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:47,938] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_20PerMin default saved in the filesystem
[2024-02-19 14:50:48,093] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_10PerMin default saved in the filesystem
[2024-02-19 14:50:48,101] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:48,101] INFO - CarbonUIServiceComponent API Developer Portal Default Context : https://localhost:9443/devportal
[2024-02-19 14:50:48,101] INFO - CarbonUIServiceComponent API Publisher Default Context : https://localhost:9443/publisher
[2024-02-19 14:50:48,125] INFO - EventJunction Producer added to the junction. Stream.org.wso2.throttle.globalThrottle.stream:1.0.0
[2024-02-19 14:50:48,125] INFO - EventJunction Consumer added to the junction. Stream.org.wso2.throttle.processed.request.stream:1.0.0
[2024-02-19 14:50:48,125] INFO - EventProcessorConfiguration Execution plan is deployed successfully and in active state : carbon.super_resource_10PerMin default
```

- Navigate to the Publisher server portal in localhost and login using credentials.



- This is the Publisher portal of WSO2.

- Create a Rest API give “Start from Scratch” there.
- Fill the details for creating API (I'm using a demo Endpoints already available in json place holder)

**Create an API**  
Create an API by providing a Name, a Version, a Context and Backend Endpoint (optional)

Name\*  
SampleAPI

Context\*  
/SampleAPI

Version\*  
1.0.0

API will be exposed in /SampleAPI/1.0.0 context at the gateway.

Endpoint  
https://jsonplaceholder.typicode.com/

\* Mandatory fields

Create Create & Publish Cancel

WSO2 API-M v4.2.0 | © 2023 WSO2 LLC

- Now the API is in developed Stage after creating.

**SampleAPI :1.0.0** CREATED State  
Created by: admin

Current API Go To Create New Version Download API Delete

**Overview** Last updated: a few seconds ago

Develop Deploy Test Publish

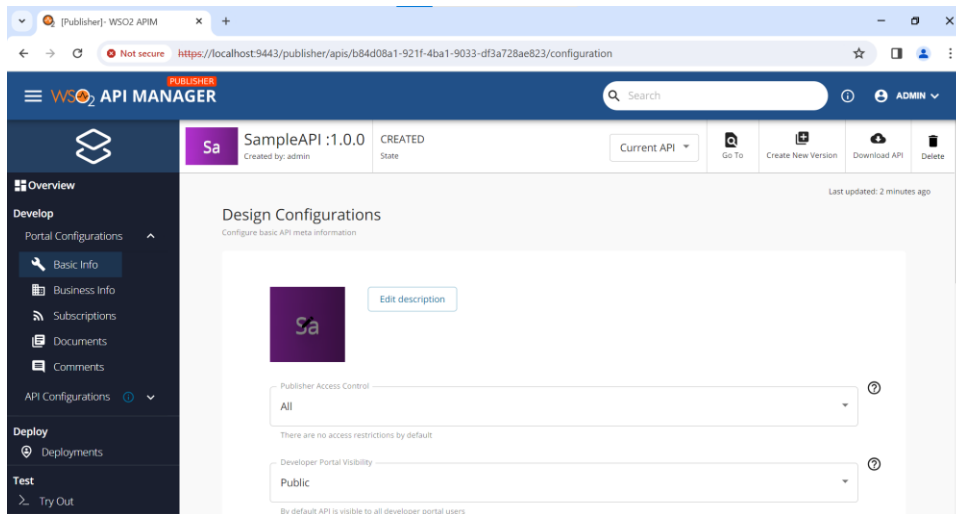
Context:

Description	-
Provider	admin
Context	/SampleAPI
Version	1.0.0

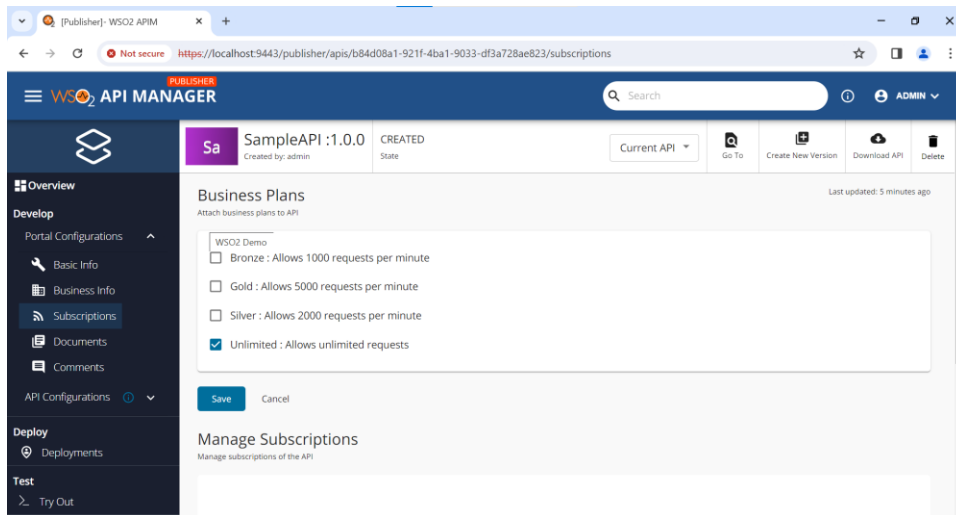
Configuration

Transports	HTTP, HTTPS
API Security	OAuth2
Access Control	None

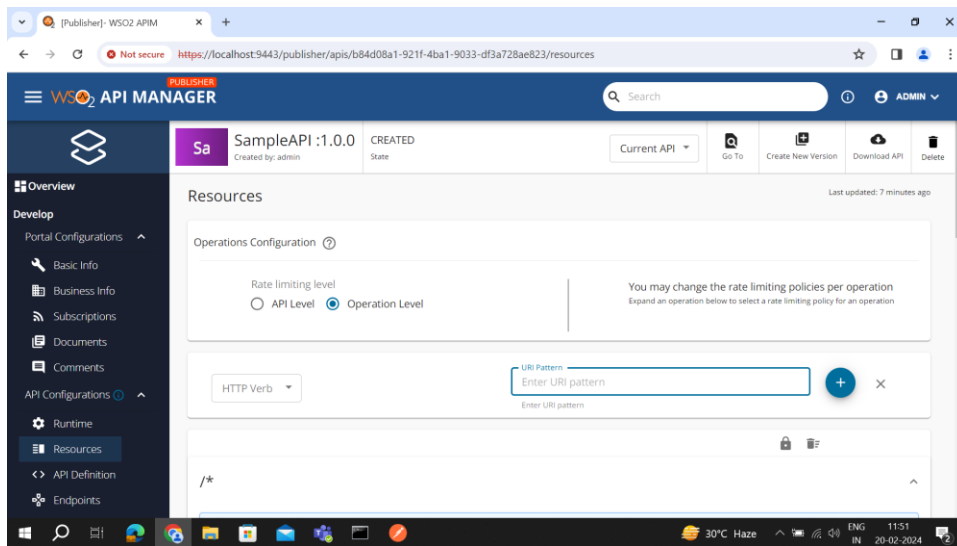
- Check the API Design and edit if needed in Portal Configurations.



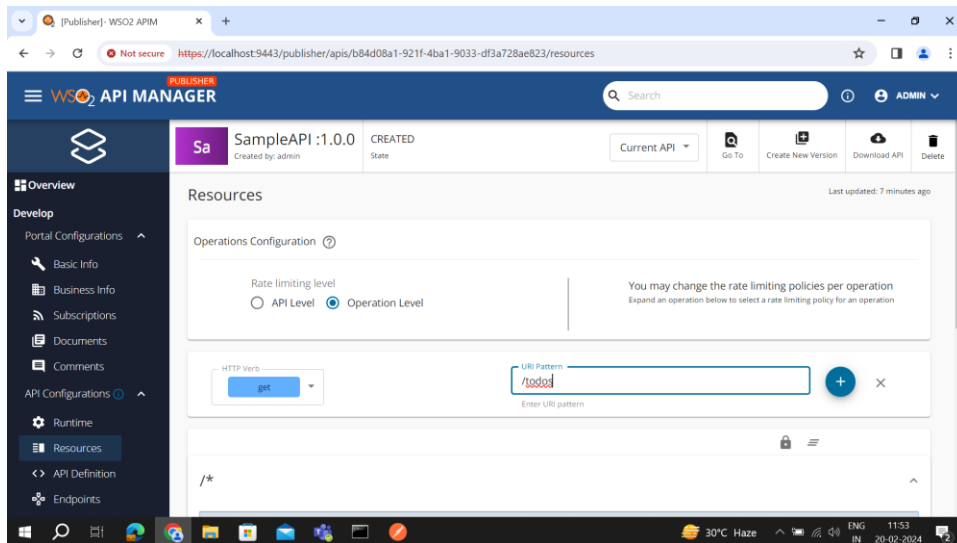
- Check the Subscription plan select based on needed.



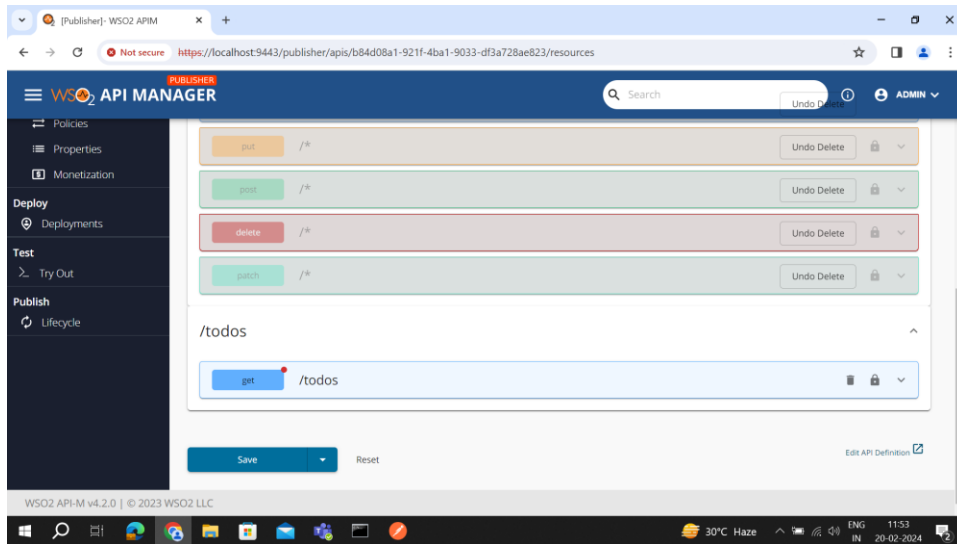
- Now start with API Configurations.



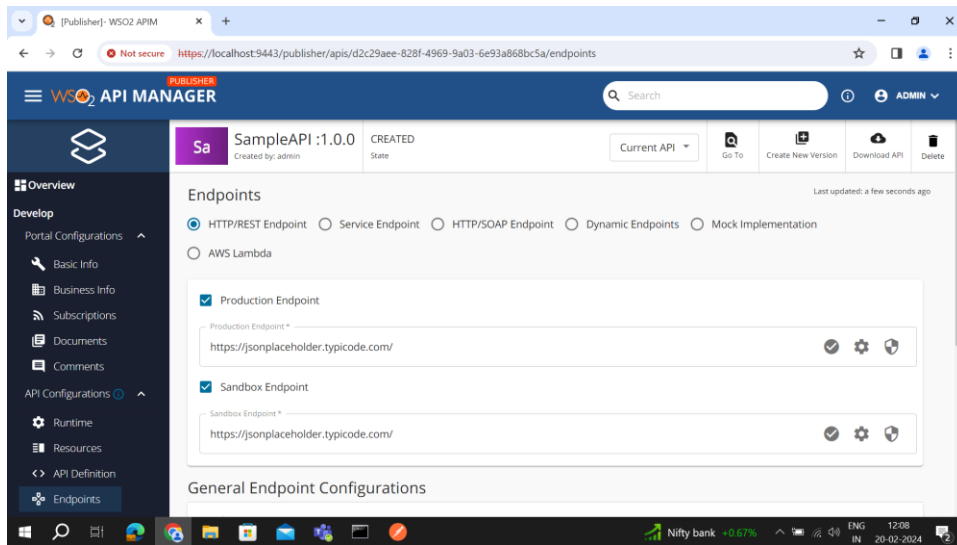
- Give the URL pattern where you want to access the data and select the HTTP Verb needed.



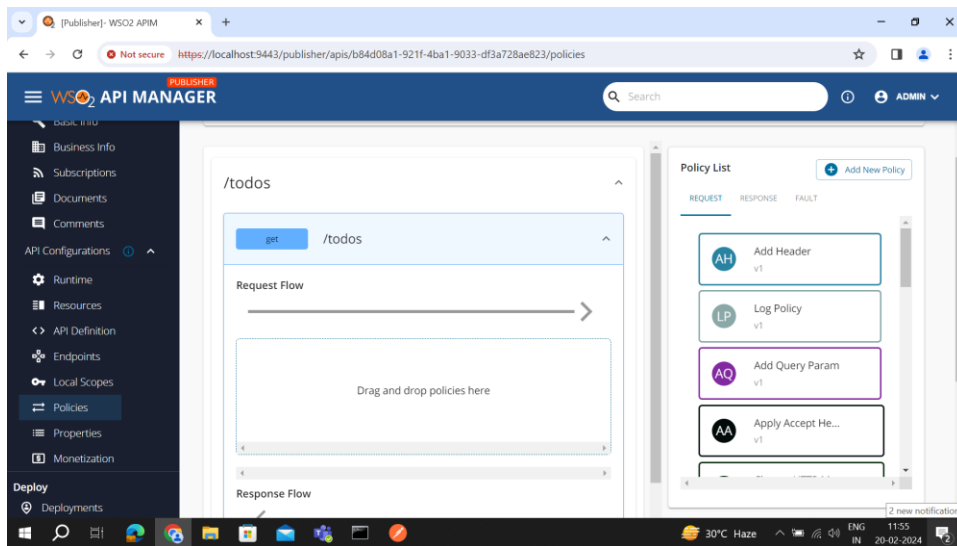
- I have given todos and created a GET method (Verb) for it.



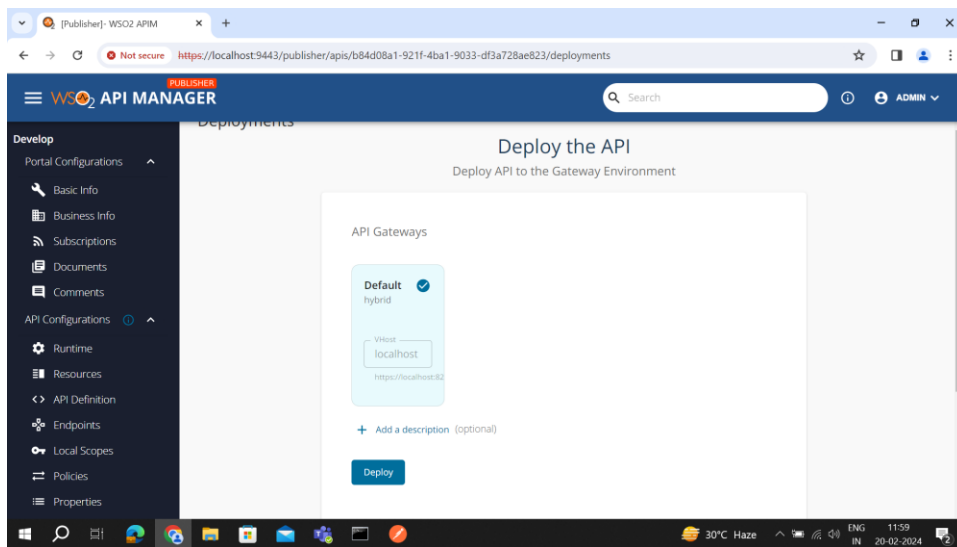
- Check the EndPoints What type of endpoint needed based on your requirement.



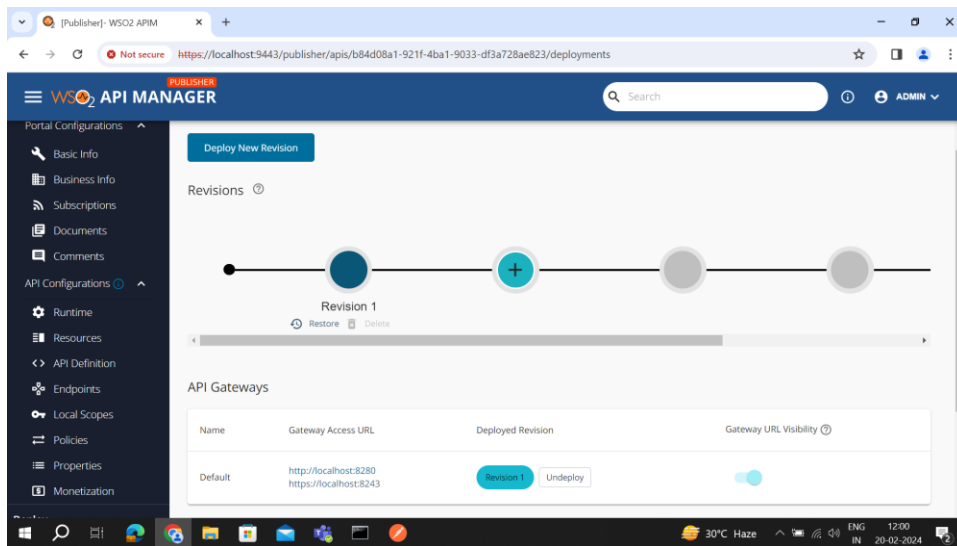
- Check the policies.



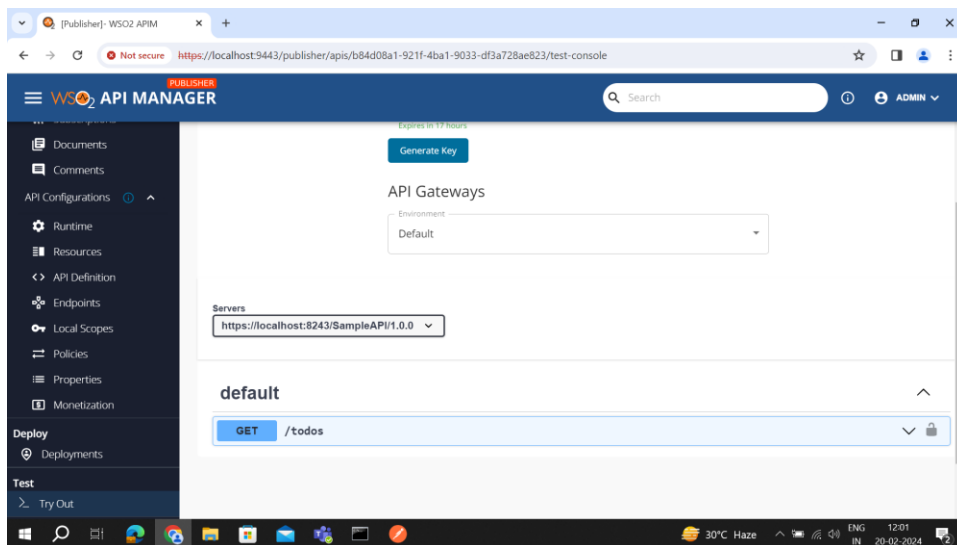
- Now you can deploy the API that you have designed and developed.



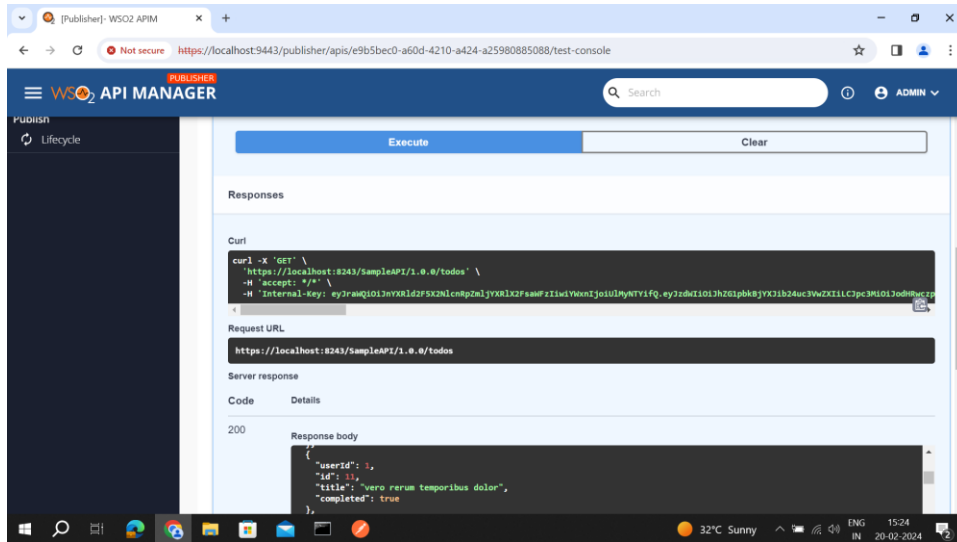
- You can deploy the API in revisions. (like after updated if u deploy u can give it revision 2).



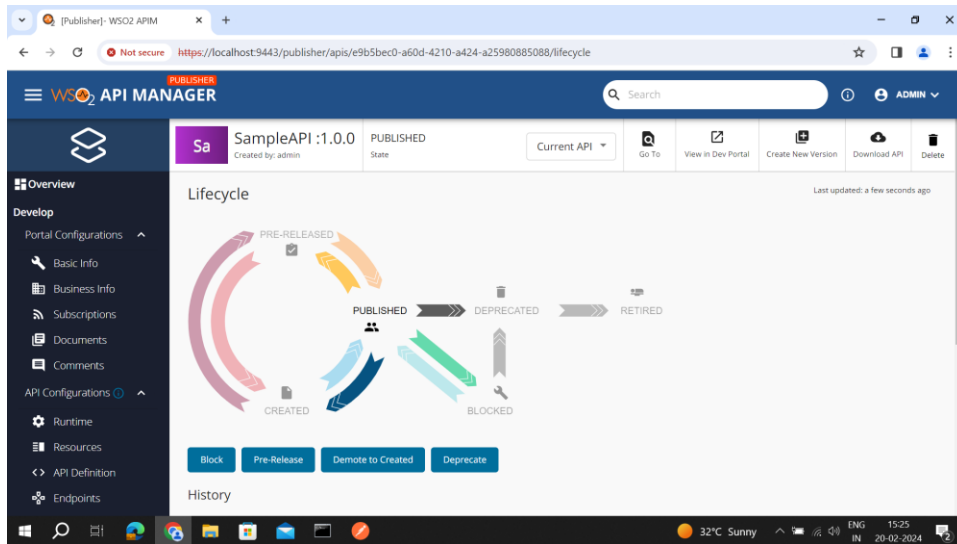
- Now you can Try out the API if you want after the deployment.



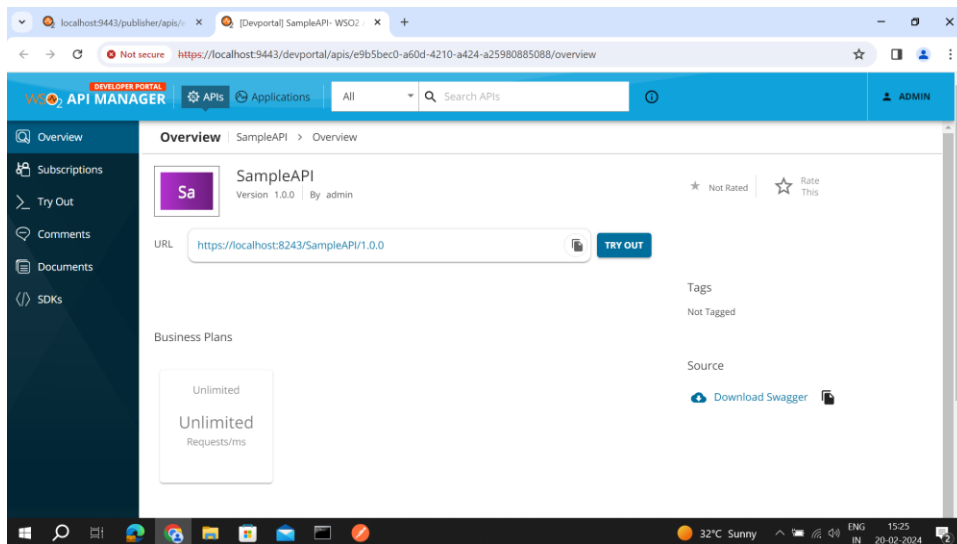
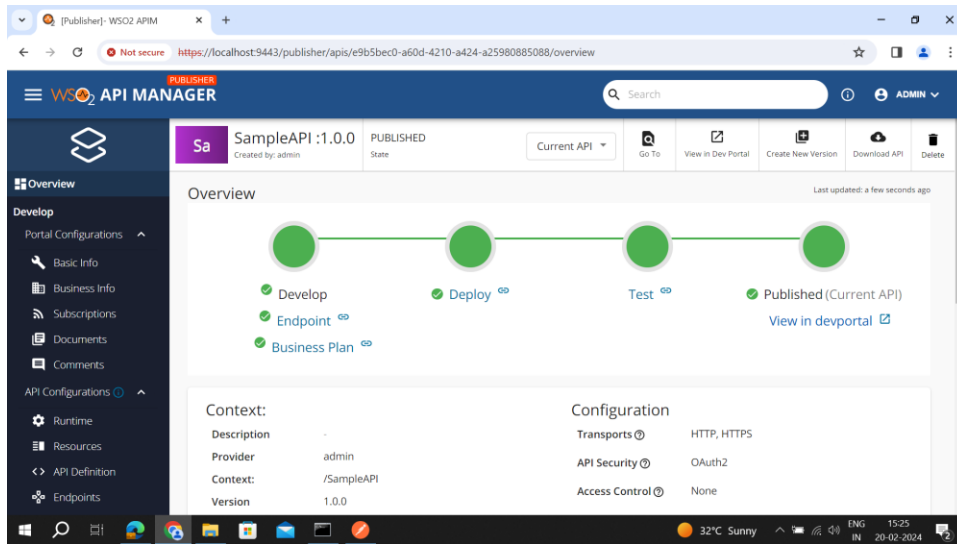
- Execute the GET request.
- See you will get response as status code 200 and OK.



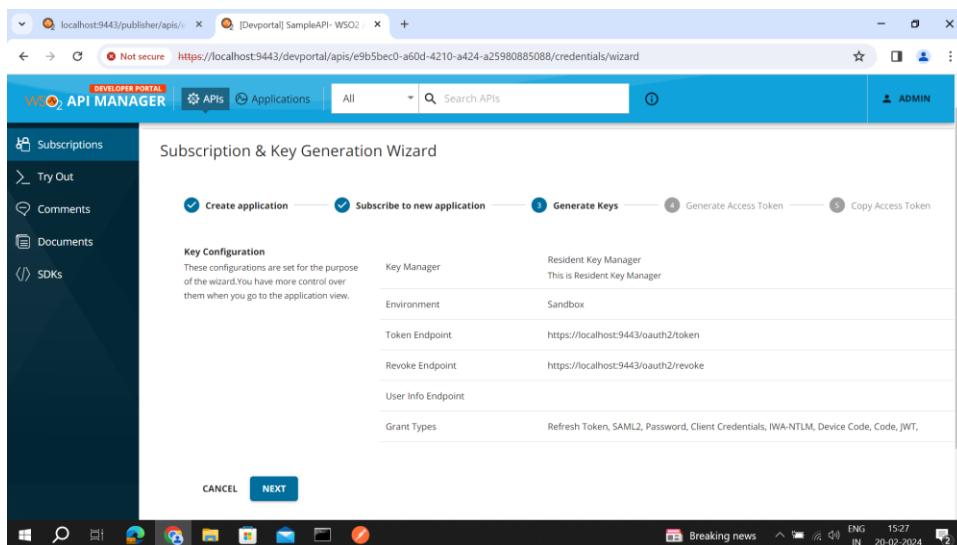
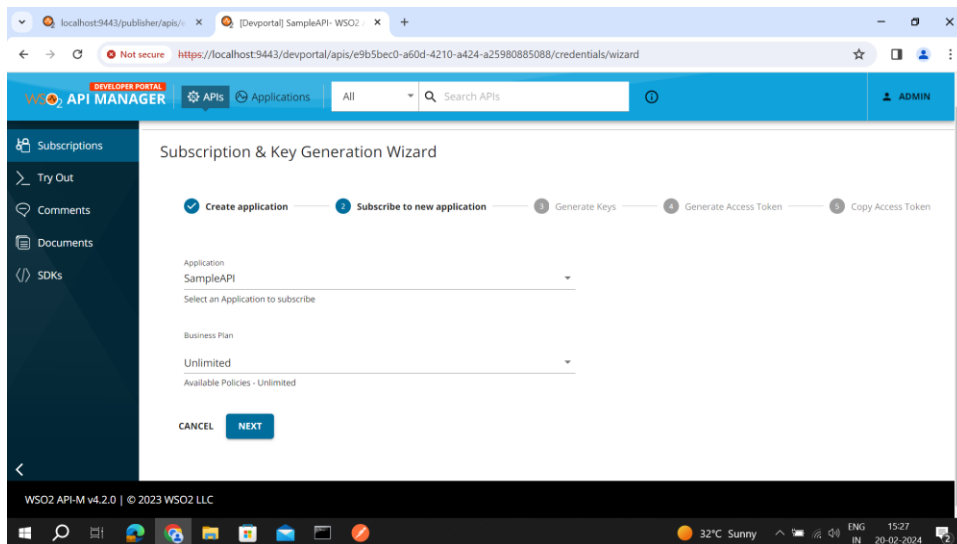
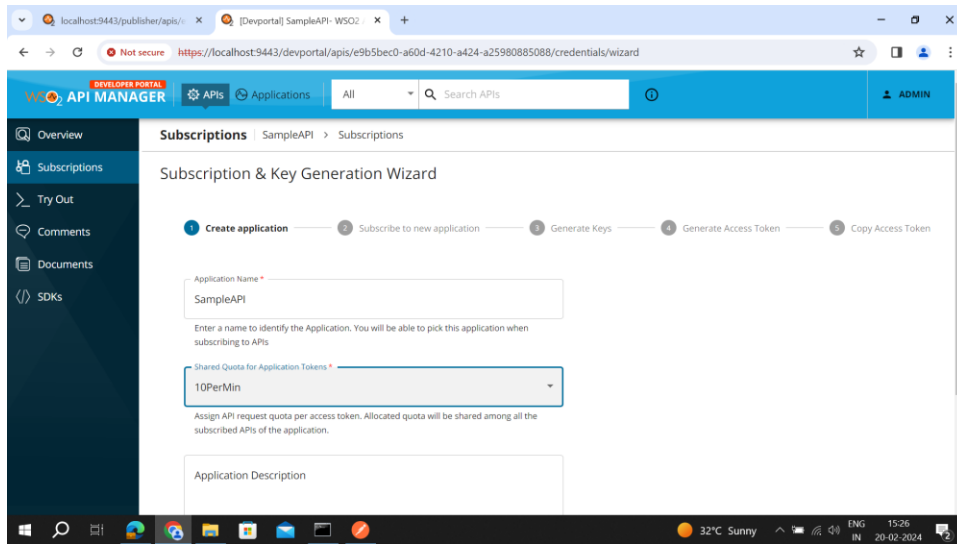
- Now publish your API.

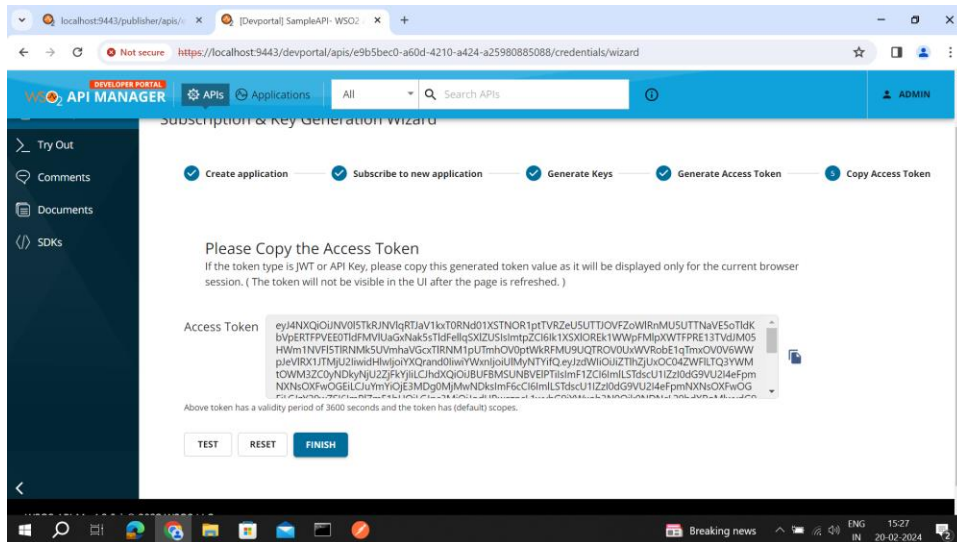




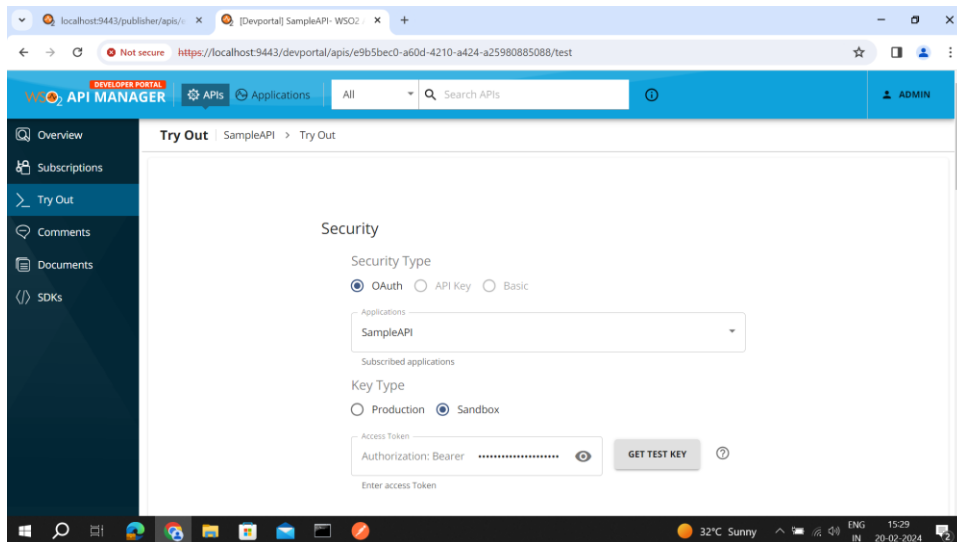


- Now navigate to developer console and do subscription before using it.





- Now Tryout your Published API with get method.



- Give Execute in GET Request.

- 
- The screenshot shows the WS02 API Manager interface. The top navigation bar includes the WS02 logo, a 'DEVELOPER PORTAL' button, and links for 'APIs', 'Applications', and 'ADMIN'. The main content area displays the details of a GET request to the endpoint `/apis/`. The response body is a JSON array containing two objects. The first object represents a completed task with ID 0, and the second object represents a task that is not completed with ID 1. Below the response body, the 'Response headers' section lists several headers, including 'cache-control', 'content-type', 'expires', and 'pragma'. At the bottom, a table shows the response status as '200 ok' with no links.
- | Code | Description | Links    |
|------|-------------|----------|
| 200  | ok          | No links |

- 
- The screenshot displays the Postman application interface. The top navigation bar includes links for Home, Workspaces, API Network, Search Postman, Invite, and Upgrade. The left sidebar shows 'My Workspace' with a 'New' button and 'Import' button. Below this are 'Collections', 'Environments', and 'History'. The main workspace area is titled 'REST API basics: CRUD, test & variable / Get data'. It shows a 'GET' request to 'https://localhost:8243/SampleAPI/1.0.0/todos'. The 'Authorization' header is set to 'Bearer ...'. The response body is shown in JSON format, indicating a successful GET request with a 200 OK status.