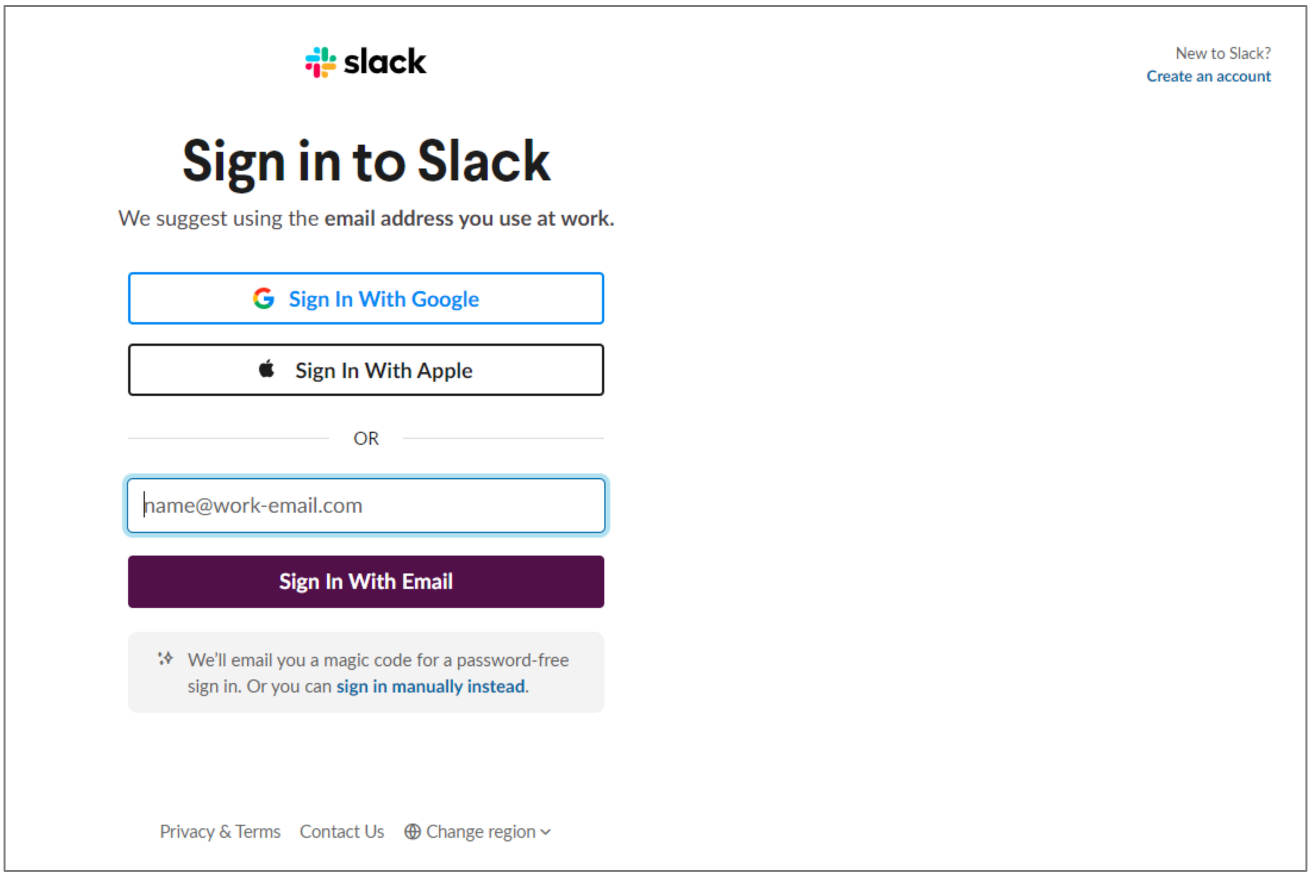
Assignment 4 **Chathushi Jayarathna**

Module No:	1	**IU No:**	5	**Exercise No.**	5

Lab Assessment Statement	**<u>Assignment 4 – Implement webhook integration with external App</u>** **Referring to Assignment-1's Project Scenario,** The scope of this assignment is to implement webhook Integration for notification using Slack for 'XYZ Cars Pte Ltd'. These are the steps provided to build an API. 1. Set up slack for Integration 2. Creating an App for webhook integration 3. Enabling Webhook in App 4. Implement webhook Integration in XYZ portal which is already developed. 5. Testing using Postman 6. Use the API in React Application. **Provide the source code of developed application.** **Provide screen capture of final result pages for API testing.**
Technical Environment	- Spring Boot, Slack for Integration, POSTMAN, React JS
Guidelines	-
Duration	120 mins

1. Set up slack for Integration

Step 1: Visit https://slack.com/signin#/signin



slack

New to Slack?
Create an account

Sign in to Slack

We suggest using the **email address you use at work.**

Sign In With Google

Sign In With Apple

OR

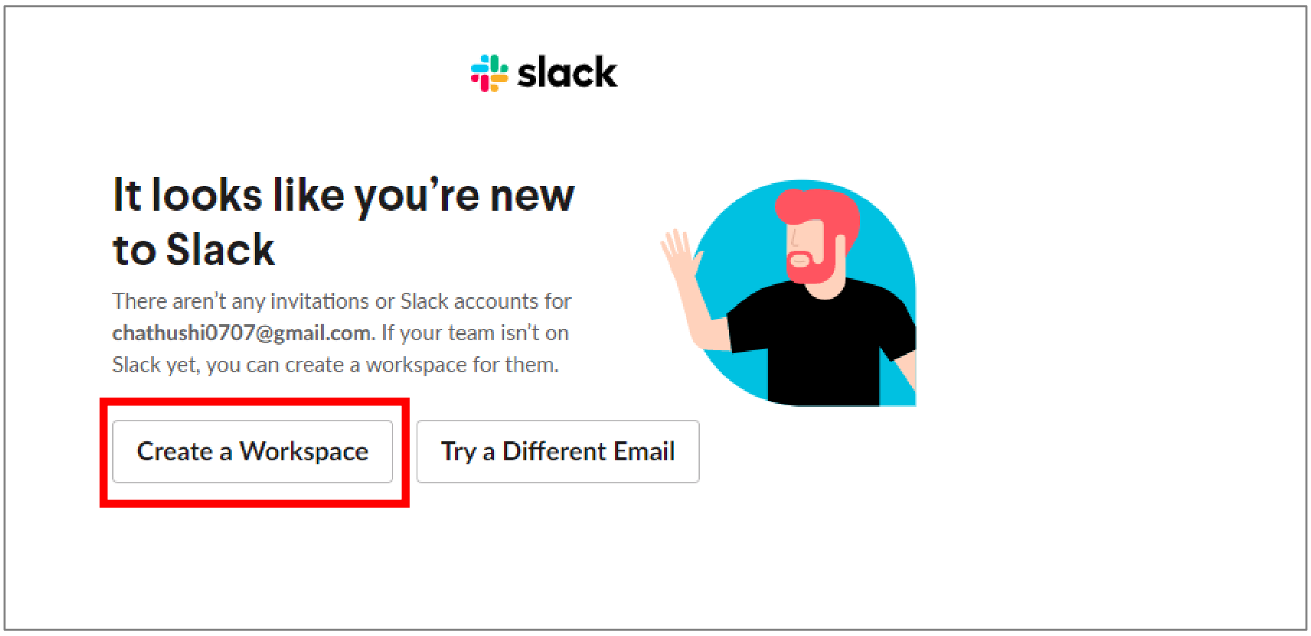
name@work-email.com

Sign In With Email

We'll email you a magic code for a password-free sign in. Or you can **sign in manually instead.**

Privacy & Terms Contact Us Change region

Step 2: Create new workspace after successful Login



slack

It looks like you're new to Slack

There aren't any invitations or Slack accounts for **chathushi0707@gmail.com**. If your team isn't on Slack yet, you can create a workspace for them.

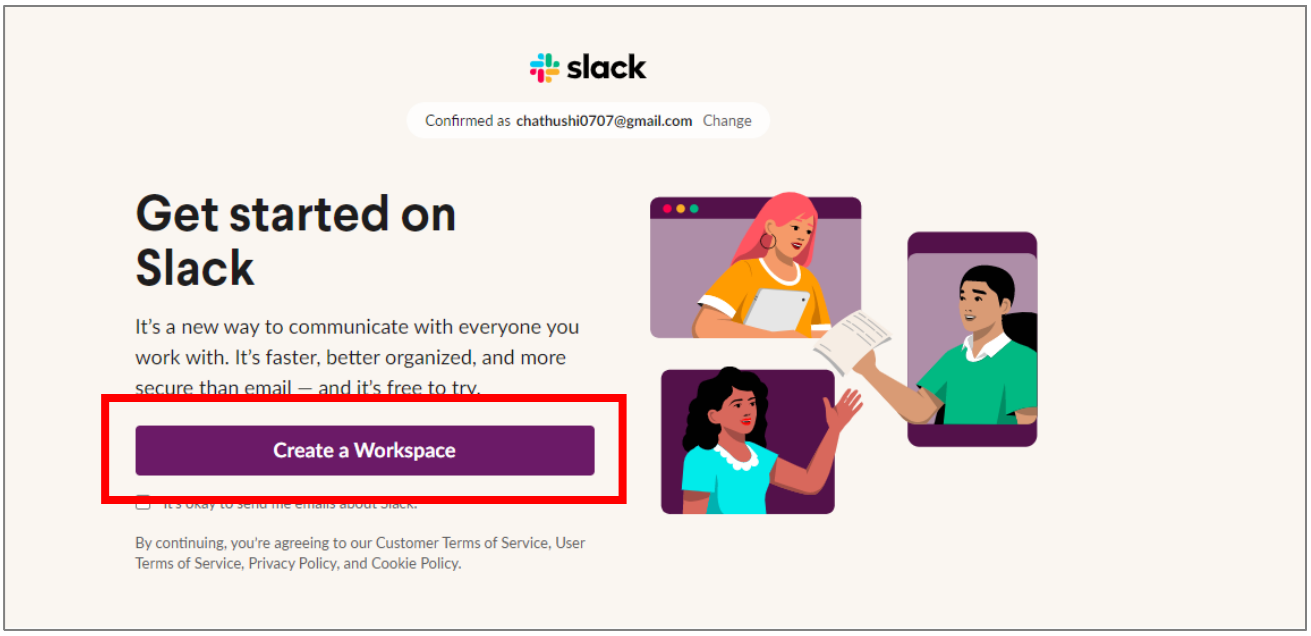
Create a Workspace

Try a Different Email

Step 3: Steps to create new Workspace

Click on create new workspace

Enter your team name

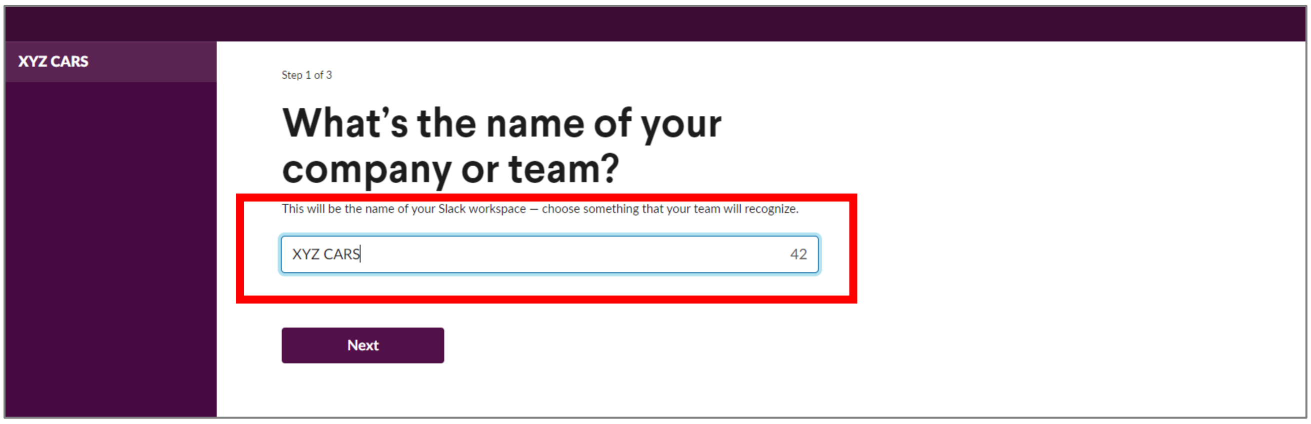

Confirmed as chathushi0707@gmail.com Change

Get started on Slack

It's a new way to communicate with everyone you work with. It's faster, better organized, and more secure than email — and it's free to try.

It's okay to send me emails about Slack.

By continuing, you're agreeing to our Customer Terms of Service, User Terms of Service, Privacy Policy, and Cookie Policy.


XYZ CARS

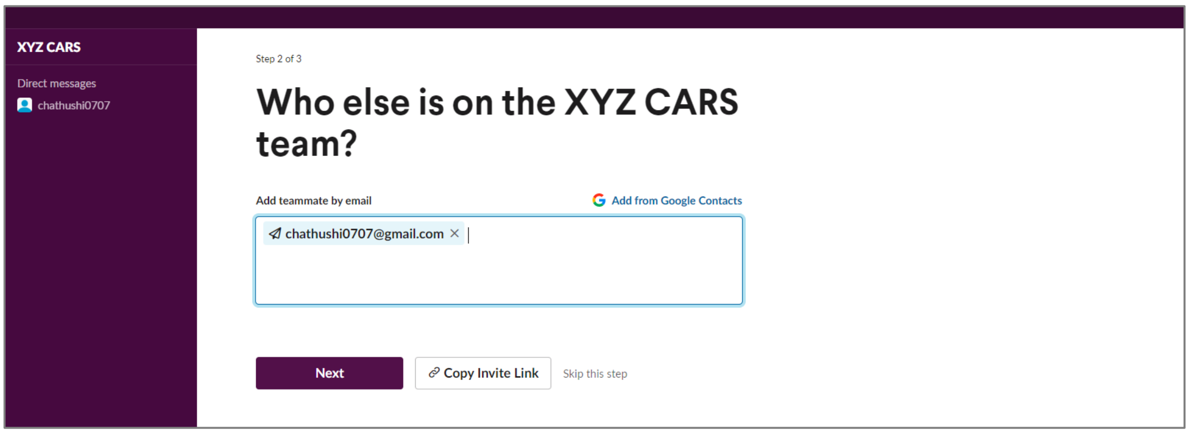
Step 1 of 3

What's the name of your company or team?

This will be the name of your Slack workspace — choose something that your team will recognize.

XYZ CARS 42

Step 4: Enter your Team working details


XYZ CARS

Direct messages

chathushi0707

Step 2 of 3

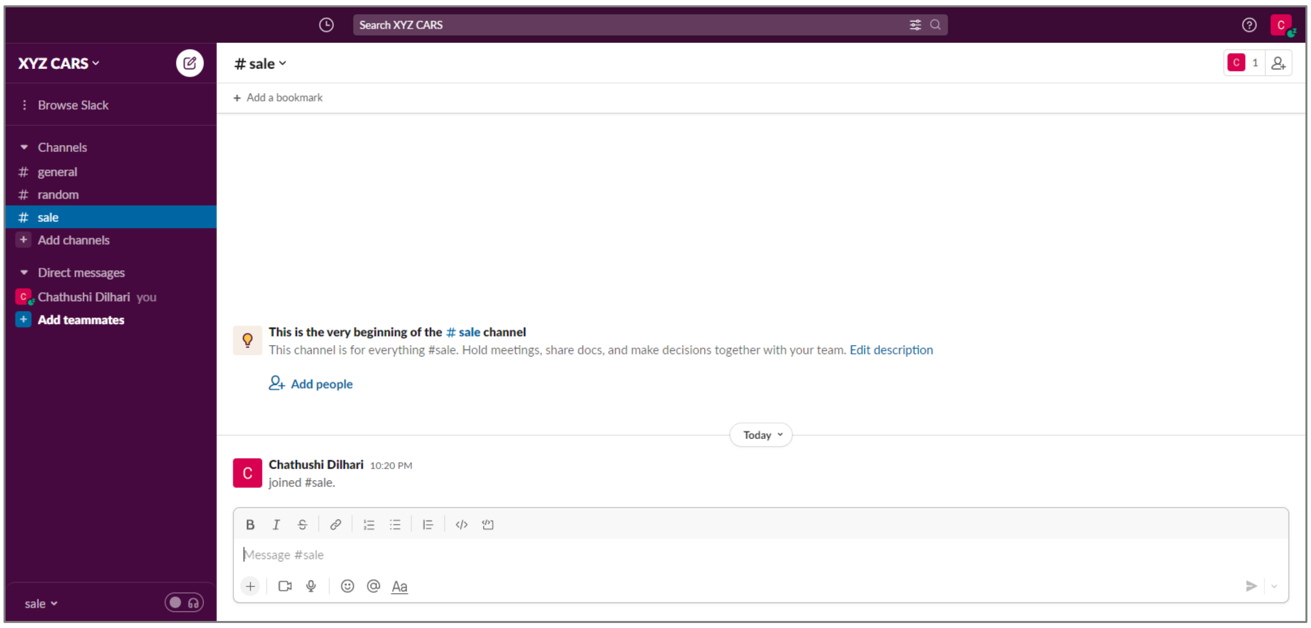
Who else is on the XYZ CARS team?

Add teammate by email

Add from Google Contacts

chathushi0707@gmail.com

Step 5: You can see the webhook-learning channel with added teammates. Here, you can send messages to your team members.

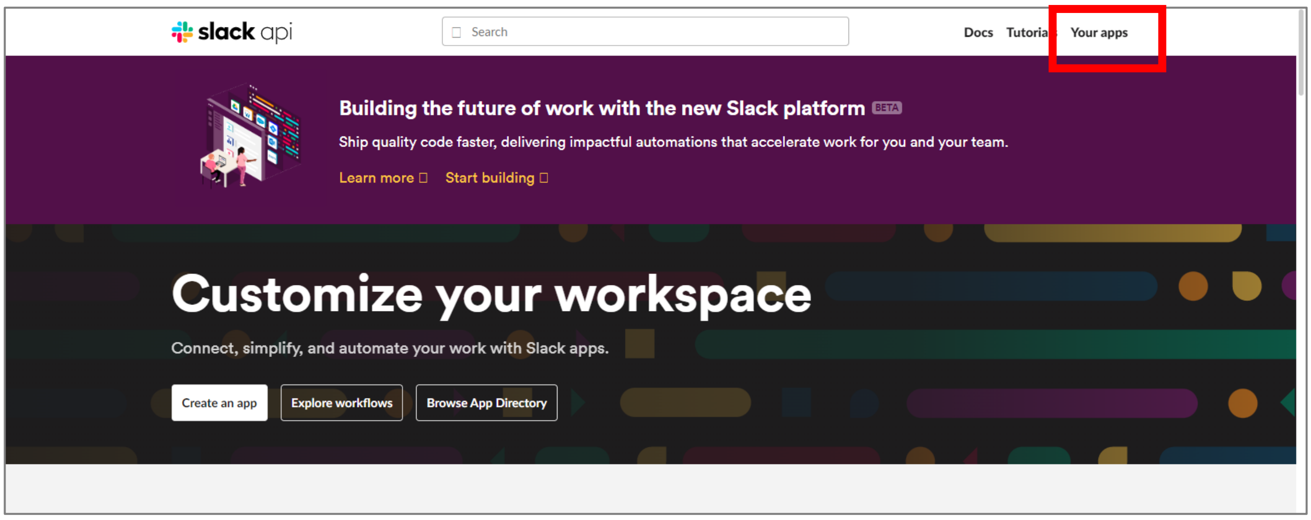


This is the very beginning of the # sale channel
This channel is for everything #sale. Hold meetings, share docs, and make decisions together with your team. Edit description

Chathushi Dilhari 10:20 PM
joined #sale.

2. Creating an App for webhook integration

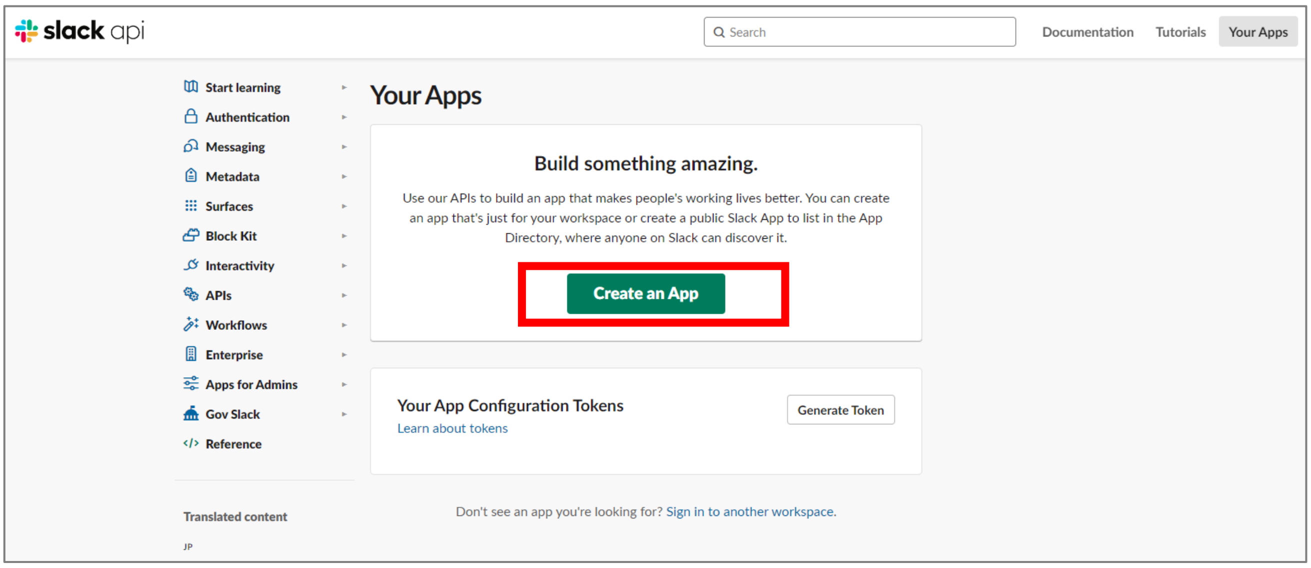
Step 1: Visit https://api.slack.com and click "your app" button



Building the future of work with the new Slack platform BETA
Ship quality code faster, delivering impactful automations that accelerate work for you and your team.

Customize your workspace
Connect, simplify, and automate your work with Slack apps.

Step 2: Click on create new app



Your Apps

Build something amazing.

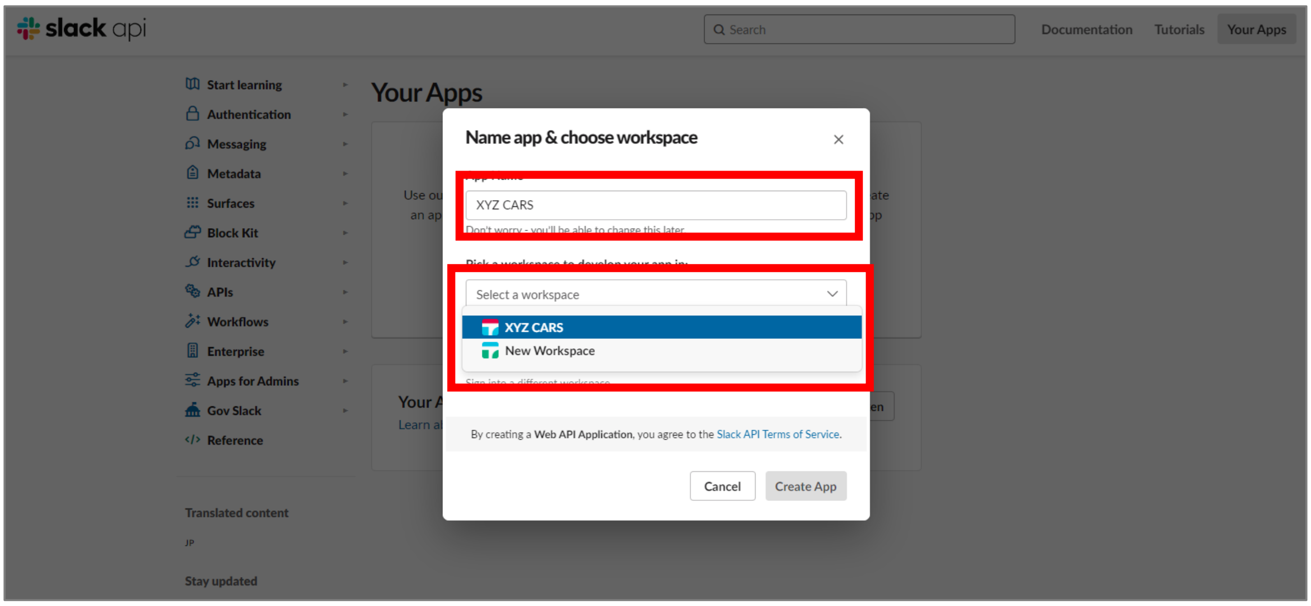
Use our APIs to build an app that makes people's working lives better. You can create an app that's just for your workspace or create a public Slack App to list in the App Directory, where anyone on Slack can discover it.

Your App Configuration Tokens

Learn about tokens

Don't see an app you're looking for? Sign in to another workspace.

Step 3: Add the App name and select the workspace



Name app & choose workspace

App Name

XYZ CARS

Don't worry - you'll be able to change this later

Pick a workspace to develop your app in:

Select a workspace

XYZ CARS

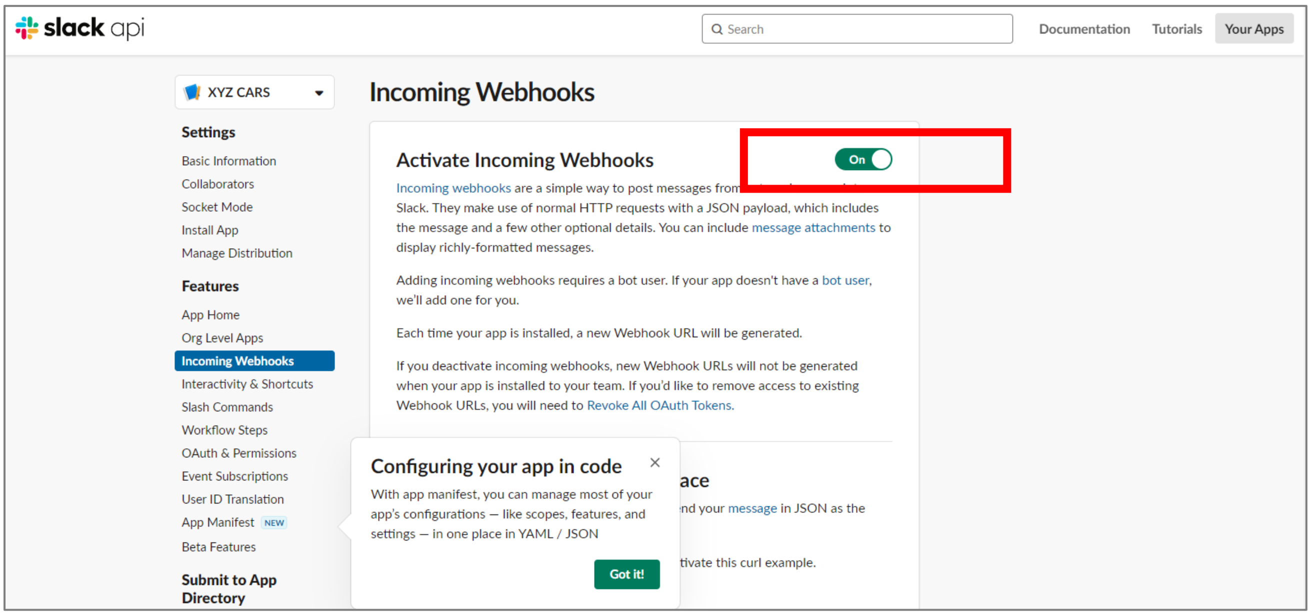
New Workspace

Sign into a different workspace

By creating a **Web API Application**, you agree to the Slack API Terms of Service.

3. Enabling Webhook in App

Step 1: Click on Incoming webhook and Activate incoming webhooks



Incoming Webhooks

Activate Incoming Webhooks

Incoming webhooks are a simple way to post messages from Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details. You can include message attachments to display richly-formatted messages.

Adding incoming webhooks requires a bot user. If your app doesn't have a bot user, we'll add one for you.

Each time your app is installed, a new Webhook URL will be generated.

If you deactivate incoming webhooks, new Webhook URLs will not be generated when your app is installed to your team. If you'd like to remove access to existing Webhook URLs, you will need to Revoke All OAuth Tokens.

Configuring your app in code

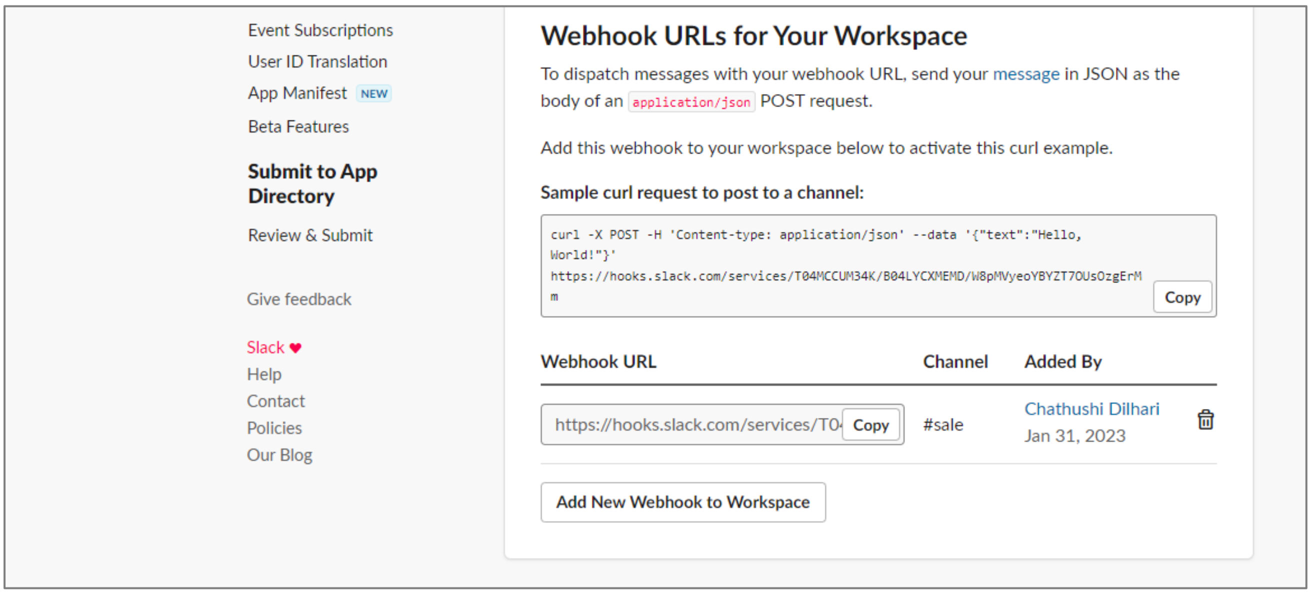
With app manifest, you can manage most of your app's configurations — like scopes, features, and settings — in one place in YAML / JSON

ace

nd your message in JSON as the

tivate this curl example.

Step 2: After activation of the webhook, you see the webhook URL of your workspace.



Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your message in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

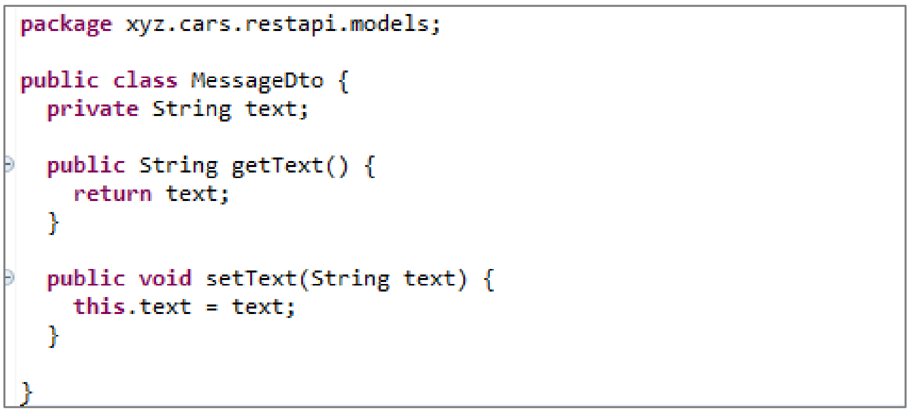
Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello,
World!"}'
https://hooks.slack.com/services/T04MCCUM34K/B04LYCXMEMD/W8pMVyeoYBYZT7OUsOzgErM
m
```

Webhook URL	Channel	Added By
https://hooks.slack.com/services/T0	#sale	Chathushi Dilhari Jan 31, 2023

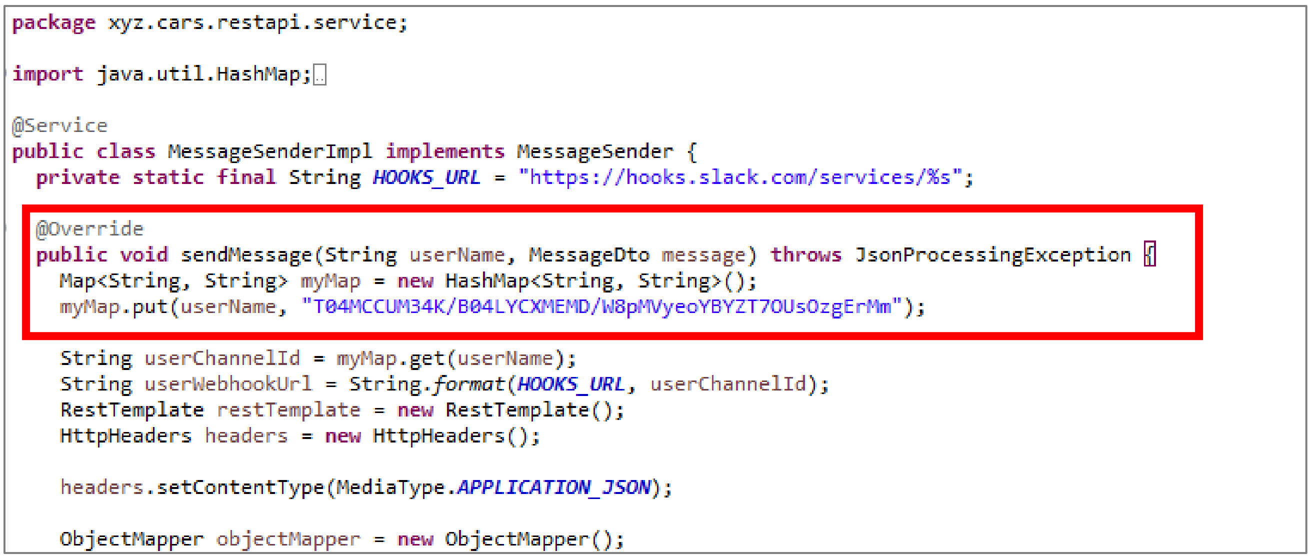
4. Implement webhook Integration in XYZ portal which is already developed.

MessageDto.java



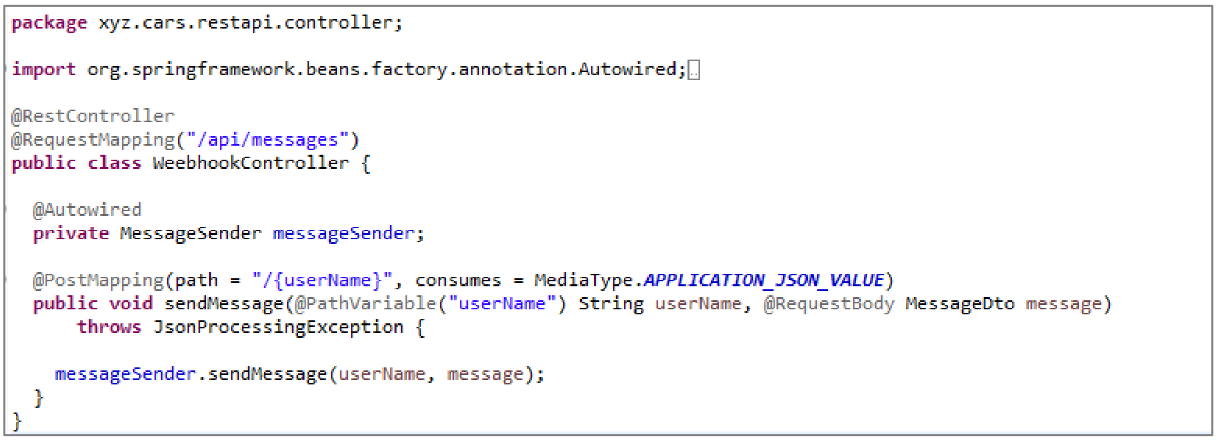
```java
package xyz.cars.restapi.models;

public class MessageDto {
  private String text;

  public String getText() {
    return text;
  }

  public void setText(String text) {
    this.text = text;
  }

}
```

MessageSenderImpl.java



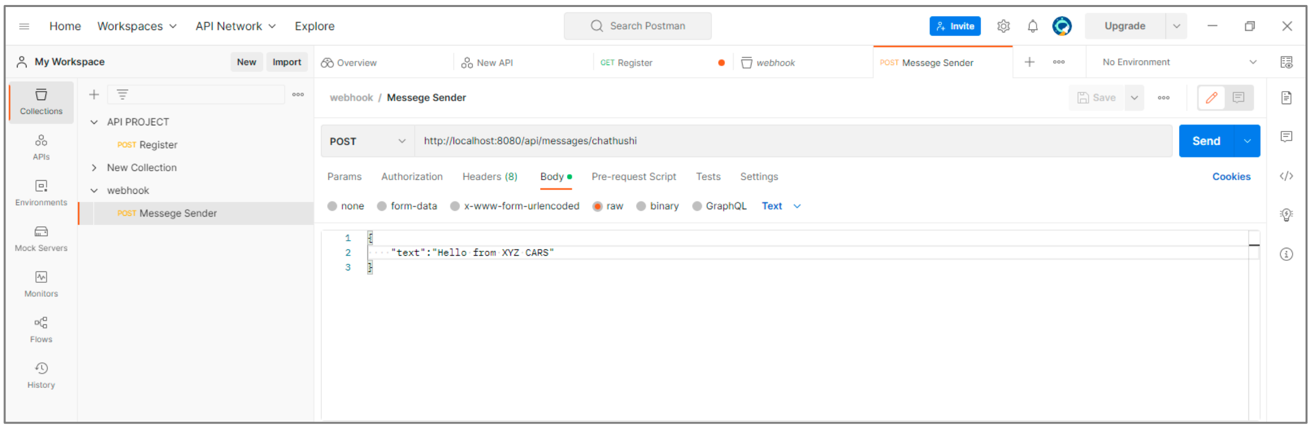
```java
package xyz.cars.restapi.service;

import java.util.HashMap;

@Service
public class MessageSenderImpl implements MessageSender {
  private static final String HOOKS_URL = "https://hooks.slack.com/services/%s";

  @Override
  public void sendMessage(String userName, MessageDto message) throws JsonProcessingException {
    Map<String, String> myMap = new HashMap<String, String>();
    myMap.put(userName, "T04MCCUM34K/B04LYCXMEMD/W8pMVyeoYBYZT7OUsOzgErMm");

    String userChannelId = myMap.get(userName);
    String userWebhookUrl = String.format(HOOKS_URL, userChannelId);
    RestTemplate restTemplate = new RestTemplate();
    HttpHeaders headers = new HttpHeaders();

    headers.setContentType(MediaType.APPLICATION_JSON);

    ObjectMapper objectMapper = new ObjectMapper();
```

WebHookController.java



```java
package xyz.cars.restapi.controller;

import org.springframework.beans.factory.annotation.Autowired;

@RestController
@RequestMapping("/api/messages")
public class WeebhookController {

  @Autowired
  private MessageSender messageSender;

  @PostMapping(path = "/{userName}", consumes = MediaType.APPLICATION_JSON_VALUE)
  public void sendMessage(@PathVariable("userName") String userName, @RequestBody MessageDto message)
      throws JsonProcessingException {

    messageSender.sendMessage(userName, message);
  }
}
```

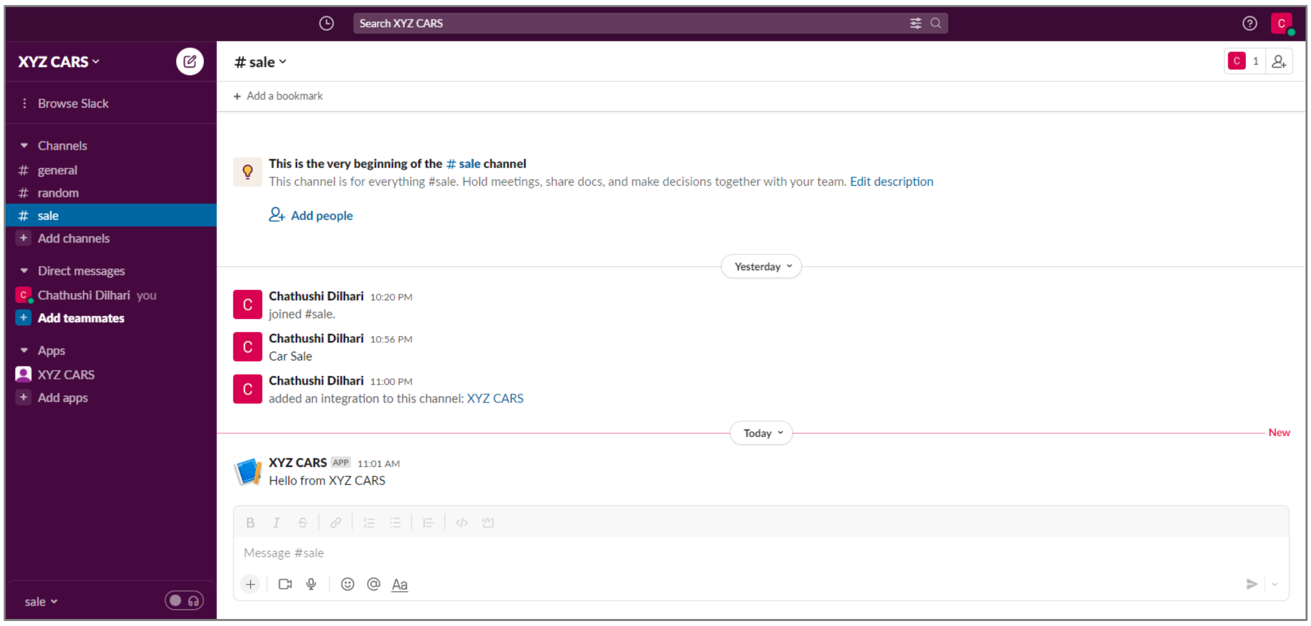
5. Testing using Postman



POST http://localhost:8080/api/messages/chathushi

Send

```
{
    "text":"Hello from XYZ CARS"
}
```



This is the very beginning of the # sale channel
This channel is for everything #sale. Hold meetings, share docs, and make decisions together with your team. Edit description

Chathushi Dilhari 10:20 PM
joined #sale.

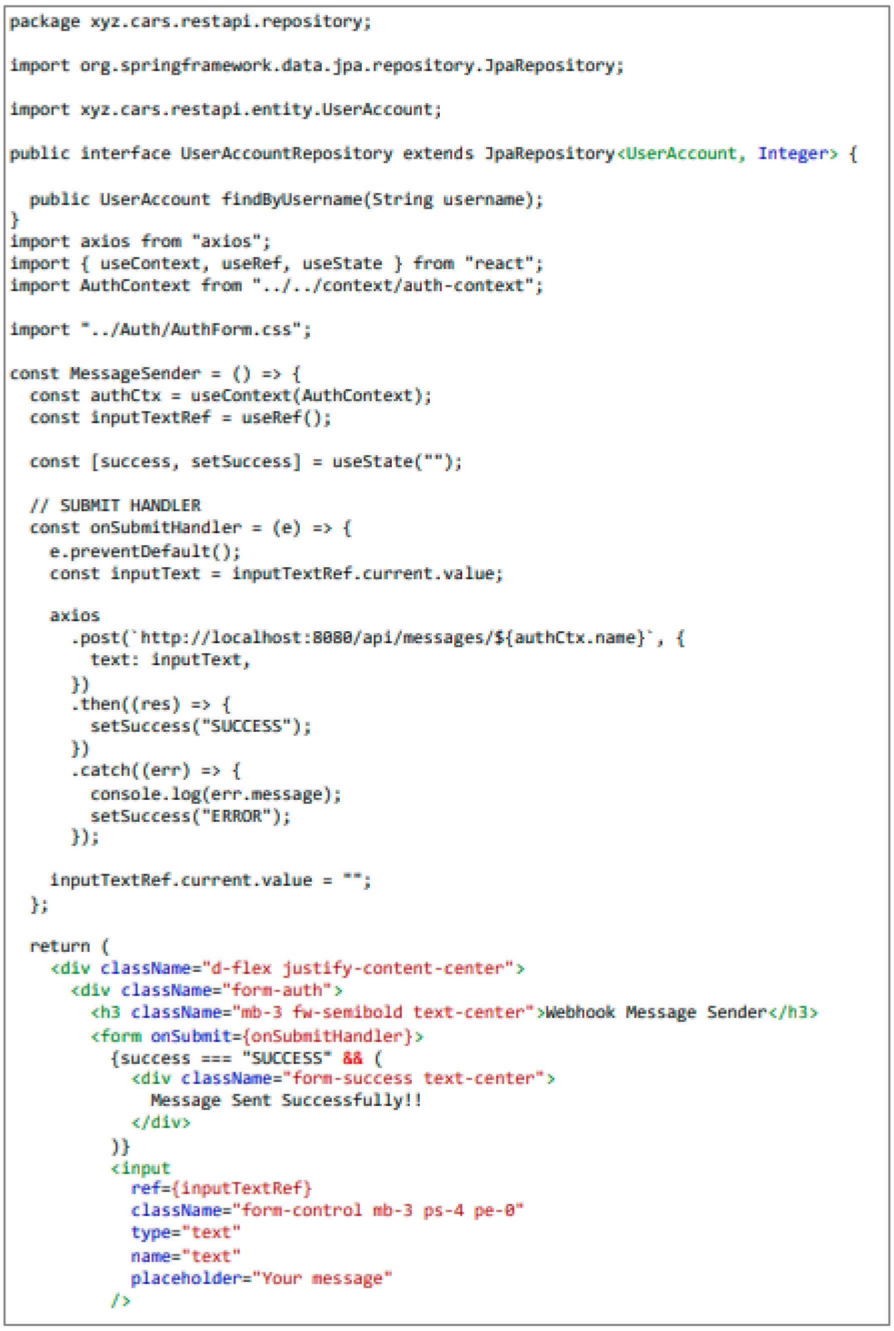
Chathushi Dilhari 10:56 PM
Car Sale

Chathushi Dilhari 11:00 PM
added an integration to this channel: XYZ CARS

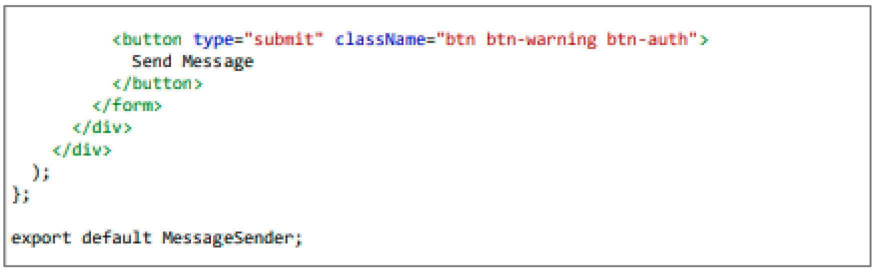
XYZ CARS APP 11:01 AM
Hello from XYZ CARS

6. Use the API in React Application.

   MessegeSender.java

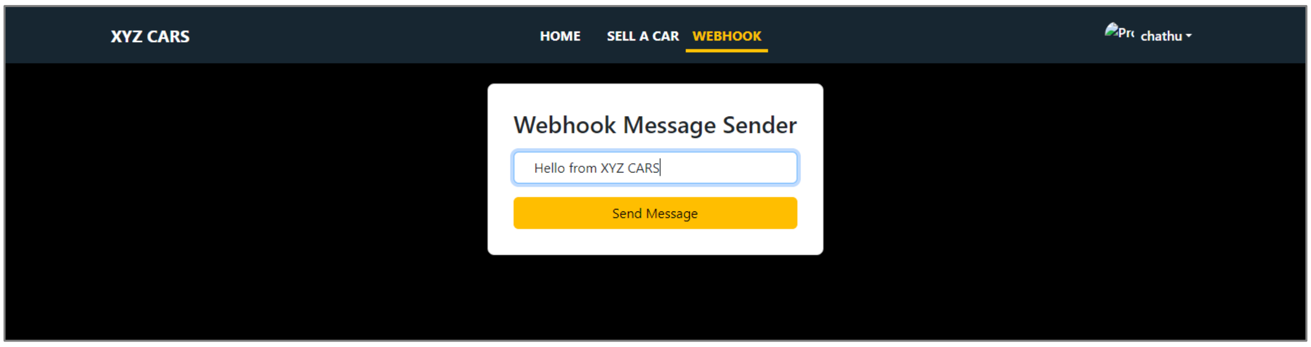


```
package xyz.cars.restapi.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import xyz.cars.restapi.entity.UserAccount;

public interface UserAccountRepository extends JpaRepository<UserAccount, Integer> {

  public UserAccount findByUsername(String username);
}
import axios from "axios";
import { useContext, useRef, useState } from "react";
import AuthContext from "../../context/auth-context";

import "../Auth/AuthForm.css";

const MessageSender = () => {
  const authCtx = useContext(AuthContext);
  const inputTextRef = useRef();

  const [success, setSuccess] = useState("");

  // SUBMIT HANDLER
  const onSubmitHandler = (e) => {
    e.preventDefault();
    const inputText = inputTextRef.current.value;

    axios
      .post(`http://localhost:8080/api/messages/${authCtx.name}`, {
        text: inputText,
      })
      .then((res) => {
        setSuccess("SUCCESS");
      })
      .catch((err) => {
        console.log(err.message);
        setSuccess("ERROR");
      });

    inputTextRef.current.value = "";
  };

  return (
    <div className="d-flex justify-content-center">
      <div className="form-auth">
        <h3 className="mb-3 fw-semibold text-center">Webhook Message Sender</h3>
        <form onSubmit={onSubmitHandler}>
          {success === "SUCCESS" && (
            <div className="form-success text-center">
              Message Sent Successfully!!
            </div>
          )}
          <input
            ref={inputTextRef}
            className="form-control mb-3 ps-4 pe-0"
            type="text"
            name="text"
            placeholder="Your message"
          />
```



```
          <button type="submit" className="btn btn-warning btn-auth">
            Send Message
          </button>
        </form>
      </div>
    </div>
  );
};

export default MessageSender;
```

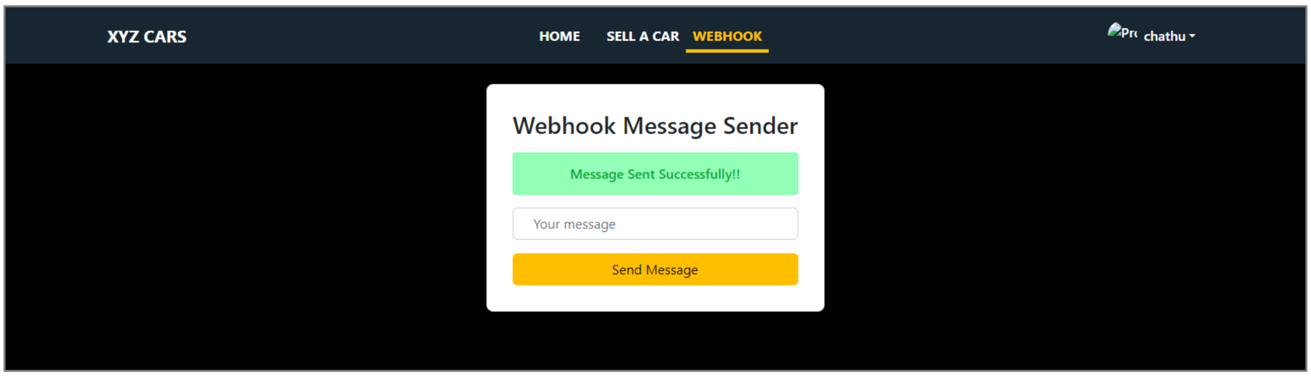
Assignment 4 **Chathushi Jayarathna**



Webhook Message Sender

Hello from XYZ CARS

Send Message

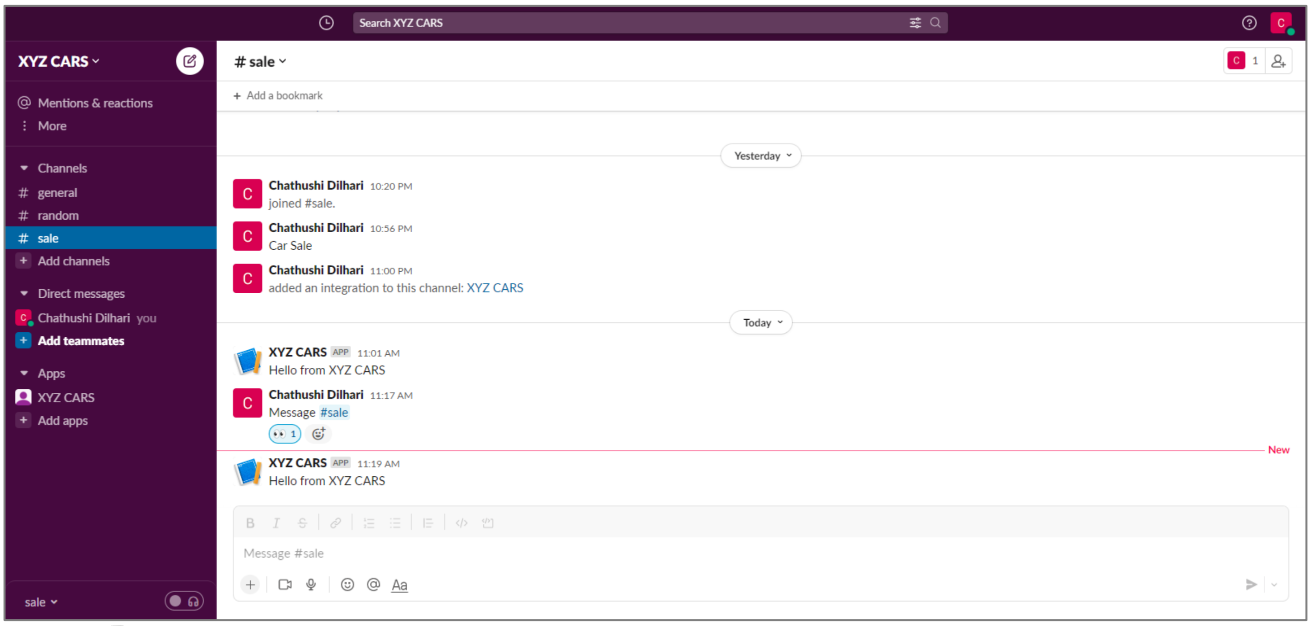


Webhook Message Sender

Message Sent Successfully!!

Your message

Send Message



Yesterday

Chathushi Dilhari 10:20 PM
joined #sale.

Chathushi Dilhari 10:56 PM
Car Sale

Chathushi Dilhari 11:00 PM
added an integration to this channel: XYZ CARS

Today

XYZ CARS APP 11:01 AM
Hello from XYZ CARS

Chathushi Dilhari 11:17 AM
Message #sale

New

XYZ CARS APP 11:19 AM
Hello from XYZ CARS

Assignment 4.zip