Here are the backend steps to fetch API data:

Step 1: Choose a Programming Language and Framework

- Choose a programming language and framework to build the backend API. For example, Node.js with Express.js, Python with Flask, or Java with Spring Boot.

Step 2: Set up the Project Structure

- Create a new project directory and set up the project structure.
- Create a **package.json** file to manage dependencies.

Step 3: Install Required Dependencies

- Install the required dependencies using npm or pip. For example, **express** for Node.js, **requests** for Python, or **okhttp** for Java.

Step 4: Create an API Route

- Create an API route to handle the API request. For example, **/api/weather** to fetch weather data.

Step 5: Make an API Request

- Use a library or module to make an API request to the weather API. For example, **axios** for Node.js, **requests** for Python, or **OkHttp** for Java.

Step 6: Handle API Response

- Handle the API response and extract the required data.
- Use a JSON parser to parse the API response.

Step 7: Return API Data

- Return the API data in a JSON format.
- Use a templating engine to render the API data in a HTML template.

Step 8: Test the API

- Test the API using a tool like Postman or cURL.
- Verify that the API returns the correct data.

Step 9: Deploy the API

- Deploy the API to a cloud platform like Heroku, AWS, or Google Cloud.
- Configure the API to handle production traffic.

```
const express = require('express');
const axios = require('axios');

const app = express();

app.get('/api/weather', async (req, res) => {
  const location = req.query.location;
  const apiKey = 'YOUR-API-KEY';

  try {
    const response = await axios.get(`https://api.weatherstack.com/current?access_key=$
{apiKey}&query=${location}`);
    const data = response.data;

    res.json({
      location: data.location.name,
      temperature: data.current.temperature,
      weatherDescription: data.current.weather_descriptions[0]
    });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Failed to fetch weather data' });
  }
});

app.listen(3000, () => {
  console.log('Server started on port 3000');
});
```