

API Call to Web Display Process

Below is a step-by-step guide on how an API call is made, tested, stored in Sanity, and displayed on a web application.

1. Making an API Call

- Use `fetch()` in JavaScript or any HTTP client like Axios.
- Send a request to the API endpoint (GET, POST, PUT, DELETE).
- Receive the response in JSON format.

2. Testing the API Response

- Use **Postman** or **Thunder Client** to manually test the API response.
- Check the status code (200, 201, 400, 500, etc.).
- Verify the response structure.

3. Pushing Data to Sanity

- Use Sanity's API to create or update content.
- Authenticate using API keys or tokens.
- Send a POST request to Sanity's dataset with the data received from the API.

4. Checking Data in Sanity Studio

- Open **Sanity Studio**.
- Navigate to the relevant schema (e.g., `posts`, `products`, etc.).
- Confirm that the new data is stored correctly.

5. Fetching Data with GROQ Query

- Use Sanity's GROQ query to fetch stored data.
- Example: `*[_type == "post"]{title, body, author}`.
- Integrate the query in a frontend framework like React, Next.js, or Vue.

6. Displaying Data on a Web Page

- Render the fetched data dynamically.
- Use a loop or map function to show multiple records.
- Ensure proper error handling.

Flowchart Representation:

+-----+

