Salesforce Project: Public Endpoint Access App with VS Code & LWC

Step 1: Set Up Salesforce DX and VS Code

1. Install VS Code and Salesforce CLI.

2. Install Salesforce Extension Pack in VS Code.

3. Authenticate to your org using:

   sfdx auth:web:login -a DevOrg

4. Create a project:

   sfdx force:project:create -n PublicAPIApp

5. Open the project in VS Code.

Step 2: Create a Lightning Web Component (LWC)

1. Run command:

   sfdx force:lightning:component:create -n PublicDataViewer -d force-app/main/default/lwc

2. LWC Files:

- PublicDataViewer.html:

```
<template>
  <lightning-card title="Public Data Viewer">
    <lightning-input label="Enter Record Id" value={recordId} onchange={handleChange}></lightning-input>
    <lightning-button label="Get Data" onclick={handleClick}></lightning-button>
    <p>{result}</p>
  </lightning-card>
</template>
```

- PublicDataViewer.js:

```
import { LightningElement } from 'lwc';
```

```
export default class PublicDataViewer extends LightningElement {
  recordId = '';
  result;

  handleChange(event) {
    this.recordId = event.target.value;
  }

  handleClick() {
    fetch('/services/apexrest/publicdata/' + this.recordId)
      .then(res => res.json())
      .then(data => {
        this.result = JSON.stringify(data);
      })
      .catch(err => {
        this.result = 'Error fetching data';
      });
  }
}
```

Step 3: Create the Apex REST Class

```
@RestResource(urlMapping='/publicdata/*')
global with sharing class PublicDataController {
  @HttpGet
```

```
  global static PublicData__c getData() {
    RestRequest req = RestContext.request;
    String id = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);
    return [SELECT Id, Name, Description__c FROM PublicData__c WHERE Id = :id LIMIT 1];
  }
}
```

Step 4: Deploy Components to Org

1. Save your changes.
2. Use the command:

   sfdx force:source:deploy -p force-app

3. Verify deployment in your org.

Step 5: Create a Lightning App Page

1. Go to Lightning App Builder in Setup.
2. Create a new App Page.
3. Drag and drop the 'PublicDataViewer' component onto the page.
4. Activate and assign it to apps/navigation.

Step 6: Make API Public via Sites (Optional)

1. Setup > Sites > Create Site.
2. Set Visualforce Page as home page (if needed).
3. Grant Guest User access to Apex Class and object permissions.

Salesforce Project: Public Endpoint Access App with VS Code & LWC

Security Note

- Use authentication for sensitive APIs.

- Always restrict Guest User access to necessary objects and fields only.