# Exercise 2: Working with the web part property pane

## Task 1: Create a new SPFx solution and web part

1. From the PowerShell command prompt, change to the **C:/LabFiles/SharePoint** directory by executing the following command: cd c:/LabFiles/SharePoint
2. Make a new directory for your SharePoint project files by executing the following command: md HelloPropertyPane
3. Navigate to the newly created SharePoint directory by executing the following command: cd HelloPropertyPane
4. Run the SharePoint Yeoman generator by executing the following command: yo @microsoft/sharepoint
5. Use the following to complete the prompt that is displayed:
   * **What is your solution name?**: HelloPropertyPane
   * **Which baseline packages do you want to target for your component(s)?**: SharePoint Online only (latest)
   * **Where do you want to place the files?**: Use the current folder
   * **Do you want to allow the tenant admin the choice of being able to deploy the solution to all sites immediately without running any feature deployment or adding apps in sites?**: No
   * **Will the components in the solution require permissions to access web APIs that are unique and not shared with other components in the tenant?**: No
   * **Which type of client-side component to create?**: WebPart
   * **What is your Web part name?**: HelloPropertyPane
   * **What is your Web part description?**: HelloPropertyPane description
   * **Which framework would you like to use?**: No JavaScript framework
6. Open the project in Visual Studio Code by executing the following command: code .
7. From the Visual Studio Code ribbon, select **Terminal > New Terminal**.
8. Verify that everything is working. Execute the following command to build, start the local web server, and test the web part in the local workbench: gulp serve
9. When the browser loads the local workbench, select the **Add a New web part** control.
10. Select the **HelloPropertyPane** web part to add the web part to the page.
11. Select the **edit web part** control to the side of the web part to display the property pane.

## Task 2: Add new properties to the web part

1. Create two new properties that will be used in the web part and property pane.
   1. From Visual Studio Code, open the file **src.ts**
   2. Locate the interface **IHelloPropertyPaneWebPartProps** after the import statements. Add the following two properties to the interface:
      * **myContinent: string;**
      * **numContinentsVisited: number;**
2. Update the web part rendering to display the values of these two properties:
   1. Within the **HelloPropertyPaneWebPart** class, locate the **render()** method.
   2. Within the **render()** method, locate the following line in the HTML output:
   * <p class="${ styles.description }">${escape(this.properties.description)}</p>
   1. Add the following two lines after the line you just located:
   * <p class="${ styles.description }">Continent where I reside: ${escape(this.properties.myContinent)}</p><p class="${ styles.description }">Number of continents I've visited: ${this.properties.numContinentsVisited}</p>
   1. At the moment the web part will render a blank string and undefined for these two fields as nothing is present in their values. This can be addressed by setting the default values of properties when a web part is added to the page.
3. Set the default property values:
   1. Open the file **src.manifest.json**
   2. Locate the following section in the file: preconfiguredEntries[0].properties.description
   3. Add a comma after the **description** property’s value.
   4. Add the following two lines after the description property:
   * "myContinent": "North America","numContinentsVisited": 4
4. Updates to the web part’s manifest file will not be picked up until you restart the local web server.
   1. In the command prompt, press CTRL+C to stop the local web server.
   2. Rebuild and restart the local web server by executing the command gulp serve.
   3. When the SharePoint Workbench loads, add the web part back to the page to see the properties.

## Task 3: Extend the property pane

Now that the web part has two new custom properties, the next step is to extend the property pane to allow users to edit the values.

1. Add a new text control to the property pane, connected to the **myContinent** property:
   1. Open the file **src.ts**
   2. Locate the method **getPropertyPaneConfiguration** and within it, locate the **groupFields** array.
   3. Add a comma after the existing **PropertyPaneTextField()** call.
   4. Add the following code after the comma:
   * PropertyPaneTextField('myContinent', {  
     label: 'Continent where I currently reside'  
     })
2. If the local web server is not running, start it by executing gulp serve
3. Once the SharePoint Workbench is running again, add the web part to the page and open the property pane.
4. Notice that you can see the property and text box in the property pane. Any edits to the field will automatically update the web part.
5. Now, add a slider control to the property pane, connected to the **numContinentsVisited** property:
   1. In the **HelloPropertyPaneWebPart.ts**, at the top of the file, add a **PropertyPaneSlider** reference to the existing import statement for the **@microsoft/sp-webpart-base** package.
   * Import { PropertyPaneSlider } from ‘@microsoft/sp-webpart-base’
   1. Scroll down to the method **getPropertyPaneConfiguration** and within it, locate the **groupFields** array.
   2. Add a comma after the last **PropertyPaneTextField()** call, and add the following code:
   * PropertyPaneSlider('numContinentsVisited', {  
     label: 'Number of continents I\'ve visited', min: 1, max: 7, showValue: true,  
     })
   1. Go back to the browser. Notice the property pane now has a slider control to control the number of continents you have visited.
6. In the command prompt, press CTRL+C to stop the local web server.
7. Close Visual Studio Code.

## Review

In this exercise, you learned how to change custom properties that exist in the web part manifest and how to add new properties.