

# Set Up Your Salesforce DX Environment

## **helloWorld.html**

```
<template>

<lightning-card title="HelloWorld" icon-name="custom:custom14">

  <div class="slds-m-around_medium">

    <p>Hello, {greeting}!</p>

    <lightning-input label="Name" value={greeting} onchange={changeHandler}></lightning-input>

  </div>

</lightning-card>

</template>
```

## **helloWorld.js**

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

export default class HelloWorld extends LightningElement {

  greeting = 'World';

  changeHandler(event) {

    this.greeting = event.target.value;

  }

}
```

## **helloWorld.js-meta.xml**

```
<?xml version="1.0" encoding="UTF-8"?>

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata"
  fqn="helloWorld">
```

```

<apiVersion>52.0</apiVersion>

<isExposed>true</isExposed>

<targets>

  <target>lightning__AppPage</target>

  <target>lightning__RecordPage</target>

  <target>lightning__HomePage</target>

</targets>

</LightningComponentBundle>

```

## Deploy LWC Files

### **bikeCard.html**

```

<template>

  <div>

    <div>Name: {name}</div>

    <div>Description: {description}</div>

    <lightning-badge label={material}></lightning-badge>

    <lightning-badge label={category}></lightning-badge>

    <div>Price: {price}</div>

    <div><img src={pictureUrl}/></div>

  </div>

</template>

```

### **bikeCard.js**

```

import { LightningElement } from 'lwc';

export default class BikeCard extends LightningElement {

  name = 'Electra X4';

  description = 'A sweet bike built for comfort.';

```

```
category = 'Mountain';  
material = 'Steel';  
price = '$2,700';  
pictureUrl = 'https://s3-us-west-1.amazonaws.com/sfdc-demo/ebikes/electrax4.jpg';  
}
```

#### **bikeCard.js-meta.xml**

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  
  <!-- The apiVersion may need to be increased for the current release -->  
  <apiVersion>52.0</apiVersion>  
  <isExposed>true</isExposed>  
  <masterLabel>Product Card</masterLabel>  
  <targets>  
    <target>lightning__AppPage</target>  
    <target>lightning__RecordPage</target>  
    <target>lightning__HomePage</target>  
  </targets>  
</LightningComponentBundle>
```

## **Add Styles and Data to a Lightning Web Component**

#### **detail.css**

```
.price {  
  color: green;  
  font-weight: bold;  
}
```

## detail.html

```
<template>

  <template if:true={product}>
    <div class="container">
      <div class="slds-text-heading_small">{product.fields.Name.value}</div>
      <div class="price">{product.fields.MSRP__c.displayValue}</div>
      <div class="description">{product.fields.Description__c.value}</div>
      <img class="product-img" src={product.fields.Picture_URL__c.value}></img>
      <p>
        <lightning-badge label={product.fields.Material__c.value}></lightning-badge>
        <lightning-badge label={product.fields.Level__c.value}></lightning-badge>
      </p>
      <p>
        <lightning-badge label={product.fields.Category__c.value}></lightning-badge>
      </p>
    </div>
  </template>

  <template if:false={product}>
    <div class="slds-text-heading_medium">Select a bike</div>
  </template>
</template>
```

## selector.js

```
import { LightningElement, wire } from 'lwc';
import { getRecord, getFieldValue } from 'lightning/uiRecordApi';
import Id from '@salesforce/user/Id';
import NAME_FIELD from '@salesforce/schema/User.Name';
const fields = [NAME_FIELD];

export default class Selector extends LightningElement {
  selectedProductId;
```

```

handleProductSelected(evt) {
    this.selectedProductId = evt.detail;
}

userId = Id;

@wire(getRecord, { recordId: '$userId', fields })

user;

get name() {
    return getFieldValue(this.user.data, NAME_FIELD);
}
}

```

### **selector.html**

```

<template>
    <div class="wrapper">
        <header class="header">Available Bikes for {name}</header>
        <section class="content">
            <div class="columns">
                <main class="main" >
                    <c-list onproductselected={handleProductSelected}></c-list>
                </main>
                <aside class="sidebar-second">
                    <c-detail product-id={selectedProductId}></c-detail>
                </aside>
            </div>
        </section>
    </div>
</template>

```