

Build your own parking app

# Title

## Subtitle

### Subsubtitle

- list item
- list item
- list item

See <https://docs.decksetapp.com/English.lproj/getting-started.html>

```
<body>  
  hello  
</body>
```

html, body, opening and closing tags, displaying text

```
<body>  
  <input>  
</body>
```

input box

```
<body>  
  <div>  
    <input>  
  </div>  
</body>
```

- divs
- opening and closing tags (again)

```
<body>  
  <div>  
    <input placeholder="Enter a postcode">  
  </div>  
</body>
```

html attributes, strings

```
<body>
  <div>
    <input placeholder="Enter a postcode">
  </div>
</body>
```

```
<script>
  console.log("test")
</script>
```

adding scripts, console, functions (calls), debugging

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>
```

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postal code">
  </div>
</body>
```

```
<script>
new Vue({
  el: '#myParkingApp'
})
</script>
```

importing vue, objects, new, Vue, Vue(), id



```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode">
      {{ postcode }}
    </div>
  </body>

<script>
new Vue({
  el: '#myParkingApp',
  data: {
    postcode: "123456"
  }
})
</script>
```

data object, interpolation

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode">
    {{ postcode }}
  </div>
</body>

<script>
new Vue({
  el: '#myParkingApp',
  data: {
    postcode: "123456"
  }
})
</script>
```

## v-model

01-carpark.html done

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}
  </div>
</body>
```

```
<script>
new Vue({
  el: '#myParkingApp',
  data: {
    postcode: ""
  },
  methods: {
    searchPostcode: function() {
      console.log("searchPostcode")
    }
  }
})
</script>
```

functions/methods, v-on:keyup-enter

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}
    {{ postcodeX }}
    {{ postcodeY }}
  </div>
</body>
```

```
<script>
new Vue({
  el: '#myParkingApp',
  data: {
    postcode: "",
    postcodeX: "",
    postcodeY: ""
  },
  methods: {
    searchPostcode: function() {
      this.postcodeX = "123"
      this.postcodeY = "456"
    }
  }
})
</script>
```

this, show desired result of this stage

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}
  </div>
</body>
```

```
<script>
new Vue({
  el: '#myParkingApp',
  data: {
    postcode: "",
    postcodeX: "",
    postcodeY: ""
  },
  methods: {
    searchPostcode: function() {
      this.postcodeX = "123"
      this.postcodeY = "456"
    }
  }
})
</script>
```

br

```
<head>  
  <script src="https://unpkg.com/vue"></script>  
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>  
</head>
```

import axios, explain what making HTTP requests is

```
searchPostcode: function() {  
  axios.get('https://developers.onemap.sg/commonapi/search', {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  })  
}
```

making HTTP request, params, looking at documentation



```
searchPostcode: function() {  
  axios.get('https://developers.onemap.sg/commonapi/search', {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    console.log(response)  
  })  
}
```

displaying response object, see what it looks like

```
searchPostcode: function() {  
  axios.get('https://developers.onemap.sg/commonapi/search', {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    console.log(response.data.results[0])  
  })  
}
```

arrays, unwrapping response object, dot notation

```
searchPostcode: function() {  
    axios.get('https://developers.onemap.sg/commonapi/search', {  
        params: {  
            searchVal: this.postcode,  
            returnGeom: "Y",  
            getAddrDetails: "N"  
        }  
    }).then(response => {  
        var unwrappedResults = response.data.results[0]  
        this.postcodeX = unwrappedResults.X  
        this.postcodeY = unwrappedResults.Y  
    })  
}
```

vars

02-carpark.html done

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">

    <div v-for="carpark in carpark">
      {{ carpark }}
    </div>
  </div>
</body>
```

...

```
data: {
  postcode: "",
  postcodeX: "",
  postcodeY: "",
  carpark: ["carpark1", "carpark2", "carpark3"]
}
```

# loops, arrays (again), displaying the result

```
methods: {
  searchPostcode: function() {
    axios.get('https://developers.onemap.sg/commonapi/search', {
      params: {
        searchVal: this.postcode,
        returnGeom: "Y",
        getAddrDetails: "N"
      }
    }).then(response => {
      var unwrappedResults = response.data.results[0]
      this.postcodeX = unwrappedResults.X
      this.postcodeY = unwrappedResults.Y
      this.getCarparkList()
    })
  },

  getCarparkList: function() {
    this.carparks = [1,2,3]
  }
}
```

## getCarparkList, displaying the result

```
getCarparkList: function() {  
    axios.get("https://data.gov.sg/api/action/datastore_search", {  
        params: {  
            resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
            limit: 2074  
        }  
    })  
}
```

making the API call, where we get the resource id and limit from

```
getCarparkList: function() {  
  axios.get("https://data.gov.sg/api/action/datastore_search", {  
    params: {  
      resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
      limit: 2074  
    }  
  }).then(response => {  
    console.log(response)  
  })  
}
```

exploring the response object



```
getCarparkList: function() {  
  axios.get("https://data.gov.sg/api/action/datastore_search", {  
    params: {  
      resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
      limit: 2074  
    }  
  }).then(response => {  
    var carparks = response.data.result.records  
  })  
}
```

unwrap the response to get carparks data. we only want to display the nearest carparks

```
getCarparkList: function() {  
  axios.get("https://data.gov.sg/api/action/datastore_search", {  
    params: {  
      resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
      limit: 2074  
    }  
  }).then(response => {  
    var carparks = response.data.result.records  
    var nearest_carparks = this.getNearestCarparks(carparks)  
    this.carparks = nearest_carparks  
  })  
}
```

hand off to another function

```
methods: {  
  searchPostcode: function() {  
    ...  
  },  
  getCarparkList: function() {  
    ...  
  },  
  getNearestCarpark: function(carparks) {  
  
  }  
}
```

```
getNearestCarpark: function(carparks) {  
  // 1. calculate and add distance to carparks  
  // 2. sort carparks by distance  
  // 3. take first ten carparks  
}
```

sketch out the way to 03-carpark.html

```
getNearestCarpark: function(carparks) {  
  // 1. calculate and add distance to carparks  
  for (var carpark of carparks) {  
    console.log(carpark)  
  }  
  // 2. sort carparks by distance  
  // 3. take first ten carparks  
}
```

## loops

```
getNearestCarpark: function(carparks) {  
    // 1. calculate and add distance to carparks  
    for (var carpark of carparks) {  
        var distance = distanceFromXY(carpark, this.postcodeX, this.postcodeY)  
        console.log(distance)  
    }  
    // 2. sort carparks by distance  
    // 3. take first ten carparks  
}
```

calculating distance

```
getNearestCarpark: function(carparks) {  
    // 1. calculate and add distance to carparks  
    for (var carpark of carparks) {  
        var distance = distanceFromXY(carpark, this.postcodeX, this.postcodeY)  
        carpark.distance = distance  
    }  
    // 2. sort carparks by distance  
    // 3. take first ten carparks  
}
```

assigning distance into carpark

```
getNearestCarpark: function(carparks) {  
    // 1. calculate and add distance to carparks  
    for (var carpark of carparks) {  
        var distance = distanceFromXY(carpark, this.postcodeX, this.postcodeY)  
        carpark.distance = distance  
    }  
    // 2. sort carparks by distance  
    var sorted_carparks = sortCarparksByDistance(carparks)  
    // 3. take first ten carparks  
}
```

**sort carparks**



```
getNearestCarpark: function(carparks) {  
    // 1. calculate and add distance to carparks  
    for (var carpark of carparks) {  
        var distance = distanceFromXY(carpark, this.postcodeX, this.postcodeY)  
        carpark.distance = distance  
    }  
    // 2. sort carparks by distance  
    var sorted_carparks = sortCarparksByDistance(carparks)  
    // 3. take first ten carparks  
    var nearest_carparks = sorted_carparks.slice(0, 10)  
}
```

slicing arrays

```
getNearestCarpark: function(carparks) {  
    // 1. calculate and add distance to carparks  
    for (var carpark of carparks) {  
        var distance = distanceFromXY(carpark, this.postcodeX, this.postcodeY)  
        carpark.distance = distance  
    }  
    // 2. sort carparks by distance  
    var sorted_carparks = sortCarparksByDistance(carparks)  
    // 3. take first ten carparks  
    var nearest_carparks = sorted_carparks.slice(0, 10)  
    return nearest_carparks  
}
```

return in function

03-carpark.html done

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carparks">
      {{ carpark }}
    </div>
  </div>
</body>
```

remind them what our HTML is showing

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carpark">
      {{ carpark.distance }}<br>
      {{ carpark.address }}<br>
    </div>
  </div>
</body>
```

display specific info only

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carpark">
      Distance: {{ carpark.distance }}m <br>
      Address: {{ carpark.address }}<br>
      <br>
    </div>
  </div>
</body>
```

nicer formatting

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carpark">
      Distance: {{ carpark.distance }}m<br>
      Address: {{ carpark.address }}<br>
      Total Lots: {{ carpark.total_lots }} <br>
      Lots Available: {{ carpark.lots_available }} <br>
      <br>
    </div>
  </div>
</body>
```

additional data we need to add into our carpark object

```
getCarparkList: function() {  
  axios.get("https://data.gov.sg/api/action/datastore_search", {  
    params: {  
      resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
      limit: 2074  
    }  
  }).then(response => {  
    var carparks = response.data.result.records  
    var nearest_carparks = this.getNearestCarparks(carparks)  
    this.carparks = nearest_carparks  
  })  
},
```

where we left off for our logic



```
getCarparkList: function() {  
  axios.get("https://data.gov.sg/api/action/datastore_search", {  
    params: {  
      resource_id: "139a3035-e624-4f56-b63f-89ae28d4ae4c",  
      limit: 2074  
    }  
  }).then(response => {  
    var carparks = response.data.result.records  
    var nearest_carparks = this.getNearestCarparks(carparks)  
    this.getCarparkAvailability(nearest_carparks)  
  })  
},
```

set it up for our next function

```
methods: {  
  searchPostcode: function() {  
    ...  
  },  
  getCarparkList: function() {  
    ...  
  },  
  getNearestCarparks: function(carparks) {  
    ...  
  },  
  getCarparkAvailability: function(carparks) {  
  
  }  
}
```

new method

```
getCarparkAvailability: function(carparks) {  
  axios.get("https://api.data.gov.sg/v1/transport/carpark-availability", {  
    headers: {  
      "api-key": "YOUR_API_KEY"  
    }  
  })  
}
```

api keys, datagovsg APIs

```
getCarparkAvailability: function(carparks) {  
  axios.get("https://api.data.gov.sg/v1/transport/carpark-availability", {  
    headers: {  
      "api-key": "YOUR_API_KEY"  
    }  
  }).then(response => {  
    console.log(response)  
  })  
}
```

explore the response from API

```
getCarparkAvailability: function(carparks) {  
  axios.get("https://api.data.gov.sg/v1/transport/carpark-availability", {  
    headers: {  
      "api-key": "YOUR_API_KEY"  
    }  
  }).then(response => {  
    var carpark_availability = response.data.items[0].carpark_data  
    console.log(carpark_availability)  
  })  
}
```

```
getCarparkAvailability: function(carparks) {  
  axios.get("https://api.data.gov.sg/v1/transport/carpark-availability", {  
    headers: {  
      "api-key": "YOUR_API_KEY"  
    }  
  }).then(response => {  
    var carpark_availability = response.data.items[0].carpark_data  
  
    // Match carpark_number in carpark_availability  
    // with car_park_no in carparks  
    this.carparks = combineCarparkData(carpark_availability, carparks)  
  })  
}
```

high level description of what we want to do

04-carparks.html done