

# Render Map in LWC

How to render Map<String, SObject> in LWC?

LWC for:each can't use Map to show elements, but it's easy to make it happen!

Get the data from the backend:

```
@AuraEnabled(cacheable=true)
public static Map<String, Account> getAccounts() {
    Map<String, Account> accounts = new Map<String, Account>();

    for (Account account : [
        SELECT Id, Name, NumberOfEmployees, (SELECT Id, Name, Email, Phone FROM Contacts)
        FROM Account
    ]) {
        accounts.put(account.Name, account);
    }

    return accounts;
}
```



Then we simply transform the map into an array:




```
@wire(getAccounts, {})
wiredAccounts({ data, error }) {
  if(data) {
    this.data = Object.entries(data).map(([key, value]) => ({
      name: key,
      employees: value.NumberOfEmployees,
      contacts: value.Contacts
    }));
  } else if(error){
    console.error(error)
  }
}
```





Simplified HTML looks like this:

```
<template>
  <template for:each={data} for:item="account">
    <template for:each={account.contacts} for:item="contact">
      <h3 key={account.Name}>{contact.Name}</h3>
    </template>
  </template>
</template>
```



The final solution can look like this:

 <b>An Account</b> Employee#: 2			
	<b>Bob Another Contact</b>	<a href="mailto:bob.contact@anaccount.company">bob.contact@anaccount.company</a>	123456789
	<b>John The Contact</b>	<a href="mailto:john.contact@anaccount.company">john.contact@anaccount.company</a>	123456789

 <b>Another Account</b> Employee#: 3			
	<b>Chris Someone</b>	<a href="mailto:chris@another.company">chris@another.company</a>	123456789
	<b>John Smith</b>	<a href="mailto:john@another.company">john@another.company</a>	123456789
	<b>Martin Uncle</b>	<a href="mailto:uncle@another.company">uncle@another.company</a>	123456789

This can be useful for any kind of nested data, such as List of Lists, Maps, or Wrappers. The principle is to parse data into an Array, which then can be rendered by LWC for:each.