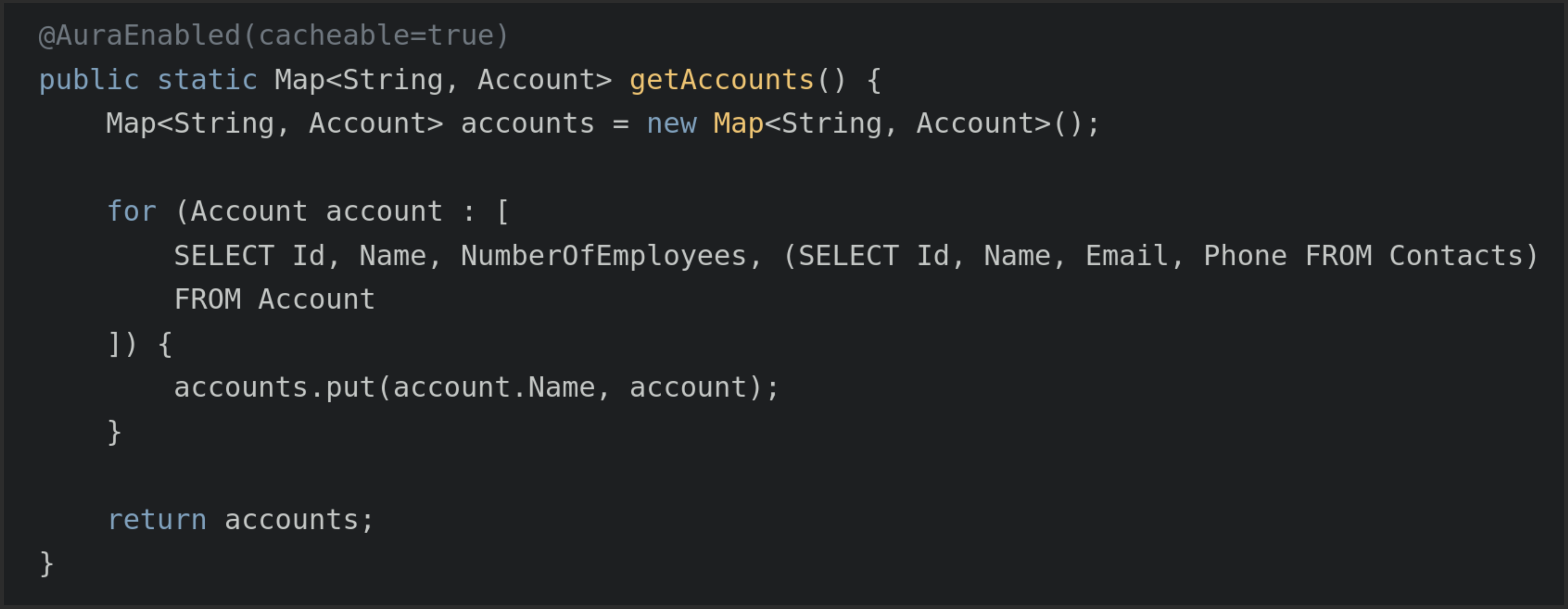
Render Map in LWC

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

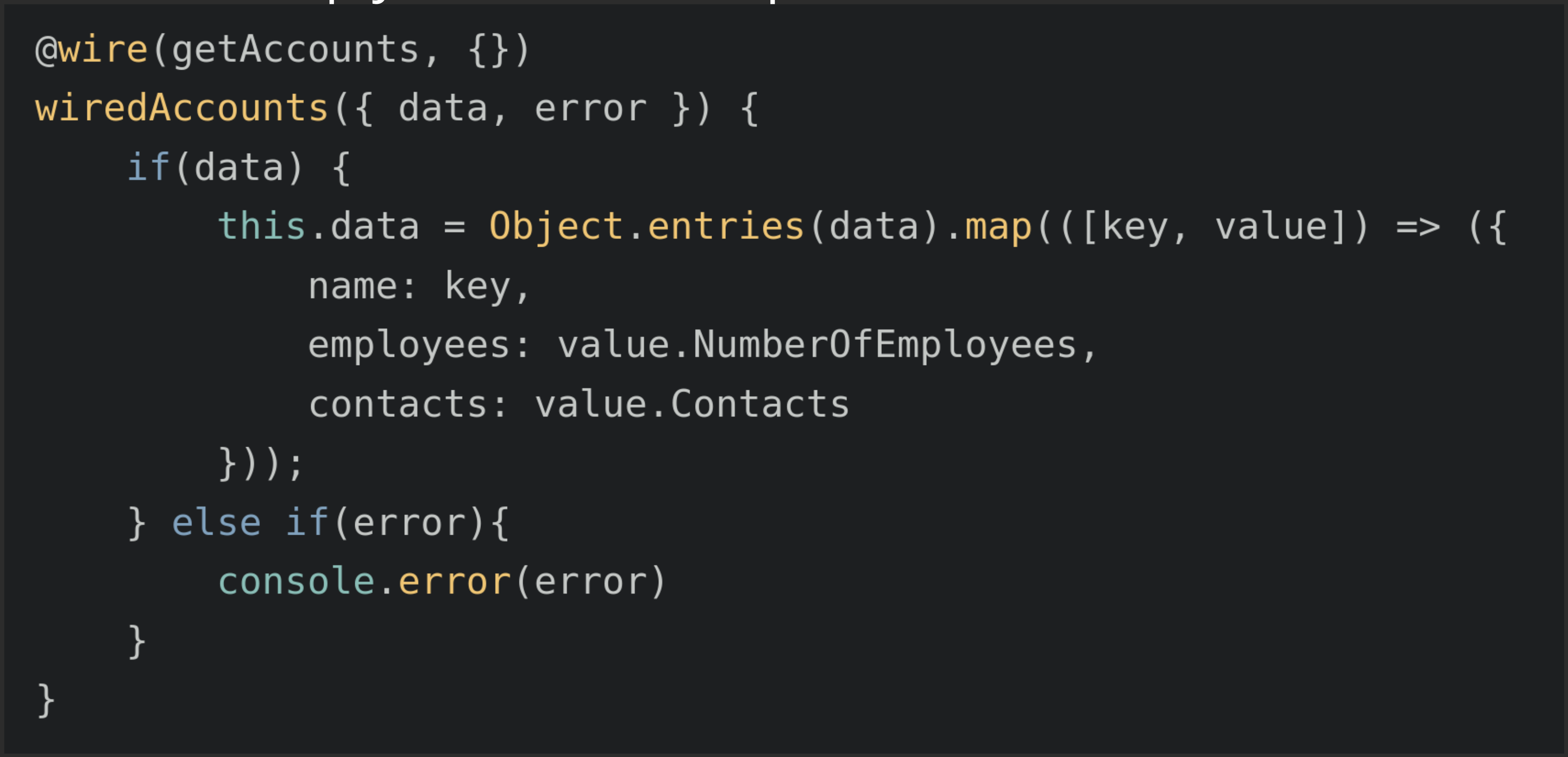
LWC for:each can't use Map to show elements, but it's easy to make it happen!

Get the data from the backend:



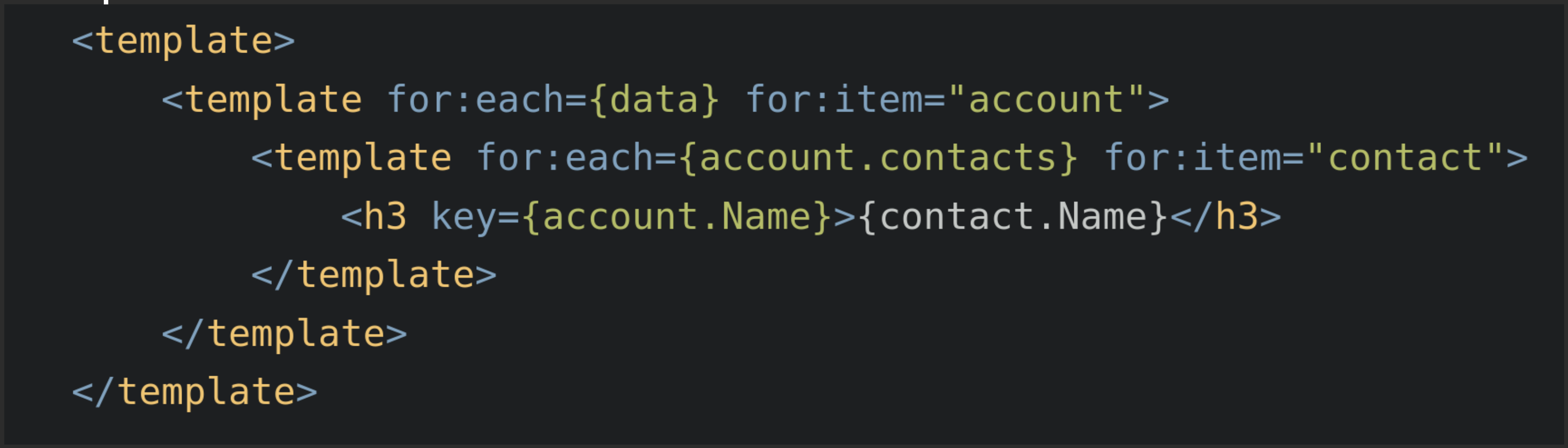
```
@AuraEnabled(cacheable=true)
public static Map<String, Account> getAccounts() {
    Map<String, Account> accounts = new Map<String, Account>();

    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 map into a list:



```
@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)
    }
}
```

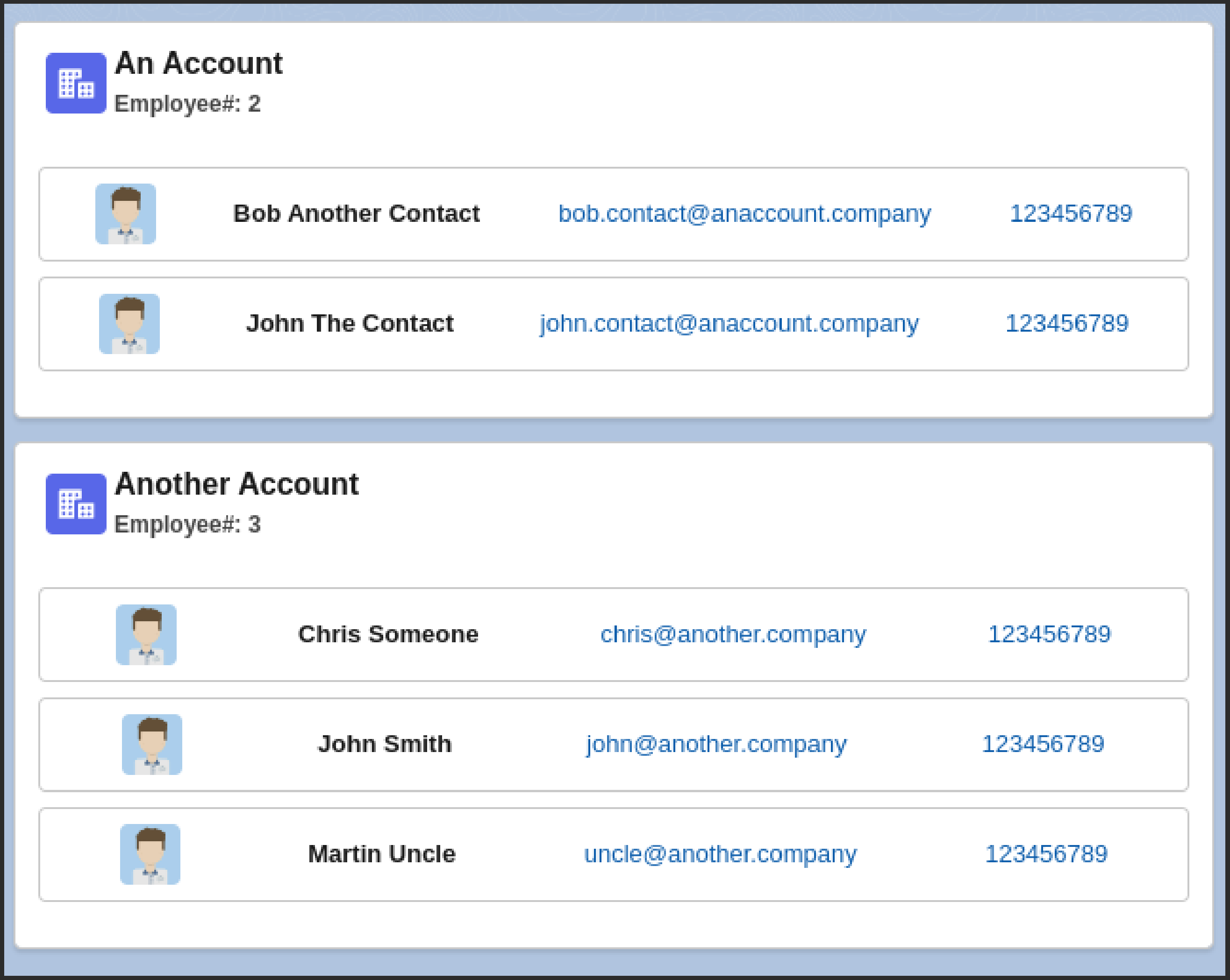
Simplified HTML looks like this:



```
<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>
```

beyond
the cloud

The final solution can look like this:



An Account
Employee#: 2

Bob Another Contact bob.contact@anaccount.company 123456789

John The Contact john.contact@anaccount.company 123456789

Another Account
Employee#: 3

Chris Someone chris@another.company 123456789

John Smith john@another.company 123456789

Martin Uncle uncle@another.company 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 a List which then can be rendered by LWC for:each.

beyond the cloud