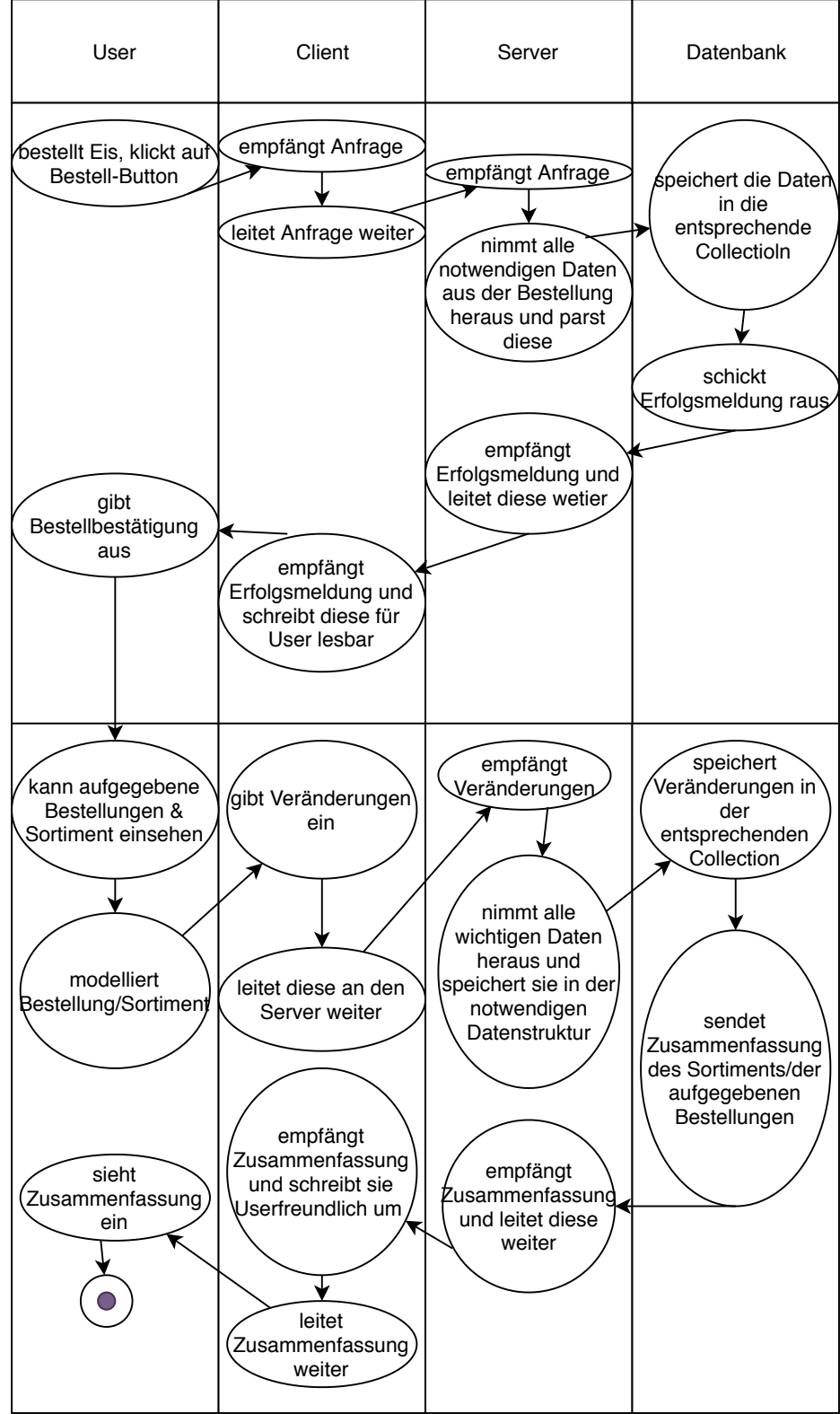


(USER)

Kunde, der eine Bestellung aufgibt:



Händlerseite:

customer.html

<fieldset id="eissorten">

<input type="number...">

<fieldset id="toppings">

<input type="checkbox...">

<fieldset id="saucen">

<input type="radio...">

<fieldset id="wOb">

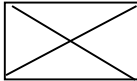
<input type="radio...">

<fieldset id="lieferoptionen">

<input type="text ...">

<button type: submit>

Dein persönlicher Eisdealer
Herzlich Willkommen



Unsere Eissorten

Preis pro Kugel: 1 €

Schwarze-Vanille

...

Toopings?

Preis pro Topping 1 €

☐ Sahne

☐ Streusel

Soßen-Toppings

Soßentoppings gibt's gratis dazu:

☐ Himbeersauce

☐ Schokoladensauce

Waffel oder Becher?

☐ Waffel

☐ Becher

Lieferoptionen

Name, Vorname

...

Eingaben überprüfen

Bestellung aufgeben

admin.html

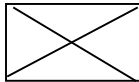
<fieldset id="input">

<input type="text ...">

<button id="insert">

<fieldset id="delete">

Admin Eisdealer



Type new data here

type
name
preis
id
value

insert to sortiment

<fieldset id="delete">

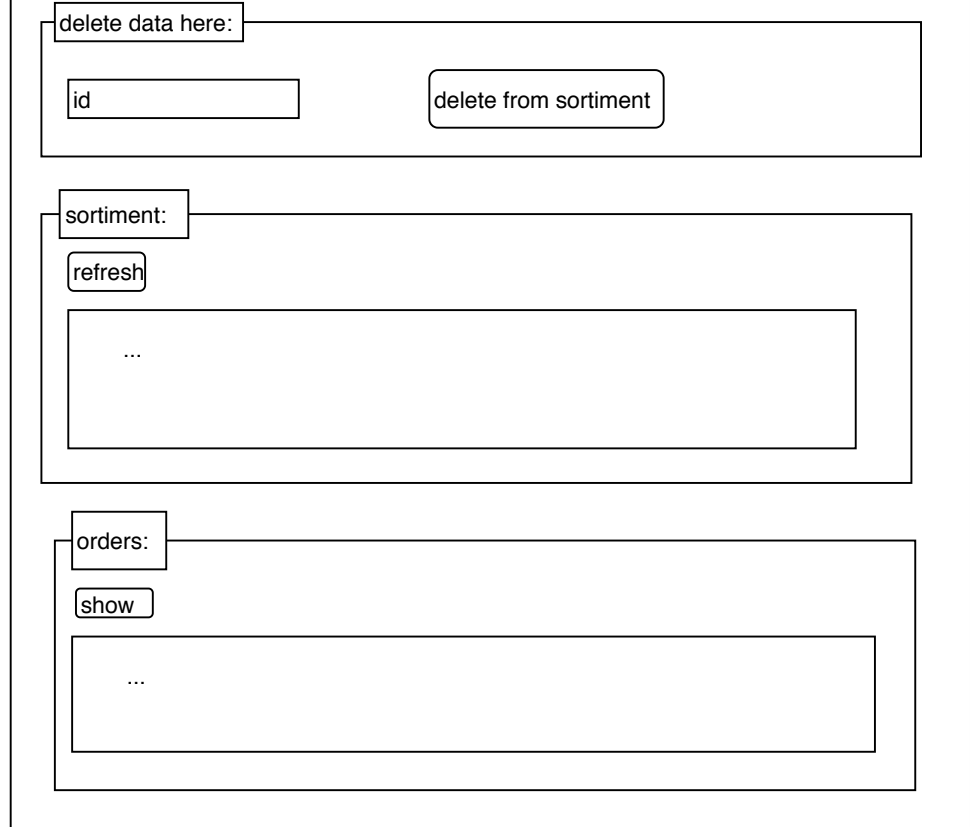

```

</fieldset id="delete">
<input type="text" ..>
<button id="deleteButton">

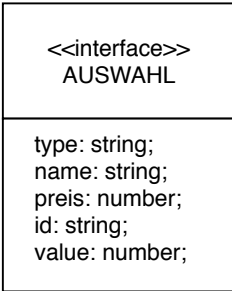
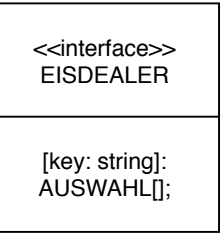
<fieldset id="sortiment">
<textarea>
<button id="refresh">

<fieldset id="orders">
<input type="text" ..>
<button id="show">

```



DATENSTRUKTUR



data.ts
(customer)

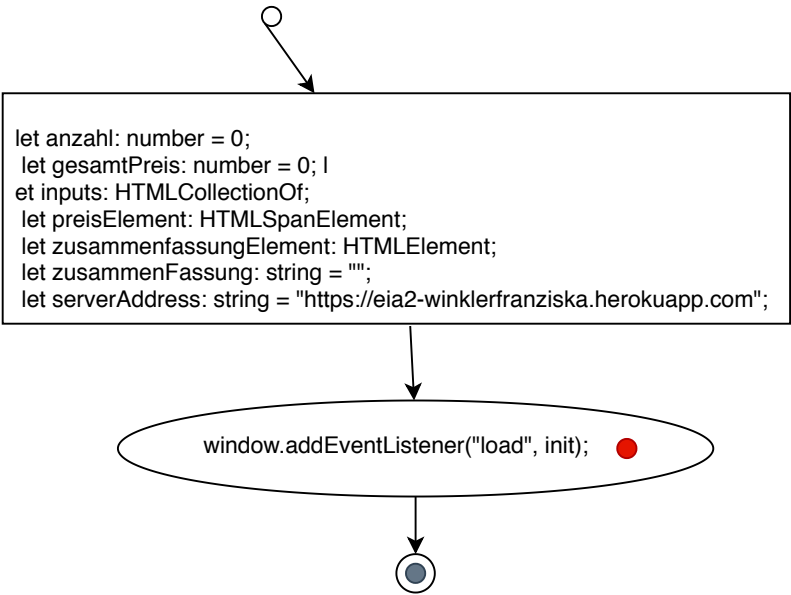
```

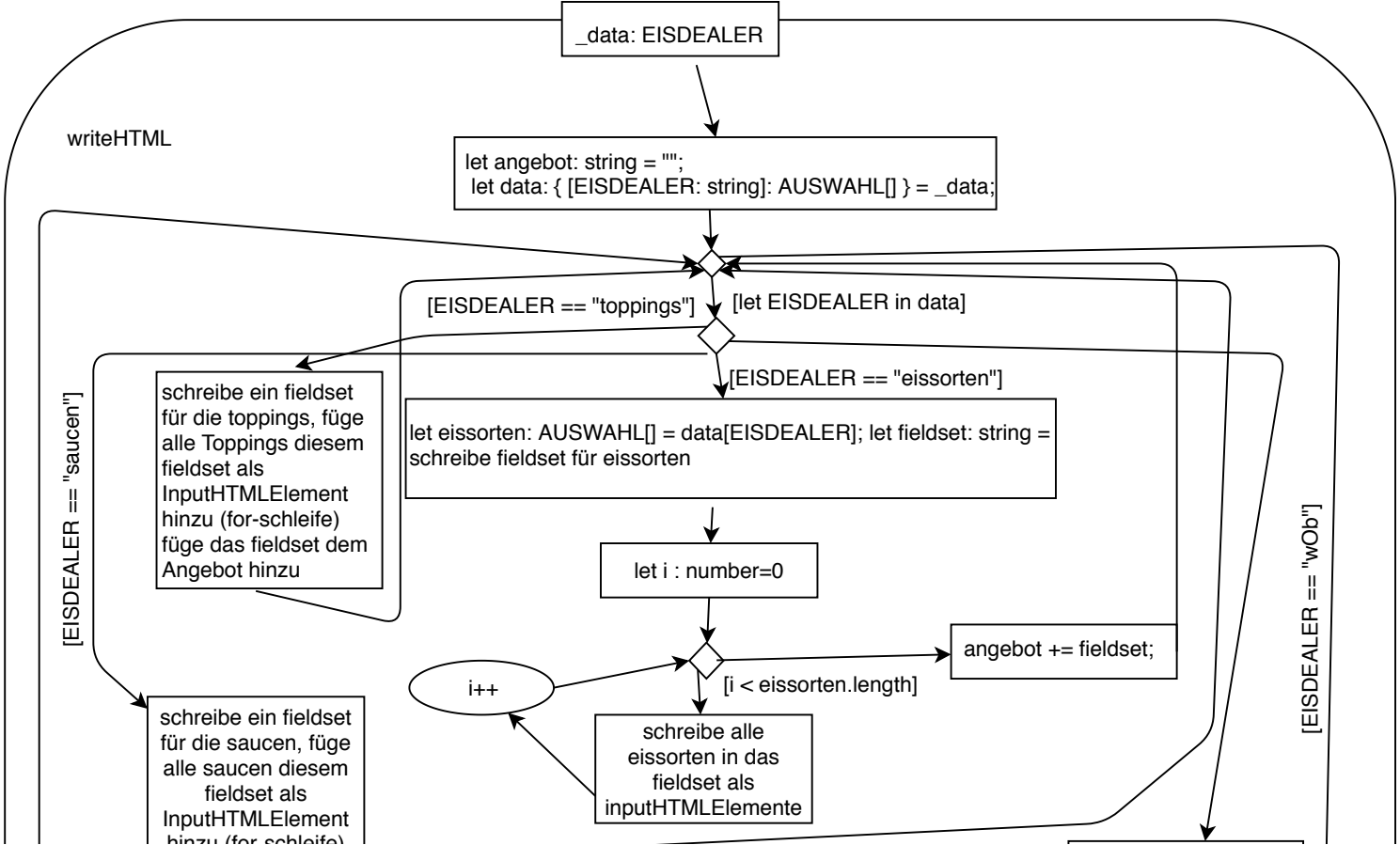
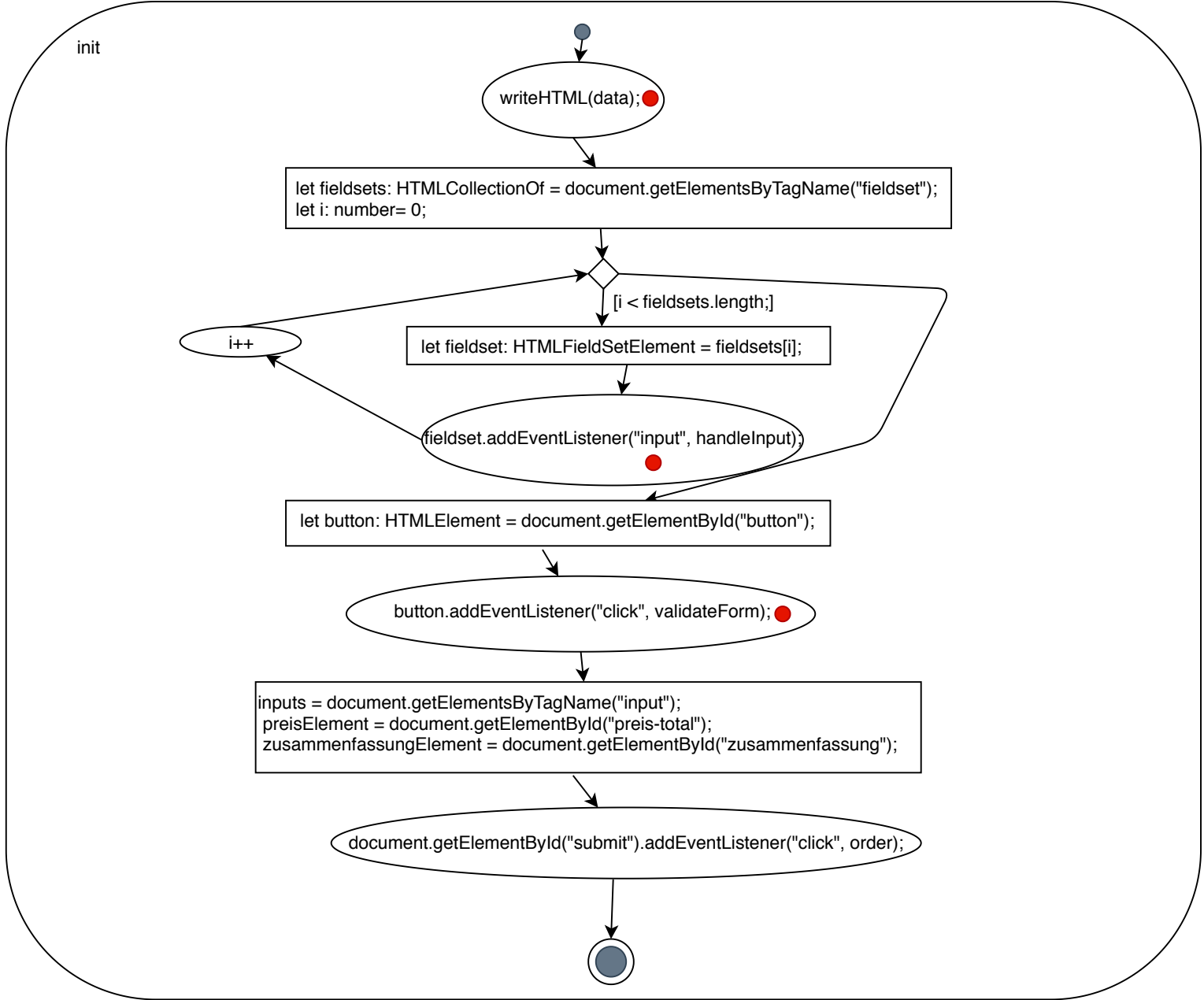
let data: EISDEALER [] = {
"eissorten"=[{type: "number", name:"eissorte", preis: 1, id: "Schwarze-Vanille", value: 0}],
"toppings"=[{}],
"saucen"=[{}],
"wOb"=[{}]}

```

● Gabel

main.ts
(customer)





hinzu (for-schleife)
füge das fieldset dem
Angebot hinzu

schreibe ein fieldset für
die behälter, füge alle
behalter diesem
fieldset als
InputHTML-Element
hinzu (for-schleife) füge
das fieldset dem
Angebot hinzu

document.getElementById("fieldset").innerHTML = offer;

handleInput

_event: Event

gesamtPreis = 0;
zusammenfassung = "";
let i: number=0;

zusammenfassungElement.innerHTML =
zusammenfassung;
preisElement.innerHTML =
String(gesamtPreis.toFixed(2));

[i < inputs.length]

anzahl = 0;
let input: HTMLInputElement = inputs[i];
let preis: number = +input.getAttribute("data-preis");

[input.checked == true || input.type == "number"]

anzahl = Number(input.value);

[input.type == "checkbox" || input.type == "radio"]

anzahl = 1;

gesamtPreis = gesamtPreis + preis * anzahl;

[anzahl != 0 || 0]

zusammenfassung = zusammenfassung + anzahl + " " + input.name + "\n";

i++

validateForm

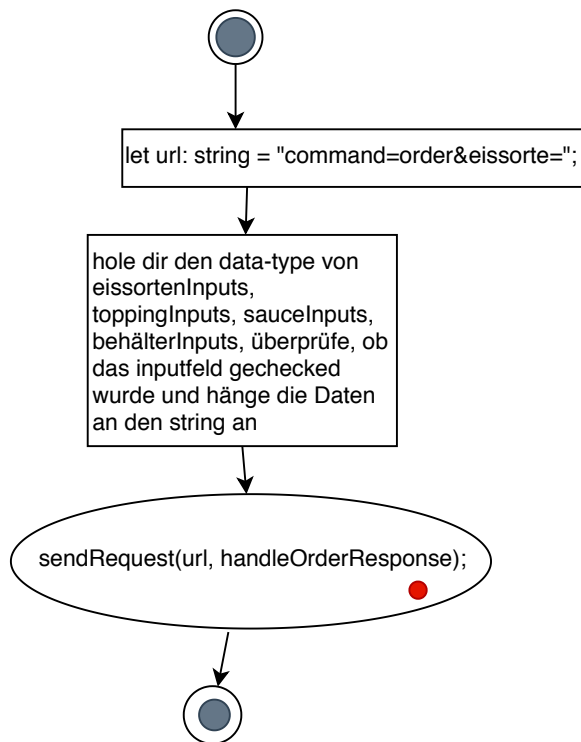
let i: number=0;

[i < inputs.length]

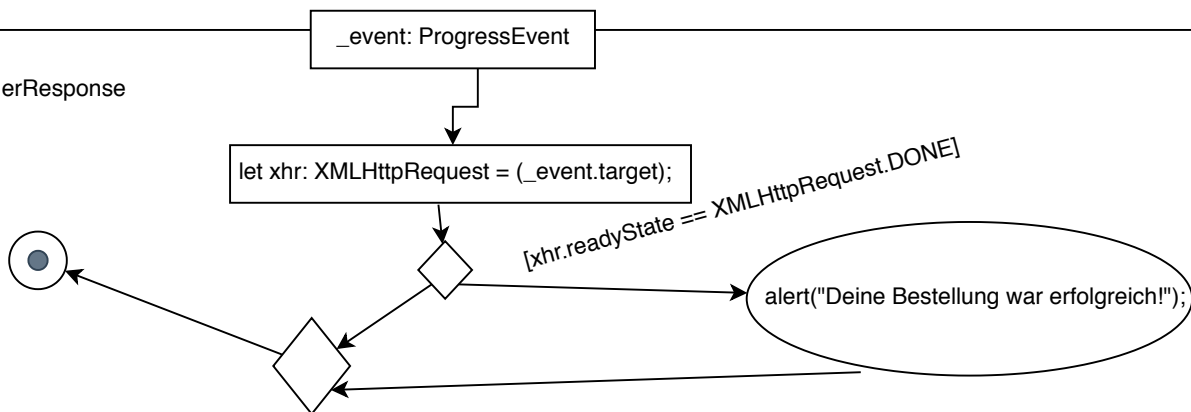
i++

let input: HTMLInputElement = inputs[i];
input.className = "validated";

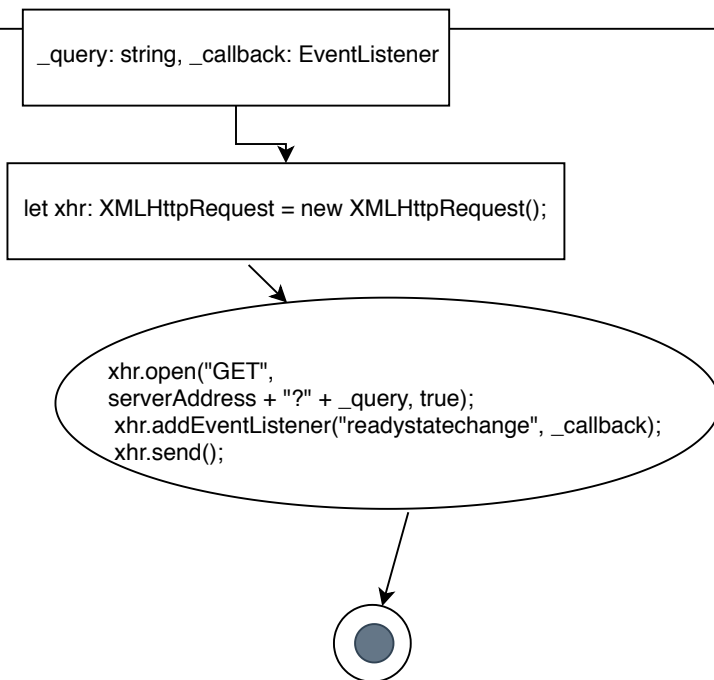
order

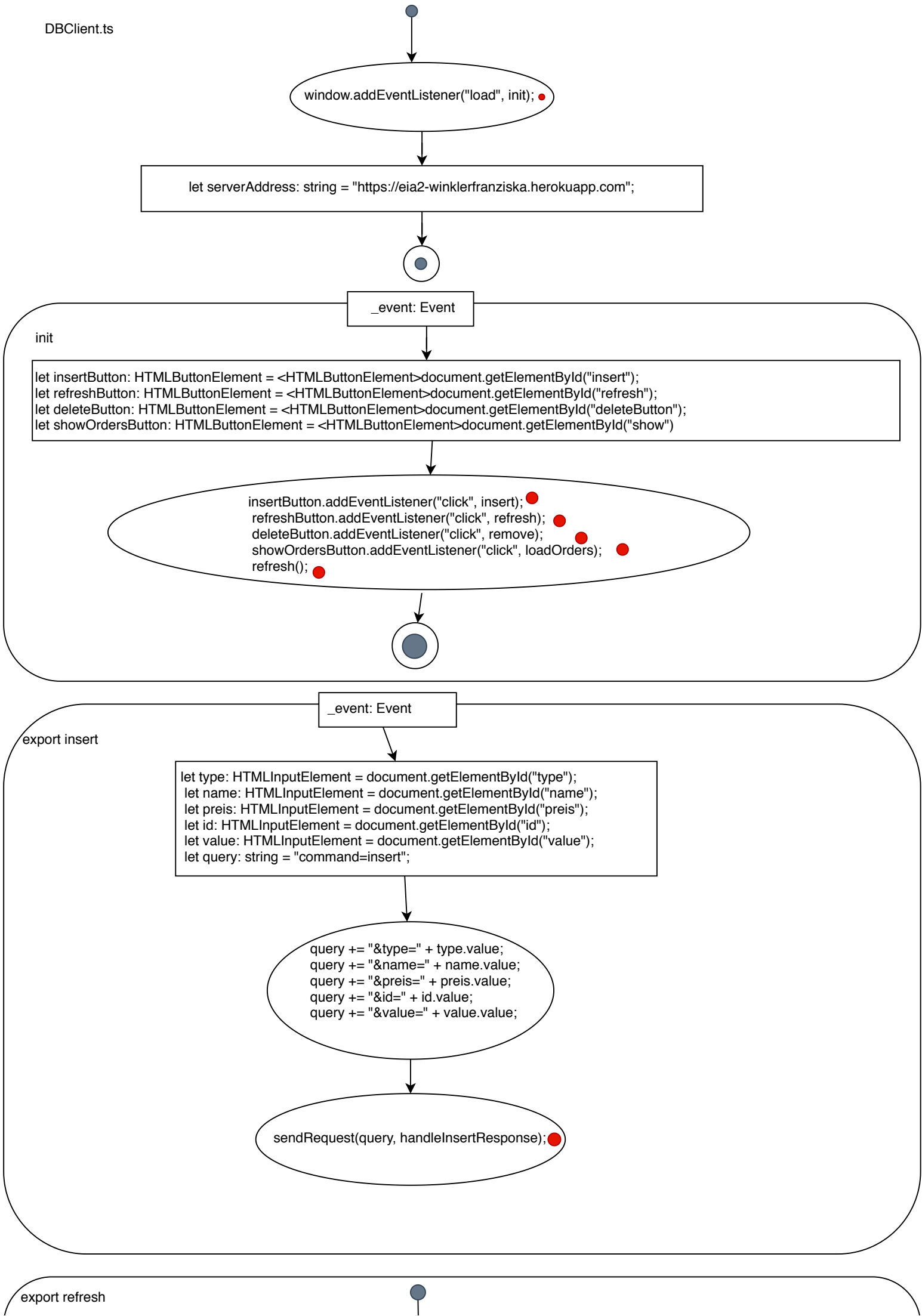


handleOrderResponse



sendRequest





let query: string = "command=refresh";

sendRequest(query, handleFindResponse);



remove

let id: HTMLInputElement = document.getElementById("delete-id");
let type: HTMLInputElement = document.getElementById("delete-type");
let query: string = "command=remove";

query += "&id=" + id.value + "&type=" + type.value;

sendRequest(query, handleRemoveResponse);



loadOrders

let query: string = "command=loadOrder";

sendRequest(query, handleLoadOrderResponse);



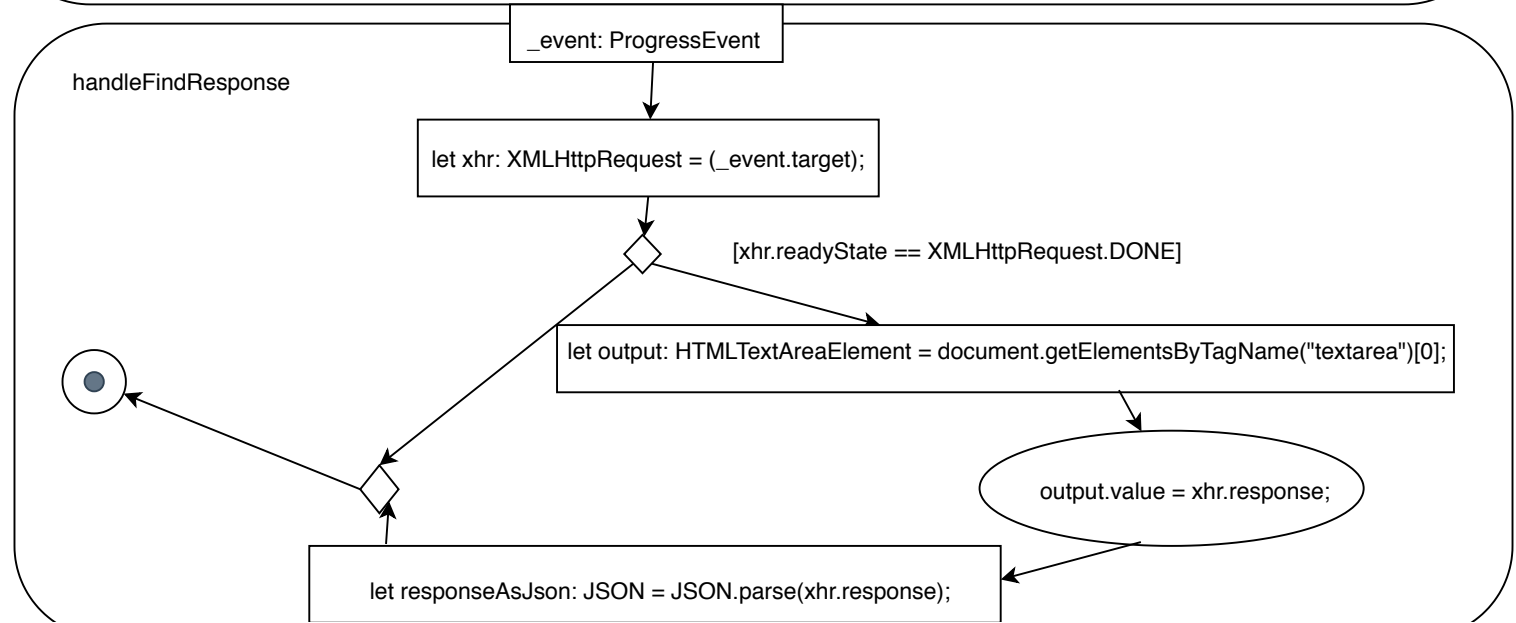
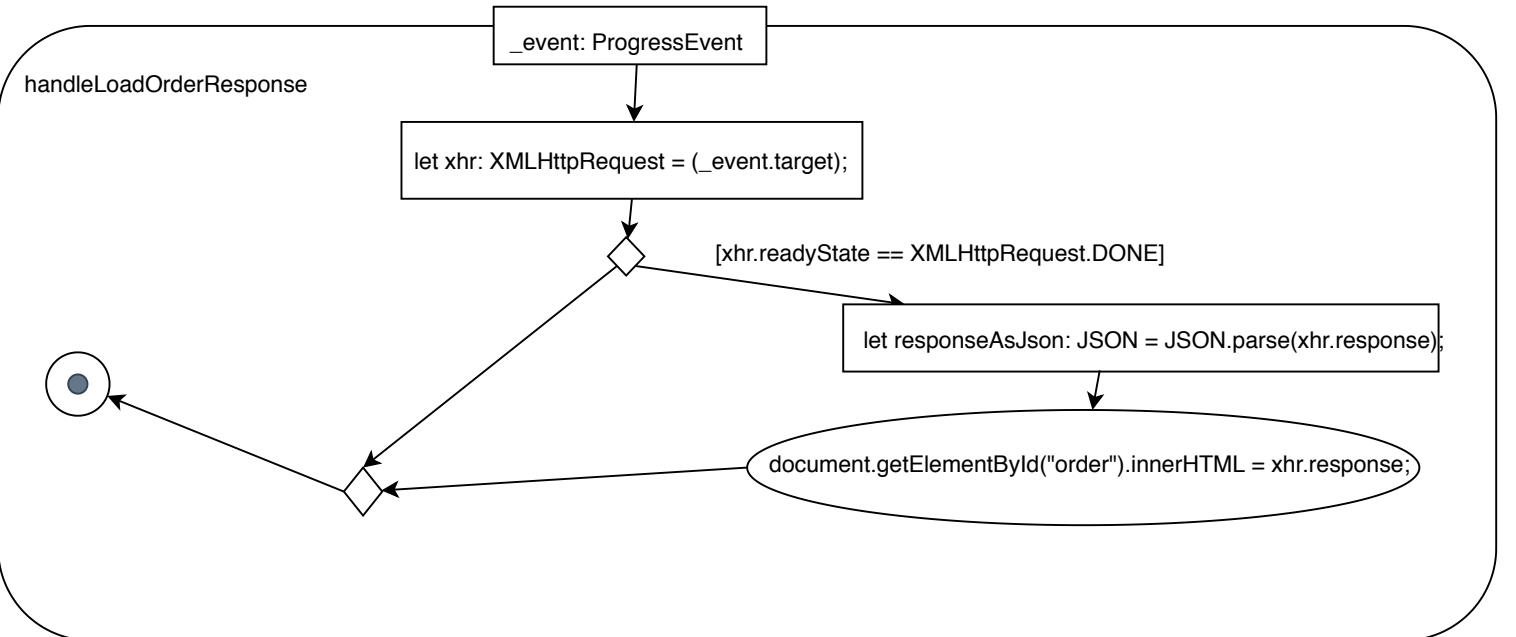
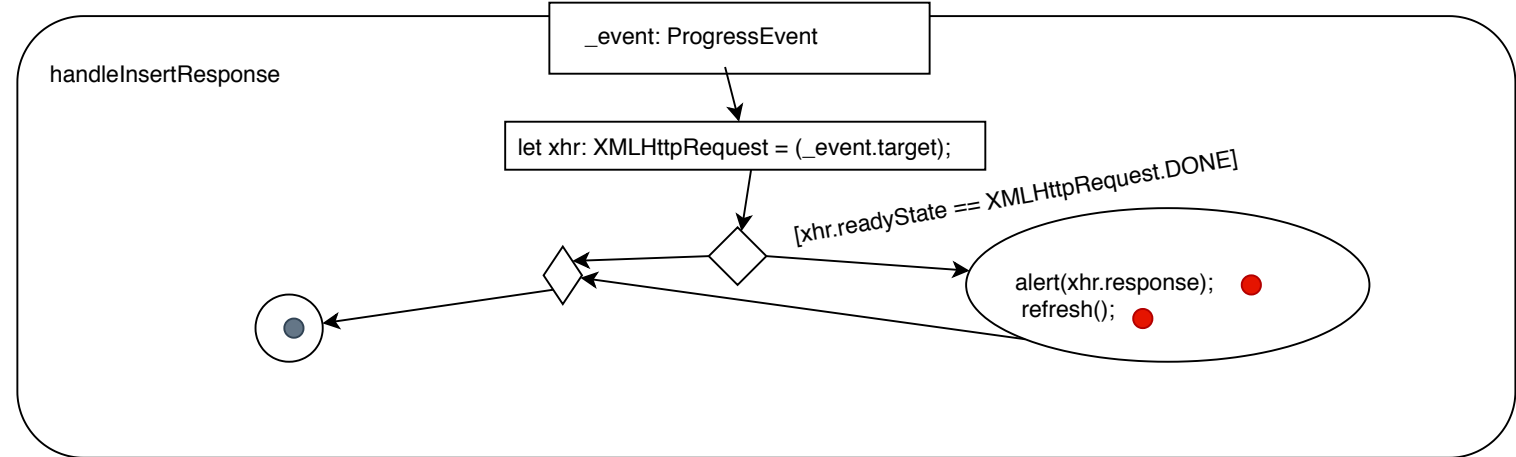
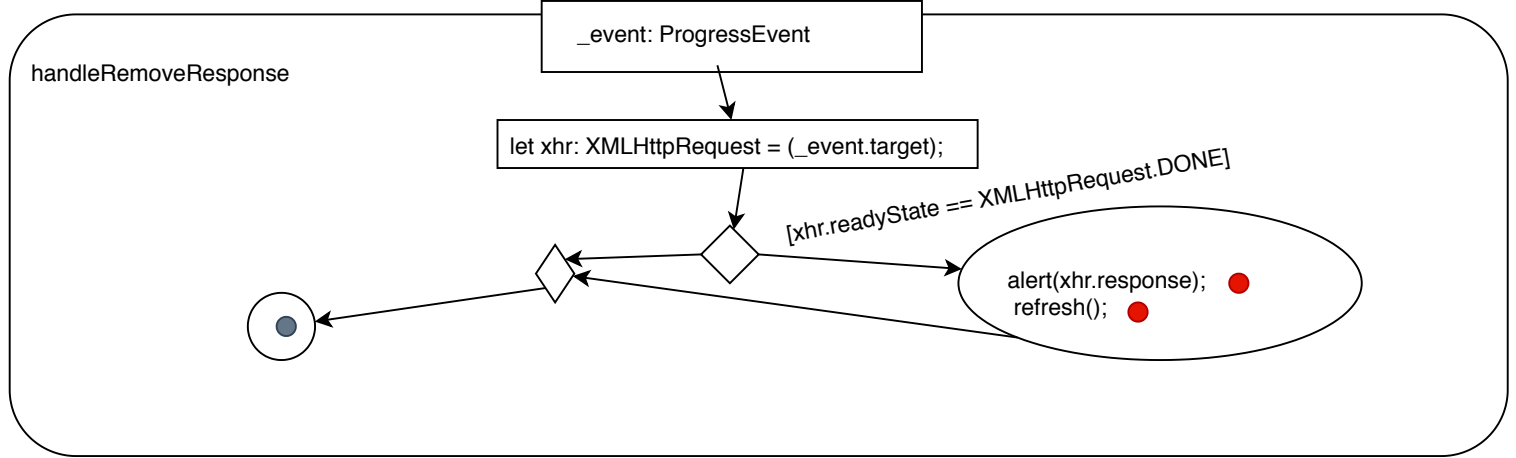
sendRequest

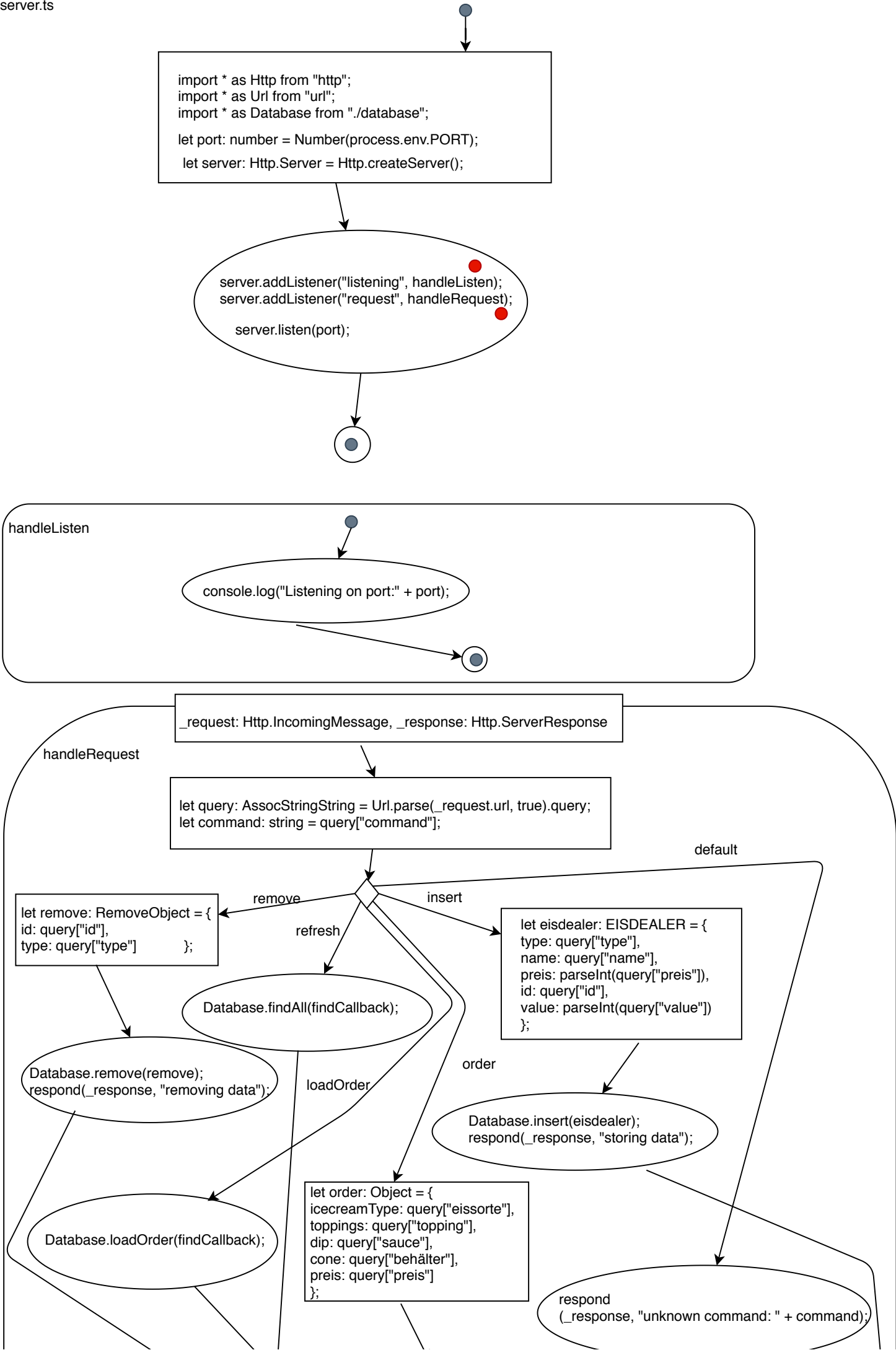
_query: string, _callback: EventListener

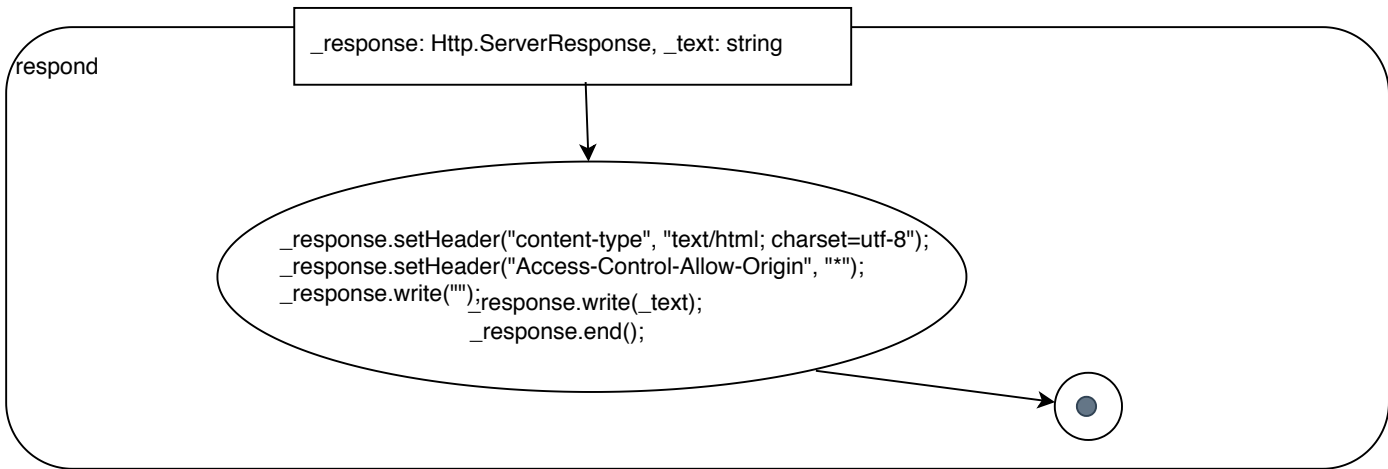
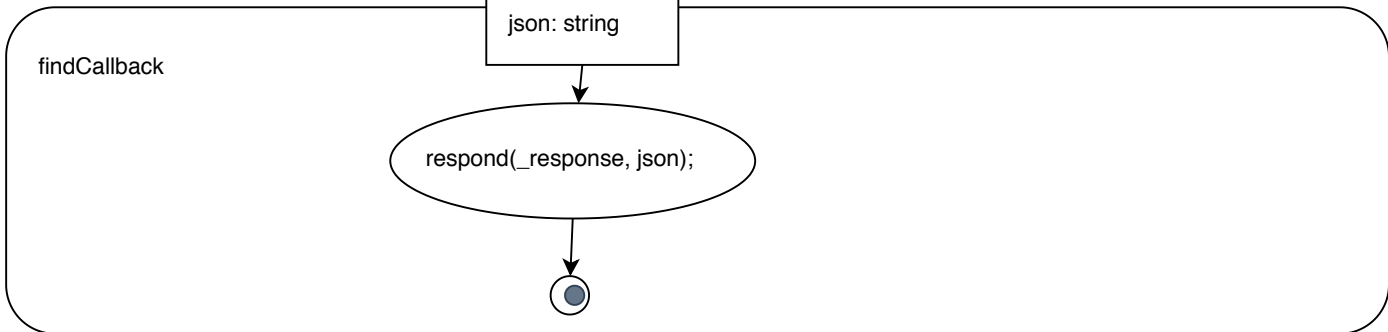
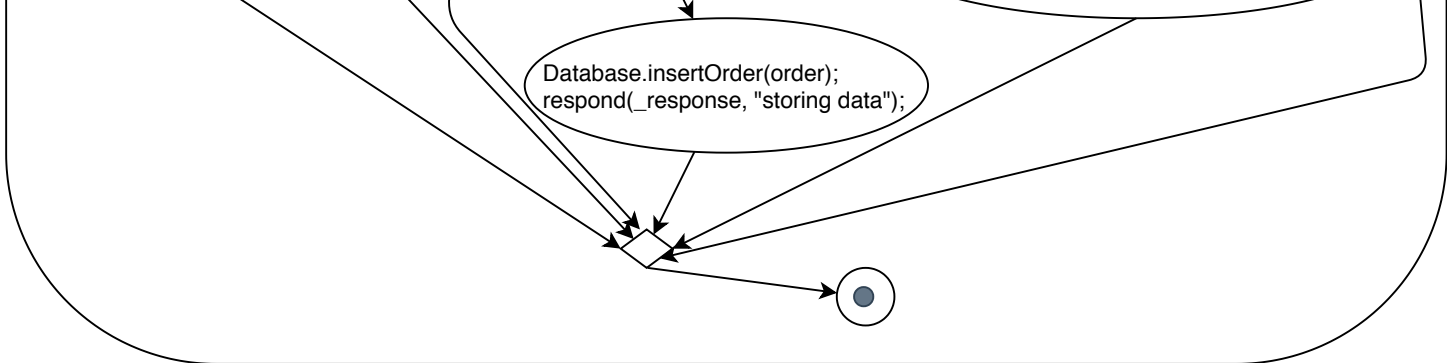
let xhr: XMLHttpRequest = new XMLHttpRequest();

xhr.open("GET", serverAddress + "?" + _query, true);
xhr.addEventListener("readystatechange", _callback);
xhr.send();

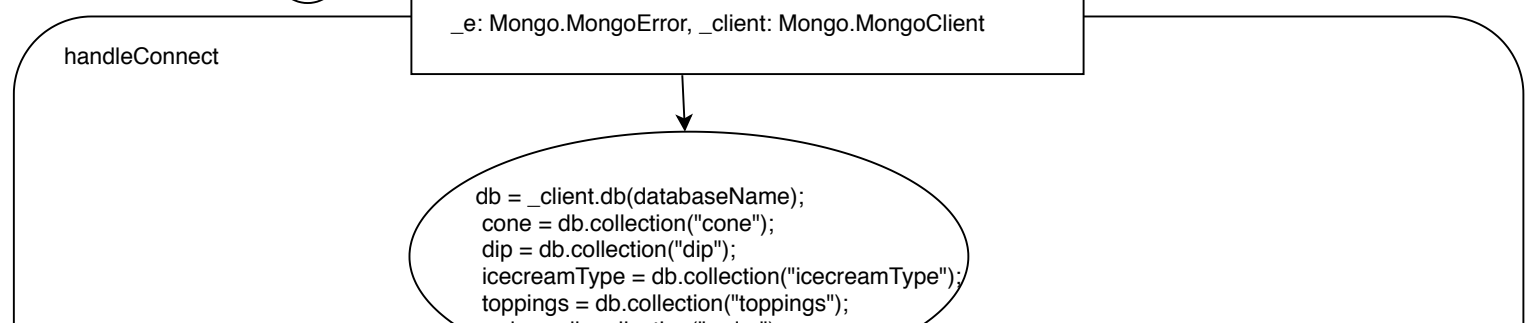
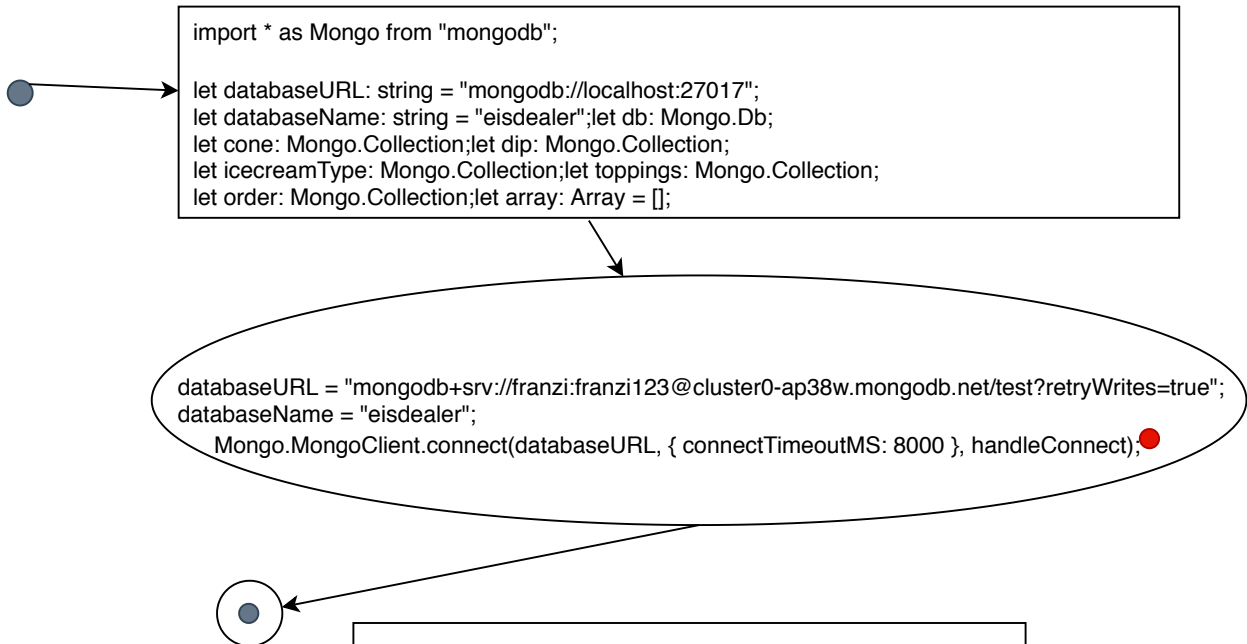








database.ts



order = db.collection("order");



export insert

_doc: EISDEALER

_doc.name

icecreamType

cone

dip

icecreamType.insertOne(_doc, handleInsert);

cone.insertOne(_doc, handleInsert);

dip.insertOne(_doc, handleInsert);

toppings

toppings.insertOne(_doc, handleInsert);



export insertOrder

_doc: Object

order.insertOne(_doc, handleInsert);



export remove

_doc: RemoveObject

```
let deleteQuery = {_id: new Mongo.ObjectId(_doc.id)}  
_doc.type
```

cone

dip

cone.deleteOne(deleteQuery, handleRemove);

dip.deleteOne(deleteQuery, handleRemove);

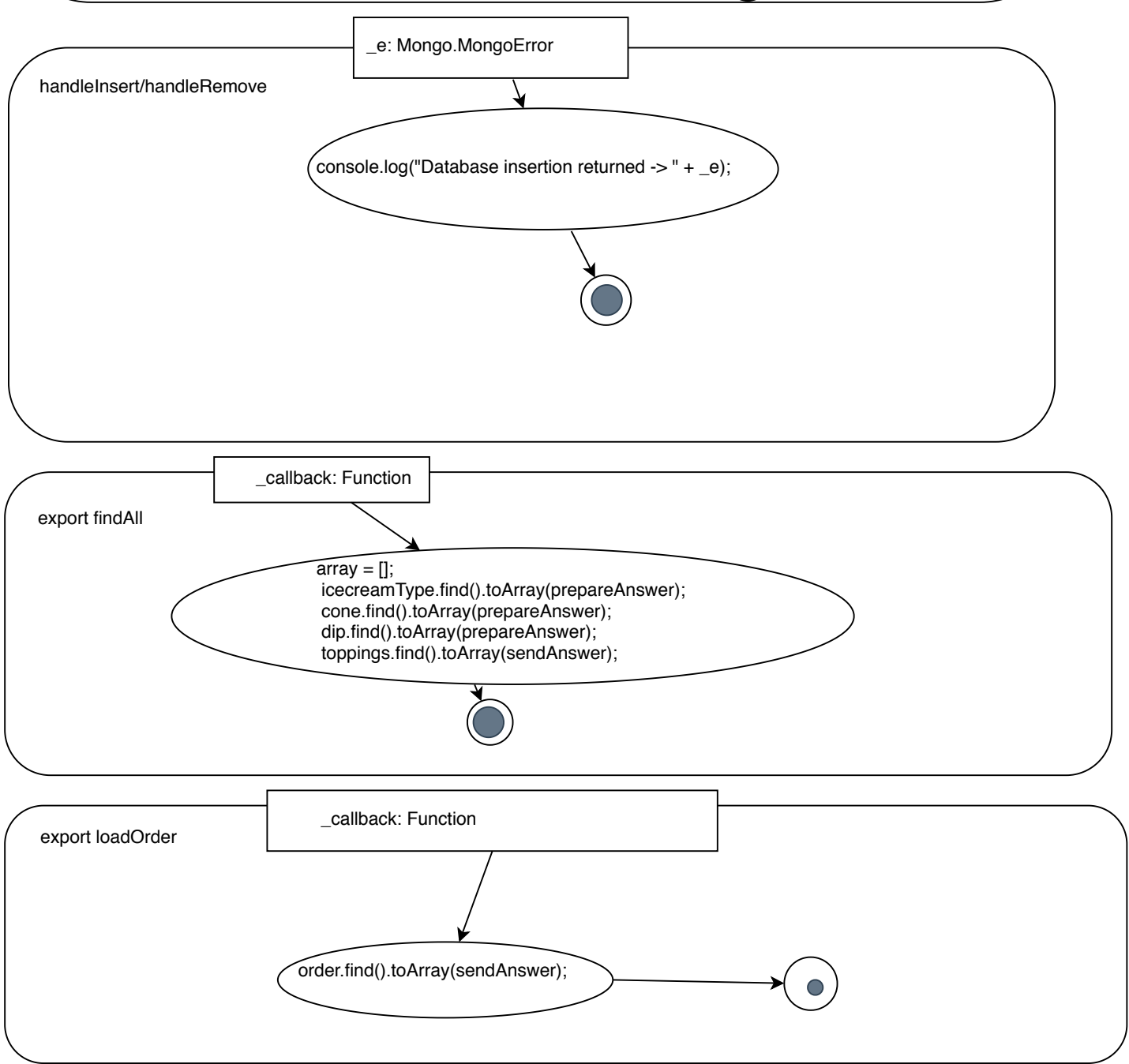
icecreamType

toppings

toppings.deleteOne(deleteQuery, handleRemove);

icecreamType.deleteOne(deleteQuery, handleRemove);





types.ts

```
interface AssocStringString {
  [key: string]: string;}

interface EISDEALER {
  type: string;
  name: string;
  preis: number;
  id: string;
  value: number;
}

interface RemoveObject {
  id: string; type: string;
}
```