Looking for a minimally viable solution where the data is saved to the db in the controller POST action.

For simplicity the Entity Models are Transaction, and ProviderSvcTransactions (1 to many)

I have created custom Models TxnNew and PstNew as container objects, refer to screen clips in pdf attached

In the Blazor UI I populate TxnNew like so:

```
   TxnNew txnNew = new TxnNew();

        txnNew.ProviderId = theProvider.ProviderId;
        txnNew.ClientId = theClient.ClientId;
        txnNew.slId = theClient.SvcLevel;
        txnNew.OwnerLocationId = (int)theClient.OwnerLocationId;

   txnNew.PstsNew = new List<PstNew>();

   foreach (var item in selectedAddOns)
            {
               txnNew.PstsNew.Add(new PstNew
               {
                  BilledRate = item.BillingRate,
                  TransactionId = txnNew.TransactionId,
                  ProviderServiceId = item.RateId
               });
            }
```

And finally:

```
   httpClient.DefaultRequestHeaders.Add("Accept", "application/json");

        JsonSerializerOptions options = new()
        {
           ReferenceHandler = ReferenceHandler.Preserve,
           WriteIndented = true
        };

        var json = JsonSerializer.Serialize(txnNew); //, options

        var content = new StringContent(json);

        string errorMessage = null;
        content.Headers.ContentType = new System.Net.Http.Headers.MediaTypeHeaderValue("application/json");
        var response = await httpClient.PostAsync($"{baseUrl}/api/Txn/AddNew", content);
   var data = await response.Content.ReadAsStringAsync();
```

Note that I have not added ",options" to the serializer

This produces the following json

```
   "ProviderId": 22,
    "ClientId": 49,
```

```
//etc.,
"PstsNew": [
  {
    "Id": 0,
    "ProviderServiceId": 11,
    "TransactionId": 0,
    "BilledRate": 90
  },
  {
    "Id": 0,
    "ProviderServiceId": 4312,
    "TransactionId": 0,
    "BilledRate": 40
  },
```

My code hits the Post action in the controller with the TxnNew object.

In there I have:

```
try
        {
            Transaction txn = new Transaction();
            txn.ProviderId = newTxn.ProviderId;
            txn.ClientId = newTxn.ClientId;
            txn.OwnerLocationId = newTxn.OwnerLocationId;


txn.ProviderSvcTransactions = new List<ProviderSvcTransaction>();
            if (newTxn.PstsNew.Count > 0)
            {
               foreach (var item in newTxn.PstsNew)
               {
                  txn.ProviderSvcTransactions.Add(new ProviderSvcTransaction
                  {
                     BilledRate = item.BilledRate,
                     TransactionId = item.TransactionId,
                     ProviderServiceId = item.ProviderServiceId
                  });
               }
            }

            context.Add(txn);
            await context.SaveChangesAsync();
            return Ok(txn);
```

This produces the desired result as in the data is saved to the db, just like I intended.

Back to the razor page:

```
var response = await httpClient.PostAsync($"{baseUrl}/api/Txn/AddNew", content);
var data = await response.Content.ReadAsStringAsync();
```

Here I now have error 500 for Json Exception and object cycle detected, I guess that is the 1 to many relationship

between Transaction and ProviderSvcTransaction

Ok, so I add options to the json serializer and the json now looks like this:

```
  //all the usual stuff here first
  "PstsNew": {
    "$id": "2",
    "$values": [
      {
        "$id": "3",
        "Id": 0,
        "ProviderServiceId": 11,
        "TransactionId": 0,
        "BilledRate": 90.0000
      },
      {
        "$id": "4",
        "Id": 0,
        "ProviderServiceId": 4312,
        "TransactionId": 0,
        "BilledRate": 40.0000
      },
```

Note the fragments @id: 2 3 and 4, and the curly brackets directly after PstsNew: {

This code does not hit the controller and I get a bad request 400, and nothing is written to the db.

The details of the error message are:
{"type":"https://tools.ietf.org/html/rfc7231#section-6.5.1","title":"One or more validation errors occurred.","status":400,"traceId":"00-e99e67c4d5da7645b6a94b0bca34affd-5e76999d6926874b-00","errors":{"$.PstsNew":["The JSON value could not be converted to System.Collections.Generic.List`1[BtApiEf5.Model.Custom.PstNew]. Path: $.PstsNew | LineNumber: 30 | BytePositionInLine: 14."]}}

What am I doing wrong here?

```
6 references
public class PstNew
{
    0 references
    public int Id { get; set; }
    3 references
    public int ProviderServiceId { get; set; }
    3 references
    public int TransactionId { get; set; }
    3 references
    public decimal BilledRate { get; set; }
}
```

```
4 references
public class TxnNew
{
    5 references
    public int TransactionId { get; set; }
    2 references
    public int ProviderId { get; set; }

    2 references
    public int ClientId { get; set; }
    2 references
    public int OwnerLocationId { get; set; }
    6 references
    public ICollection<PstNew> PstsNew { get; set; }
```