

```

<div class="row p-4 c-controls">
  <div class="col-md-2 ps-3">
    <label>Product category</label>
  </div>
  <div class="col-md-4 pe-4">
    <ComboBox @bind-Value="SearchData.ProductCategoryId" TValue="int"
      TEntity="ProductCategory" TextExpression="t => t.Title" />
  </div>
  <div class="col-md-1 ps-4">
    <label>Status</label>
  </div>
  <div class="col-md-3 pe-3">
    <DropDownList @bind-Value="SearchData.ActiveType" />
  </div>
  <div class="col-md-12">
    <DataGrid @ref="CrudGrid" TEntity="Product" Search="SearchData"
      TableHeight="400" HideInsertIcon>
      <Columns>
        <GridColumn Field="t => t.ProductCategory.Title"
          Title="Product category" />
        <GridColumn Field="t => t.Title" />
        <GridColumn Field="t => t.Price.ToString()">
          <Template>
            <NumericTextBox TValue="int"
              Focused="selectedId == context.Data.Id &&
                col == 1"
              Value="context.Data.Price"
              ValueChanged="value => context.Data.Price = value"
              @onkeyup="async e => await UpdatePrice(e,
                context.Data.Id, context.Data.Price)" />
          </Template>
        </GridColumn>
        <GridColumn Field="t => t.TakeoutPrice.ToString()">
          <Template>
            <NumericTextBox TValue="int"
              Focused="selectedId == context.Data.Id &&
                col == 2"
              Value="context.Data.TakeoutPrice"
              ValueChanged="value => context.Data.TakeoutPrice =
                value"
              @onkeyup="async e => await UpdateTakeoutPrice(e,
                context.Data.Id, context.Data.TakeoutPrice)" />
          </Template>
        </GridColumn>
      </Columns>
    </DataGrid>
  </div>
</div>

```

```

@code
{
    int? selectedId;
    int? col;

    async Task UpdatePrice(KeyboardEventArgs e, int productId, int price)
    {
        if (e.Code == "Enter" || e.Code == "NumpadEnter")
        {
            using var scope = ServiceScopeFactory.CreateScope();
            var productService = new ProductService(scope.ServiceProvider);
            await productService.UpdatePrice(productId, price);
            await productService.SaveChangesAsync();
            ///After the price is updated, Take out price should be focused
            selectedId = productId;

            col = 2;
        }
    }

    async Task UpdateTakeoutPrice(KeyboardEventArgs e, int productId, int
        takeoutPrice)
    {
        if (e.Code == "Enter" || e.Code == "NumpadEnter")
        {
            using var scope = ServiceScopeFactory.CreateScope();
            var productService = new ProductService(scope.ServiceProvider);
            await productService.UpdateTakeoutPrice(productId, takeoutPrice);
            await productService.SaveChangesAsync();
            //Select next row for focus
            await CrudGrid.SelectNextRow();
            if (CrudGrid.SelectedRowId.HasValue)
            {
                //Selected Row id seted and price focused
                selectedId = CrudGrid.SelectedRowId.Value;

                col = 1;
            }
        }
    }
}

```