

```

<div class="c-controls c-header row m-3 align-items-center">
  <div class="col-md-2 ps-2">
    <label>Product Category</label>
  </div>
  <div class="col-md-7">
    <ComboBox TValue="int?" TEntity="ProductCategory"
      @bind-Value="productCategoryId" TextExpression="t => t.Title"
      OnValueChanged="() => CrudGrid.EnableLoading()" />
  </div></div>
<div class="c-body ms-3 me-3">
  @if (productCategoryId.HasValue)
  {
    <DataGrid @ref="CrudGrid" TEntity="Product" ConditionExpr="@t =>
      t.ProductCategoryId == productCategoryId.Value)" Search="SearchData"
      TableHeight="350">
      <Columns>
        <GridColumn Field="t => t.Title" />
        <GridColumn Field="t => t.ActiveType.FaText()" />
        <GridColumn Field="t => t.Price.ToString()" />
        <GridColumn Field="t => t.TakeoutPrice.ToString()" />
        <GridColumn CommandColumns/>
      </Columns>
    </DataGrid>
    <Window @ref="UpsertWindow" Title="Product" Style="width:750px;">
      <Content>
        <CaspianForm @ref="UpsertForm" Model="UpsertData"
          OnSubmit="SetCategoryId">
          <CaspianValidationValidator
            ValidatorType="typeof(ProductService)" />
          <div class="row c-controls ps-3 pe-3 align-items-self-start">
            <div class="col-md-7">
              <div class="row">
                <div class="col-md-4">
                  <label>Title</label>
                </div>
                <div class="col-md-8">
                  <StringTextBox @bind-Value="UpsertData.Title" />
                </div>
                <div class="col-md-4">
                  <label>Price</label>
                </div>
                <div class="col-md-8">
                  <NumericTextBox @bind-Value="UpsertData.Price"
                    OnChange="SetCategoryId" />
                </div>
                <div class="col-md-4">
                  <label>Take out price</label>
                </div>
                <div class="col-md-8">
                  <NumericTextBox @bind-
                    Value="UpsertData.TakeoutPrice" />
                </div>
                <div class="col-md-4">
                  <label>Meal</label>
                </div>
                <div class="col-md-8">
                  <MultiSelectBox @bind-Value="UpsertData.Meal" />
                </div>
              </div>
            </div>
          </div>
        </CaspianForm>
      </Content>
    </Window>
  }

```

```

        <div class="col-md-4">
            <label>Discountable</label>
        </div>
        <div class="col-md-8">
            <CheckBox @bind-Value="UpsertData.Discountable"
                />
        </div>
        <div class="col-md-4">
            <label>Status</label>
        </div>
        <div class="col-md-8">
            <DropDownList @bind-
                Value="UpsertData.ActiveType" />
        </div>
        <div class="col-md-12">
            <WindowCommandButtons />
        </div>
    </div>
</div>
<div class="col-md-5 ps-4">
    <div style="text-align:-webkit-right" class="col-md-12">
        <Fileupload OnChange="async (e) => await
            SetImage(e)">
            <PictureBox Data="UpsertData.Image"
                style="width: 250px; height: 250px;" />
        </Fileupload>
    </div>
    <div class="col-md-12"></div>
</div>
</div>
</CaspianForm>
</Content>
</Window>
}
</div>

```

@code

```

{
    int? productId;

    async Task SetImage(InputFileChangeEventArgs e)
    {
        UpsertData.Image = await e.File.GetByteArrayAsync();
    }

    void SetCategoryId()
    {
        UpsertData.ProductCategoryId = productId.Value;
    }
}

```

```
protected override async Task<bool> BeforUpsertAsync(Product data)
{
    if (data.Price == 0)
        return await Confirm("Do you want to register the product with zero amount?");
    return await base.BeforUpsertAsync(data);
}
```