

WebCheat sheet

1/ Creation of services instances :

program.cs (Web):

```
builder.Services.AddScoped<IServiceEntity,ServiceEntity>();
```

2/ Create controller EntitynameController

3/ Inject services in controller :

IServiceEntity serviceEntity; , generate constructor

4/ In action name , right click , generate view, razor view,
select the right view template (list for index create for create)
select the right entity name

5/Add view



Controller example with all crud + Views :

1-Create

```
// POST: CompteController/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(Entity e)
{
    try {
        serviceCompte.Add(c);
        serviceCompte.Commit();
        return RedirectToAction(nameof(Index));
    } catch { return View(); } }
```

!!! if u have a Dynamic list in the create:

2-Create with static enum or dynamic List:

// Static enum :

***In create View :**

****Change the simple input with :**

```
<select asp-for="Type"
class="form-control"asp-items="Html.GetEnumSelectList<App
licationCore.Domain.enumName> ()"></select>
```

enumName: enum in database

// Dynamic List :

Step1:Config in controller (get create) : creation of view bag

```
// GET: CompteController/Create
public ActionResult Create()
{
```

```
ViewBag.viewbagname = new SelectList(serviceEntity.GetAll(),
"primary-key-Entity", "display-prop");
return View();
}
```

Step2 :

1/generate view

2/ change simple input to :

```
<select asp-for="entityFK" class="form-control" asp-items="
ViewBag.viewbagname"></select>
```

3- Search by : date, type ,enum :

// Search by int /String :

***In index View Add:**

```
<form asp-action="index">
  <fieldset>
    <legend>Recherche </legend>
    <input type="int ou string" name="name"/>
    <input type="submit" value="Serach" />
  </fieldset>
</form>
```

***Get Controller:**

if Search by int:

```
// GET: PlaneController
public ActionResult Index(int? name)
{
    if(name==null)
        return View(serviceEntity.GetAll());
    else
        return View(serviceEntity.GetMany(p=>p.entity-filed >
            name));
}
```

or if **Search by String** :

```
// GET: PlaneController
public ActionResult Index(string? name)
{
    if(name==null)
        return View(serviceEntity.GetAll());
    else
        return View(serviceEntity.GetMany(p=>p.entity-filed .Equals(name)));
}
```

Search by date

***In index View Add:**

```
<form asp-action="index">
    <fieldset>
        <legend>Recherche</legend>
        Saisir une date <input type="date" name="name" />
        <input type="submit" value="Search" />
    </fieldset>
</form>
```

***Get Controller:**

```
// GET: FlightController
public ActionResult Index(DateTime? name)
{
    if (name == null)
        return View(serviceEntity.GetAll());
    else
```

```
return View(serviceEntity.GetMany(f =>
f.propDate.Date.Equals(date))); }
```

Search by enum

*In index View Add:

```
<form asp-action="index">
<fieldset>
  <legend>Recherche</legend>
  Saisir un type:<select
asp-items="Html.GetEnumSelectList<ApplicationCore.Domain.
enumname>()" name="type" >
</select>
<input type="submit" value="Search" />
</fieldset>
</form>
```

*Get Controller:

```
public ActionResult Index(enumname? type)
{
  if(type==null)
    return View(serviceEntity.GetAll());
  return View(serviceEntity.GetMany(p =>
p.Type.Equals(type))); }
```

4-Get Index Simple :

```
// GET: CompteController
ActionResult Index() {
    return View(serviceEntity.GetAll());
}
```

5-Get Index with Order:

```
// GET: CompteController
ActionResult Index() {
    return View(serviceEntity.GetAll().OrderBy(f=>f.orderprop));
```

```
//OR OrderByDescending
}
```

6-Details :

```
// GET: CompteController/Details/5
public ActionResult Details(String id)
{
    return View(serviceEntity.GetById(id));
}
```

!!!!!!! in order to test the edit/delete/details, go to the bottom of the index file (action links)and remove the comment on 'id=item.PrimaryKey' then replace PrimaryKey by the actual primary key of the entity as shown below :

```
@Html.ActionLink("Edit", "Edit", new { id=item.EntityPrimarykey }) |  
@Html.ActionLink("Details", "Details", new { id=item.PrimaryKey })  
@Html.ActionLink("Delete", "Delete", new { id=item.PrimaryKey })
```

Redirection:

```
@Html.ActionLink("Prestation", "Index", "Prestation", new { }, null)
```

6-Edit :

```
// GET: Comptecontroller/Edit/5 Get edit  
public ActionResult Edit(String id)  
{  
    ViewBag.BanqueList = new SelectList(serviceEntity.GetAll(),  
    "Code", "Nom"); // only if u have a Dynamic list , otherwise  
    leave it empty  
    return View(serviceEntity.GetByld(id));  
}  
  
// POST: Comptecontroller/Edit/5  
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult Edit(Type of primary key id, Compte c1)  
{  
    try  
    {  
        serviceCompte.Update(c1);  
        serviceCompte.Commit();  
        return RedirectToAction(nameof(Index));  
    } catch{  
        return View();  
    }  
}
```

6-Delete :

```
// GET: CompteController/Delete/5
public ActionResult Delete(int id)
{
    return View(); }

// POST: CompteController/Delete/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Delete(Type of primary key id,
    IFormCollection collection)
{
    try
    {
        serviceCompte.Delete(serviceCompte.GetByld(id));
        serviceCompte.Commit();
        return RedirectToAction(nameof(Index)); }
    catch{
        return View(); }}
```