

Résolution d'incidents et de problèmes sur les applications internes

La consigne:

FRmonitor
Récupérer et afficher dans un tableau la liste des utilisateurs dont la date de connexion >= 2 ans
⇒ L'affichage se faire dans le menu SDPR => Application => Webshop
Pour cela aller dans la base de données DERLVAIO268 => Webshop => aspnet_Membership

CustomerNumber	CustomerName	Brand	Email	CreateDate	LastLoginDate	LastLoginDate	Action
Faire jointure entre la table aspnet_Membership et ReferentialSAP_Customer_User par userid et sélectionner CustomerNumber	Faire jointure entre ReferentialSAP_Customer_User et par CustomerNumber et [ReferentialSAP].[dbo].[Customer] sélectionner CustomerName	Faire jointure entre la table aspnet_Membership et ReferentialSAP_Customer_User par userid et sélectionner IdBCL et IdSDD SD : 21 1 VA : 30 3	aspnet_Membership	aspnet_Membership	aspnet_Membership	aspnet_Membership	

Requetes linq

```
0 références | 0 modification | 0 auteur, 0 modification
public class SDPRUserController : Controller
{
    private static FRModuleEntities db = new FRModuleEntities();

    // GET: SDPRUser
0 références | 0 modification | 0 auteur, 0 modification
    public ActionResult Index()
    {
        List<SDPRUserModelView> model = new List<SDPRUserModelView>();

        DateTime dateTime = DateTime.Now.AddYears(-2);
        List<aspnet_Users> users01 = (from user in db.aspnet_Users
                                     where user.aspnet_Membership.LastLoginDate <= dateTime
                                     select user).Distinct().ToList();

        // List<aspnet_Users> users02 = db.aspnet_Users.Where(user => user.aspnet_Membership.LastLoginDate <= dateTime).Distinct()
```

On renseigne la base de données.

On initialise le model.

On utilise le DISTINCT pour ne pas avoir deux fois les mêmes lignes.

```
        where user.aspnet_Membership.LastLoginDate <= dateTime
        select user).Distinct().ToList();

// List<aspnet_Users> users02 = db.aspnet_Users.Where(user => user.aspnet_Membership.LastLoginDate <= dateTime).Distinct()

List<aspnet_Users> UsersSD = users01.Where(user => user.ReferentialSAP_Customer_User.Any(refs => refs.idBCL == 21)).ToList()
List<aspnet_Users> UsersVA = users01.Where(user => user.ReferentialSAP_Customer_User.Any(refs => refs.idBCL == 30)).ToList()

if (UsersSD.Any())
{
    model.AddRange(UsersSD.Select(user => new SDPRUserModelView
    {
        Brand = "Saunier Duval",
        CreateDate = user.aspnet_Membership.CreateDate,
        CustomerName = "",
        CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
        Email = user.aspnet_Membership.Email,
        LastLoginDate = user.aspnet_Membership.LastLoginDate,
        UserId = user.UserId.ToString()
    }).ToList());
}

if (UsersVA.Any())
{
    model.AddRange(UsersVA.Select(user => new SDPRUserModelView
    {
        Brand = "Saunier Duval",
        CreateDate = user.aspnet_Membership.CreateDate,
        CustomerName = "",
        CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
        Email = user.aspnet_Membership.Email,
        LastLoginDate = user.aspnet_Membership.LastLoginDate,
        UserId = user.UserId.ToString()
    }).ToList());
}

List<aspnet_Users> UsersSD = users01.Where(user => user.ReferentialSAP_Customer_User != null && user.ReferentialSAP_Customer_User.Any(refs => refs.idBCL
List<aspnet_Users> UsersVA = users01.Where(user => user.ReferentialSAP_Customer_User != null && user.ReferentialSAP_Customer_User.Any(refs => refs.idBCL
List<aspnet_Users> UsersNA = users01.Where(user => user.ReferentialSAP_Customer_User == null ).ToList();

if (UsersSD.Any())
{
    model.AddRange(UsersSD.Select(user => new SDPRUserModelView
    {
        Brand = "Saunier Duval",
        CreateDate = user.aspnet_Membership.CreateDate,
        CustomerName = "",
        CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
        Email = user.aspnet_Membership.Email,
        LastLoginDate = user.aspnet_Membership.LastLoginDate,
        UserId = user.UserId.ToString()
    }).ToList());
}

if (UsersVA.Any())
{
    model.AddRange(UsersVA.Select(user => new SDPRUserModelView
    {
        Brand = "Saunier Duval",
        CreateDate = user.aspnet_Membership.CreateDate,
        CustomerName = "",
        CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
        Email = user.aspnet_Membership.Email,
        LastLoginDate = user.aspnet_Membership.LastLoginDate,
        UserId = user.UserId.ToString()
    }).ToList());
}

if (UsersNA.Any())
{
    model.AddRange(UsersNA.Select(user => new SDPRUserModelView
    {
        Brand = "N/A",
        CreateDate = user.aspnet_Membership.CreateDate,
        CustomerName = "",
        CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
        Email = user.aspnet_Membership.Email,
        LastLoginDate = user.aspnet_Membership.LastLoginDate,
        UserId = user.UserId.ToString()
    }).ToList());
}

return View(model);
}
```

```

private static FRModuleEntities db = new FRModuleEntities();

// GET: SDRUser
0 références | 0 modification | 0 auteur, 0 modification
public ActionResult Index( int filter = 25)
{
    List<SDPRUserModelView> model = new List<SDPRUserModelView>();

    DateTime dateTime = DateTime.Now.AddYears(-2);
    List<aspnet_Users> users01 = (from user in db.aspnet_Users
                                where user.LastActivityDate <= dateTime
                                select user).OrderByDescending( o => o.LastActivityDate).Take(filter).ToList();

    // List<aspnet_Users> users02 = db.aspnet_Users.Where(user => user.aspnet_Membership.LastLoginDate <= dateTime).Distinct().ToList();

    List<aspnet_Users> UsersSD = users01.Where(user => user.ReferentialSAP_Customer_User != null && user.ReferentialSAP_Customer_User.Any(refs => refs.
    List<aspnet_Users> UsersVA = users01.Where(user => user.ReferentialSAP_Customer_User != null && user.ReferentialSAP_Customer_User.Any(refs => refs.
    List<aspnet_Users> UsersNA = users01.Where(user => user.ReferentialSAP_Customer_User == null ).ToList();

    if (UsersSD.Any())
    {
        model.AddRange(UsersSD.Select(user => new SDPRUserModelView
        {
            Brand = "Saunier Duval",
            CreateDate = user.aspnet_Membership.CreateDate,

    if (UsersSD.Any())
    {
        model.AddRange(UsersSD.Select(user => new SDPRUserModelView
        {
            Brand = "Saunier Duval",
            CreateDate = user.aspnet_Membership.CreateDate,
            CustomerName = "",
            CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
            Email = user.aspnet_Membership.Email,
            LastLoginDate = user.aspnet_Membership.LastLoginDate,
            UserId = user.UserId.ToString()
        }).ToList());

    if (UsersVA.Any())
    {
        model.AddRange(UsersVA.Select(user => new SDPRUserModelView
        {
            Brand = "Saunier Duval",
            CreateDate = user.aspnet_Membership.CreateDate,
            CustomerName = "",
            CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
            Email = user.aspnet_Membership.Email,
            LastLoginDate = user.aspnet_Membership.LastLoginDate,
            UserId = user.UserId.ToString()
        }).ToList());

    if (UsersNA.Any())
    {
        model.AddRange(UsersNA.Select(user => new SDPRUserModelView
        {
            Brand = "N/A",
            CreateDate = user.aspnet_Membership.CreateDate,
            CustomerName = "",

    if (UsersNA.Any())
    {
        model.AddRange(UsersNA.Select(user => new SDPRUserModelView
        {
            Brand = "N/A",
            CreateDate = user.aspnet_Membership.CreateDate,
            CustomerName = "",
            CustomerNumber = user.ReferentialSAP_Customer_User.FirstOrDefault().CustomerNumber,
            Email = user.aspnet_Membership.Email,
            LastLoginDate = user.aspnet_Membership.LastLoginDate,
            UserId = user.UserId.ToString()
        }).ToList());

    return View(model);
}

```

Dans la page de vue, on fait notre tableau qui renvoie les données stockés dans le fichier de ressources :

```

<div>
    <table class="table">
        <tr>
            <th>
                N°
            </th>
            <th>
                @Resources.CustomerNumber
            </th>
            <th>
                @Resources.CustomerName
            </th>
            <th>
                @Resources.Brand
            </th>
            <th>
                @Resources.Email
            </th>
            <th>
                @Resources.CreateDate
            </th>
            <th>
                @Resources.LastLoginDate
            </th>
        </tr>
    </table>

```

```

@{
    int i = 0;
    {
        foreach (var item in Model)
        {
            <tr>
                <td>
                    @{
                        i++;
                    }
                    @i
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.CustomerNumber)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.CustomerName)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Brand)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Email)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.CreateDate)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.LastLoginDate)
                </td>
            </tr>
        }
    }
}

```