private const string DateFromLastEmailSettingName = "GroupwareTicketHandler\_DateFromLastEmail";

protected virtual DateTime DateFromLastEmail

{

get

{

//Setting Property holen

var session = Session.Current;

var sp = session.GetService<SettingsPersistencyService>();

var ss = session.GetService<dotNetBF.Services.SettingsService>();

DateTime dateTimeValue = ObjectBroker.GetNullValue<DateTime>();

SystemSettingProperty lu = sp.GetSettingProperty(session, DateFromLastEmailSettingName);

if (lu == null)

return dateTimeValue;

if (!ss.Settings.Contains(DateFromLastEmailSettingName))

{

var systemsetting = SystemSetting.QueryExtent(session.Broker, "Property == param1", new QueryParameter("param1", lu)).First;

if (systemsetting != null)

{

dateTimeValue = PropertyService.GetValue(systemsetting) as DateTime? ?? dateTimeValue;

ss.Settings.Add(new Setting(DateFromLastEmailSettingName, dateTimeValue));

}

}

else

dateTimeValue = sp.GetValue(session, lu) as DateTime? ?? dateTimeValue;

return dateTimeValue;

}

set

{

//Setting Property setzen

var session = Session.Current;

var sp = session.GetService<SettingsPersistencyService>();

var ss = session.GetService<dotNetBF.Services.SettingsService>();

DateTime dateTimeValue = value;

if (!ss.Settings.Contains(DateFromLastEmailSettingName))

ss.Settings.Add(new Setting(DateFromLastEmailSettingName, null));

SystemSettingProperty lu = sp.GetOrCreateSettingProperty(session, DateFromLastEmailSettingName);

lu.DataType = dotNetBF.Modules.Services.Properties.PropertyService.DateTimeDataType;

sp.SetValue(session, lu, dateTimeValue);

}

}

