

Полезное

Разные интересные примеры и важные решения

- [Get response from PostAsJsonAsync](#)
- [WARNING: Unable to download from URI 'https://go.microsoft.com/fwlink/?LinkID=627338&clid=0x409' to ''.](#)
- [Installing Docker on Windows 2016 Server](#)
- [Установка RabbitMQ на Ubuntu 20.04](#)
- [Слияние дублей Юридических лиц](#)
- [Processing LINUXHOST Error: Failed to call RPC function 'FclsExists': The user name or password is incorrect.](#)
- [LDAP + Имя и Отчество в поле Имя](#)
- [Удаление у документа существующих атрибутов](#)

Get response from PostAsJsonAsync

```
var httpClient = new HttpClient()

var task = httpClient.PostAsJsonAsync(posturi, model)
    .ContinueWith( x => x.Result.Content.ReadAsAsync<bool>().Result);

// 1. GETTING RESPONSE - NOT ASYNC WAY
task.Wait(); //THIS WILL HOLD THE THREAD AND IT WON'T BE ASYNC ANYMORE!
bool response = task.Result

// 2. GETTING RESPONSE - TASK ASYNC WAY (usually used in < .NET 4.5
task.ContinueWith( x => {
    bool response = x.Result
});

// 3. GETTING RESPONSE - TASK ASYNC WAY (usually used in >= .NET 4.5
bool response = await task;
```

WARNING: Unable to download from URI 'https://go.microsoft.com/fwlink/?LinkId=627338&clid=0x409' to ''.

Well, I had the same problem. Set my Powershell to TLS 1.2 and it worked for me.

To test this :

1. Open Powershell (As Admin)
2. `[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12`
3. Try it again!

Installing Docker on Windows 2016 Server

To install Docker Enterprise Edition on Windows Server 2016 use the following PowerShell commands:

```
Install-Module DockerMsftProvider -Force  
Install-Package Docker -ProviderName DockerMsftProvider -Force
```

Установка RabbitMQ на Ubuntu 20.04

Внешняя ссылка: <https://losst.ru/ustanovka-rabbitmq-v-ubuntu-20-04>

Слияние дублей Юридических лиц

В процессе работы зачастую возникают ситуации с дублированием контрагентов - Юридических лиц. В результате вести аналитику и навести порядок не так то просто. Для решения данного вопроса можно использовать следующий процесс: [Миграция ЮЛ](#)

Processing LINUXHOST Error: Failed to call RPC function 'FclsExists': The user name or password is incorrect.

1) Stop the service:

```
# systemctl stop veeamservice
```

2) Make database backup:

```
# mv /var/lib/veeam/veeam_db.sqlite /var/lib/veeam/veeam_db.sqlite.bak
```

3) Start the service:

```
# systemctl start veeamservice
```

4) Wait a few minutes for the database to rebuild

5) Rescan machine in the protection group

LDAP + Имя и Отчество в поле Имя

При настройках интеграции с LDAP зачастую получается ситуация, что Имя и Отчество пользователя при импорте в ELMA записывается в поле Имя, а поле Отчество остается пустым.

Для решения данной проблемы необходимо в глобальный модуль добавить следующий listener, который будет корректировать данную проблему при добавлении нового пользователя из LDAP:

```
using EleWise.ELMA.ComponentModel;
using EleWise.ELMA.Runtime.NH.Listeners;
using NHibernate.Event;
using EleWise.ELMA.Security.Models;

namespace MyGlobalModule
{
    [Component]
    public class UserListener : EntityEventsListener
    {
        public override bool OnPreInsert(PreInsertEvent @event)
        {
            var user = @event.Entity as IUser;
            if (user != null && !string.IsNullOrEmpty(user.FirstName))
            {
                var gn = user.FirstName.Split(' ');
                if (gn.Length > 1 && string.IsNullOrEmpty(user.MiddleName))
                {
                    user.FirstName = gn[0];
                    user.MiddleName = gn[1];
                }
            }
            return base.OnPreInsert(@event);
        }
    }
}
```

```
public override bool OnPreUpdate(PreUpdateEvent @event)
{
    var user = @event.Entity as IUser;
    if (user != null && !string.IsNullOrEmpty(user.FirstName))
    {
        // Fix обновления из LDAP
        var gn = user.FirstName.Split(' ');
        if (gn.Length > 1)
        {
            user.FirstName = gn[0];
            if (string.IsNullOrEmpty(user.MiddleName))
            user.MiddleName = gn[1];
        }
    }
    return base.OnPreUpdate(@event);
}
}
```

Удаление у документа существующих атрибутов

Для атрибутов документов (объектов), созданных в Дизайнере при их удалении из модули, фактического удаления не происходит. Более того, возможны варианты, когда атрибуты являются системными и их не возможно удалить. На примере признака документа "Контроль отправки" рассмотрим вариант как вернуть все обратно.

Список типов документов | Документ со штрих-кодом

Описание | Атрибуты | Жизненный цикл | Маршруты документа | Дополнительные | Формы (предс...

Название типа документа

Документ со штрих-кодом

Шаблон названия документа

Документ (с вычисляемым штрих-кодом)

Разрешать менять название документа

Добавление версии документа

Загрузка версии из файла

Кнопка сканирования версии

Генерация версии по шаблону

Дополнительные настройки

Контроль отправки

Разрешать регистрировать документы этого типа несколько раз

Ограничить работу с документом через веб-интерфейс

Использовать жизненный цикл для документов этого типа

Описание

Необходимо реализовать процесс с ручным запуском, сделать скрипт со следующим кодом:

```
public virtual void Process(Context context)
{
    // 744 - пример Id нужного нам типа документа из базы
    var mh = MetadataItemHeaderManager.Instance.Load(744);
```

```

var m = mh.Current; //.Published;
var metadata = (DocumentMetadata)m.Metadata;

Logger.Log.ErrorFormat("Start update metadata for {0}", metadata.Name);

metadata.ControlOnSending = false;

var sendStatusProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendStatus);
RemoveSendProperty(sendStatusProperty, metadata);

var sendDateProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendDate);
RemoveSendProperty(sendDateProperty, metadata);

var senderProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendUser);
RemoveSendProperty(senderProperty, metadata);

var sendDocumentTypeProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendDocumentType);
RemoveSendProperty(sendDocumentTypeProperty, metadata);

var sendNumberProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendNumber);
RemoveSendProperty(sendNumberProperty, metadata);

var sendScanCopyProperty = LinqUtils.NameOf<ISendableDocument>(d => d.SendScanCopy);
RemoveSendProperty(sendScanCopyProperty, metadata);

m.Metadata = metadata;
m.Save();
mh.IsDirtyItem = true;
mh.Save();
var mm = new ModelManager();
mm.Publish("Fix control on send", mh.Id);

Logger.Log.ErrorFormat("Metadata for {0} published", metadata.Name);
}

private void RemoveSendProperty(string propertyName, DocumentMetadata metadata)
{
var allProperties = metadata.Properties; //MetadataLoader.GetBaseProperties(metadata);
Logger.Log.ErrorFormat("Props count: {0}", allProperties.Count);
var toRemove = allProperties.Where(p1 => p1.Name == propertyName && p1.IsSystem).ToList();

```

```
foreach (var p in toRemove)
{
    Logger.Log.ErrorFormat("Removed: {0}", p.Name);
    metadata.Properties.Remove(p);
}
}
```

Порядок применения:

- Опубликовать процесс
- Запустить и дождаться его завершения
- Перезапустить сервер ELMA

Обязательно сделайте резервную копию базы данных.