



Взаимодействие с веб-сервисом

Последнее обновление: 17.09.2016



Теперь создадим проект Xamarin Forms по типу Portable Class Library, который назовем MobileClient. Первым делом добавим в проект пакеты System.Web.Http и Newtonsoft.Json. Особенности добавления System.Web.Http были описаны в статье [Класс HttpClient и отправка запросов](#).

После добавления выше перечисленных пакетов определим в главном проекте класс, который будет

представлять основную модель приложения - класс Friend:

```
namespace MobileClient
{
    public class Friend
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }

        public override bool Equals(object obj)
        {
            Friend friend = obj as Friend;
            return this.Id==friend.Id;
        }
    }
}
```

И также добавим в проект вспомогательный класс, через который мы будем взаимодействовать с сервером - класс FriendService:

```
using Newtonsoft.Json;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;

namespace MobileClient
{
    public class FriendsService
    {
```

```
const string Url =
    "http://192.168.1.18:3000/api/friends/";
// настройка клиента
private HttpClient GetClient()
{
    HttpClient client = new HttpClient();
    client.DefaultRequestHeaders.Add("Accept
    "application/json");
    return client;
}

// получаем всех друзей
public async Task<IEnumerable<Friend>> Get()
{
    HttpClient client = GetClient();
    string result = await
client.GetStringAsync(Url);
    return
JsonConvert.DeserializeObject<IEnumerable<Friend>>
(result);
}

// добавляем одного друга
public async Task<Friend> Add(Friend friend)
{
    HttpClient client = GetClient();
    var response = await client.PostAsync(Url
        new StringContent(
            JsonConvert.SerializeObject(frie
            Encoding.UTF8, "application/json

    if (response.StatusCode != HttpStatusCode
        return null;

    return JsonConvert.DeserializeObject<Fri
```

```
        await
response.Content.ReadAsStringAsync());
    }
    // обновляем друга
    public async Task<Friend> Update(Friend frie
    {
        HttpClient client = GetClient();
        var response = await client.PutAsync(Url
"/" + friend.Id,
        new StringContent(
            JsonConvert.SerializeObject(frie
            Encoding.UTF8, "application/json

        if (response.StatusCode != HttpStatusCode
            return null;

        return JsonConvert.DeserializeObject<Fri
            await
response.Content.ReadAsStringAsync());
    }
    // удаляем друга
    public async Task<Friend> Delete(int id)
    {
        HttpClient client = GetClient();
        var response = await client.DeleteAsync(
"/" + id);
        if (response.StatusCode != HttpStatusCode
            return null;

        return JsonConvert.DeserializeObject<Fri
            await
response.Content.ReadAsStringAsync());
    }
}
}
```

Здесь применяется рассмотренный в прошлых темах класс `HttpClient` и его методы. В данном случае предполагается, что веб-сервис запущен по адресу `http://192.168.1.18:3000`, в вашем случае адрес может быть иным.

Первый метод `GetClient()` используется для создания и конфигурации объекта `HttpClient`. Здесь мы можем сразу установить различные заголовки. В данном случае мы устанавливаем заголовок `Accept`, чтобы сервер отдавал нам данные в формате `json`.

Метод `Get` загружает все данные и преобразует их в объекты `Friend` посредством метода `JsonConvert.DeserializeObject()`:

```
public async Task<IEnumerable<Friend>> Get()
{
    HttpClient client = GetClient();
    string result = await
client.GetStringAsync(Url);
    return
JsonConvert.DeserializeObject<IEnumerable<Friend>>
(result);
}
```

Метод `Add` посылает серверу новые данные:

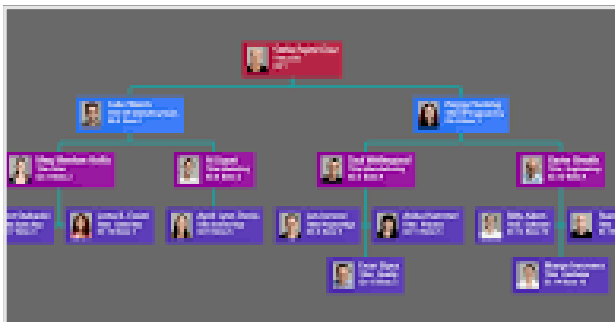
```
public async Task<Friend> Add(Friend friend)
{
    HttpClient client = GetClient();
    var response = await client.PostAsync(Url,
        new StringContent(
```

```
JsonConvert.SerializeObject(friend),  
Encoding.UTF8, "application/json"));  
  
if (response.StatusCode != HttpStatusCode.OK)  
    return null;  
  
return JsonConvert.DeserializeObject<Friend>(  
    await response.Content.ReadAsStringAsync()).  
}
```

Для отправки объект сериализуется с помощью метода `JsonConvert.SerializeObject()` в формат json.

Так как созданный в прошлой теме сервер после добавления возвращает добавленный объект, то соответственно здесь мы его можем получить и вернуть в качестве результата метода. При этом мы ожидаем, что ответ от сервера будет иметь статусный код 200. Если же статусный код другой, например, 40х, то возвращаем null.

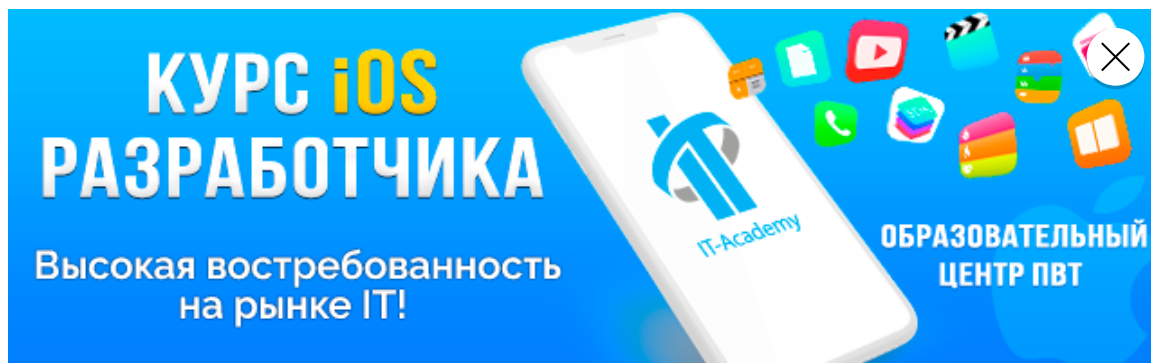
Методы `Update` и `Delete`, предназначенные для обновления и удаления данных, в целом аналогичны методу `Add`.



JS Diagram Lib

gojs.net

[Назад](#) [Содержание](#) [Вперед](#)



Комментарии

Сообщество



Войти



Рекомендовать



Поделиться

Лучшее в начале



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS



Имя



Сергей Лавров • 3 месяца назад

Здравствуйте, спасибо за статью, но я не могу вызвать метод `Get(int Id)`. В примере не показано как его вызывать. Я пытаюсь сделать подобно `Delete`, но все равно вызывается всегда метод `Get()`. Помогите пожалуйста новичку!!! Спасибо

^ | v • Ответить • Поделиться ›



Metanit Модератор ➔ Сергей Лавров
• 3 месяца назад

не совсем понятно, в статье нет метода `Get(int Id)`, есть только метод `Get()`

^ | v • Ответить • Поделиться ›



Denis Lohachov • 6 месяцев назад

имеется метод

```
public async Task<IEnumerable<recipientlist>>
GetAllList()
{
```

```
    HttpClient _client = GetClient();
    string _url = Url + "getalllist";
    var _result = await _client.GetStringAsync(_url);
    return res =
    JsonConvert.DeserializeObject<IEnumerable<re
    ( result):
```



```
}
```

в ходе десериализации ловлю ошибку
"FATAL UNHANDLED EXCEPTION:
Newtonsoft.Json.JsonSerializationException:
Error converting value "[{"Id":2,"Name":"Yulia"},
{"Id":3,"Name":"ккк"},{"Id":4,"Name":""},
{"Id":5,"Name":"Дима"}]" to type 'System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<int, string>>'.

показать больше

^ | v • Ответить • Поделиться ›



Denis Lohachov ➔ Denis Lohachov

• 6 месяцев назад

Нашел в чем проблема, сервер уже отдает данные в json, и клиент просит отдавать в json так как Асцепт "application /json " по этому идет двойная сериализация , а как сделать что бы в столовом представлении возвращался ответ? Пробовал "text/*" не получилось(

^ | v • Ответить • Поделиться ›



Yurii Nosach • 8 месяцев назад

Можно в Xamarin использовать NInject для внедрения зависимостей?

^ | v • Ответить • Поделиться ›



Metanit Модератор ➔ Yurii Nosach

• 8 месяцев назад

Через Nuget можно подключать в Xamarin сторонние библиотеки

^ | v • Ответить • Поделиться ›



Balalamb • 9 месяцев назад

При первой попытке соединения в ходе исполнения метода `string result = await client.GetStringAsync(URL)` ловлю ошибку "Произошла ошибка в ходе отправки запроса"

Как можно узнать установилось ли соединение в принципе или нет?

И еще, в заключительной части адреса "`http://192.168.1.107:51613/api/friends/`", а именно `/api/friends/`, можно узнать, что это вообще за обращение, может тут что-то неправильно указано?

И как вообще можно узнать, что серверная часть корректно работает?

^ | v • Ответить • Поделиться ›



Metanit Модератор ➔ Balalamb

• 9 месяцев назад

через отладку попробуйте посмотреть заголовки ответа.

По поводу адреса

`http://192.168.1.107:51613/` - это локальный адрес приложения в моей локальной подсети. Соответственно в вашем случае все может отличаться. `/api/friends/` - это адрес ресурса на сервера.

Ну тут необходимы хотя бы базовые значения asp net web api

^ | v • Ответить • Поделиться ›



Balalamb → Metanit

• 9 месяцев назад

Проблема была в том, что порт не был доступен не смотря на команды из статьи про запуск веб службы, помогла обычная пропись порта в брандмауере. Теперь возвращается ошибка 404.

В результате я вместо поиска в базе, прямо в методе веб-службы создаю два объекта Friend, упаковываю их в List (`db.Friends.ToList()`; я так понимаю именно это и делает) и возвращаю этот List. Но все равно в телефоне мне выдает

Not found 404, не понимаю, что
не так?

^ | v • Ответить • Поделиться ›



Metanit Модератор ➔

Balalamb

• 9 месяцев назад

не, неправильно, не

AdChoices 



[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Google+](#) |
[Youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2018. Все права защищены.