

Расширение API через коннектор

Последние изменения: 2024-03-26

Есть возможность расширить API, создав свой коннектор в серверной части платформы.

Более подробно о создании коннектора можно посмотреть [здесь](#).

Для этого при создании коннектора нужно унаследовать и от интерфейса `IApiExtenderPlugin`, например:

```
public class ApiExtenderConnector : IConnector, IApiExtenderPlugin
{
    public IList<IApiGroup> ApiGroup { get; private set; }
}
```

Свойство `ApiGroup` возвращает массив классов, унаследованных от `IApiGroup`, которые будут обрабатывать все запросы расширения из внешней системы.

```
public interface IApiGroup
{
    string Id { get; }
    object CreateNewEntity();
    string GetEntityIdFieldName();
    IEnumerable<object> GetList();
    object Get(string id);
    object Post(object input);
    object Put(string id, object input);
    void Delete(string id);
}
```

Id — идентификатор, который должен быть уникальным в системе. Он может содержать как буквы, так и цифры. Для обращения к этому расширению в url строку нужно будет вставить этот идентификатор, например:

Создаем класс коннектора:

```
public class ApiExtenderConnector : IConnector, IApiExtenderPlugin
{
    public ApiExtenderConnector() {
        ApiGroup = new List<IApiGroup>();
        ApiGroup.Add(new ApiTestGroup());
    }
    public bool IsSelfTimeoutBehavior { get { return false; } }
    private string id;
    [System.ComponentModel.Description("Идентификатор.")]
    public string Id
    {
```

```

get { return this.id; }
set { this.id = value; }
}
bool initialized = false;
public bool Initialized
{
get
{
return this.initialized;
}
}
private int timeout = 0;
public int Timeout
{
get { return this.timeout; }
set { this.timeout = value; }
}
private bool enabled = true;
public bool Enabled
{
get
{
return this.enabled;
}
set
{
this.enabled = value;
}
}
private TimeoutBehavaior timeoutBehavaior = TimeoutBehavaior.ThrowException;
public TimeoutBehavaior TimeoutBehavaior
{
get { return this.timeoutBehavaior; }
set { this.timeoutBehavaior = value; }
}
public bool IsSupportDeviceInfoInArgs
{get { return false; }
}
[Cleverence.DataCollection.Xml.XmlSerializable(Cleverence.DataCollection.Xml.XmlSerializationType.
None)]
public IList<IApiGroup> ApiGroup { get; private set;}
public void CheckLicenseLimitations(List<LicenseExternalSystem> supportedSystems)
{
}
public void Initialize()
{
}
public object InvokeMethod(string methodName, object[] args)
{
return null;
}
}

```

Далее необходимо откомпилировать и положить готовую библиотеку в папку сервера. В результате после удачного запуска сервера проверяем ответ сервера по пути

<https://localhost:<порт базы>/api/v1/>

В ответе должна содержаться группа

```
{"name»:"Plugins/ApiTestGroup»,"kind»:"EntitySet»,"url»:"Plugins/ApiTestGroup"}
```

Далее при переходе по пути

<https://localhost:<порт базы>/api/v1/Plugins/ApiTestGroup>

получаем список наших сущностей (результат выполнения функции GetList ())

```
{
"@odata.context":"https://localhost:9005/MobileSMARTS/api/v1/$metadata#Plugins",
"value":[
{"id":"1","name":"1111","valueInt":0,"valueBool":false},
{"id":"2","name":"2223","valueInt":0,"valueBool":false},
{"id":"3","name":"3333","valueInt":0,"valueBool":false}]
}
```

Далее POST запрос можем проверить в swagger. Для этого заходим

<https://localhost:<порт базы>/swagger>

Находим там нашу группу Plugins/ApiTestGroup, выбираем POST, вставляем в параметр body:

```
{"id»:"4»,"name»:"new4"}
```

Нажимаем кнопку «Попробовать!»

Plugins/ApiTestGroup

Показать/Скрыть | Операции кратко | Операции подробно

GET /api/v1/Plugins/ApiTestGroup

[Получить все записи](#)

POST /api/v1/Plugins/ApiTestGroup

[Отправить значения в функцию](#)

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
objectWrapper	<pre>{"id": "4", "name": "new4"}</pre>		body	Описание Пример <pre>{ "id": "string" }</pre>

Content Type параметра: [Попробовать!](#)[Спрятать ответ](#)

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"id": "4", "name": "new4"}' 'https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup'
```

URL запроса

<https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup>

Тело ответа

```
{
  "@odata.context": "https://localhost:9005/MobileSMARTS/api/v1/$metadata#Plugins/$entity",
  "id": "4",
  "name": "new4",
  "valueInt": 0,
  "valueBool": false
}
```

HTTP код ответа

201

Далее заходим в блок GET и видим что наша новая сущность сохранилась:

[Попробовать!](#)[Спрятать ответ](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup'
```

URL запроса

<https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup>

Тело ответа

```
{
  "id": "2",
  "name": "2223",
  "valueInt": 0,
  "valueBool": false
},
{
  "id": "3",
  "name": "3333",
  "valueInt": 0,
  "valueBool": false
},
{
  "id": "4",
  "name": "new4",
  "valueInt": 0,
  "valueBool": false
}
]
```

Для удаления заходим в блок DELETE, вводим наш новый идентификатор 4 и выполняем.

DELETE

/api/v1/Plugins/ApiTestGroup('{id}')

Запрос на удаление значений

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
id	4	key: id	path	string
If-Match	*	If-Match header	header	string

Что может прийти в ответ

HTTP код	Причина	Структура ответа	Заголовки
204	NoContent		

Попробовать!

Проверяем в блоке GET и видим, что новая сущность исчезла.

Попробовать!

Скрыть ответ

Curl

curl -X GET --header 'Accept: application/json' 'https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup'

URL запроса

https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup

Тело ответа

```
{
  {
    "id": "2",
    "name": "2223",
    "valueInt": 0,
    "valueBool": false
  },
  {
    "id": "3",
    "name": "3333",
    "valueInt": 0,
    "valueBool": false
  }
}
```

Не нашли что искали?

?

Задать вопрос в техническую поддержку