

# Взаимодействие системы с адаптером по протоколу HTTP

Взаимодействие системы с адаптером и адаптера с ситемой осуществляется при помощи POST запроса.

URL запроса зависит от конфигурации системы и не несёт полезной информации. Он просто определяет точку входа (endpoint) для HTTP сервера.

Например:

```
https://xx.xx:yyy/ipc/request
```

Тело запроса (payload) - строка в формате JSON.

Стандартный формат JSON не содержит информации о типе данных, которые в нём содержатся. Чтобы при запросе не возникало необходимости передавать информацию о типе объекта, можно использовать массив, первым элементом которого является имя класса объекта, а вторым - сам объект.

Например:

```
[
  "net.oilix.udsm.db.UpdateEntity",
  {
    "requestId" : "BF7262...D4",
    "stamp" : "2024-03-01T13:24:19",
    ...
  }
]
```

Ответ - строка в формате JSON.

JSON представляет из себя объект вида:

```
[
  "net.oilix.udsm.common.Reply",
  {
    "code" : "0",
    "message" : "OK",
    "data":
    [
      "net.oilix.udsm.db.EntityCreated",
      {
        "requestId" : "BF7262...D4",
        "entityType" : "net.oilix.udsm.galaktika.Absence",
        "entityId" : "GL0_1234"
      }
    ]
  }
]
```

Здесь

- **code** - результат выполнения запроса
- **message** - текстовое сообщение, описывающее результат
- **data** - данные, передаваемые в ответе.

Поле **data** не обязательно должно присутствовать в ответе. Его нужно

передавать в том случае, когда изменение объектов в системе происходит синхронно в процессе выполнения запроса.

Ниже будут описаны процедуры подтверждения и уведомления об ошибках. В поле **data** можно передать объекты, генерируемые при подтверждении/уведомлении. Это - любой из объектов `net.oilix.udsm.db.EntityCreated`, `net.oilix.udsm.db.ProcessEntityError` или `net.oilix.udsm.db.EntityEventList`.

## Запросы

Обработка запроса на стороне адаптера или системы определяется классом передаваемого объекта. Обработка запроса не должна занимать длительное время. Все длительные процессы, связанные с обработкой запроса, должны происходить автономно в фоновом режиме.

Положительный ответ обработчика означает, что запрос принят в обработку. Результаты этой обработки могут быть переданы отправителю отдельным запросом. Не существует ограничений на время обработки.

### Объект предметной области

Будем называть объектами предметной области объекты, которые существуют в системах, которые не имеют отношения к взаимодействию систем. Например объект `net.oilix.udsm.db.UpdateEntity` указанный выше, не является объектом предметной области, а `net.oilix.udsm.galaktika.Absence`, который будет описан позже, им является.

Можно назвать "объект предметной области" **бизнес-объектом**, но это несколько сужает его сущность.

### Создание (изменение) "объекта предметной области"

Для создания (изменения) "объекта предметной области" служит запрос ``net.oilix.udsm.db.UpdateEntity``:

```
class net.oilix.udsm.db.UpdateEntity
{
    String requestId;
    String stamp;
    Object entity;
}
```

Здесь:

- **stamp** - Время совершения операции.
- **requestId** - GUID запроса.
- **entity** - Объект.

Тип объекта определён в описании самого объекта.

В виде JSON он выглядит так:

```
[
    "net.oilix.udsm.db.UpdateEntity",
    {
```

```

    "requestId" : "BF7262...D4",
    "stamp" : "2024-03-01T13:24:19",
    "entity":
    [
        "net.oilix.udsm.galaktika.Absence",
        {
            "id": "GLO_1234",
            ...
        }
    ]
}
]

```

К объекту, передаваемому в поле `entity` предъявляются следующее требование:

- Одно из его полей должно быть идентификатором.

Правила определения такого поля в разных системах могут быть разными. В адаптере это решается при помощи аннотации `entityId`. Обработка запроса `UpdateEntity` в адаптере и в системе осуществляется по-разному.

### Обработка запроса `UpdateEntity` в адаптере

При получении запроса `UpdateEntity` от системы, адаптер:

- определит поле, которое содержит аннотацию `entityId`
- если в таблице соответствия отсутствует идентификатор, находящийся в этом поле, сгенерирует уникальный идентификатор "шины"
- добавит его в таблицу соответствия идентификаторов
- заменит оригинальный идентификатор объекта на сгенерированный (полученный из таблицы соответствия).
- поместит модифицированный объект в очереди доставки.

При перемещении объекта между адаптерами, идентификатором объекта является идентификатор сгенерированный в шине.

### Обработка запроса `UpdateEntity` в системе

- Если при получении объекта поле идентификатора оставлено пустым (`null`) - это новый объект и он должен быть создан в системе. После этого система должна отправить адаптеру сообщение `net.oilix.udsm.db.EntityCreated` (будет описано ниже). Время отправки такого сообщения не регламентируется. Логика работы адаптера в этом случае будет описана ниже
- Если это поле не пустое (`not null`) - объект должен быть обновлён. При этом отправителю (адаптеру) ничего отправлять не нужно.

### Логика работы адаптера при создании объекта

При получении от системы источника объекта, идентификатора которого нет в таблице соответствия, адаптер помещает идентификатор "шины" этого объекта в таблицу соответствия и оставляет поле идентификатора "системы" пустым. Кроме этого в данную таблицу помещается идентификатор запроса, который привёл к созданию этого объекта.

Bus Reference Info			
BUS ID	SYSTEM ID	RequestId	TYPE
GAL_983647		BF7262...D4	net.oilix.udsm.galaktika.Absence

При получении новых запросов на модификацию (удаление) объекта с этим идентификатором или запросов, в которых объекты ссылаются на этот идентификатор, такие запросы переключаются во временную очередь для хранения объектов с неразрешёнными зависимостями.

После получения от системы запроса `net.oilix.udsm.db.EntityCreated` и обновления таблицы соответствия, объекты из временной очереди отправляются в систему.

## Удаление объекта

Для удаления "объекта предметной области" служит запрос

``net.oilix.udsm.db.DeleteEntity`:`

```
class net.oilix.udsm.db.DeleteEntity
{
    String requestId;
    String stamp;
    String entityType;
    String entityId;
}
```

Здесь:

- **stamp** - Время совершения операции.
- **requestId** - GUID запроса.
- **entityType** - Имя класса передаваемого объекта
- **entityId** - Идентификатор объекта.

В виде JSON он выглядит так:

```
[
  "net.oilix.udsm.db.DeleteEntity",
  {
    "requestId" : "BF7262...D4",
    "stamp" : "2024-03-01T13:24:19",
    "entityType" : "net.oilix.udsm.galaktika.Absence",
    "entityId" : "GL0_1234"
  }
]
```

Этот запрос может быть применён как при передаче от системы-отправителя к адаптеру, так и от адаптера к системе-получателю.

При передаче системе-получателю, после обработки этого запроса никаких подтверждений адаптеру слать не нужно.

Если в адаптер придёт запрос `DeleteEntity` в котором для поля `entityId` в таблице соответствия нет значения - это ошибка.

В дальнейшем необходимо предусмотреть механизмы её исправления.

## Пакетная обработка

Для оптимизации передачи данных, предусмотрен пакетный метод. Он может быть осуществлён при помощи объекта `net.oilix.udsm.db.EntityProcessList`:

```
class net.oilix.udsm.db.EntityProcessList
{
    List<Object> list;
}
```

Здесь:

- **list** - Список запросов обработки объектов. Элементами этого списка могут быть объекты `net.oilix.udsm.db.UpdateEntity` и `net.oilix.udsm.db.DeleteEntity`.

В виде JSON он выглядит так:

```
[
  "net.oilix.udsm.db.EntityProcessList",
  {
    "list":
    [
      [
        "net.oilix.udsm.db.UpdateEntity",
        {
          "requestId": "0c001d56bbaa40258c9437c96926a29e",
          "stamp": "2024-03-29T22:57:00",
          "entity":
          [
            "net.oilix.udsm.galaktika.Absence",
            {
              "id": "GLO_1234",
              ...
            }
          ]
        }
      ],
      [
        "net.oilix.udsm.db.DeleteEntity",
        {
          "requestId": "d1a52d64cc2b4193a6c805aed4ec9adf",
          "stamp": "2024-03-29T22:57:00",
          "entityType": "net.oilix.udsm.galaktika.Absence",
          "entityId": "GLO_1234"
        }
      ],
      [
        "net.oilix.udsm.db.UpdateEntity",
        {
          "requestId": "100dc2dcdf8746db92882c2632ec6c1d",
          "stamp": "2024-03-29T22:57:00",
          "entity":
          [
            "net.oilix.udsm.galaktika.Absence",
            {
              "id": "GLO_1235",
              ...
            }
          ]
        }
      ]
    ]
  }
]
```

```
}  
]
```

Пакетный метод может быть применён как при передаче от системы-отправителя к адаптеру, так и от адаптера к системе-получателю.

## Подтверждение создания объекта

Для подтверждения создания объекта служит запрос

`net.oilix.udsm.db.EntityCreated:`

```
class net.oilix.udsm.db.EntityCreated  
{  
    String requestId;  
    String entityType;  
    String entityId;  
}
```

Здесь:

- **requestId** - GUID запроса.
- **entityType** - Имя класса созданного объекта
- **entityId** - Идентификатор объекта.

В виде JSON он выглядит так:

```
[  
  "net.oilix.udsm.db.EntityCreated",  
  {  
    "requestId" : "BF7262...D4",  
    "entityType" : "net.oilix.udsm.galaktika.Absence",  
    "entityId" : "GLO_1234"  
  }  
]
```

Этот запрос отправляется системой адаптеру после завершения процедуры создания объекта.

## Уведомление об ошибке

Если при создании/обновлении/удалении объекта в системе-получателе возникает ошибка, система может уведомить об этом адаптер. Тогда он сможет передать эту информацию в систему мониторинга.

Для уведомления об ошибке служит запрос `net.oilix.udsm.db.ProcessEntityError:`

```
class net.oilix.udsm.db.ProcessEntityError  
{  
    String code;  
    String message;  
    Object request;  
}
```

Здесь:

- **code** - код ошибки
- **message** - текстовое сообщение, описывающее ошибку
- **request** - запрос, который вызвал ошибку. Этот запрос передаётся адаптеру полностью в том виде, в котором он был получен от адаптера.

В виде JSON он выглядит так:

```
[
  "net.oilix.udsm.db.ProcessEntityError",
  {
    "code" : "33",
    "message" : "Undefined reference",
    "request" :
    [
      "net.oilix.udsm.db.UpdateEntity",
      {
        "requestId":"0c001d56bbaa40258c9437c96926a29e",
        "stamp":"2024-03-29T22:57:00",
        "entity":
        [
          "net.oilix.udsm.galaktika.Absence",
          {
            "id":"GLO_1234",
            ...
          }
        ]
      }
    ]
  }
]
```

Уведомление об ошибке (или подтверждение создания объекта) может быть передано адаптеру несколько раз (в случае, если в системе существуют собственные механизмы исправления ошибок и система будет повторять попытки изменения объекта)

## Пакетная передача уведомлений адаптеру

Уведомления могут быть переданы адаптеру в виде списка:

```
class net.oilix.udsm.db.EntityEventList
{
    List<Object> list;
}
```

Здесь:

- **list** - Список запросов уведомлений. Элементами этого списка могут быть ОБЪЕКТЫ net.oilix.udsm.db.EntityCreated И net.oilix.udsm.db.ProcessEntityError.

В виде JSON он выглядит так:

```
[
  "net.oilix.udsm.db.EntityEventList",
  {
    "list":
    [
      [
        "net.oilix.udsm.db.EntityCreated",
        {
          "requestId" : "BF7262...D4",
          "entityType" : "net.oilix.udsm.galaktika.Absence",
          "entityId" : "GLO_1234"
        }
      ],
      [
        "net.oilix.udsm.db.ProcessEntityError",
        {
          "code" : "33",
          "message" : "Undefined reference",

```

```

        "request" :
        [
            "net.oilix.udsm.db.UpdateEntity",
            {
                "requestId":"0c001d56bbaa40258c9437c96926a29e",
                "stamp":"2024-03-29T22:57:00",
                "entity":
                [
                    "net.oilix.udsm.galaktika.Absence",
                    {
                        "id":"GL0_1234",
                        ...
                    }
                ]
            }
        ],
        ...
    ]
}
]

```

## Объекты предметной области

### Отсутствия сотрудников

Информация об отсутствии сотрудника может быть описана структурой:

```

class net.oilix.udsm.galaktika.Absence
{
    @EntityId
    java.lang.String id;
    java.lang.String tn;
    java.lang.String from;
    java.lang.String till;
    java.lang.String reason;
}

```

или в формате YML

```

- type: class
  name: net.oilix.udsm.galaktika.Absence
  fields:
    - name: id
      className: java.lang.String
      annotations:
        - type: entityId
    - name: tn
      className: java.lang.String
    - name: from
      className: java.lang.String
    - name: till
      className: java.lang.String
    - name: reason
      className: java.lang.String

```

Здесь



- **id** - Идентификатор события в формате, принятом в системе "Галактика". Этот идентификатор понадобится в случае, если впоследствии эта информация изменится и её нужно будет обновить в других системах (например если даты отсутствия изменятся, либо отсутствие будет удалено)
- **tn** - Табельный номер сотрудника.
- **from** - Дата начала отсутствия (включительно) в формате yyyy-MM-dd (например "2024-03-01").
- **till** - Дата окончания отсутствия (включительно) в формате yyyy-MM-dd (например "2024-03-05").
- **reason** - Код причины отсутствия.

В виде JSON этот объект выглядит так:

```
[
  "net.oilix.udsm.galaktika.Absence",
  {
    "id": "GLO_1234",
    "tn": "33",
    "from": "2024-03-01",
    "till": "2024-03-05",
    "reason": 8
  }
]
```