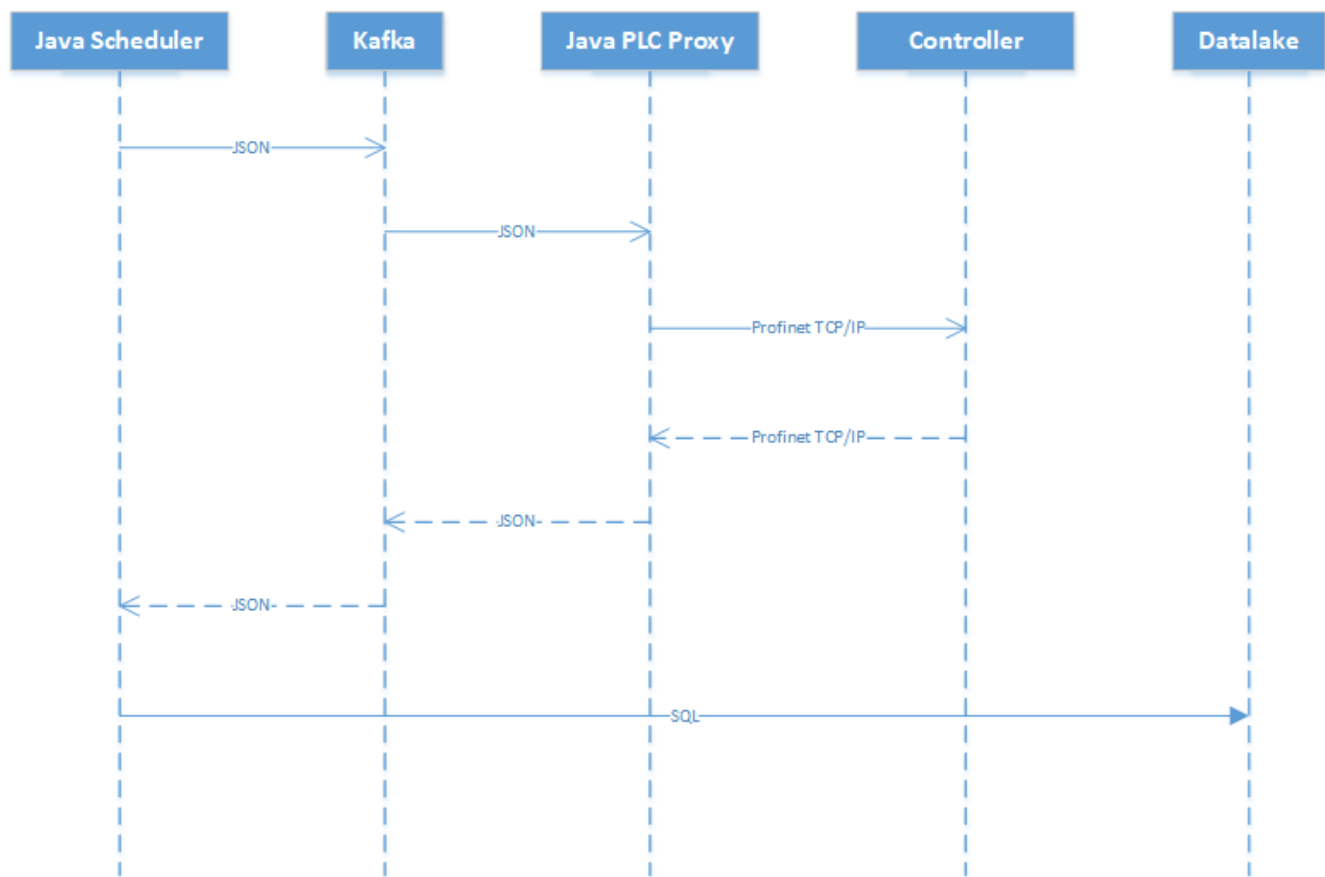


Диаграмма обмена данными с контроллерами



- 1. По расписанию запускается задача, которая в формате JSON отправляет запрос в Kafka для опроса контроллера
- 2. Kafka транслирует запрос в Java PLC Proxy, который обрабатывает запрос и опрашивает необходимые контроллеры и ячейки памяти
- 3. Контроллер предоставляет данные по запросу в Java PLC Proxy
- 4. Java PLC Proxy формирует JSON и передает в Kafka для ответа на исходный запрос.
- 5. Java Scheduled вычитывает очереди, обрабатывает его и сохраняет в базу данных

Описание полей запроса

Элемент	Описание
request_id	Идентификатор запроса
task_id	Идентификатор задачи
request-datetime	Дата и время запроса
request	Данные для обращения к контроллерам
controller_id	Идентификатор контроллера
data	Набор элементов описывающих какие данные нужно получить с контроллера
data-block	Имя блока данных
data-vals	Набор описывающий список данных из этого блока, которые мы хотим получить
operation	Операция над данными. допустимо два значения: read прочитать данные, write записать данные
data-type	Тип данных которые хотим получить
data-offset	сдвиг в байтах относительно начала блока данных, где хранятся нужные нам данные
bitmask	Обязательный для data-type=bit определяет маску бита которым нам нужно получить из байта, для остальных значений data-type не обязательное поле.

id	внутренний идентификатор строки в базе данных, в которой храниться наименование параметра и адрес. данные в таблицу подгружаются из xml в котором должно быть описано актуальное расположение контроллеров и датчиков которые они контролируют. данное значение возвращается в ответом сообщении и позволяет нам понять, значение какого параметра мы получили
----	--

Описание полей ответа

Элемент	Описание
request-datetime	Полученное в запросе дата и время запроса формируется заполняется при формировании запроса к прокси
request-datetime-proxy	Дата и время запроса формируется при получении в прокси
response-datetime	Дата и время передачи ответа
request_id	Идентификатор запроса
task_id	Идентификатор задачи
response	Данные в ответе на запрос
id	см описание параметра запроса
data	Значение параметра полученное из контроллера
controller-datetime	Дата и время полученные из контроллера
status	Статус запроса. 1 запрос выполнен успешно, 0 возникли ошибки
errors	В случае отсутствия ошибок, пустой массив, иначе массив ошибок

Примеры

Пример запроса

```
{
  "request_id": 34234234234234
, "task_id": 12345
, "request-datetime": "2020-02-01T12:03:23.3556"
, "request": [
  {
    "controller_id": 123
    , "data": [
      {
        "data-block": "DB001"
        , "data-vals": [
          { "operation": "read", "data-type": "bool",
            "offset": 0, "bitmask": "00010000", "id": 1345 }
          , { "operation": "read", "data-type": "bool",
            "offset": 1, "bitmask": "00010000", "id": 1356 }
          , { "operation": "write", "data-type": "bool",
            "offset": 24, "bitmask": "00010000", "id": 1334 }
        ]
      }
    ]
    ,
    {
      "data-block": "DB002"
      , "data-vals": [
```

```

        {"operation":"read","data-type":"byte", "offset":10, "id":1345}
        ,{"operation":"read","data-type":"bool",
"offset":13,"bitmask":"00010000", "id":1336}
        ,{"operation":"read","data-type":"bool",
"offset":43,"bitmask":"00010000", "id":1330}
    ]
    }
]
}
,
{
    "controller_id":12
    , "data":[
        {
            "data-block":"DB101"
        , "data-vals":[
            {"operation":"read","data-type":"real", "offset":10, "id":1245}
            ,{"operation":"read","data-type":"word", "offset":11, "id":1256}
            ,{"operation":"read","data-type":"byte", "offset":224, "id":1234}
        ]
        }
    ,
        {
            "data-block":"DB102"
        , "data-vals":[
            {"operation":"read","data-type":"bool",
"offset":11,"bitmask":"00010000", "id":1245}
            ,{"operation":"write","data-type":"bool",
"offset":23,"bitmask":"00010000", "id":1236}
            ,{"operation":"write","data-type":"bool",
"offset":56,"bitmask":"00010000", "id":1230}
        ]
        }
    ]
}
]
}
}

```

Пример ответа

```
{
  , "request-datetime": "2020-02-02T13:00:00.45"
  , "request-datetime-proxy": "2020-02-02T13:00:00.45"
  , "response-datetime": "2020-02-02T13:00:01.45"
  , "request_id": 23123123123
  , "task_id": 123
  , "response": [
    {
      "id": 1345
      , "data": false
      , "controller-datetime": "2020-02-02T13:00:00.45"
      , "status": 1
      , "errors": []
    }
    , {
      "id": 1344
      , "data": true
      , "controller-datetime": "2020-02-02T13:00:00.45"
      , "status": 1
      , "errors": []
    }
    , {
      "id": 1343
      , "data": true
      , "controller-datetime": "2020-02-02T13:00:00.45"
      , "status": 1
      , "errors": []
    }
    , {
      "id": 2345
      , "data": 1.2
      , "controller-datetime": "2020-02-02T13:00:00.45"
      , "status": 0
      , "errors": [
        {
          "1": " "
          , "2": " "
        }
      ]
    }
  ]
}
```