

PLC Proxy Server

Руководитель проекта	Безруков Антон
Аналитик	Бяков Сергей
Ссылка на задачу в JIRA	https://jira.dds.lanit.ru/browse/NIOKR-4

Проблема / Задача	Что нужно сделать	Что для заказчика будет являться показателем решения проблемы / задачи	Риски
	Предоставить возможность клиентам получать данные с контроллеров для дальнейшей обработки		

- Чтение конфигурации контроллеров
- Обработка конфигурации контроллеров
- Обработка запроса клиента
- Формирование ответа

PLC Proxy Server предоставляет возможность клиентам получать данные с контроллеров для дальнейшей обработки.

На вход сервер получает JSON содержащий информацию о том, кто, для какой задачи и какие данные хочет получить. Проанализировав запрос, прокси сервер обращается к контроллерам и получает от них необходимые данные, после чего в ответом JSON передает полученные результаты.

Чтение конфигурации контроллеров

При запуске сервер должен запросить и получить данные в которых описаны параметры соединения с контроллерами.

Запрос и ответ осуществляются через message broker Kafka.

Описание формата данных приведено на странице [Загрузка конфигурации прокси](#)

Необходимо реализовать чтение конфигурации и соответственно механизм предоставления данных по конфигурации из базы данных.

В случае успешного чтения конфигурации, прокси должен использовать полученные данные для обращения к контроллеру.

При изменении конфигурации в базе данных, потребует перезапуск прокси сервера, что бы измененные данные были учтены в работе прокси. Автоматическая проверка изменений не реализуется умышлено, так как данные изменения будут крайне редкими.

Обработка конфигурации контроллеров

Описание формата данных приведено на странице [Загрузка конфигурации прокси](#)

Основным параметром, на который необходимо обратить внимание является параметр permanent_connection

В случае если значение параметра = true, то после открытия соединения с контроллером, необходимо осуществлять контроль соединения, т.е. при данном значении параметра, соединение с контроллером всегда должно быть открыто, и в случае обрыва соединения, по истечении времени указанном в параметре timeout необходимо пытаться установить соединение с контроллером повторно. Данный режим работы нужен для обеспечения скорости чтения данных с контроллера и предоставления данных клиенту. По результатам нашего теста, 75% времени обработки запроса тратиться на установку соединения с контроллером.

При значении параметра permanent_connection = false с контроллером нужно соединяться только в случае запроса от клиента, после получения данных, соединение необходимо закрывать, что бы не исчерпать лимит подключений к контроллеру, для контроллеров трехсотой серии установлены лимиты в 20 активных подключений.

Обработка запроса клиента

Прокси сервер получает запросы от клиентов через message broker Kafka в формате JSON. Описание формата запроса и диаграмма потока данных приведена на странице [Диаграмма обмена данными с контроллерами](#).

request_id	Идентификатор запроса. Данное значение вернется в ответе, что бы можно было связать результаты запроса и ответа
task_id	Идентификатор задачи. Данное значение вернется в ответе, используется для исключения лишних запросов по определению задачи в рамках которой получены данные

request_datetime	Дата и время формирования запроса. Данное значение вернется в ответе для оценки скорости работы
controller_id	Идентификатор контроллера. Необходимо проверить, что данное значение присутствует в конфигурации прокси сервера. Если значение найдено, то читаем данные из указанного контроллера.
data-block	Указывает блок данных с которым необходимо работать, в контроллере указанном в controller_id
operation	Тип операции над данными. read - чтение данных из контроллера. write запись данных в контроллер. При значении read необходимо учитывать конфигурацию прокси, там запись может быть запрещена. Если запись в конфиге прокси разрешена, то проверить, что она разрешена и на контроллере указанном в controller_id
data-type	Указывает тип данных которые следует прочитать или записать в контроллер. Допустимые типы по ссылке выше
data-offset	Сдвиг в байтах относительно начала блока данных, от куда нужно начать читать или куда записать данные.
bitmask	Номер бита в байте, для data-type = bool который следует прочитать или записать
id	Внутренний идентификатор в таблице controllers_data.id который описывает данные, которые мы читаем. Возвращается обратно в ответе. Позволяет понять, значение каких данных мы получили.

В случае возникновения ошибок, при работе с определенными данными, обработка всего запроса продолжается. Ошибка фиксируется и передается в ответном сообщении.

Если у контроллера указано постоянное соединение, но оно не доступно, то необходимо обработать запрос к данному контроллеру по схеме permanent_connection = false, т.е. попытаться установить соединение с контроллером. Если соединение установится, произвести операцию на данными, если не установится, зафиксировать ошибку соединения и продолжить обработку запроса.

Формирование ответа

После обработки всех элементов запроса, необходимо сформировать ответное сообщение и передать его в message broker для клиента. Описание формата ответа приведено на странице [Диаграмма обмена данными с контроллерами](#).

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, заполняется значением из запроса
task_id	Идентификатор задачи, заполняется значением из запроса
id	заполняется значением из 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

Массив ошибок формируется в виде "порядковый номер ошибки": "текст ошибки" при этом счетчик ошибок обнуляется для каждого элемента запроса (id).