

## **ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЕ РАСПОЛОЖЕННЫХ В РАЗНЫХ ДОМЕНАХ С ИСПОЛЬЗОВАНИЕМ CORS**

Силков М.С.

Научный руководитель – Куприянов А.Б., к.т.н., доцент

### **Введение**

В современном мире веб-приложения все чаще сталкиваются с необходимостью взаимодействия между собой, даже если они расположены на разных доменах. Это может быть вызвано, например, потребностью в интеграции данных или функциональности разных сервисов. Однако, безопасность браузеров ограничивает такие запросы из соображений безопасности. Здесь на помощь приходит CORS (Cross-Origin Resource Sharing) - механизм, который позволяет веб-приложениям запрашивать данные из других источников, не нарушая политики безопасности браузера.

### **Решение**

Прежде чем двигаться дальше, важно понять, что такое CORS. Это механизм, который позволяет веб-странице запросить ресурс с другого домена, чем тот, с которого сама страница была загружена. Без CORS браузеры блокируют такие запросы из соображений безопасности.

Для разрешения CORS запросов на сервере необходимо установить заголовки CORS в ответ на запросы от клиента. При настройке CORS на сервере с ASP.NET Core, возможно установить различные параметры, чтобы точно определить, какие запросы разрешены, а какие - нет. Вот некоторые из основных параметров, которые могут быть использованы:

- **Origins (Источники):** Предоставляет возможность указать один или несколько доменов, с которых разрешены CORS запросы. Это позволяет точно контролировать источники, которым разрешено взаимодействовать с вашим сервером.
- **Methods (Методы):** Это определяет HTTP методы, которые разрешены для CORS запросов. Обычно это GET, POST, PUT, DELETE и так далее.
- **Headers (Заголовки):** Этот параметр определяет список заголовков, которые могут быть включены в CORS запросы. Это может быть полезно, если приложение отправляет кастомные заголовки, которые нужно разрешить.
- **Exposed Headers (Открытые заголовки):** В данном параметре можно указать список заголовков, которые могут быть доступны на клиенте после выполнения CORS запроса. Например, если ваш сервер отправляет определенные заголовки в ответе, вы можете разрешить клиентам получать доступ к ним.

- **Credentials (Учетные данные):** Этот параметр указывает, разрешено ли отправлять и принимать учетные данные (например, куки или авторизационные заголовки) в CORS запросах. Обычно это устанавливается в true только в случае, если оба домена доверяют друг другу.

Используя эти параметры, становится возможным точно настроить политику CORS на веб-сервере, чтобы обеспечить безопасное и эффективное взаимодействие между веб-приложением и другими доменами.

Вот пример более подробной настройки политики CORS с использованием ASP.NET Core:

```
services.AddCors(options =>
{
    options.AddPolicy("AllowSpecificOrigin",
        builder => builder
            .WithOrigins("http://example.com")
            .AllowAnyMethod()
            .AllowAnyHeader()
            .AllowCredentials());
});
```

Рис. 1. Настройка политики CORS

Этот код устанавливает политику CORS, разрешающую запросы только с указанного источника - `http://example.com`. Существует возможность настроить его так, чтобы разрешить запросы с разных источников, указав `"*"`, но с данной конфигурацией нужно быть осторожным, так как это может представлять угрозу для безопасности.

#### **Вывод**

Использование CORS - это важный аспект разработки веб-приложений, особенно при работе с распределенными системами или взаимодействии между различными доменами. Правильная настройка CORS позволяет обеспечить безопасное и эффективное взаимодействие между веб-приложениями, расположенными на разных доменах, открывая новые возможности для разработчиков и улучшая пользовательский опыт.

#### *Литература*

1. Троелсен Э., Джепикс Ф., Язык программирования C#7 и платформы .Net и .Net Core – Санкт-Петербург: Питер, 2018. – 1316 с.

2. Рихтер Дж., CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# – Москва: Питер, 2018. – 893 с.

УДК 004.932

## АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ АРТЕЗИАНСКОЙ СКВАЖИНОЙ АСУ АС

Шишковец Ф.Д.

Научный руководитель – Куприянов А.Б., к. т. н., доцент

Автоматизированная система управления артезианскими скважинами предназначена для дистанционного и локального управления оборудованием артезианских скважин. Система автоматизации скважины осуществляет контроль состояния оборудования и технологических параметров с центрального (или локального) диспетчерского пункта посредством кабельной линии связи, модема или GSM канала, а также оповещает оператора о возникновении перебоев в работе системы.

Автоматизированная система обеспечивает автоматическое местное и дистанционное включение и выключение оборудования скважины по заданному алгоритму, контроль наличия сетевого напряжения в павильоне артезианской скважины. Также контроль давления воды на выходе насоса, контроль статического и динамического уровня воды в скважине при наличии пьезометрического датчика, контроль температуры и других параметров поступающей воды. При возникновении аварийной ситуации включается звуковая и световая сигнализация на пульте управления, также оповещается отдел, отвечающий за работу АСУ АС.

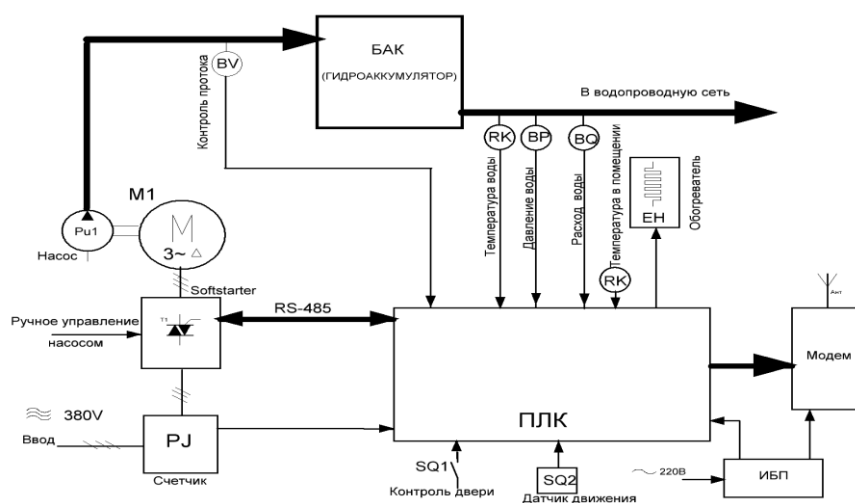


Рис. 1. Структурная схема автоматизации управления скважиной