



ООО «КВАНТУМ АРТ»

115184, Москва, Озерковский переулок, д. 12

тел. (495) 783-65-74

# Модуль Angular для продукта QP8.WidgetPlatform

---

Руководство по установке (Linux)

Москва  
2023

## ИСТОРИЯ ИЗМЕНЕНИЙ

Версия	Дата	Автор	Описание
1.1	16.10.2023	Молькова М.Е.	Обновление структуры документа
1.0	05.07.2023	Григорьева М.А.	Первичное техническое описание

## Оглавление

---

<b>1. ОБОЗНАЧЕНИЯ</b> .....	<b>4</b>
<b>2. ОПРЕДЕЛЕНИЯ И ТЕРМИНЫ, ИСПОЛЬЗУЕМЫЕ В ДОКУМЕНТЕ</b> .....	<b>5</b>
2.1. ОБЩИЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ .....	5
2.2. ТЕРМИНЫ QR .....	5
<b>3. ОБЩИЕ СВЕДЕНИЯ</b> .....	<b>7</b>
3.1. СОСТАВ МОДУЛЯ .....	7
<b>4. СИСТЕМНЫЕ ТРЕБОВАНИЯ</b> .....	<b>8</b>
<b>5. УСТАНОВКА QR8.WIDGETPLATFORM.ANGULAR (С ИСПОЛЬЗОВАНИЕМ DOCKER)</b> .....	<b>9</b>
<b>6. УСТАНОВКА QR8.WIDGETPLATFORM.ANGULAR (С ИСПОЛЬЗОВАНИЕМ KUBERNETES)</b> .....	<b>10</b>
<b>7. УСТАНОВКА QR8.WIDGETPLATFORM.ANGULAR (БЕЗ ИСПОЛЬЗОВАНИЯ DOCKER)</b> .....	<b>11</b>
7.1. УСТАНОВКА WP.API .....	11
7.2. УСТАНОВКА WP.DEMOSITERUS.ANGULAR .....	12
<b>8. НАСТРОЙКА NGINX</b> .....	<b>14</b>
8.1. NGINX С ИСПОЛЬЗОВАНИЕМ DOCKER.....	14
8.2. NGINX БЕЗ ИСПОЛЬЗОВАНИЯ DOCKER .....	14

## 1. Обозначения

Обозначение	Описание	Пример использования
Технические данные	Используется для выделения различных технических данных в тексте: URL, названия свойств и методов, имена файлов и т.п.	ГПИ Системы доступен по URL <code>https://www.domainname.zone/</code> .
Код	Пример кода.	<code>public DataTable Data { get; set; }</code>
Переменная	Используется для указания переменного значения.	Формат URL: <i>Базовый URI/Псевдоним объекта</i>
Требует дополнения	TBD (to be determined). Указывает, что необходима доработка текста – проверка корректности утверждения, детализация, правка после внесения изменений в документ и т.п.	Система работает с одной БД.
Примечание:	Дополнительные данные справочного характера.	<b>Примечание:</b> используется при генерации классов LINQ to SQL.
Внимание:	Важные данные, которые требуется обязательно учитывать.	<b>Внимание:</b> опция поддерживается только ASP-сборкой в целях совместимости.

## 2. Определения и термины, используемые в документе

### 2.1. Общие термины и определения

В таблице ниже приведено описание используемых терминов и определений.

Термин или определение	Описание
<b>API</b>	«Application Programming Interface» (интерфейс программирования приложений) – набор правил по использованию функциональных возможностей Системы, предоставляемый разработчикам для организации взаимодействия сторонних программных продуктов с Системой
<b>QP8.CMS</b> или <b>QP8.CMS с поддержкой PostgreSQL</b> (далее «QP»)	Программный продукт, обладающий широким спектром возможностей для разработки программной части Системы различной сложности
<b>QP8.WidgetPlatform</b> (также «Виджетная платформа»)	Расширяет возможности QP. Позволяет через бэкенд наполнять веб-страницы Системы самостоятельно разработанными модульными приложениями. Виджетная платформа и виджеты основаны на шаблоне архитектуры MVC (от англ. «Model-View-Controller», «Модель-Представление-Контроллер»)
<b>URL</b>	«Uniform Resource Locator» – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса
<b>БД</b>	База данных
<b>ГПИ</b>	Графический пользовательский интерфейс
<b>REST</b>	«Representational State Transfer» – архитектурный стиль, определяющий правила взаимодействия между клиентом и сервером
<b>HTTP</b>	«HyperText Transfer Protocol» – протокол передачи гипертекста
<b>Docker</b>	Программная платформа для быстрой сборки, отладки и развертывания приложений с помощью контейнеров
<b>Kubernetes</b>	Инструмент оркестрации контейнеров с открытым исходным кодом, который позволяет беспрепятственно развертывать и масштабировать контейнерные приложения, управлять ими в различных производственных средах
<b>nginx</b>	Программное обеспечение с открытым исходным кодом, используемое в качестве почтового или обратного прокси-сервера
<b>NPM</b>	«Node Package Manager» – пакетный менеджер для JavaScript, работающий на Node.js
<b>JSON</b>	«JavaScript Object Notation» — текстовый формат хранения и передачи структурированных данных в формате «ключ:значение»
<b>DNS</b>	«Domain Name System» — система доменных имен
<b>CORS</b>	«Cross-Origin Resource Sharing» — механизм обеспечения безопасности и защиты пользователей, позволяющий определить список ресурсов, к которым страница может получить доступ

### 2.2. Термины QP

В таблице ниже приведены термины QR.

Термин или определение	Описание
<b>GraphQL</b>	Отдельный сервис для доступа к данным QR при разработке новых сайтов/продуктов, подключаемый к QR как плагин
<b>Бэкенд</b>	Копия QR; обладает ГПИ для работы с содержимым БД Системы
<b>Контент</b>	Раздел сайта, отвечающий за содержание пользовательской таблицы БД Системы и ее настройки
<b>Поле</b>	Атрибут контента. С использованием полей формируется структура данных для контента.
<b>Сайт</b>	Набор данных в бэкенде. Допускается создание нескольких сайтов. Содержимое каждого сайта определяется созданным в нём контентом.

## 3. Общие сведения

Модуль **Angular** для продукта **QP8.WidgetPlatform** (или `QP8.WidgetPlatform.Angular`) – это набор средств, позволяющих разработать сайт на технологии *Angular* с использованием возможностей виджетной платформы `QP8.WidgetPlatform`.

### 3.1. Состав модуля

Модуль `Angular` для `QP8.WidgetPlatform` состоит из:

- библиотеки [@quantumart/qa-engine-page-structure-angular](https://www.npmjs.com/package/@quantumart/qa-engine-page-structure-angular) (доступна в репозитории `npm.js`, включается в состав сайтов, разрабатываемых на технологии `Angular`);
- сервиса API виджетной платформы;

Также в состав дистрибутива входит демо-сайт, показывающий возможности виджетной платформы на технологии `Angular`.

**Внимание:** Модуль `Angular` не является мультитенантным и устанавливается для конкретного *customer code*.

**Внимание:** Компонент `WP.API` может использоваться только либо для `live`, либо для `stage` (по умолчанию). Если необходимы оба режима, нужно установить ещё один экземпляр сервиса, поменяв название сервиса и порт.

**Внимание:** Компонент `WP.API` (сервис API виджетной платформы) входит также в состав модуля `React`. В случае одновременной установки обоих модулей можно использовать уже установленный компонент и закомментировать соответствующие разделы манифестов / не выполнять соответствующие шаги установки, либо установить ещё один экземпляр сервиса, поменяв название сервиса и порт.

## 4. Системные требования

Для установки и использования модуля Angular требуется установка продукта «QP8.WidgetPlatform».



## 5. Установка QP8.WidgetPlatform.Angular (с использованием Docker)

1. Скачать архив [widget-angular-config.tar](#) и распаковать его содержимое в `/etc/widget-angular-config`.
2. Перейти в папку `/etc/widget-angular-config/compose`.
3. В файле `docker-compose.yml` задать:
  - строку подключения к ранее развёрнутой базе демо-сайта в параметре `QpSettings__ConnectionString` (можно скопировать из конфигурационного файла QP);
  - внешний URL приложения WP.API в параметре `WIDGET_PLATFORM_API_URL` вместо `http://localhost:6200`, если этот этот внешний URL известен на текущий момент, и если планируется, что пользователь будет открывать приложение через браузер с другого компьютера;
  - внешний URL приложения QP.GraphQL в параметре `GRAPHQL_DATA_API_URL` вместо `http://localhost:6300`, если этот этот внешний URL известен на текущий момент, и если планируется, что пользователь будет открывать приложение через браузер с другого компьютера.

**Внимание:** URL приложения QP.GraphQL должен заканчиваться на `/graphql` или `/graphql/stage`

4. Выполнить команду:

```
sudo docker-compose up -d
```

## 6. Установка QP8.WidgetPlatform.Angular (с использованием Kubernetes)

1. Скачать архив [widget-angular-config.tar](#) и распаковать его содержимое в `/etc/widget-angular-config`.
2. Перейти в папку `/etc/widget-angular-config/k8s`.
3. В файле `widget.yml` можно настроить:
  - строку подключения к ранее развёрнутой базе демо-сайта в параметре `QpSettings__ConnectionString` (можно скопировать из конфигурационного файла QP);
  - внешний DNS приложения WP.API в параметре `WIDGET_PLATFORM_API_URL` вместо `wp-demosite-rus-api.test`;
  - внешний DNS приложения QP.GraphQL в параметре `GRAPHQL_DATA_API_URL` вместо `graphql-demosite-rus-api.test`.

**Внимание:** URL приложения QP.GraphQL должен заканчиваться на `/graphql` или `/graphql/stage`

4. Развернуть сервисы командой:

```
kubectl apply -f widget.yml
```

5. При необходимости можно настроить ингрессы к `wp-demosite-rus-api` и `wp-demosite-rus-angular` и в файле `ing.yml`, задав свои DNS вместо `wp-demosite-rus-api.test` и `wp-demosite-rus-angular.test` и выполнив команду:

```
kubectl apply -f ing.yml
```

## 7. Установка QP8.WidgetPlatform.Angular (без использования Docker)

### При установке из скомпилированного кода:

1. Скачать [архив](#), содержащий бинарные файлы API виджетной платформы (WP.API) и демо-сайта Angular (WP.DemositeRus.Angular).
2. Распаковать полученный архив в домашний каталог пользователя, созданного в рамках установки QP8.CMS (по умолчанию – /home/qp). Должна получиться следующая структура из двух каталогов:
  - WP.API;
  - WP.DemositeRus.Angular.

### 7.1. Установка WP.API

#### При сборке из исходников:

1. Вытянуть исходники из [репозитория](#) на github.
2. В папке проекта QA.WidgetPlatform.Api (где должен быть файл QA.WidgetPlatform.Api.csproj) выполнить команду на публикацию:

```
dotnet publish "QA.WidgetPlatform.Api.csproj" -c Release -o bin/release/publish/ -r linux-x64 --self-contained=false
```

3. В папке /home/qp создать подпапку WP.API и скопировать туда всё содержимое каталога bin/release/publish/, в который осуществлялась публикация. Следует проверить, что у пользователя QP есть права на чтение и исполнение содержимого папки WP.API.

#### При всех вариантах установки:

1. Перейти в папку WP.API. В файле appsettings.json в секции QpSettings изменить значение параметра ConnectionString на строку подключения к ранее развёрнутой базе демо-сайта в параметре (можно скопировать из конфигурационного файла QP).
2. В файле NLog.config найти internalLogFile и <variable name="logDirectory" value= и прописать там путь /var/log/wp-api/.

**Внимание:** для internalLogFile должен сохраниться конечный файл, т.е. правильно строка будет выглядеть так: /var/log/wp-api/internal-nlog.txt

3. В файле NLog.config в секции rules во всех логгерах, в которых в параметре writeTo задано значение console, заменить его на fileStructured.
4. Создать на сервере директорию /var/log/wp-api/ и выдать пользователю QP права на владение директорией.
5. Создать файл wp-api.service в папке /usr/lib/systemd/system/ и заполнить его следующим содержимым:

```
[Unit]
Description=WP API
After=qp.service
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=on-failure
RestartSec=5
User=qp
WorkingDirectory=/home/qp/WP.API/
ExecStart=/bin/dotnet QA.WidgetPlatform.Api.dll --urls http://*:6200

[Install]
WantedBy=multi-user.target
```

**Внимание:** если в конкретном дистрибутиве Linux отсутствует папка `/usr/lib/systemd/system/`, то все изменения следует вносить в папку `/lib/systemd/system/`.

6. Прописать сервис в автозапуск, выполнив команду:

```
systemctl enable wp-api.service
```

7. Запустить сервис, выполнив команду:

```
systemctl start wp-api.service
```

## 7.2. Установка WP.DemoSiteRus.Angular

**При сборке из исходников:**

1. Вытянуть исходники из [репозитория](#) на github.
2. В папке `Demosite.Angular/demosite` выполнить установку *npm*-пакетов командой:

```
npm ci
```

3. Собрать фронт в подпапку `dist` командой:

```
npm run build:ssr --configuration=production
```

4. В папке `/home/qp` создать подпапку `WP.DemoSiteRus.Angular` и скопировать туда всё содержимое каталога `dist`, в который осуществлялась публикация. Следует проверить, что у пользователя `QP` есть права на чтение и исполнение содержимого папки `WP.DemoSiteRus.Angular`.

**При всех вариантах установки:**

1. Создать файл `wp-demosite-rus-angular.service` в папке `/usr/lib/systemd/system/` и заполнить его следующим содержимым:

```
[Unit]
Description=QP DemositeRus.Angular
```

```

After=wp-api.service qp-graphql-service.service
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=on-failure
RestartSec=5
User=qp
Environment=NODE_ENV=production
Environment=WIDGET_PLATFORM_API_URL=http://localhost:6200
Environment=GRAPHQL_DATA_API_URL=http://localhost:6300/graphql/stage
WorkingDirectory=/home/qp/WP.DemositeRus.Angular/
ExecStart=/bin/node dist/demosite/server/main.js

[Install]
WantedBy=multi-user.target
  
```

2. При этом необходимо задать:

- внешний URL приложения WP.API в параметре `WIDGET_PLATFORM_API_URL` вместо `http://localhost:6200`, если этот этот внешний URL известен на текущий момент, и если планируется, что пользователь будет открывать приложение через браузер с другого компьютера;
- внешний URL приложения QP.GraphQL в параметре `GRAPHQL_DATA_API_URL` вместо `http://localhost:6300`, если этот этот внешний URL известен на текущий момент, и если планируется, что пользователь будет открывать приложение через браузер с другого компьютера.

**Внимание:** URL приложения QP.GraphQL должен заканчиваться на `/graphql` или `/graphql/stage`

**Внимание:** если в конкретном дистрибутиве Linux отсутствует папка `/usr/lib/systemd/system/`, то все изменения следует вносить в папку `/lib/systemd/system/`.

3. Прописать сервис в автозапуск, выполнив команду:

```
systemctl enable wp-demosite-rus-angular.service
```

4. Запустить сервис, выполнив команду:

```
systemctl start wp-demosite-rus-angular.service
```

## 8. Настройка nginx

Для корректной работы WP.API и QP.GraphQL необходимо решить вопрос безопасности, связанный с CORS-ограничениями. Для их обхода следует формировать запросы на тот же домен, на котором располагается WP.DemoSiteRus.Angular. Осуществить это можно с помощью сервера nginx.

### 8.1. Nginx с использованием docker

Если nginx при установке QP8.CMS был запущен в docker, то следует:

1. Выполнить команду:

```
docker-compose down
```

2. Открыть на редактирование файл `nginx.conf`, предположительно расположенный по пути `/etc/qpconfig/nginx/nginx.conf`;
3. В конфигурационный файл добавить секции из файла `/etc/widget-angular-config/nginx/wp-nginx.conf`;
4. Задать свои DNS вместо `wp-demosite-rus-api.test`, `graphql-demosite-rus-api.test` и `wp-demosite-rus-angular.test`;
5. Выполнить команду:

```
docker-compose up -d
```

### 8.2. Nginx без использования docker

Если nginx при установке QP8.CMS был запущен не в docker, то следует:

1. Открыть на редактирование файл `nginx.conf`, предположительно расположенный по пути `/etc/nginx/nginx.conf`;
2. В конфигурационный файл добавить секции из файла `/home/qp/widget-angular-config/nginx/wp-nginx.conf`;
3. Задать свои DNS вместо `wp-demosite-rus-api.test`, `graphql-demosite-rus-api.test` и `wp-demosite-rus-angular.test`;
4. Проверить корректность конфигурации командой:

```
nginx -t
```

5. если всё корректно, то прочитать обновленную конфигурацию nginx командой:

```
nginx -s reload
```