

## **Программный комплекс**

**Универсальная интеграционная платформа для  
построения систем информационного мониторинга и  
анализа данных интегрированных подсистем сбора  
информации (ПК УИП «Глория»)**

**Руководство администратора**

## СОДЕРЖАНИЕ

1 Состав.....	3
2 Назначение.....	4
3 Запуск ПО .....	5
3.1 Запуск ПО в Windows .....	5
3.2 Запуск ПО в AstraLinux.....	6
4 Остановка ПО .....	8
5 Конфигурирование параметров соединения с БД.....	9
6 Настройка связи между модулями.....	10

## 1 Состав

Программный комплекс универсальная интеграционная платформа для построения систем информационного мониторинга и анализа данных интегрированных подсистем сбора информации (ПК УИП «Глория») (далее ПО) состоит из следующих основных модулей:

- сервер бизнес-логики (далее Сервер);
- сервер публикации Web-интерфейса (далее Web-сервер);

## 2 Назначение

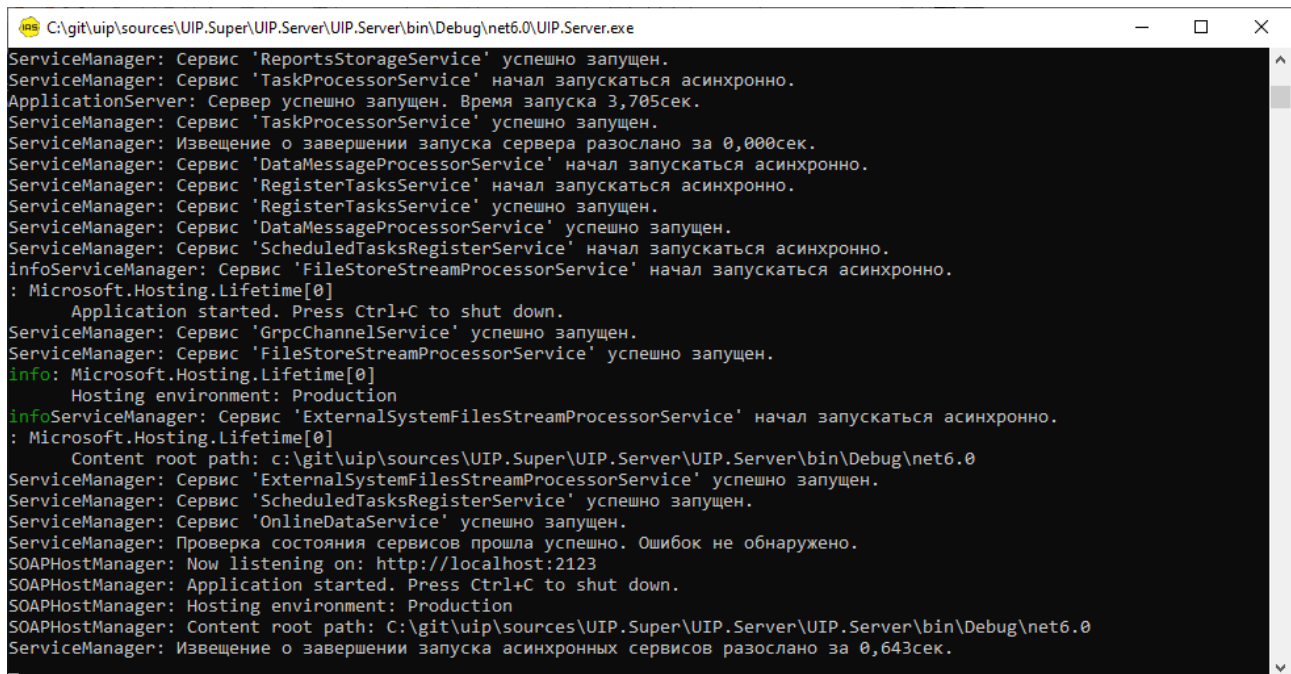
Программный комплекс универсальная интеграционная платформа для построения систем информационного мониторинга и анализа данных интегрированных подсистем сбора информации (ПК УИП «Глория») предназначен для построения распределённой системы интегрирующей данные из различных источников и предоставляющей инструменты мониторинга и аналитики поступающих данных.

Универсальная интеграционная платформа для построения систем информационного мониторинга и анализа данных интегрированных подсистем сбора информации - ПК УИП «Глория» - это комплекс сервисных программ и модулей, доступных разработчику. Сервер предоставляет интегрированную среду для развертывания и выполнения высокопроизводительных серверных бизнес-приложений и управления ими. Эти приложения могут обслуживать запросы, принимаемые от удаленных клиентских систем, в том числе, подключающихся из Интернета, корпоративной сети или интрасети. Предоставляет разработчикам упрощенную модель программирования сетевых серверных приложений. Разработчики могут использовать встроенные библиотеки для реализации в приложениях множества функций, таких как ввод-вывод, обработка численных данных и текста, доступ к базам данных, обработка XML-кода, управление транзакциями, бизнес-правила и веб-службы.

## 3 Запуск ПО

### 3.1 Запуск ПО в Windows

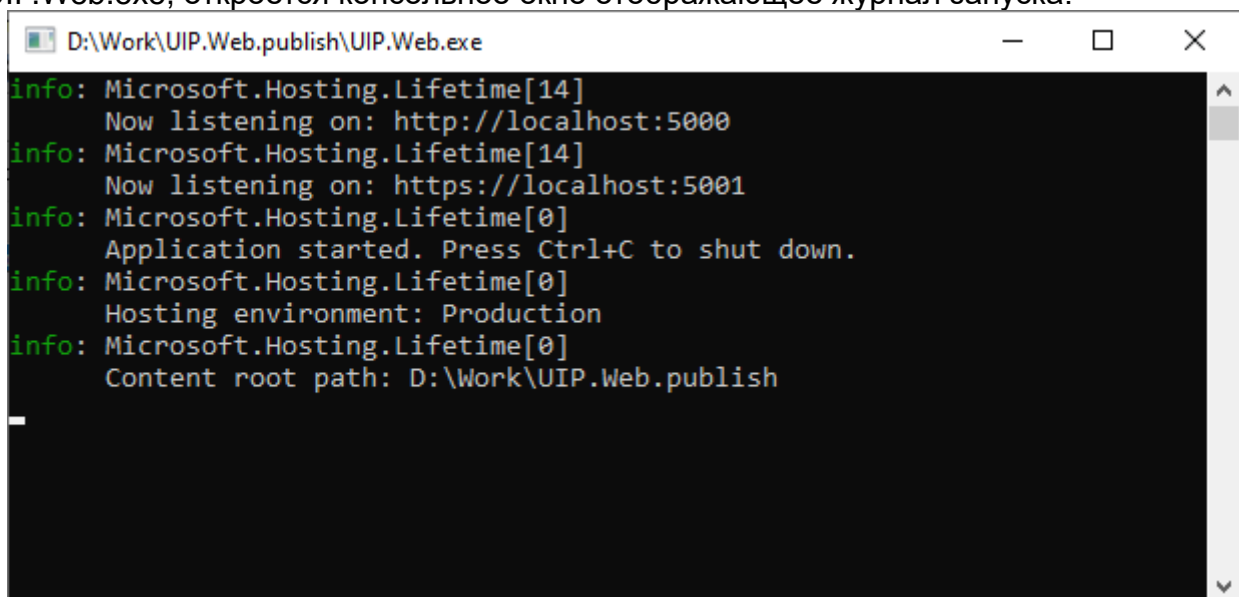
Для запуска Сервера в соответствующей папке нужно запустить файл UIP.Server.exe, откроется консольное окно отображающее журнал запуска.



```
C:\git\uiip\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0\UIP.Server.exe
ServiceManager: Сервис 'ReportsStorageService' успешно запущен.
ServiceManager: Сервис 'TaskProcessorService' начал запускаться асинхронно.
ApplicationServer: Сервер успешно запущен. Время запуска 3,705сек.
ServiceManager: Сервис 'TaskProcessorService' успешно запущен.
ServiceManager: Извещение о завершении запуска сервера разослано за 0,000сек.
ServiceManager: Сервис 'DataMessageProcessorService' начал запускаться асинхронно.
ServiceManager: Сервис 'RegisterTasksService' начал запускаться асинхронно.
ServiceManager: Сервис 'RegisterTasksService' успешно запущен.
ServiceManager: Сервис 'DataMessageProcessorService' успешно запущен.
ServiceManager: Сервис 'ScheduledTasksRegisterService' начал запускаться асинхронно.
InfoServiceManager: Сервис 'FileStoreStreamProcessorService' начал запускаться асинхронно.
: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
ServiceManager: Сервис 'GrpcChannelService' успешно запущен.
ServiceManager: Сервис 'FileStoreStreamProcessorService' успешно запущен.
Info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Production
InfoServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' начал запускаться асинхронно.
: Microsoft.Hosting.Lifetime[0]
  Content root path: c:\git\uiip\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' успешно запущен.
ServiceManager: Сервис 'ScheduledTasksRegisterService' успешно запущен.
ServiceManager: Сервис 'OnlineDataService' успешно запущен.
ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.
SOAPHostManager: Now listening on: http://localhost:2123
SOAPHostManager: Application started. Press Ctrl+C to shut down.
SOAPHostManager: Hosting environment: Production
SOAPHostManager: Content root path: C:\git\uiip\sources\UIP.Super\UIP.Server\UIP.Server\bin\Debug\net6.0
ServiceManager: Извещение о завершении запуска асинхронных сервисов разослано за 0,643сек.
```

Если всё установлено и сконфигурировано корректно, то в последних строках журнала можно увидеть строку «ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.»

Для запуска Web-сервера в соответствующей папке нужно запустить файл UIP.Web.exe, откроется консольное окно отображающее журнал запуска.

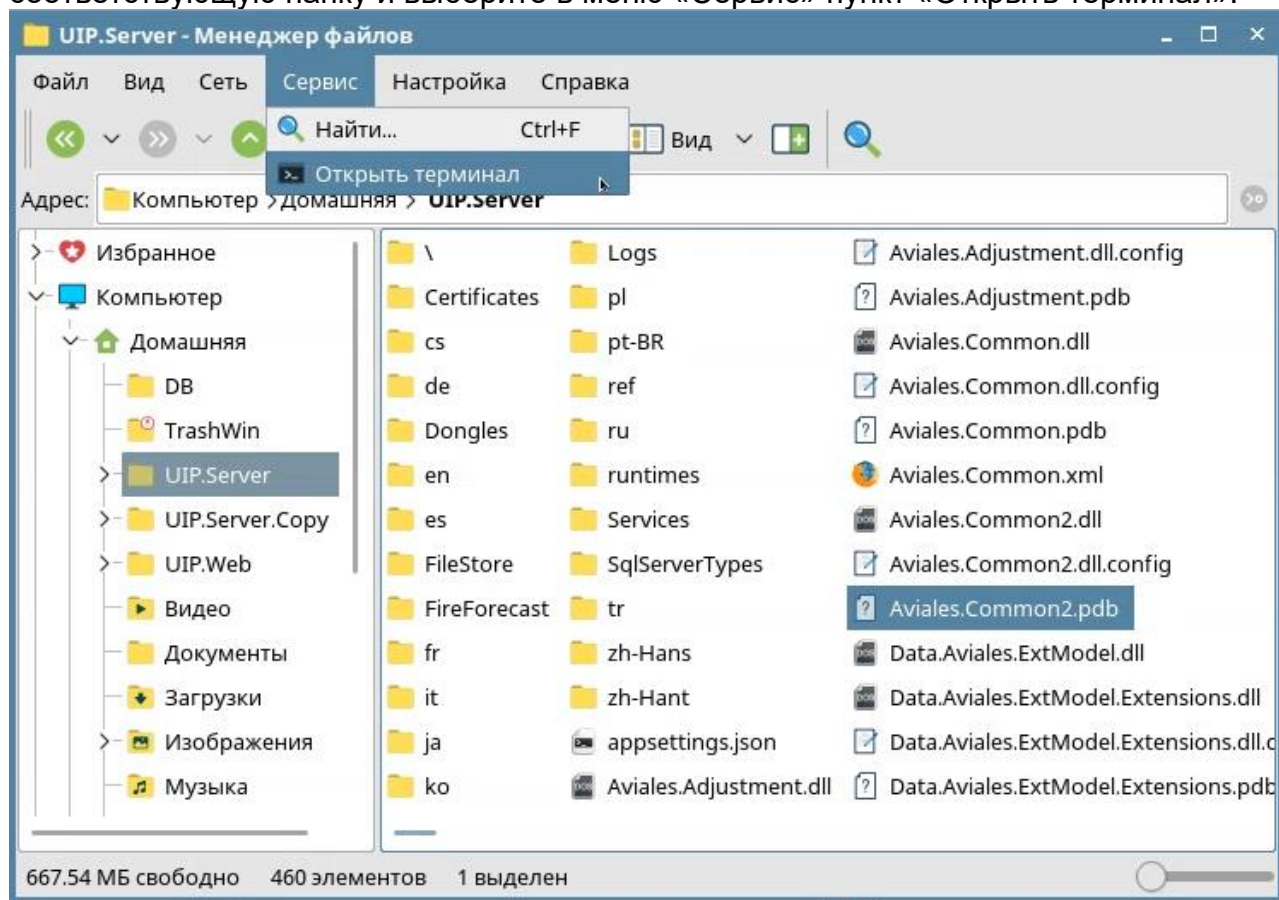


```
D:\Work\UIP.Web.publish\UIP.Web.exe
Info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5000
Info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:5001
Info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Production
Info: Microsoft.Hosting.Lifetime[0]
  Content root path: D:\Work\UIP.Web.publish
```

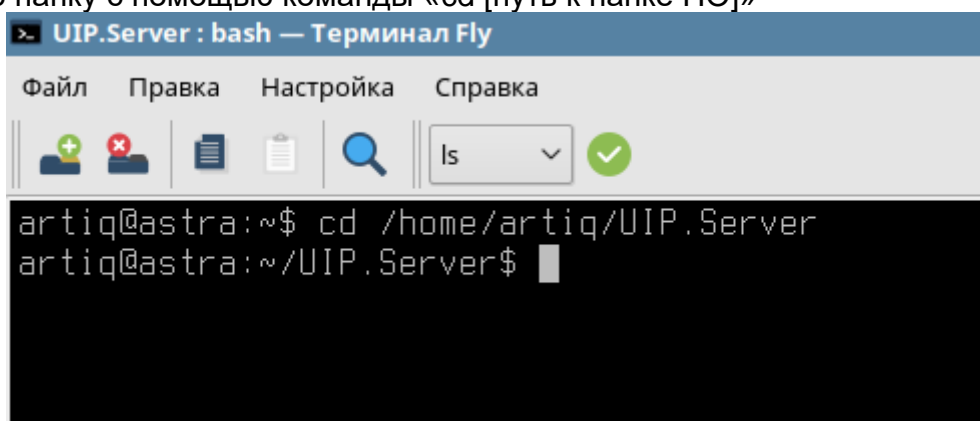
Если всё установлено и сконфигурировано корректно, то в открытом консольном будет присутствовать строка «Application started. Press Ctrl+C to shut down.»

### 3.2 Запуск ПО в AstraLinux

Для запуска ПО под AstraLinux нужно открыть по окну терминала для каждой папки куда разархивировано ПО. Для этого в менеджере файлов перейдите в соответствующую папку и выберите в меню «Сервис» пункт «Открыть терминал».



Также можно открыть терминал из меню приложений системы и перейти в нужную папку с помощью команды «cd [путь к папке ПО]»



Для запуска Сервера в соответствующем терминале нужно выполнить команду «dotnet UIP.Server.dll»

```
ServiceManager: Сервис 'FileStoreStreamProcessorService' успешно запущен.  
ServiceManager: Сервис 'GrpcChannelService' успешно запущен.  
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' начал запускаться ас  
ServiceManager: Сервис 'ScheduledTasksRegisterService' успешно запущен.  
ServiceManager: Сервис 'ExternalSystemFilesStreamProcessorService' успешно запущен.  
ServiceManager: Сервис 'OnlineDataService' успешно запущен.  
ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.  
SOAPHostManager: Now listening on: http://localhost:2123  
SOAPHostManager: Application started. Press Ctrl+C to shut down.  
SOAPHostManager: Hosting environment: Production  
SOAPHostManager: Content root path: /home/artiq/UIP.Server  
ServiceManager: Извещение о завершении запуска асинхронных сервисов разослано за 0,573с
```

Если всё установлено и сконфигурировано корректно, то в последних строках журнала можно увидеть строку «ServiceManager: Проверка состояния сервисов прошла успешно. Ошибок не обнаружено.»

Для запуска Сервера в соответствующем терминале нужно выполнить команду «dotnet UIP.Web.dll»

```
root@astra:/home/artiq/UIP.Web# dotnet UIP.Web.dll  
info: Microsoft.Hosting.Lifetime[14]  
Now listening on: http://localhost:5000  
info: Microsoft.Hosting.Lifetime[0]  
Application started. Press Ctrl+C to shut down.  
info: Microsoft.Hosting.Lifetime[0]  
Hosting environment: Production  
info: Microsoft.Hosting.Lifetime[0]  
Content root path: /home/artiq/UIP.Web
```

Если всё установлено и сконфигурировано корректно, то в открытом консольном будет присутствовать строка «Application started. Press Ctrl+C to shut down.»

После запуска приложений в браузере можно будет открыть сайт по адресу <http://localhost:5000> и увидеть ПО в работе.

## 4 Остановка ПО

Вне зависимости от ОС в которой запущено ПО, целесообразно сначала останавливать Web-сервер а затем Сервер. Поскольку в обоих случаях приложения работают в консолях, оба останавливаются посредством комбинации клавиш Ctrl+C.

После остановки Web-сервер выведет сообщение о скором закрытии, после чего в Windows консольное окно закроется, а в AstraLinux вернётся строка ввода в терминале.

```
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
```

После остановки Сервер, запустит процесс остановки всех запущенных сервисов, и в конце будет строка сообщающая о завершении остановки.

```
ServiceManager: Сервис 'MService' деинициализирован.
ServiceManager: Сервис 'HandshakeService' деинициализирован.
ServiceManager: Сервис 'ReportsStorageService' деинициализирован.
ServiceManager: Для сервиса 'ReportsStorageService' освобождены ресурсы.
ServiceManager: Сервис 'FireForecastParametersService' деинициализирован.
ServiceManager: Сервис 'InvokerService' деинициализирован.
ServiceManager: Сервис 'SOAPHostManagerService' деинициализирован.
ServiceManager: Для сервиса 'SOAPHostManagerService' освобождены ресурсы.
ServiceManager: Сервис 'MonitorCacheService' деинициализирован.
ServiceManager: Для сервиса 'MonitorCacheService' освобождены ресурсы.
ApplicationServer: Сервер остановлен.
```

В отличие от Web-сервера, Сервер после остановки сервисов, уже не может использоваться для работы ПО, но остаётся открытым, и повторное нажатие сочетания клавиш Ctrl+C останавливает его окончательно, также либо закрывая консольное окно, либо возвращая в строку ввода терминала.



## 5 Конфигурирование параметров соединения с БД

В папке с файлами Сервера найдите файл `appsettings.json` и откройте его в любом текстовом редакторе. В файле содержится единственный параметр — строка подключения к БД.

```
{  
  "ConnectionStrings": {  
    "Default": "host=localhost;port=5432;database=ui;username=postgres;password=postgres;"  
  }  
}
```

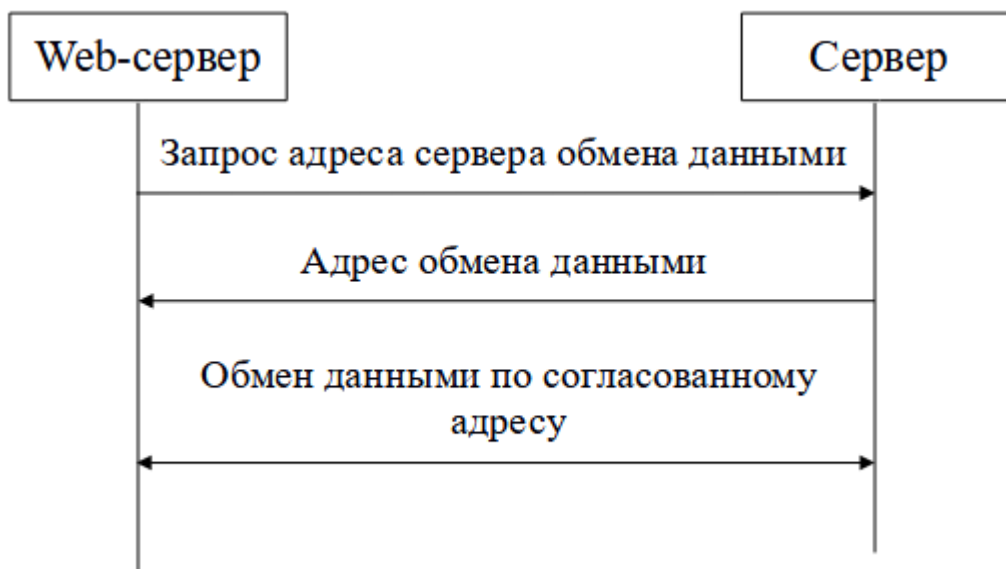
Если вы устанавливали PostgreSQL на этом же компьютере тогда в строке подключения нужно заменить параметры `database={название созданной БД}` и пароль пользователя `password={заданный при установке пароль}`.

Если вы решили использовать другой сервер PostgreSQL то укажите его параметры.

## 6 Настройка связи между модулями

Если Сервер и Web-сервер запускаются на одном компьютере, то необходимости настраивать что-то дополнительно нет. Однако, само по себе разделение ПО на модули, подразумевает возможность установки их на разные ЭВМ ради безопасности Сервера и данных в БД. В этом случае следует согласовать конфигурацию сервера и сетевой безопасности.

В общем виде взаимодействие Сервера и Web-сервера выглядит так:



Со стороны Сервера конфигурация удалённого доступа настраивается в файле «{Папка Сервера}\Services\System\GrpcHostManager.service.config», который выглядит так:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <service
    name="GrpcHostManager"
    assemblyName="UIP.GrpcChannel.Server"
    className="UIP.GrpcChannel.Server.Service.GrpcHostManagerService">
    <parameter name="ServiceUrl" value="localhost"/>
    <parameter name="GrpcPorts" value="2778,2779"/>
    <parameter name="HttpPort" value="3777"/>
    <dependOn name="LoggerService"/>
    <dependOn name="PreRequirementsService"/>
  </service>

  <service
    name="HandshakeService"
    assemblyName="UIP.GrpcChannel.Server"
    className="UIP.GrpcChannel.Server.Service.HandshakeService">
    <parameter name="Address" value="localhost"/>
    <parameter name="Port" value="2777"/>
  </service>
</configuration>
```

```

        <parameter name="GrpcHostManagerName" value="GrpcHostManager"/>
        <parameter name="ExternalAddresses" value=""/>
        <dependOn name="GrpcHostManager"/>
        <dependOn name="LoggerService"/>
        <dependOn name="PreRequirementsService"/>
    </service>
</configuration>

```

Выше описана конфигурация двух сервисов внутри Сервера:

- GrpcHostManager – сервис определяющий параметры соединения к Серверу для подключения и обеспечивает взаимодействие с Web-сервером
- HandshakeService – сервис «рукопожатия» предоставляет публичный адрес на который соединяется Web-сервер, после чего получает уже адрес обмена данными

Актуальные параметры GrpcHostManager:

- ServiceUrl – адрес по которому может доступен Сервер для обмена данными
- GrpcPorts – порты, через запятую, которые будут добавляться к адресу в ServiceUrl при отправке данных о подключении после рукопожатия

Актуальные параметры HandshakeService:

- Address – сетевой адрес компьютера для публикации, актуально заполнять если на ЭВМ с запускаемым Сервером более одного сетевого интерфейса
- Port – порт по которому Web-сервер сможет запросить адрес для обмена данными

Конфигурация Web-сервера находится в файле «{папка Web-сервера\appsettings.json}», который выглядит так:

```

{
  "ServerChannel": {
    "handshakeAddress": "localhost",
    "handshakePort": 2777
  },
  "SystemService": {
    "systemServiceLogin": "systemService",
    "systemServicePassword": "*****"
  },
  "SiteName": "УИП",
  "YandexSearch": {
    "YandexApiKey": "00000000-0000-0000-0000-000000000000"
  },
  "TempFile": {
    "Path": "tempFiles"
  },
  "NLog": {
    "Targets": [

```

```

        {
            "Name": "AllLog",
            "FileName": "D:/Logs/WebLogs/UIP/${shortdate}.txt",
            "Layout": "${date:format=HH\\:mm\\:ss\\.fff}
${message} ${exception:format=tostring}"
        }
    ],
    "Rules": [
        {
            "Name": "nlog",
            "Level": "Error",
            "WriteTo": "AllLog"
        },
        {
            "Name": "nlog",
            "Level": "Warn",
            "WriteTo": "AllLog"
        },
        {
            "Name": "nlog",
            "Level": "Info",
            "WriteTo": "AllLog"
        },
        {
            "Name": "nlog",
            "Level": "Debug",
            "WriteTo": "AllLog"
        }
    ]
},
"Logging": {
    "LogLevel": {
        "Default": "Information",
        "Microsoft": "Warning",
        "Microsoft.Hosting.Lifetime": "Information"
    }
},
"GeoProxy": {
    "BasicPassword": ""
},
"AllowedHosts": "*"
}

```

Параметры соединения с Сервером содержатся в секции «ServiceChannel»:

- handshakeAddress – адрес Сервера
- handshakePort – порт HandshakeService Сервера

Следует также учитывать параметр «FileName» секции «Nlog», в нём указывается путь к файлу(ам) журналов Web-Сервера. Нужно указать актуальный путь с учётом используемой ОС и наличия дисков. Если указать путь к конкретному файлу, будет записываться один файл журналов, если заменить только часть «D:/Logs/WebLogs/UIP/» то на каждый день работы Web-сервера будет формироваться отдельный файл.