# Техническое задание на программу для ЭВМ «Сервис обновлений» версия 1.

# 1. Описание

Программа представляет собой web-сервер для предоставления клиентским приложениям наборов обновлений. Клиентские приложения через web-api запрашивают информацию об имеющихся обновлениях и по запросу получают ссылку на скачивание архива.

Так же реализуется ARM приложения, позволяющий добавлять клиентские приложения и наборы обновлений.

# 2. Логика работы

Перед началом работы необходимо в базу клиентов добавить нового. Для клиента задается: наименование, описание, логин – пароль, каталог хранения обновлений. После добавления клиента он может начинать отправлять запросы обновления.

Первым запросом обязательно идет сообщения авторизации. В ответ сервер возвращает токен, с помощью которого уже можно отправлять запросы об обновлениях. Срок жизни токена определяется в настройках.

После авторизации клиент отправляет запрос, который может содержать текущую версию клиента (тогда в ответ будут выгружены только более новые пакеты), архитектуру ОС хоста приложения (в этом случае в ответ будут выгружены только соответствующие пакеты).

После получения ответа клиент может сделать запрос на загрузку необходимого пакета обновлений (по id).

ARM сервера позволяет добавлять новых клиентов и пакеты обновлений.

# 3. Требования

Сервер работает по протоколу http(s), должен обрабатывать не менее 300 запросов проверки обновлений в секунду.

Доступ к данным о пакетах обновлений, обновлениях и доступ в ARM только авторизованным пользователям (клиентам). Ограничение авторизаций клиента: не более трёх неудачных попыток за 10 минут. Уведомление о попытках неудачных авторизаций после десяти подряд.

Доступ в ARM ограничен в первой версии только одной учетной записью администратора сервера. Роли в первой версии не предусмотрены.

# 4. Технологии

## 4.1. Сервер

* EntityFramework
* ASP .NetCore
* .NetCore 3.x

## 4.2. Базы данных

* PostgreSQL

## 4.3. Тестирование

* XUnit
* Moq

## 4.4. Web-client

* ASP .NetCore
* ReactJS или AngularJS
* Html
* Css
* JQuery

## 4.5. Установщик

* WixInstaller

# 5. Лицензирование

Продукт распространяется по лицензии Apache 2.0.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object

form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain

separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including

the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright 2021 Dmitriy Rokoth

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Наиболее современная и сбалансированная из разрешительных лицензий. Написана человеческим языком, но с оглядкой на современное правоприменение, в частности, упомянутые выше патентные отношения (пункт 3 лицензии). GNU советуют применять именно эту лицензию, когда вам необходима разрешительная лицензия.

Для применения лицензии Apache 2.0 к вашему проекту, нужно добавить в него файл LICENSE, содержащий текст лицензии. Кроме того, в APPENDIX лицензии нам предлагают добавлять в качестве шапки в каждый файл исходного кода следующий текст:

Copyright 2021 Dmitriy Rokoth

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Но при этом сама лицензия выдвигает следующие требования:

made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below)

copyright notice — это как раз строка, указывающая правообладателя. А «made available under the License, as indicated» означает, что еще должна быть явно указана лицензия. То есть, допустимо что-то вида:

///Copyright 2021 Dmitriy Rokoth

///Licensed under the Apache License, Version 2.0

Причем, совсем необязательно в исходном коде — Apache 2.0 позволяет для этого использовать файл NOTICE («or attached to the work»).

И еще о файле NOTICE: если в вашей работе вы используете чужой проект под лицензией Apache 2.0, содержащий свой файл NOTICE, то в этом случае вы обязаны копировать в производную работу содержимое файла NOTICE, в одно из трех мест: либо в аналогичный файл NOTICE, либо в исходные коды или документацию, распространяемую вместе с производной работой, либо в вывод производной работы (например в about-диалог); все согласно пункту 4 (d) лицензии. Заметьте, что, вопреки расхожему мнению, обязательного наличия файла NOTICE лицензия не требует.

При распространении в бинарной форме, вы, кроме того, должны предоставлять копию лицензии вместе с программой.

# 6. Версионирование

Исходные коды хранятся на Github. Очередной релиз выпускается после окончания всех работ по очередному ТЗ и прохождения всех тестов. Мажорная версия выпускается если доработки не содержат обратной совместимости. Иначе выпускается минорная версия. Для исправлений и мелких доработок выпускается патч.

Нумерация:

<Мажорная версия>.<минорная версия>.<Патч>

# 7. Модели данных

## 7.1. Клиентские модели

### 7.1.1. Базовые классы

#### 7.1.1.1. Entity

***Entity***

***{***

***public Guid Id;***

***}***

#### 7.1.1.2. EntityHistory

***EntityHistory***

***{***

***public Guid Id;***

***public long Hid;***

***public DateTimeOffset ChangeDate;***

***public bool IsDeleted;***

***}***

### 7.1.2. Пользователи

#### 7.1.2.1. User

***User: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***}***

#### 7.1.2.2. UserUpdater

***UserUpdater***

***{***

***Public Guid Id;***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.1.2.3. UserHistory

***UserHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***}***

#### 7.1.2.4. UserCreator

***UserCreator***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

### 7.1.3. Клиент

#### 7.1.3.2. Client

***Client: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string BasePath;***

***public Guid UserId;***

***}***

### 7.1.5. История загрузок

#### 7.1.5.1. LoadHistory

***LoadHistory: Entity***

***{***

***public Guid ClientId;***

***public Guid ReleaseId;***

***public Guid ArchitectId;***

***public DateTime LoadDate;***

***public bool Success;***

***}***

#### 7.1.5.1. LoadHistoryCretor

***LoadHistoryCreator***

***{***

***public Guid ClientId;***

***public Guid ReleaseId;***

***public Guid ArchitectId;***

***public DateTime LoadDate;***

***public bool Success;***

***}***

### 7.1.6. Архитектуры обновления

#### 7.1.6.1. ReleaseArchitect

***ReleaseArchitect: Entity***

***{***

***public Guid ReleaseId;***

***public string Name;***

***public string Path;***

***}***

#### 7.1.6.2. ReleaseArchitectCreator

***ReleaseArchitectCreator***

***{***

***public Guid ReleaseId;***

***public string Name;***

***public string Path;***

***}***

#### 7.1.6.3. ReleaseArchitectHistory

***ReleaseArchitectHistory: EntityHistory***

***{***

***public Guid ReleaseId;***

***public string Name;***

***public string Path;***

***}***

#### 7.1.6.2. ReleaseArchitectUpdater

***ReleaseArchitectUpdater***

***{***

***public Guid Id;***

***public Guid ReleaseId;***

***public string Name;***

***public string Path;***

***}***

### 7.1.7. Обновления

#### 7.1.7.1. ReleaseClient

***ReleaseClient***

***{***

***public Guid Id;***

***public string Version;***

***public List<ReleaseArchitectClient> Architects;***

***}***

## 7.2. Серверные модели

### 7.2.2. Пользователь

#### 7.2.2.1. User

***User: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.2.2.2. UserHistory

***UserHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

### 7.2.4. Клиент

#### 7.2.4.1. Client

***Client: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***public string BasePath;***

***public Guid UserId;***

***}***

#### 7.2.4.2. ClientHistory

***ClientHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***public string BasePath;***

***public Guid UserId;***

***}***

### 7.2.5. Обновление

#### 7.2.4.1. Release

***public class Release : Entity***

***{***

***public string Path ;***

***public string Version ;***

***public int Number;***

***public Client Client ;***

***public Guid ClientId ;***

***}***

#### 7.2.4.2. ReleaseHistory

***public class ReleaseHistory : EntityHistory***

***{***

***public string Path ;***

***public string Version ;***

***public int Number;***

***public Client Client ;***

***public Guid ClientId ;***

***}***

### 7.2.6. История загрузок

#### 7.2.6.1. LoadHistory

***LoadHistory: Entity***

***{***

***public Guid ClientId;***

***public Guid ReleaseId;***

***public Guid ArchitectId;***

***public DateTime LoadDate;***

***public bool Success;***

***}***

### 7.2.7. Архитектуры обновления

#### 7.2.7.1. ReleaseArchitect

***ReleaseArchitect: Entity***

***{***

***public Guid ReleaseId;***

***public string Name;***

***public string Path;***

***}***

## 7.3. Модели базы данных

### 7.3.1. client

***client:***

***id uuid not null primary key,***

***name varchar(100) not null,***

***description varchar(1000) null,***

***login varchar(100) not null,***

***password varchar(100) not null,***

***base\_path varchar(1000) not null,***

***version\_date timestamptz not null,***

***is\_deleted boolean not null default false,***

***userid uuid not null,***

***constraint fk\_client\_user\_id foreign key user\_id on user,***

***unique index (login) where not is\_deleted***

### 7.3.2. user

***user:***

***id uuid not null primary key,***

***name varchar(100) not null,***

***description varchar(1000) null,***

***login varchar(100) not null,***

***password varchar(100) not null,***

***version\_date timestamptz not null,***

***is\_deleted boolean not null default false,***

***unique index (login) where not is\_deleted***

### 7.3.3. h\_user

***h\_user:***

***h\_id long not null primary key,***

***id uuid null,***

***name varchar(100) null,***

***description varchar(1000),***

***login varchar(100) null,***

***password varchar(100) null,***

***version\_date timestamptz null,***

***is\_deleted boolean null,***

***change\_date timestamptz not null,***

***user\_id uuid null,***

***constraint idx\_ h\_user \_id on table h\_user (id),***

***constraint idx\_ h\_user \_ change\_date on table h\_user (change\_date),***

### 7.3.4. h\_client

***h\_client:***

***h\_id long not null primary key,***

***id uuid null,***

***name varchar(100) null,***

***description varchar(1000) null,***

***login varchar(100) null,***

***password varchar(100) null,***

***base\_path varchar(1000) null,***

***version\_date timestamptz null,***

***is\_deleted boolean,***

***userid uuid null,***

***is\_deleted boolean null,***

***change\_date timestamptz not null,***

***user\_id uuid null,***

***constraint idx\_ h\_client \_id on table h\_client (id),***

***constraint idx\_ h\_client \_ change\_date on table h\_client (change\_date),***

### 7.3.5. release\_architect

***release\_architect:***

***id uuid not null primary key***

***, release\_id uuid not null***

***, "name" varchar not null***

***, "path" varchar not null***

***, "file\_name" varchar not null***

***, version\_date timestamptz not null***

***, is\_deleted boolean not null default false***

***, constraint fk\_release\_architect\_release\_id***

***foreign key(release\_id)***

***references release(id)***

***on delete no action on update no action;***

***create unique index uidx\_release\_architect\_name***

***on release\_architect(release\_id, "name") where not is\_deleted;***

***create unique index uidx\_release\_architect\_path***

***on release\_architect(release\_id, "path") where not is\_deleted and "path" is not null;***

***create index idx\_release\_architect\_release\_id***

***on release\_architect(release\_id);***

***create index idx\_release\_architect\_is\_deleted***

***on release\_architect(is\_deleted);***

### 7.3.6. h\_release

***h\_release:***

***h\_id long not null primary key,***

***id uuid null,***

***path varchar(1000) null,***

***version varchar(1000) null,***

***number int null,***

***client\_id uuid null,***

***version\_date timestamptz null,***

***is\_deleted boolean,***

***userid uuid null,***

***is\_deleted boolean null,***

***change\_date timestamptz not null,***

***user\_id uuid null,***

***constraint idx\_h\_release\_id on table h\_release (id),***

***constraint idx\_ h\_release \_ change\_date on table h\_release (change\_date),***

# 8. Сервер

## 8.1. Описание

Сервер представляет собой приложение, работающее в режиме службы. Предоставляет API для запросов от клиентских приложений и АРМ для настройки приложения. API требует предварительной авторизации клиента.

Под клиентом подразумевается некое приложение, в котором реализован соответствующий функционал, выполняющий обновление хостового приложения.

Три основных вызова API: авторизация, запрос списка пакетов обновлений по фильтру, скачивание нужного пакета.

АРМ позволяет добавлять новых клиентов и пользователей, редактировать их, добавлять и редактировать пакеты обновлений.

## 8.2. Сервисы

### 8.2.1. Сервис раскатки БД

Сервис представляет собой обертку над библиотекой раскатки БД Deployer. Содержит один публичный метод Deploy:

***void Deploy(int? num = null)***

где параметр num – номер апдейта, начиная с которого следует выполнить раскатку.

### 8.2.2. Сервис работы с данными

#### 8.2.2.1. Авторизация пользователя

Метод принимает в качестве параметра логин и пароль пользователя, проверяет на соответствие, в случае успеха возвращает объект ClaimsIdentity для куки-идентификации. Сигнатура:

***ClaimsIdentity Auth(UserIdentity login)***

#### 8.2.2.2. Авторизация клиента

Метод принимает в качестве параметра логин и пароль клиента, проверяет на соответствие, в случае успеха возвращает объект ClaimsIdentity для идентификации с помощью токена. Сигнатура:

***ClaimsIdentity Auth(ClientIdentity login)***

#### 8.2.2.3. Получение списка элементов с постраничной отдачей

Метод принимает в качестве параметра экземпляр фильтра данной модели, формирует фильтр в репозиторий, делает запрос, полученный результат маппит в клиентскую модель, дообогащает при необходимости. Сигнатура:

***PagedResult<T> Get(Filter<T> filter)***

#### 8.2.2.4. Получение элемента

Метод принимает в качестве параметра id модели, делает запрос в репозиторий, полученный результат маппит в клиентскую модель, дообогащает при необходимости. Сигнатура:

***T Get(Guid id)***

#### 8.2.2.5. Обновление элемента

Метод принимает в качестве параметра клиентскую модель обновления (TUpdater), маппит в модель БД, делает запрос в репозиторий, полученный результат маппит в клиентскую модель, дообогащает при необходимости. Сигнатура:

***T Update(TUpdater updater)***

#### 8.2.2.6. Добавление элемента

Метод принимает в качестве параметра клиентскую модель создания (TCreator), маппит в модель БД, делает запрос в репозиторий, полученный результат маппит в клиентскую модель, дообогащает при необходимости. Сигнатура:

***T Add(TCreator creator)***

## 8.3. API

### 8.3.1. Авторизация

Метод принимает параметрами логин и пароль клиента. В случае удачной авторизации возвращает токен. Сигнатура:

***POST /api/v1/client/auth***

***Body:***

***{***

***“Login”: string***

***“Password”: string***

***}***

### 8.3.2. Получение списка обновлений

Метод принимает параметрами номер текущей версии клиента. Возвращает список доступных версий выше версии клиента (с архитектурами). Сигнатура:

***GET /api/v1/release?[version=null]***

# 9. Клиент

## 9.1. Описание

Клиент представляет собой службу, работающее на стороне обновляемого клиентского приложения, устанавливается вместе с приложением в один каталог. В фоновом режиме служба проверяет наличие обновлений, при наличии обновлений в зависимости от конфигурации либо уведомляет о наличии обновления, либо устанавливает обновление автоматически (сразу же, либо по расписанию).

Процесс установки обновления:

- бекап бинарников и (в случае соответствующей настройки ) базы данных;

- остановка приложения (службы), проверка, что все связанные процессы были завершены;

- установка обновлений в базу данных;

- запуск инсталлятора либо замена бинарников;

- модификация файла конфигурации;

- выполнение других действий, заданных в скрипте обновления;

- запуск приложения;

- в случае любой ошибки при установке либо критических ошибок после запуска приложения производится откат базы и приложения и уведомление об ошибке установки в сервис сбора ошибок.

Самообновление производится отдельно, в автоматическом режиме: приложение копирует себя в отдельный каталог (temp), устанавливает необходимую конфигурацию и запускает. Запущенный процесс останавливает службу обновления, скачивает и устанавливает свежую версию, после чего запускает службу и завершает работу.

## 9.2. HttpClient

### 9.2.1. Описание

Библиотека реализует методы http-запросов к серверу: авторизация, пинг, запрос списка обновлений, загрузка выбранного обновления.

### 9.2.2. Ping

Сервис с некоторой периодичностью вызывает метод ping api сервера. В случае удачного вызова выставляется флаг доступности = true, уоторый разрешает вызов других методов api, в противном случае флаг выставляется в false.

## 9.3. Сервисы

### 9.3.1. Сервис резервного сохранения

Сервис выполняет резервное копирование в установленный каталог рабочей (текущей) версии приложения и делает внеочередной бекап базы данных. Вызывается после остановки приложения.

### 9.3.2. Сервис создания локальной базы данных

Сервис создаёт локальную базу данных и заполняет её данными по умолчанию. Запускается автоматически при первом запуске сервиса и при обновлении сервиса (при обновлении выполняется обновление существующей базы).

### 9.3.3. Сервис самообновления

Сервис выполняет подготовку бинарников к обновлению сервиса обновления и запускает сервис самообновления (себя же в режиме самообновления – не запускаются назначенные задания, только сервис самообновления). Запущенная программа останавливает сервис обновления, обновляет бинарники и локальную базу, запускает сервис обновления, проверяет, что служба работает и после этого останавливается.

### 9.3.4. Сервис обновления

Сервис выполняет обновление бинарников обслуживаемого сервиса в одном из режимов (в первой версии: режим замены бинарников, режим запуска инсталлятора). В случае успеха возвращается true, сбоя – false.

### 9.3.5. Сервис отката обновления

Сервис выполняет откат базы данных и бинарников из резервного каталога. В случае успеха возвращается true, сбоя – false.

## 9.4. Скрипты обновления

### 9.4.2. Скрипт самообновления

Представляет собой частный случай скрипта обновления. Поддерживаются те же конструкции, методы и переменные. Предназначается для настройки процесса самообновления клиентского приложения «Сервис обновлений», с добавлением команды переключения на временное приложение.

### 9.4.3. Парсер скриптов обновления

Выполняет разбор скрипта обновления. Предполагает следующую командную строку скрипта:

<идентификатор строки>: <команда> <аргументы> <условие>

Где:

<идентификатор строки> - уникальный номер команды

<команда> - одно из Backup, Install, Rollback, CMD, Stop, Start

<аргументы> - набор аргументов (для команды CMD)

<условие> - условие выполнения, может содержать номера предыдущих команд, литералы not, or, and и скобки

# 10. Web Клиент

## 10.1. Описание

Представляет собой ARM для управления клиентами и обновлениями.

Функции:

- Просмотр, добавление, редактирование, удаление клиентов

- Просмотр, добавление, редактирование, удаление наборов обновлений (возможность разделения на архитектуры)

- Просмотр истории загрузок обновлений

## 10.2. Формы

### 10.2.1. Главная

#### 10.2.1.1. Управляющие элементы

- Начать работу

- Авторизация

- Развернуть/обновить базу

- О программе

- Отправить сообщение об ошибке

#### 10.2.1.2. Логика

На главной странице несколько управляющих элементов – кнопок и логотип приложения.

По кнопке «Начать работу» открывается окно клиентов текущего пользователя

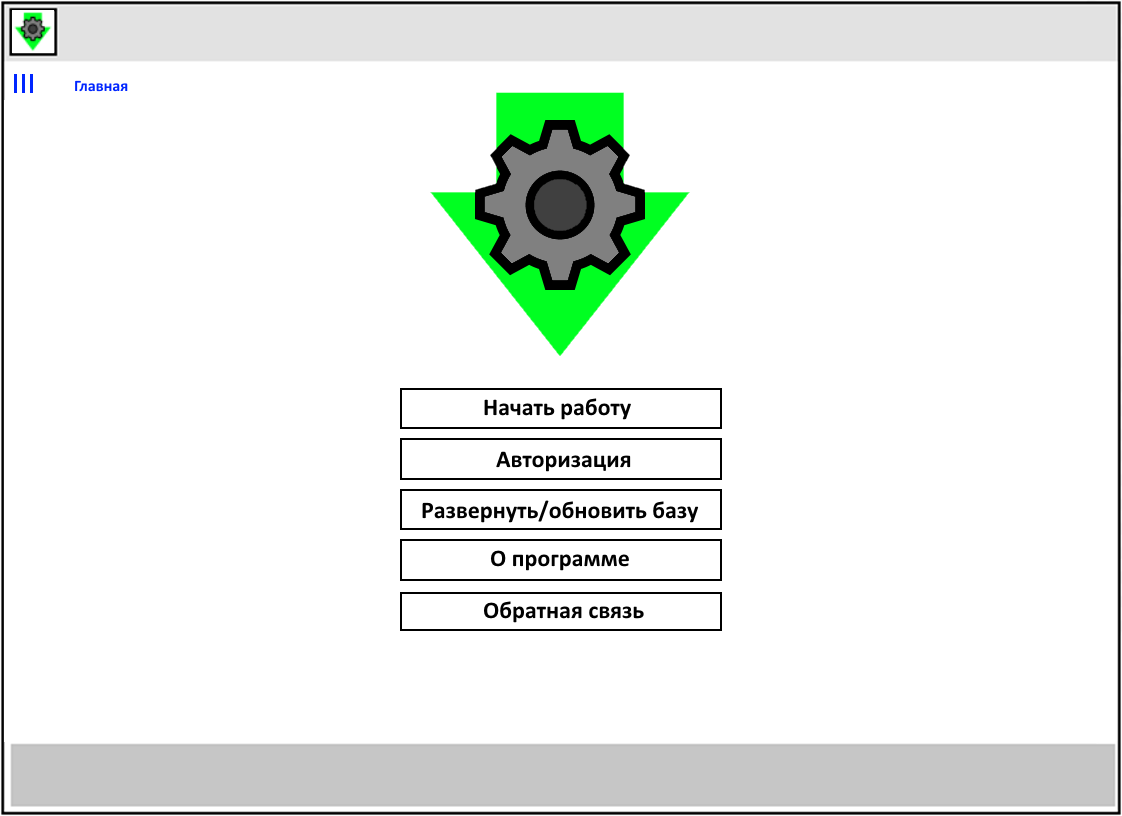
По кнопке «Авторизация» открывается окно авторизации (если пользователь еще не авторизован, переход осуществляется автоматически при открытии любой страницы работы с данными).

По кнопке «Развернуть/обновить базу данных» открывается окно обновления базы данных, доступно только администратору.

По кнопке «О программе» открывается окно описания приложения

По кнопке «Отправить сообщение об ошибке» открывается окно обратной связи

#### 10.2.1.3. Дизайн



### 10.2.2. Авторизация

#### 10.2.2.1. Управляющие элементы

- Кнопка «Войти»

- Кнопка «Вернуться»

#### 10.2.2.2. Логика

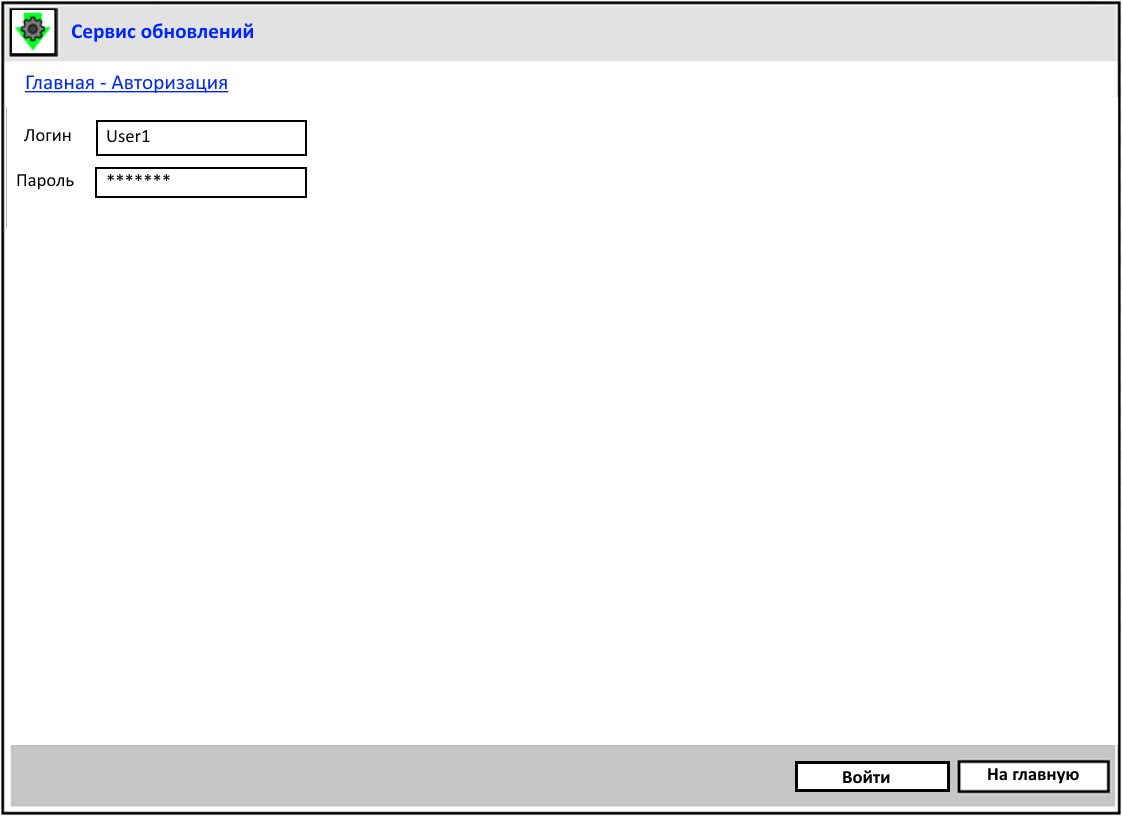
Форма предназначена для авторизации пользователей (не клиентов). После ввода корректных данных авторизации страница перенаправляется на страницу, с которой произошел запрос авторизации.

#### 10.2.2.3. Поля

- Логин – текстовое поле

- Пароль – поле типа password

#### 10.2.2.4. Дизайн



# 11. Установщик

## 11.1. Сервер Windows

Для построения установщика используется WixInstaller. Содержит следующие окна:

* Описание
* Лицензионное соглашение
* Выбор папки установки
* Выбор поставщика базы данных
* Подключение к базе данных
* Выбор режима работы (служба/приложение)
* Установка
* Выход

## 11.2. Сервер Linux

Приложение поставляется в виде архива tar.gz. Конфигурируется вручную, описание настройки и лицензионное соглашение располагаются в файле readme.

## 11.3. Docker

Приложение поставляется в виде контейнера. Конфигурируется вручную, описание настройки и лицензионное соглашение располагаются в файле readme.

# 12. Тестирование

## 12.1. Интеграционное

### 12.1.1. API

#### 12.1.1.1. Сценарий 1

Проверяется работа авторизации (положительный сценарий).

Последовательность теста:

- создание окружения

- добавление в базу пользователя, клиента

- отправка данных авторизации

- проверка что получен положительный ответ

- проверка, что сообщения принимаются сервером

#### 12.1.1.2. Сценарий 2

Проверяется работа авторизации (отрицательный сценарий).

Последовательность теста:

- создание окружения

- добавление в базу пользователя, клиента

- отправка неверных данных авторизации

- проверка что получен отрицательный ответ

- проверка, что запросы на получение обновлений не принимаются сервером

## 12.2. Модульное

### 12.2.1. API

#### 12.2.1.1. Auth

Проверяется работа метода Auth контроллера CommonController.

Последовательность теста:

- Создание окружения

- Добавление пользователя, клиента

- Вызов метода с логином и паролем клиента

- Проверка, что метод вернул токен

### 12.2.2. Сервисы

#### 12.2.2.1. AddUser

Проверяется работа метода добавления пользователя. Последовательность теста:

* Создание окружения
* Добавление пользователя через метод AddГыук
* Получение пользователя через репозиторий
* Проверка на соответствие полученного пользователя базовому

### 12.2.3. Репозитории

#### 12.2.3.1. Get

Проверяется работа метода Get репозитория для любой из моделей. Последовательность теста:

* Создание окружения
* Добавление нескольких сущностей через контекст
* Получение сущностей по фильтру через репозиторий
* Проверка на соответствие полученных сущностей фильтру