# Техническое задание на программу для ЭВМ «Агрегатор сообщений об ошибках» версия 1.

# 1. Описание

Программа представляет собой web-сервер для сбора сообщений из разных источников (в первой версии реализуется только web-api). После получения сообщения программа преобразует его, используя соответствующее выражение для источника сообщения и помещает в базу данных.

Для каждого из источников и уровня сообщения можно задать формирование уведомления о получении сообщения.

Реализован web-интерфейс для просмотра сообщений и управления сервером.

# 2. Логика работы

Перед началом работы необходимо в базу клиентов добавить нового. Для клиента задается: наименование, описание, логин – пароль, опционально – формат сообщения и логика маппинга во внутреннюю структуру сообщений. После добавления клиента он может начинать отправлять сообщения.

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

Сервер принимает сообщения, маппит их во внутреннюю структуру, сохраняет в базу данных, опционально отправляет уведомление о получении.

ARM сервера позволяет обрабатывать сообщения (перемещать по этапам обработки: «Новый», «В работе», «Выполнено», «Отложено», «В архив»), добавлять заметки о процессе обработки (дата, название, описание), добавлять и изменять клиентов, изменять настройки сервера.

# 3. Требования

Сервер в первой версии принимает сообщения только через web-интерфейс.

Сервер должен обрабатывать не менее 300 запросов в секунду.

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 [yyyy] [name of copyright owner]

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 [yyyy] [name of copyright owner]

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. Базовый класс

#### 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. UserHistory

***UserHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***}***

#### 7.1.2.3. UserUpdater

***User***

***{***

***public Guid Id;***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.1.2.3. UserCreator

***User***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

### 7.1.3. Клиент

#### 7.1.3.1. ClientCreator

***ClientCreator***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***public string MappingRules;***

***public Guid UserId;***

***}***

#### 7.1.3.2. Client

***Client: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string MappingRules;***

***public Guid UserId;***

***}***

#### 7.1.3.3. ClientHistory

***ClientHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string MappingRules;***

***public Guid UserId;***

***}***

#### 7.1.3.4. ClientUpdater

***ClientUpdater***

***{***

***public Guid Id;***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***public string MappingRules;***

***public Guid UserId;***

***}***

### 7.1.4. Статус сообщения

#### 7.1.4.1. MessageStatus

***MessageStatus: Entity***

***{***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

#### 7.1.4.2. MessageStatusCreator

***MessageStatusCreator***

***{***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

#### 7.1.4.3. MessageStatusUpdater

***MessageStatusUpdater***

***{***

***public Guid Id;***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

#### 7.1.4.4. MessageStatusHistory

***MessageStatusHistory: EntityHistory***

***{***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

### 7.1.5. Сообщение

#### 7.1.5.1. Message

***Message: Entity***

***{***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public string AddFileds;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

}

#### 7.1.5.2. MessageHistory

***MessageHistory: EntityHistory***

***{***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public string AddFileds;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

}

#### 7.1.5.3. MessageCreator

***MessageCreator***

***{***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public string AddFileds;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

}

#### 7.1.5.3. MessageUpdater

***MessageUpdater***

***{***

***public Guid Id;***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public string AddFileds;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

}

### 7.1.6. Уведомление

#### 7.1.6.1. Notification

***Notification***

***{***

***public string NotificationType;***

***public string Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public string AddFileds;***

***public string Client;***

***public DateTimeOffset CreatedDate;***

}

## 7.2. Серверные модели

### 7.2.1. Базовый класс

#### 7.2.1.1. Entity

***Entity***

***{***

***public Guid Id;***

***public DateTimeOffset VersionDate;***

***public bool IsDeleted;***

***}***

#### 7.2.1.2. EntityHistory

***EntityHistory***

***{***

***public long Hid;***

***public Guid Id;***

***public DateTimeOffset VersionDate;***

***public bool IsDeleted;***

***public DateTimeOffset ChangeDate;***

***}***

### 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;***

***}***

### 7.2.3. Сообщение

#### 7.2.3.1. Message

***Message: Entity***

***{***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

***public string AddFileds;***

***}***

#### 7.2.3.2. MessageHistory

***MessageHistory: EntityHistory***

***{***

***public int Level;***

***public string Title;***

***public string Description;***

***public string FeedbackContact;***

***public Guid ClientId;***

***public DateTimeOffset CreatedDate;***

***public string AddFileds;***

***}***

### 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 MappingRules;***

***public Guid UserId;***

***}***

#### 7.2.4.2. ClientHistory

***ClientHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***public string MappingRules;***

***public Guid UserId;***

***}***

### 7.2.5. Статус сообщения

#### 7.2.5.1. MessageStatus

***MessageStatus: Entity***

***{***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

#### 7.2.5.2. MessageStatusHistory

***MessageStatusHistory: EntityHistory***

***{***

***public Guid MessageId;***

***public int StatusId;***

***public Guid UserId;***

***public string Description;***

***public DateTimeOffset StatusDate;***

***}***

## 7.3. Модели базы данных

### 7.3.1. client\_history

***client\_history:***

***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,***

***user\_id uuid not null,***

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

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

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

### 7.3.2. 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,***

***map\_rules varchar(1000) null,***

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

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

***user\_id uuid not null,***

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

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

### 7.3.3. 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***

### 7.3.4. user\_history

***user\_history:***

***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,***

### 7.3.5. message

***message:***

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

***level smallint not null,***

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

***description varchar not null,***

***feedback\_contact varchar not null,***

***add\_fields json null,***

***client\_id uuid not null,***

***created\_date timestamptz not null,***

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

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

***constraint fk\_message\_client\_id foreign key client\_id on user,***

### 7.3.6. message\_history

***message\_history:***

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

***id uuid,***

***level smallint null,***

***title varchar(100) null,***

***description varchar null,***

***feedback\_contact varchar null,***

***add\_fields json null,***

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

***created\_date timestamptz null,***

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

***is\_deleted Boolean null,***

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

### 7.3.7. message\_status

***message\_status:***

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

***message\_id uuid not null,***

***status\_id smallint not null,***

***description varchar not null,***

***user\_id uuid not null,***

***status\_date timestamptz not null,***

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

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

***constraint fk\_message\_status\_message\_id foreign key message\_id on message,***

***constraint fk\_message\_ status\_user\_id foreign key user\_id on user,***

### 7.3.7. message\_status\_history

***message\_status\_history:***

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

***id uuid null,***

***message\_id uuid null,***

***status\_id smallint null,***

***description varchar null,***

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

***status\_date timestamptz null,***

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

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

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

# 8. Сервер

## 8.1. Описание

Сервер представляет собой приложение, работающее в режиме службы. Предоставляет API для получения сообщений от клиентов и АРМ для настройки приложения. API требует предварительной авторизации клиента.

Под клиентом подразумевается некое приложение, в котором реализован соответствующий функционал отправки сообщений.

Сообщение должно содержать обязательные поля (уровень сообщения: ошибка, пожелание, замечание, заголовок, текст, данные для обратной связи) и дополнительные в формате json, которые будут размещены в базе в общем текстовом поле, либо у клиента в настройках должны быть заданы правила маппинга полей сообщения в поля приложения (в формате json).

АРМ позволяет добавлять новых клиентов и пользователей, редактировать их, просматривать и обрабатывать сообщения.

## 8.2. Сервисы

### 8.2.1. GetUser

Метод получает по идентификатору из базы пользователя, дообогащает при необходимости, маппит в клиентскую модель и возвращает ее.

Сигнатура:

***Task<UserModel> GetUserAsync(Guid id)***

### 8.2.2. GetUsers

Метод получает по фильтру список пользователей из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<IEnumerable<UserModel>> GetUsersAsync(Filter<User> filter)***

### 8.2.3. GetClients

Метод получает по фильтру список клиентов из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<IEnumerable<ClientModel>> GetClientsAsync(Filter<Client> filter)***

### 8.2.4. GetMessage

Метод получает по id сообщение из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<MessageModel> GetMessageAsync(Guid id)***

### 8.2.5. GetClient

Метод получает по id клиента из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<ClientModel> GetClientAsync(Guid id)***

### 8.2.6. GetMessageStatus

Метод получает по id статус сообщения из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<MessageStatusModel> GetMessageStatusAsync(Guid id)***

### 8.2.7. GetMessages

Метод получает по фильтру список cообщений из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<IEnumerable<MessageModel>> GetMessagesAsync(Filter<Message> filter)***

### 8.2.8. GetMessageStatuses

Метод получает по фильтру список статусов cообщения из БД, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<IEnumerable<MessageStatusModel>> GetMessageStatusesAsync(Filter<MessageStatus> filter)***

### 8.2.9. AddClient

Метод принимает параметром экземпляр ClientCreator, выполняет проверку уникальности, формирует хеш пароля, добавляет в базу данных нового клиента, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<ClientModel> AddClientAsync(ClientCreator creator)***

### 8.2.10. AddMessage

Метод принимает параметром экземпляр MessageCreator, добавляет в базу данных новое сообщение, отправляет уведомление, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<MessageModel> AddMessageAsync(MessageCreator creator)***

### 8.2.11. AddMessageStatus

Метод принимает параметром экземпляр MessageStatusCreator, добавляет в базу данных новый статус сообщения, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<MessageStatusModel> AddMessageStatusAsync(MessageStatusCreator creator)***

### 8.2.12. UpdateUser

Метод принимает параметром экземпляр UserUpdater, выполняет проверку уникальности, формирует хеш пароля, если он изменился, обновляет в базе данных пользователя, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<UserModel> UpdateUserAsync(UserUpdater updater)***

### 8.2.13. AddUser

Метод принимает параметром экземпляр UserCreator, выполняет проверку уникальности, формирует хеш пароля, добавляет в базу данных нового пользователя, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<UserModel> AddUserAsync(UserCreator creator)***

### 8.2.14. UpdateMessage

Метод принимает параметром экземпляр MessageUpdater, обновляет в базе данных сообщает, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<MessageModel> UpdateMessageAsync(MessageUpdater updater)***

### 8.2.15. UpdateClient

Метод принимает параметром экземпляр ClientUpdater, изменяет запись в базе данных, дообогащает при необходимости, маппит в клиентскую модель и возвращает.

Сигнатура:

***Task<ClientModel> UpdateClientAsync(ClientUpdater updater)***

## 8.3. API

### 8.3.1. Ping

Метод не принимает параметров. Возвращает OK если доступен. Сигнатура:

***GET /api/v1/ping/***

### 8.3.2. Авторизация

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

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

***Body:***

***{***

***“Login”: string***

***“Password”: string***

***}***

### 8.3.3. Отправка сообщения

Метод принимает параметром сообщение, в заголовке реквеста должен быть токен. Возвращает ОК либо сообщение об ошибке. Сигнатура:

***POST /api/v1/message/send***

***Body:***

***{***

***[“key”:”value”]***

***}***

Сообщение должно содержать обязательные поля (см. клиентскую модель MessageCreator), либо в настройках клиента должны быть указаны правила маппинга полей в клиентскую модель (поля регистронезависимые)

## 8.4. Репозитории

### 8.4.1. GetItem

Метод принимает параметром Id сущности. Возвращает найденную запись, если найдена, иначе null. Сигнатура:

***Task<T> GetItem<T>(Guid Id)***

### 8.4.2. Add

Добавление новой записи. При добавлении используется оптимистичная блокировка по Id. Сигнатура:

***Task<T> AddAsync(T entity, bool withSaved)***

### 8.4.3. Update

Обновление записи. При изменении используется оптимистичная блокировка по Id. Сигнатура:

***Task<T> UpdateAsync(T entity, bool withSaved)***

### 8.4.4. Get

Метод принимает параметром фильтр. Возвращает список записей, соответствующих фильтру. Сигнатура:

***Task<IEnumerable<T>> Get<T>(Filter<T> filter)***

### 8.4.5. Delete

Удаление записи из базы. Метод принимает параметром id. Возвращает удаленную запись. Сигнатура:

***Task<T> Delete<T>(Guid id)***

## 8.5. Сервис уведомлений

В первой версии сервис формирует уведомления «на лету», без сохранения в БД. Уведомления отправляются на почту. Формируются следующие уведомления:

1. Сообщение получено

2. Таймаут взятия в работу (параметр из конфига для каждого типа сообщений)

3. Таймаут обработки (параметр из конфига для каждого типа сообщений)

4. Событие просрочено (параметры из конфига для каждого типа сообщений: таймаут плюс периодичность напоминания)

# 9. Web клиент

## 9.1. Главная

### 9.1.1. Управляющие элементы

- Начать работу

- Авторизация

- Развернуть/обновить базу

- О программе

- Отправить сообщение об ошибке

### 9.1.2. Логика

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

По кнопке «Начать работу» открывается окно списка сообщений

По кнопке «Авторизация» открывается окно авторизации (если пользователь еще не авторизован, переход осуществляется автоматически).

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

### 9.1.3. Дизайн



## 9.2. Пользователи

### 9.2.1. Список

#### 9.2.1.1. Управляющие элементы

- Фильтры: кнопка «Применить»

- Пагинатор: управляющие элементы пагинатора (см. раздел «Пагинатор»)

- Кнопка «Добавить»

- Напротив каждого элемента: кнопки «Редактировать», «Удалить»

#### 9.2.1.2. Поля

- Таблица – список элементов

- Поля фильтров:

- Наименование

- Поля пагинатора:

- Количество элементов на странице

#### 9.2.1.3. Дизайн



#### 9.2.1.4. Логика

Страница представляет собой список пользователей с фильтром по имени и постраничной отдачей. Сортировка по имени. Каждая строка таблицы содержит кнопки «Редактировать», «Удалить», по нажатию на которые открывается соответствующая форма. По клику по имени открывается форма просмотра пользователя. По кнопке «Добавить» открывается окно добавления нового пользователя.

### 9.2.2. Изменение

#### 9.2.2.1. Поля

- Имя: текстовое поле

- Описание: текстовое поле

- Логин: текстовое поле

- Пароль: текстовое поле

#### 9.2.2.2. Управляющие элементы

- Кнопка «Сохранить»

- Кнопка «Отмена»

#### 9.2.2.3. Дизайн



#### 9.2.2.4. Логика

По нажатию на кнопку «Сохранить» выполняется валидация полей и попытка сохранения пользователя в базу. Если валидация не пройдена, некорректное поле помечается соответствующей ошибкой. Если при сохранении возникает ошибка, происходит перенаправление на страницу вывода ошибок. В случае успеха происходит возврат на страницу просмотра пользователя. При нажатии на кнопку «Отмена» происходит возврат на страницу просмотра пользователя.

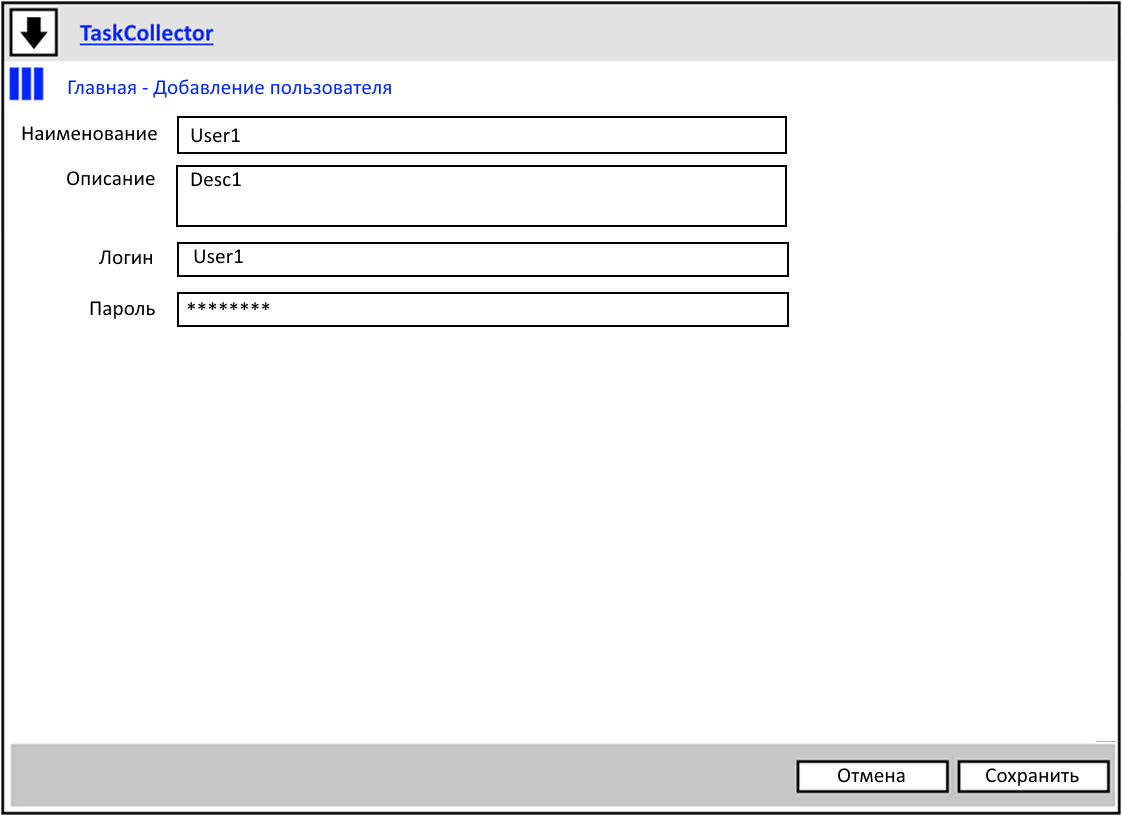
### 9.2.3. Добавление

#### 9.2.3.1. Управляющие элементы

- Кнопка «Сохранить»

- Кнопка «Отмена»

#### 9.2.3.2. Дизайн



#### 9.2.3.3. Поля

- Имя: текстовое поле

- Описание: текстовое поле

- Логин: текстовое поле

- Пароль: текстовое поле

### 9.2.4. Удаление

#### 9.2.4.1. Управляющие элементы

- Кнопка «Удалить»

- Кнопка «Отмена»

#### 9.2.4.2. Дизайн



#### 9.2.4.3. Поля

- Имя: текстовое поле

- Описание: текстовое поле

- Логин: текстовое поле

#### 9.2.4.4. Логика

На странице отображаются все поля удаляемого полльзователя и две кнопки: подтверждения (по нажатию пользователь удаляется) и отмены удаления (возврат на предыдущую страницу).

### 9.2.5. Просмотр

#### 9.2.5.1. Управляющие элементы

- Кнопка «Удалить»

- Кнопка «Редактировать»

- Кнопка «Вернуться»

### 9.2.6. История

#### 9.2.6.1. Поля

- Поля фильтров:

- Наименование

- Пользователь

- Дата от

- Дата до

- Поля пагинатора:

- Количество элементов на странице

- Таблица – список записей

## 9.3. Авторизация

Авторизация представлена контроллером и одним представлением для ввода данных (логина и пароля). Схема аутентификации на основе куки. В АРМ можно войти только под учетными данными пользователя (не клиента).

### 9.3.2. Logout

Метод не принимает параметров. Закрывает сессию текущего пользователя, удаляя из хранилища его Claims, и перенаправляет на домашнюю страницу. Сигнатура

***IActionResult Logout()***

## 9.4. Клиенты

### 9.4.1. История

#### 9.4.1.1. Поля

- Поля фильтров:

- Наименование

- Клиент

- Дата от

- Дата до

- Поля пагинатора:

- Количество элементов на странице

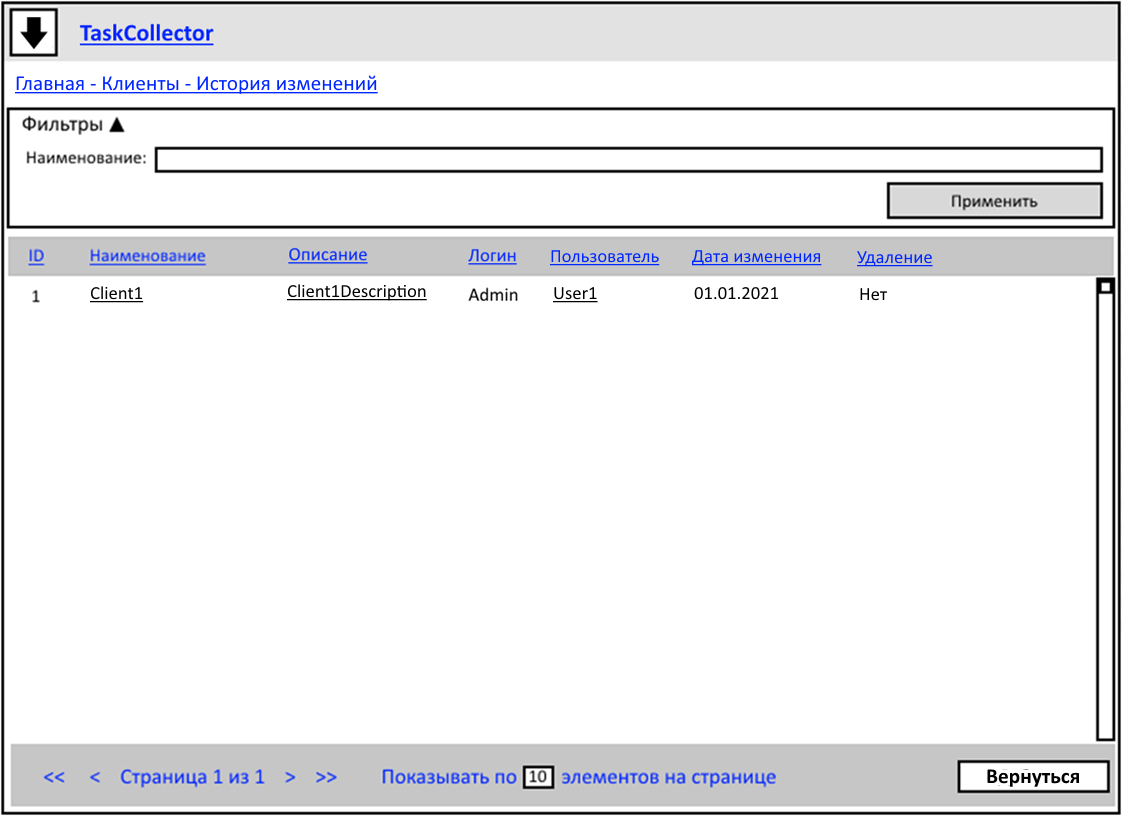
- Таблица – список записей

#### 9.4.1.2. Управляющие элементы

- Фильтры: кнопка «Применить»

- Пагинатор: управляющие элементы пагинатора (см. раздел «Пагинатор»)

#### 9.4.1.3. Дизайн



### 9.4.2. Просмотр

#### 9.4.2.1. Управляющие элементы

- Кнопка «Редактировать»

- Кнопка «Удалить»

- Кнопка «Вернуться»

#### 9.4.2.2. Поля

- Наименование – текст

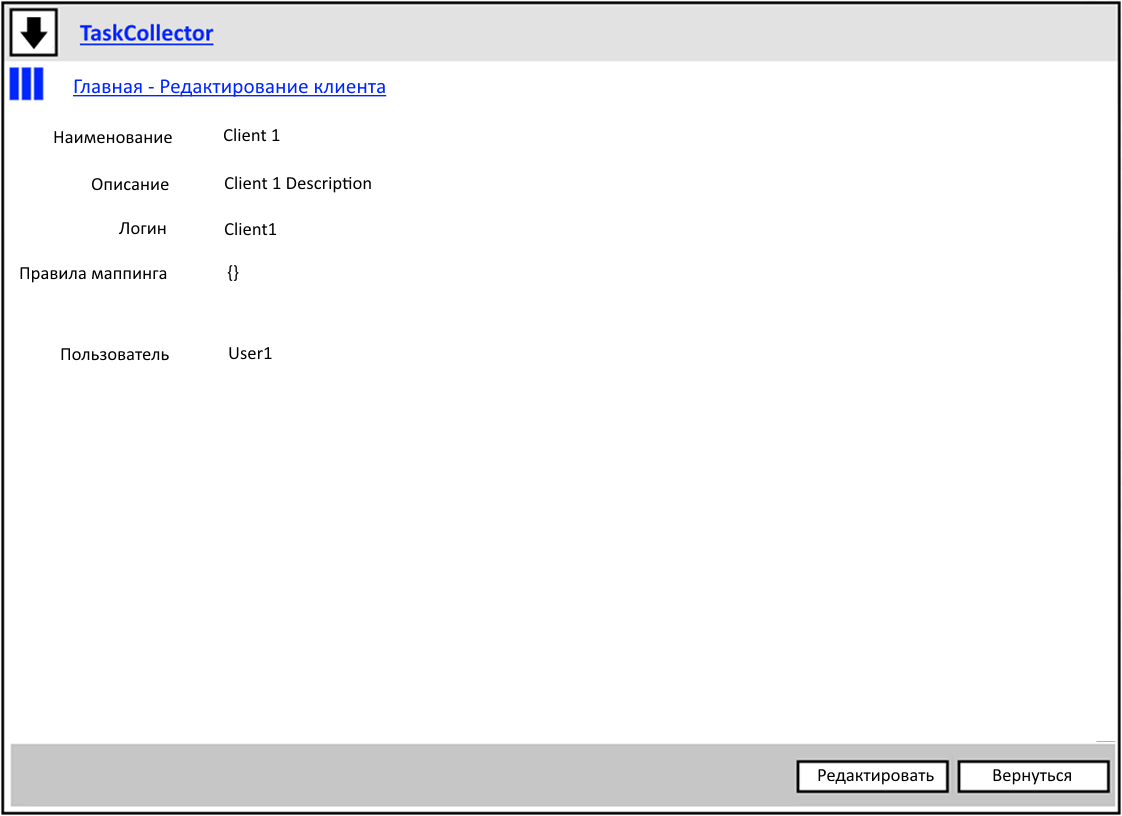
- Описание – текст

- Логин – текст

- Правила маппинга – текст (json)

- Пользователь - текст

#### 9.4.2.3. Дизайн



#### 9.4.2.4. Логика

На странице отображаются все поля выбранного для просмотра клиента. Так же есть две кнопки: Редактировать – открывается страница редактирования клиента и Вернуться – возврат к списку клиентов

### 9.4.3. Список

#### 9.4.3.1. Управляющие элементы

- Фильтры: кнопка «Применить»

- Пагинатор: управляющие элементы пагинатора (см. раздел «Пагинатор»)

- Кнопка «Добавить»

- Напротив каждого элемента: кнопки «Редактировать», «Удалить»

#### 9.4.3.2. Дизайн



#### 9.4.3.3. Логика

Страница представляет собой список клиентов с фильтром по имени и постраничной отдачей. Сортировка по имени. Каждая строка таблицы содержит кнопки «Редактировать», «Удалить», по нажатию на которые открывается соответствующая форма. По клику по имени открывается форма просмотра клиента. По кнопке «Добавить» открывается окно добавления нового клиента.

#### 9.4.3.4. Поля

- Таблица – список элементов

- Поля фильтров:

- Наименование

- Поля пагинатора:

- Количество элементов на странице

### 9.4.4. Удаление

#### 9.4.4.1. Управляющие элементы

- Кнопка «Удалить»

- Кнопка «Отмена»

#### 9.4.4.2. Дизайн



#### 9.4.4.3. Поля

- Наименование – текст

- Описание – текст

- Логин – текст

- Правила маппинга – текст (json)

- Пользователь - текст

#### 9.4.3.3. Логика

На странице отображаются все поля удаляемого клиента и две кнопки: подтверждения и отмены удаления.

### 9.4.5. Добавление

#### 9.4.5.1. Управляющие элементы

- Кнопка «Сохранить»

- Кнопка «Отмена»

#### 9.4.5.2. Поля

- Наименование – текст

- Описание – текст

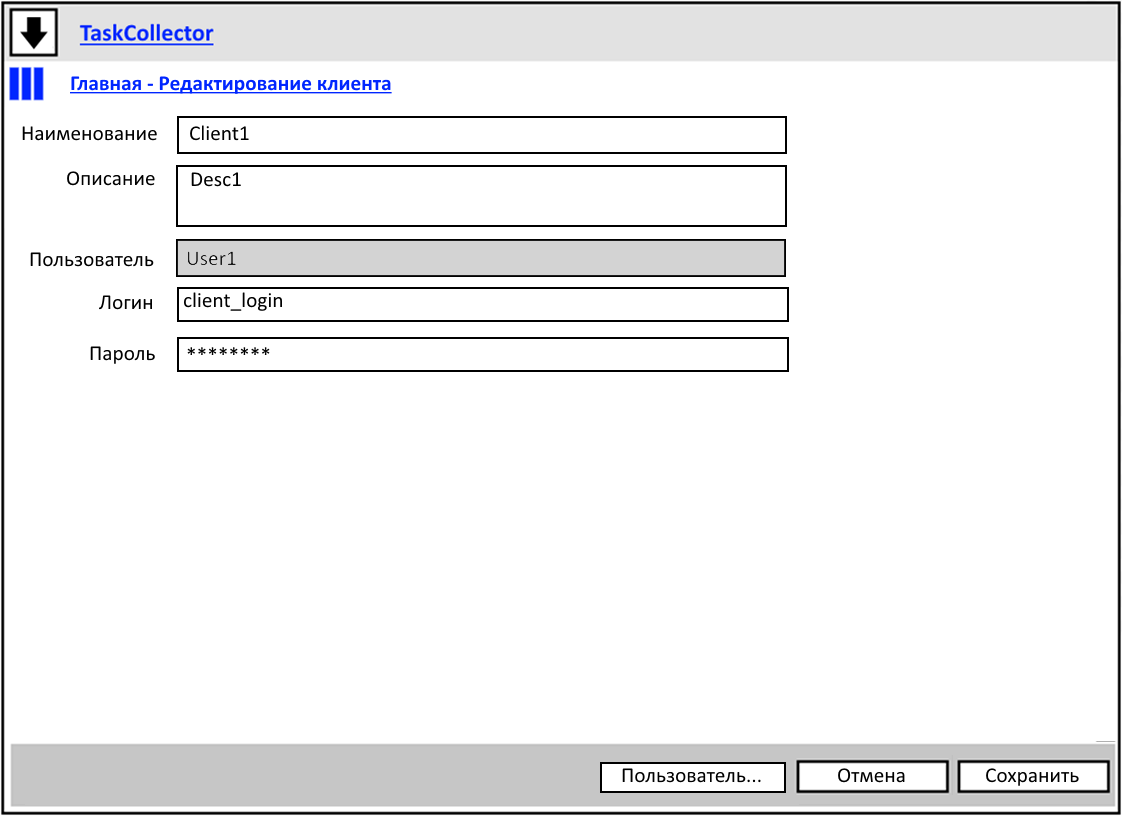
- Логин – текст

- Правила маппинга – текст (json)

- Пользователь - текст

### 9.4.6. Изменение

#### 9.4.6.1. Дизайн



#### 9.4.6.2. Управляющие элементы

- Кнопка «Сохранить»

- Кнопка «Отмена»

- Кнопка «Выбрать пользователя»

#### 9.4.6.3. Логика

По нажатию на кнопку «Сохранить» выполняется валидация полей и попытка сохранения клиента в базу. Если валидация не пройдена, некорректное поле помечается соответствующей ошибкой. Если при сохранении возникает ошибка, происходит перенаправление на страницу вывода ошибок. В случае успеха происходит возврат на страницу просмотра клиента. При нажатии на кнопку «Отмена» происходит возврат на страницу просмотра клиента.

#### 9.4.6.4. Поля

- Наименование – текст

- Описание – текст

- Логин – текст

- Пароль – текст (пароль)

- Правила маппинга – текст (json)

## 9.5. Статусы

### 9.5.1. Список

#### 9.5.1.1. Управляющие элементы

- Фильтры: кнопка «Применить»

- Пагинатор: управляющие элементы пагинатора (см. раздел «Пагинатор»)

- Кнопка «Добавить»

- Напротив каждого элемента: кнопки «Редактировать», «Удалить»

## 9.6. Сообщения

### 9.6.1. Просмотр

#### 9.6.1.1. Поля

- Заголовок – текст

- Уровень - текст

- Описание – текст

- Обратная связь – текст

- Дополнительные поля – текст (json)

- Клиент – текст

- Дата создания – дата

### 9.6.2. Добавление

#### 9.6.2.1. Поля

- Заголовок – текст

- Уровень – выбор из списка

- Описание – текст

- Обратная связь – текст

- Дополнительные поля – текст (json)

- Клиент – выбор из списка

#### 9.6.2.2. Управляющие элементы

- Кнопка «Сохранить»

- Кнопка «Отмена»

# 10. Установщик

## 10.1. 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

## 10.2. Сервер Linux

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

## 10.3. Docker

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

# 11. Тестирование

## 11.1. Модульное

### 11.1.1. Репозитории

#### 11.1.1.1. 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

#### 11.1.1.2. GetItem

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

* Создание окружения
* Добавление сущности через контекст
* Получение сущности по id через репозиторий
* Проверка на соответствие полученной сущности базовой

#### 11.1.1.3. Add

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

* Создание окружения
* Добавление сущности через репозиторий
* Получение сущности через контекст
* Проверка на соответствие полученной сущности базовой

#### 11.1.1.4. Update

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

* Создание окружения
* Добавление сущности через репозиторий
* Получение сущности через контекст
* Проверка на соответствие полученной сущности базовой
* Обновление сущности
* Проверка на соответствие полученной сущности измененной

#### 11.1.1.5. Delete

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

* Создание окружения
* Добавление сущности через репозиторий
* Получение сущности через контекст
* Проверка на соответствие полученной сущности базовой
* Удаление сущности через репозиторий
* Получение сущности через контекст, проверка, что выставлен флаг IsDeleted

### 11.1.2. Сервис работы с данными

#### 11.1.2.1. GetUsers

Проверяется работа метода GetUsers сервиса данных. Последовательность теста:

* Создание окружения
* Добавление нескольких пользователей
* Получение пользователей по фильтру
* Проверка на соответствие полученных сущностей фильтру

#### 11.1.2.2. GetUser

Проверяется работа метода GetUser сервиса данных. Последовательность теста:

* Создание окружения
* Добавление пользователя через репозиторий
* Получение пользователя по id через метод
* Проверка на соответствие полученной сущности созданной

#### 11.1.2.3. GetClients

Проверяется работа метода GetClients сервиса данных. Последовательность теста:

* Создание окружения
* Добавление пользователя и нескольких клиентов
* Получение клиентов по фильтру
* Проверка на соответствие полученных сущностей фильтру

#### 11.1.2.4. GetClient

Проверяется работа метода GetClient сервиса данных. Последовательность теста:

* Создание окружения
* Добавление пользователя и клиента через репозиторий
* Получение клиента по id через метод
* Проверка на соответствие полученной сущности созданной

#### 11.1.2.5. UpdateUser

Проверяется работа метода UpdateUser сервиса данных. Последовательность теста:

* Создание окружения
* Добавление пользователя через репозиторий
* Получение пользователя по id через репозиторий
* Проверка на соответствие полученной сущности созданной
* Изменение нескольких полей методом UpdateUser сервиса данных
* Проверка на соответствие полученной сущности измененной

### 11.1.3. API

#### 11.1.3.1. ClientController

##### **11.1.3.1.1. Auth**

Проверяется работа метода Auth контроллера ClientController.

Последовательность теста:

- Создание окружения

- Добавление пользователя, клиента

- Вызов метода с логином и паролем клиента

- Проверка, что метод вернул токен

#### 11.1.3.2. MessageController

##### **11.1.3.2.1. Send**

Проверяется работа метода Send контроллера MessageController.

Последовательность теста:

- Создание окружения

- Добавление пользователя, клиента

- Авторизация

- Отправка сообщения

- Проверка, что вернулся ОК с идентификатором сообщения

- Проверка, что сообщение и новый статус были добавлены в БД

#### 11.1.3.3. CommonController

##### **11.1.3.3.1. Ping**

Проверяется работа метода Ping контроллера CommonController

Последовательность теста:

- Создание окружения

- Отправка запроса

- Проверка, что вернулся ОК

##### **11.1.3.3.2. ErrorNotify**

Проверяется работа метода ErrorNotify контроллера CommonController

Последовательность теста:

- Создание окружения

- Отправка запроса

- Проверка, что вернулся ОК

- Проверка, что сервис сформировал и отправил сообщение сервису сбора ошибок

##### **11.1.3.3.3. Deploy**

Проверяется работа метода Deploy контроллера CommonController

Последовательность теста:

- Создание окружения

- Отправка запроса

- Проверка, что вернулся ОК

- Проверка, что база создалась корректно

### 11.1.4. Сервис раскатки БД

Проверяется работа метода Deploy сервиса раскатки базы данных. Последовательность теста:

* Создание окружения
* Вызов метода
* Проверка что метод выполнился без ошибок, база данных создана

### 11.1.5. Сервис уведомлений

#### 11.1.5.1. SendNotify

Проверяется работа метода отправки уведомлений указанным провайдером.

Последовательность теста:

- Создание окружения

- Вызов метода

- Проверка, что метод выполнился без ошибок, уведомление доставлено

## 11.2. Интеграционное

### 11.2.1. Web клиент

#### 11.2.1.1. Сценарий 1

Проверяется работа страниц управления клиентами (положительный сценарий).

Последовательность теста:

- Создание окружения

- Добавление через репозиторий нескольких клиентов

- Открытие страницы списка клиентов, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра клиента

- Открытие страницы редактирования клиента, внесение изменений

- Возвращение на предыдущую страницу, проверка что данные изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления клиента

- Добавление клиента

- Проверка, что добавление прошло удачно

- Проверка, что открылась страница просмотра клиента и правильность данных

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра клиентов

- Удаление клиента

- Проверка, что клиент удалился

#### 11.2.1.2. Сценарий 2

Проверяется работа страниц управления клиентами (отрицательный сценарий).

Последовательность теста:

- Создание окружения

- Проверка работы неудачной авторизации

- Добавление через репозиторий нескольких пользователей и клиентов, привязанных к разным пользователям

- Открытие страницы списка клиентов, проверка работы фильтров (недопустимые символы, пустая строка, sql-инъекции), постраничной отдачи (попытка выйти за пределы диапазона страниц)

- Открытие страницы просмотра клиента, замена id в маршруте на несуществующий, на id клиента, привязанного к другому пользователю

- Открытие страницы редактирования клиента, внесение недопустимых изменений, проверка правильности ошибки

- Возвращение на предыдущую страницу, проверка что данные не изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления клиента

- Попытка добавления клиента с некорректными полями

- Проверка, что добавление не прошло

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра клиентов

- Попытка удаления несуществующего клиента, клиента, привязанного к другому пользователю

- Проверка, что клиент не удалился

-

#### 11.2.1.3. Сценарий 3

Проверяется работа страниц управления пользователями (положительный сценарий).

Последовательность теста:

- Создание окружения

- Добавление через репозиторий нескольких пользователей

- Открытие страницы списка пользователей, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра пользователя

- Открытие страницы редактирования пользователя, внесение изменений

- Возвращение на предыдущую страницу, проверка что данные изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления пользователей

- Добавление пользователя

- Проверка, что добавление прошло удачно

- Проверка, что открылась страница просмотра пользователя и правильность данных

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра пользователей

- Удаление пользователя

- Проверка, что пользователь удалился

#### 11.2.1.4. Сценарий 4

Проверяется работа страниц управления пользователями (отрицательный сценарий).

Последовательность теста:

- Создание окружения

- Проверка работы неудачной авторизации

- Добавление через репозиторий нескольких пользователей

- Открытие страницы списка пользователей, проверка работы фильтров (недопустимые символы, пустая строка, sql-инъекции), постраничной отдачи (попытка выйти за пределы диапазона страниц)

- Открытие страницы просмотра пользователя, замена id в маршруте на несуществующий,

- Открытие страницы редактирования пользователя, внесение недопустимых изменений, проверка правильности ошибки

- Возвращение на предыдущую страницу, проверка что данные не изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления пользователя

- Попытка добавления пользователя с некорректными полями

- Проверка, что добавление не прошло

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра пользователей

- Попытка удаления несуществующего пользователя

- Проверка, что пользователь не удалился

-

#### 11.2.1.5. Сценарий 5

Проверяется работа страниц работы с сообщениями (положительный сценарий).

Последовательность теста:

- Создание окружения

- Добавление пользователя, клиента, нескольких сообщений (через базу либо API)

- Открытие страницы сообщений, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра сообщения

- Открытие страницы обработки сообщения, изменение статуса

- Возвращение на предыдущую страницу, проверка что данные изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

#### 11.2.1.6. Сценарий 6

Проверяется работа страниц работы с сообщениями (отрицательный сценарий).

Последовательность теста:

- Создание окружения

- Проверка работы неудачной авторизации

- Добавление через репозиторий нескольких пользователей и клиентов, привязанных к разным пользователям. Добавление нескольких сообщений к каждому из клиентов с разными статусами.

- Открытие страницы сообщений, проверка работы фильтров (недопустимые символы, пустая строка, sql-инъекции), постраничной отдачи (попытка выйти за пределы диапазона страниц)

- Открытие страницы просмотра сообщения, замена id в маршруте на несуществующий, на id сообщения клиента, привязанного к другому пользователю

- Открытие страницы работы с сообщением, внесение недопустимых изменений, проверка правильности ошибки

- Возвращение на предыдущую страницу, проверка что данные не изменились, новый статус не добавился

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

### 11.2.2. API

#### 11.2.2.1. 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

#### 11.2.2.2. 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 сообщения не принимаются сервером

#### 11.2.2.3. Сценарий 3

Проверяется работа получения сообщения (положительный сценарий, без маппинга полей).

Последовательность теста:

- создание окружения

- добавление, пользователей, клиентов

- авторизация

- формирование и отправка сообщения

- проверка, что сообщение сохранено в базе, проверка правильности полей, проверка что добавился статус

#### 11.2.2.4. Сценарий 4

Проверяется работа получения сообщения (отрицательный сценарий, без маппинга полей).

Последовательность теста:

- создание окружения

- добавление, пользователей, клиентов

- авторизация (с неправильными данными)

- формирование и отправка сообщения

- проверяем, что сообщение не сохранилось

- авторизация

- формирование и отправка некорректного сообщения (не json, некорректные поля, sql-инъекция)

- проверка, что сообщения не сохранились в базе

## 11.3. Ручное

### 11.3.1. Web клиент

#### 11.3.1.2. Сценарий 1

Проверяется работа страниц управления клиентами (положительный сценарий).

Последовательность теста:

- Добавление нескольких пользователей, клиентов

- Открытие страницы списка клиентов, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра клиента

- Открытие страницы редактирования клиента, внесение изменений

- Проверка, что изменения применились, открылась страница просмотра клиента

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления клиента

- Добавление клиента

- Проверка, что клиент добавился

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Удаление клиента

- Проверка, что клиент удалился

- Проверка, что вернулись на страницу списка клиентов, фильтры сохранились

#### 11.3.1.2. Сценарий 2

Проверяется работа страниц управления клиентами (отрицательный сценарий).

Последовательность теста:

- Добавление нескольких пользователей, клиентов

- Открытие страницы списка клиентов, проверка работы фильтров (пустой фильтр, использование недопустимых символов, sql-запросов), постраничной отдачи (попытка выйти за границы списка)

- Открытие страницы просмотра клиента

- Открытие страницы редактирования клиента, внесение изменений (пустые поля, некорректные символы)

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления клиента

- Попытки добавления клиентов (пустые поля, некорректные символы)

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра клиентов

#### 11.3.1.3. Сценарий 3

Проверяется работа страниц управления пользователями (положительный сценарий).

Последовательность теста:

- Создание окружения

- Добавление нескольких пользователей

- Открытие страницы списка пользователей, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра пользователя

- Открытие страницы редактирования пользователя, внесение изменений

- Возвращение на предыдущую страницу, проверка что данные изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

- Открытие страницы добавления пользователя

- Добавление пользователя

- Проверка, что добавление прошло удачно

- Проверка, что открылась страница просмотра пользователя и правильность данных

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра пользователей

- Удаление пользователя

- Проверка, что пользователь удалился

#### 11.3.1.4. Сценарий 4

Проверяется работа страниц управления пользователями (отрицательный сценарий).

Последовательность теста:

- Создание окружения

- Добавление нескольких пользователей

- Открытие страницы списка пользователей, проверка работы фильтров (некорректные символы, sql-инъекции), постраничной отдачи (попытка выхода за границы)

- - Открытие страницы редактирования пользователя, внесение некорректных изменений (пустые поля, некорректные символы, дублирование существующих имени, логина)

- Возвращение на предыдущую страницу, проверка правильности перехода

- Открытие страницы добавления пользователя

- Попытка добавления пользователя (пустые поля, некорректные символы, дублирование существующих имени, логина)

- Проверка, что добавление не прошло

- Возврат на предыдущую страницу, проверка, что открылась страница просмотра пользователей

- Попытка удаления несуществующего или удаленного пользователя

#### 11.3.1.5. Сценарий 5

Проверяется работа страниц работы с сообщениями (положительный сценарий).

Последовательность теста:

- Создание окружения

- Добавление пользователя, клиента, нескольких сообщений (через базу либо API)

- Открытие страницы сообщений, проверка работы фильтров, постраничной отдачи

- Открытие страницы просмотра сообщения

- Открытие страницы обработки сообщения, изменение статуса

- Возвращение на предыдущую страницу, проверка что данные изменились

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

#### 11.2.1.6. Сценарий 6

Проверяется работа страниц работы с сообщениями (отрицательный сценарий).

Последовательность теста:

- Создание окружения

- Проверка работы неудачной авторизации

- Добавление через репозиторий нескольких пользователей и клиентов, привязанных к разным пользователям. Добавление нескольких сообщений к каждому из клиентов с разными статусами.

- Открытие страницы сообщений, проверка работы фильтров (недопустимые символы, пустая строка, sql-инъекции), постраничной отдачи (попытка выйти за пределы диапазона страниц)

- Открытие страницы просмотра сообщения, замена id в маршруте на несуществующий, на id сообщения клиента, привязанного к другому пользователю

- Открытие страницы работы с сообщением, внесение недопустимых изменений, проверка правильности ошибки

- Возвращение на предыдущую страницу, проверка что данные не изменились, новый статус не добавился

- Возвращение на предыдущую страницу, проверка правильности перехода и сохранения фильтров

### 11.3.2. API

#### 11.3.2.1. 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

#### 11.3.2.2. 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 сообщения не принимаются сервером

## 11.5. Регресс

### 11.5.1. API

### 11.5.2. Сервер

#### 11.5.2.1. Сценарий 1

Запуск модульных тестов (репозитории, сервис работы с данными, API, сервиса раскатки БД и сервиса уведомлений), интеграционныx тестов API и сервера (регулярные фоновые задания). Если поддерживает репозиторий, запуск тестов выполняется после чекина изменений, иначе все тесты запускаются вручную перед чекином в релизные ветки. При возможности настраивается периодический автоматический запуск тестов.

### 11.5.3. Web клиент

#### 11.5.3.1. Сценарий 1

Запуск модульных тестов (репозитории, сервис работы с данными), интеграционныx тестов Web клиента. Если поддерживает репозиторий, запуск тестов выполняется после чекина изменений, иначе все тесты запускаются вручную перед чекином в релизные ветки. При возможности настраивается периодический автоматический запуск тестов.

.