# Техническое задание на программу для ЭВМ «Планировщик задач» версия 1.

# 1. Описание

Программа представляет собой программу для планирования задач согласно заданного алгоритма. Проекты, под которые планируются задачи, хранятся в базе данных в виде дерева. Проект содержит ряд полей, использующихся для вычисления очередного проекта, добавляющегося в расписание, минимального и/или максимального времени выполнения добавленной задачи расписания.

Программа состоит из сервера, осуществляющего работу с данными, web-клиента и десктопного клиента.

# ======== Actualize ========

# 2. Логика работы

В приложении две основных сущности – «проект» и «расписание». Проекты – базовые структуры, хранящиеся в виде дерева. Выбор проекта для добавления в расписание осуществляется с помощью формулы на основании нескольких полей проекта. Добавление выбранного проекта в расписание осуществляется на основании политики добавления.

Политика представляет собой набор условий и действий при выполнении условий. Одно из условий может быть на добавление проекта в расписание. Если условие не выполняется, выборка проекта не сохраняется, все действия откатываются. Если условие на добавление в расписание отсутствует, проект добавляется в расписание безусловно.

К проекту в один момент времени может быть привязана одна политика. Если проект без политики, используется политика по умолчанию для данного пользователя.

Дерево проектов можно привязать к файловой системе на клиентской (только при использовании десктопного клиента) или серверной стороне.

Добавление проектов в расписание выполняется либо вручную, либо можно настроить периодическое задание (настраивается как периодичность выполнения, так и условия очередного выполнения).

# 3. Требования

Сервер должен обрабатывать не менее 300 запросов в секунду. Выбор и добавление проекта в расписание должно занимать не более пяти секунд при количестве проектов не более 10000.

Ограничение авторизаций пользователя: не более трёх неудачных попыток за 10 минут. Уведомление о попытках неудачных авторизаций после десяти подряд.

Роли в первой версии не предусмотрены.

# 4. Технологии

## 4.1. Сервер

* EntityFramework
* ASP .NetCore
* .NetCore 3.x
* Swagger
* Jwt bearer
* Serilog

## 4.2. Базы данных

* PostgreSQL (сервер)
* SqLite (клиент)

## 4.3. Тестирование

* XUnit
* Moq

## 4.4. Web-client

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

## 4.5. Установщик

* WixInstaller

## 4.6. Клиент

* SqLite (+ EntityFramework?)
* .Net Framework 4.7
* WPF
* HttpClient

# 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. Модели базы данных

### 7.1.1. 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***

### 7.1.2. project

***project:***

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

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

***path\_name varchar(255) not null,***

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

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

***parent\_id uuid null,***

***priority int not null default 1,***

***period int null,***

***cycle\_period int not null default 4320,***

***add\_time int null,***

***last\_used\_date timestamptz not null,***

***select\_count int not null default 0,***

***is\_leaf bool not null default true,***

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

***is\_private bool not null default true,***

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

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

***unique index (parent\_id, path\_name) where not is\_deleted***

***constraint fk\_project\_parent\_id foreign key (parent\_id, user\_id) references project(id, user\_id),***

***constraint fk\_project\_user\_id foreign key (user\_id) references user(id)***

***index idx\_project\_user\_id (user\_id)***

***index idx\_project\_parent\_id (parent\_id)***

***index idx\_project\_ last\_used\_date (last\_used\_date)***

### 7.1.3. 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,***

### 7.1.4. settings

***settings:***

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

***param\_name varchar(100) not null,***

***param\_value varchar(1000) null,***

### 7.1.5. schedule

***schedule:***

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

***project\_id uuid not null,***

***userid uuid not null,***

***order bigint not null,***

***begin\_date timestamptz not null,***

***end\_date timestamptz not null,***

***elapsed\_time int not null default 0,***

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

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

***unique index (userid, order) where not is\_deleted***

***constraint fk\_schedule\_user\_id foreign key (user\_id) references user(id, user\_id),***

***constraint fk\_schedule\_project\_id foreign key (project\_id) references project(id)***

***index idx\_schedule\_user\_id (userid)***

***index idx\_schedule\_project\_id (project\_id)***

***index idx\_ schedule \_ begin\_date (begin\_date)***

***index idx\_ schedule \_ end\_date (end\_date)***

***index idx\_ schedule \_ order (order)***

### 7.1.6. project\_history

***project\_history:***

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

***id uuid null,***

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

***path\_name varchar(255) null,***

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

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

***parent\_id uuid null,***

***priority int null,***

***period int null,***

***cycle\_period int null,***

***add\_time int null,***

***last\_used\_date timestamptz null,***

***select\_count int null ,***

***is\_leaf bool null,***

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

***is\_private bool null,***

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

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

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

***index idx\_project\_history\_user\_id (user\_id)***

***index idx\_project\_history\_id (id)***

***index idx\_project\_history\_change\_date (change\_date)***

### 7.1.7. h\_schedule

***h\_schedule:***

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

***id uuid null,***

***project\_id uuid null,***

***userid uuid null,***

***order bigint null,***

***begin\_date timestamptz null,***

***end\_date timestamptz null,***

***elapsed\_time int null,***

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

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

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

### 7.1.8. formula

***formula:***

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

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

***text varchar(1000),***

***is\_default Boolean not null,***

***version\_id uuid not null,***

***version\_num int not null,***

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

***is\_deleted boolean not null***

***constraints:***

***unique name where not is\_deleted***

***unique version\_id***

***unique (version\_num, id )***

### 7.1.9. h\_formula

***h\_formula:***

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

***id uuid null,***

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

***text varchar(1000),***

***is\_default Boolean null,***

***version\_id uuid null,***

***version\_num int null,***

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

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

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

## 7.2. Серверные

### 7.2.1. User

#### 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;***

***}***

### 7.2.2. Проекты

#### 7.2.2.1. ProjectHistory

***ProjectHistory: EntityHistory***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

#### 7.2.2.2. Project

***Project: Entity***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

### 7.2.3. Формулы

#### 7.2.3.1. Formula

***Formula: Entity***

***{***

***public string Name;***

***public string Text;***

***public bool IsDefault;***

***}***

#### 7.2.3.2. FormulaHistory

***FormulaHistory: EntityHistory***

***{***

***public string Name;***

***public string Text;***

***public bool IsDefault;***

***}***

### 7.2.4. Настройки

#### 7.2.4.1. Settings

***Settings***

***{***

***public int Id;***

***public string ParamName;***

***public string ParamValue;***

***}***

### 7.2.5. Расписание

#### 7.2.5.1. Schedule

***Schedule: Entity***

***{***

***public Guid ProjectId ;***

***public Guid UserId ;***

***public int Order ;***

***public DateTimeOffset BeginDate ;***

***public DateTimeOffset EndDate ;***

***public bool IsRunning ;***

***}***

#### 7.2.5.1. ScheduleHistory

***ScheduleHistory: EntityHistory***

***{***

***public Guid ProjectId ;***

***public Guid UserId ;***

***public int Order ;***

***public DateTimeOffset BeginDate ;***

***public DateTimeOffset EndDate ;***

***public bool IsRunning ;***

***}***

## 7.3. Клиентские

### 7.3.1. Пользователь

#### 7.3.1.1. User

***User: Entity***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.3.1.2. UserIdentity

***UserIdentity: IIdentity***

***{***

***public string Login;***

***public string Password;***

***}***

#### 7.3.1.3. UserCreator

***UserCreator***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.3.1.4. UserHistory

***UserHistory: EntityHistory***

***{***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

#### 7.3.1.5. UserUpdater

***UserUpdater***

***{***

***public Guid Id;***

***public string Name;***

***public string Description;***

***public string Login;***

***public string Password;***

***}***

### 7.3.2. Проекты

#### 7.3.2.1. ProjectHistory

***ProjectHistory: EntityHistory***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

#### 7.3.2.2. ProjectCreator

***ProjectCreator***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

#### 7.3.2.3. Project

***Project: Entity***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

#### 7.3.2.4. ProjectUpdater

***ProjectUpdater***

***{***

***public string Name ;***

***public string Path ;***

***public Guid? ParentId ;***

***public bool IsLeaf ;***

***public DateTimeOffset? LastUsedDate ;***

***public int? Period ;***

***public int Priority;***

***}***

### 7.3.3. Базовый класс

#### 7.3.3.1. Entity

***Entity***

***{***

***public Guid Id;***

***}***

#### 7.3.3.2. EntityHistory

***EntityHistory***

***{***

***public Guid Id;***

***public long Hid;***

***public DateTimeOffset ChangeDate;***

***public bool IsDeleted;***

***}***

### 7.3.4. Расписание

#### 7.3.4.1. Schedule

***Schedule: Entity***

***{***

***public Guid ProjectId ;***

***public Guid UserId ;***

***public int Order ;***

***public DateTimeOffset BeginDate ;***

***public DateTimeOffset EndDate ;***

***public bool IsRunning ;***

***}***

#### 7.3.4.2. ScheduleHistory

***ScheduleHistory: EntityHistory***

***{***

***public Guid ProjectId ;***

***public Guid UserId ;***

***public int Order ;***

***public DateTimeOffset BeginDate ;***

***public DateTimeOffset EndDate ;***

***public bool IsRunning ;***

***}***

#### 7.3.4.3. ScheduleCreator

***ScheduleCreator***

***{***

***public Guid? ProjectId ;***

***public Guid UserId ;***

***public DateTimeOffset? BeginDate ;***

***}***

# 8. Сервер

## 8.1. Описание

Сервер представляет собой приложение, работающее в режиме службы. Предоставляет API для работы клиентского приложения и АРМ. Функции АРМа и API дублируются за исключением некоторых вызовов (управление пользователями в вызовах API отсутствуют). API требует предварительной авторизации клиента, все остальные вызовы обрабатываются только при наличии в запросе актуального токена.

Также с помощью регулярных заданий сервер осуществляет периодическую обработку данных в базе данных.

## 8.2. Работа с данными

### 8.2.1. Get

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

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

### 8.2.2. GetItem

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

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

### 8.2.3. Add

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

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

## 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. project

#### 8.3.3.1. Get

Метод принимает параметрами страницу, размер страницы, поле сортировки, фильтры: по имени, по дате последнего использования (от - до), по родительскому проекту, по флагу «листовой элемент», токен авторизации (в заголовках). Возвращается список проектов текущего пользователя. Сигнатура:

***GET /api/v1/project/?name=null&size=10&page=0&sort=null&isLeaf=null&lastUsedDateBegin=null& lastUsedDateEnd=null&parentId=null***

***Response:***

***{***

***IEnumerable<Project> Projects;***

***}***

#### 8.3.3.2. GetItem

Метод принимает параметром id проекта. Возвращается проект текущего пользователя по id. Сигнатура:

***GET /api/v1/project/{id}***

***Response:***

***{***

***Project Project;***

***}***

## 8.4. Сервисы

### 8.4.1. Сервис выбора задачи

Содержит следующие методы:

- переключение на следующий проект: сдвигает расписание на текущее время, завершает текущий проект, запускает следующий

- сдвиг расписания (вызывается при превышении даты окончания текущей задачи или при переключении на следующий проект): сдвигает все проекты начиная с текущего по дате на заданный промежуток времени

- добавление новой задачи в расписание: добавляет переданный в параметрах проект или делает выбор проекта из дерева проектов и добавляет его в расписание (в конец расписания либо вставляет в расписание в заданное время со сдвигом последующих задач)

### 8.4.2. Сервис авторизации

Содержит два метода: авторизация клиента и авторизация web-клиента. Каждый метод возвращает свой набор Claims в качестве результата (для авторизации через Cookie и токен).

Методы проверяют переданные учётные данные и возвращают либо набор Claims либо null;

# 10. Клиент

## 10.1. Главное окно

### 10.1.1. Меню

* Файл
  + Подключиться к серверу…
  + ----
  + Добавить проект
  + Добавить задачу
  + ----
  + Выход
* Окно
  + Проекты
  + Формулы
  + Настройки
* Помощь
  + О программе
  + Сообщение об ошибке

### 10.1.2. Управляющие элементы

Управляющие элементы представлены в виде панели инструментов, частично дублирующих меню:

* Обновить (обновить таблицу)
* Добавить проект
* Добавить задачу
* Редактировать выбранную задачу
* Завершить текущую задачу (переключиться на следующую)
* Выход

### 10.1.3. Логика

Главное окно содержит расписание текущего пользователя в виде таблицы.

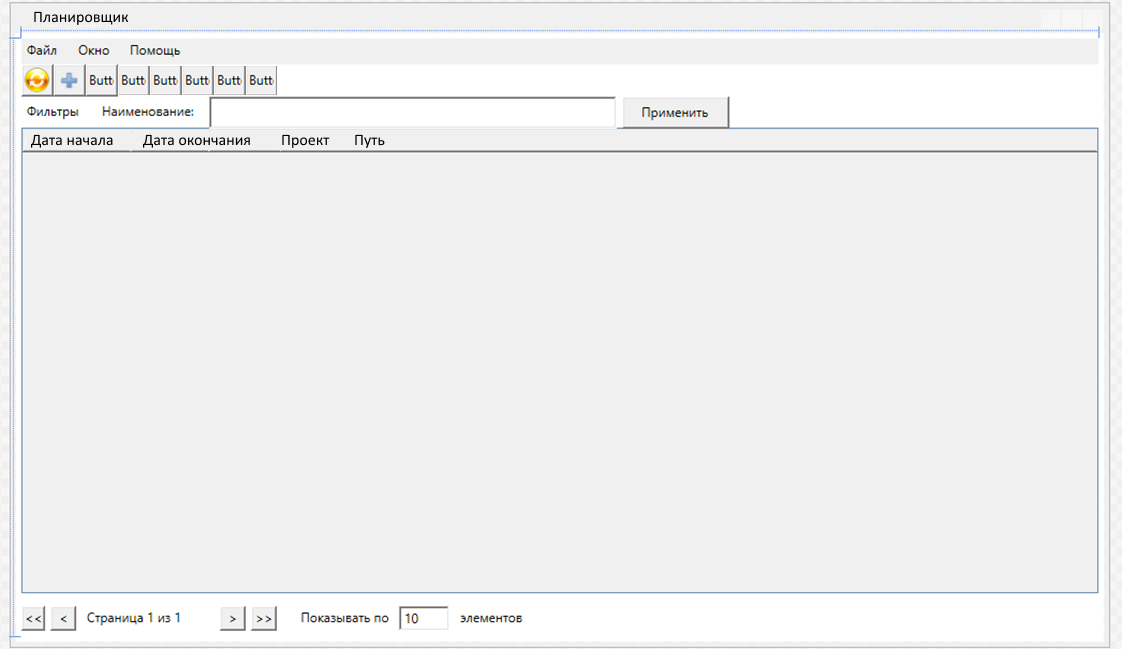
Пункты меню позволяют совершить следующие действия:

* Файл
  + Подключиться к серверу… - подключение к серверу (сохраняется в локальных настройках)
  + Добавить проект – добавление нового проекта
  + Добавить задачу – добавить задачу в расписание (без переключения) – конкретный проект или по формуле выбора
  + Выход - закрытие программы
* Окно
  + Проекты – окно списка проектов пользователя
  + Формулы – окно списка формул выбора следующего проекта
  + Настройки – настройки текущего пользователя (сохраняются на сервере)
* Помощь
  + О программе – окно о программе
  + Сообщение об ошибке –отправка сообщения об ошибке в сервис сбора ошибок

Управляющие элементы главного окна позволяют совершить следующие действия:

* Обновить - обновить таблицу главного окна (расписание)
* Добавить проект – Добавить новый проект (для пользователя)
* Добавить задачу – Добавить новую задачу в расписание (без переключения) – конкретный проект или по формуле выбора
* Редактировать выбранную задачу – изменить параметры текущей задачи
* Завершить текущую задачу (переключиться на следующую)
* Выход

### 10.1.4. Дизайн



## 10.2. Описание

Клиент представляет собой десктопное приложение, работающее в режиме тонкого клиента. Вся бизнес-логика расположена в серверной части, клиент выполняет только отображение данных.

## 10.3. Библиотека запросов к серверу

Библиотека представляет собой обертку над библиотекой HttpClient. Реализуются следующие методы:

### 10.3.1. Метод Auth

Выполняет запрос к методу Auth контроллера AccountController. Выполняется автоматически перед началом работы с другими методами.

### 10.3.2. Метод Get

Выполняет запрос к методу Get контроллера модели ([Model]Controller). Параметры передаются в запросе. Полученный результат десериализуется в массив элементов [Model].

## 10.3. Сервис локальных настроек

Сервис позволяет сохранять и получать из локальной БД SqLite необходимые настройки (сервер подключения, имя пользователя, пароль и пр.).

### 10.3.1. Метод GetSettings

Получает из базы настройку (модель Settings) по имени параметра (ParamName).

# 11. Установщик

## 11.1. Docker

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

## 11.2. Сервер Linux

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

## 11.3. 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

# 12. Тестирование

## 12.1. Регресс

Запуск модульных и интеграционныx тестов (автоматически). 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.