# Brief description of the project:

The project was created in order to automatically fill in the atzarat-briut shave in certain schools / circles, etc. /.

The project uses RabbitMQ to pass requests between micro services.

1. In this project, every morning at 7.15 a.m. the scheduler-service turns on and checks whether it is necessary to fill in the shave atzarat today.

There are 2 planning options to fill out. These are regular filling (checked by the day of the week - for example - every Tuesday) and one-time filling (checked by the current date)

1. If the scheduler-service finds an entry to populate, it sends a seeding request to the appropriate service. The request can be sent to tsofim-service, school-service, gymnast-service)

When requested, the scheduler-service sends a UuidChild to the corresponding service (that is, the identifier of the child for whom you need to fill in the shave atzarat briut)

1. All services that "receive" requests for filling out the document work according to the same scheme.

The service "listens" to RabbitMQ with certain settings, and when it "hears" its request:

1. UuidChild receives from the user-service all the data it needs about the parents / responsible person of the child (such as name, passport number, etc.) - using HystrixApi
2. UuidChild receives the necessary data about the child from the children-service HystrixApi
3. If necessary, from the database, he receives information about a specific child, which is necessary to fill in the atzarat briut shave.
4. After receiving all the necessary data, the service goes to the specified URL to the desired page. And fills in there all the fields required to fill the document (Used by Selenium + Firefox driver)
5. Filling in any atzarat briut shave ends with a message that "the document has been successfully completed." The service takes a screenshot of this message
6. The service sends to the email-service all data of the child, his parents and a screenshot in Base64 format (using RabbitMQ)

2. Email-service "listens" to RabbitMQ and if it hears that it is necessary to send a message, it sends a letter to the email specified in the message. The letter contains information about the user (surname / name), a screenshot of the completed document, the place for which the document was filled out.

3. If something is wrong with the letter, then the email-service sends a letter to the project administrator, in which it tells what exactly happened and why the letter was not sent.

# Security nuances:

# The project uses Spring Security. Authentication takes place using a token.

# To receive a token, the user must register in the application. When registering, the user receives a letter with a temporary registration token to the specified email address. The user must follow the link with the token and thus confirm his email address. Only after that he will be able to "enter the site" to configure the project (enter all the necessary data)

If the user has forgotten his password, he can change it. To do this, he must enter his email address, which will receive an email with a temporary token to change the password. by clicking on the link in the letter, the user will be able to enter a new password for his account.

# List of micro services

|  |  |  |
| --- | --- | --- |
| The name of the micro service | Description of the micro service | Port |
| [config-service](#_config-service)  spring.application.name  =config-server | Сервис конфигурации. Хранит ссылку на git-hub. По ссылке хранятся все application-properties всех микро сервисов | 8012 |
| [eureka-service](#_eureka-service)  spring.application.name  =eureka-service | Eureka service. Each Micro service is registered on the Eureka server, and the Eureka server knows all the client applications running at each port and IP address. | 8761 |
| [zuul-service](#_zuul-service)  spring.application.name  =zuul-service | Agateway that handles all requests and performs dynamic routing of microservice applications | 8011 |
| [user-service](#_user-service)  spring.application.name  = user-service | A service that stores information about the user or aboutthe person responsible for each particular child.  Функционал – регистрация, логин, обновление пароля + CRUD пользователя | 8090 |
| [children-service](#_children-service)  spring.application.name  = children-service | A service that stores all information about the child. His name, surname, theudat zeut, link to the parent or the responsible person | 8092 |
| [scheduler-service](#_scheduler-service)  spring.application.name  = scheduler-service | Сервис, который раз в день проверяет базу данных и смотрит, нужно ли сегодня заполнять атцарат бриют. И если нужно – то какой. После того, как scheduler-service выполняет проверку, он отправляет запрос на заполнение документа в необходимый сервис (school-service, tsofim-service или gymnast-service) | 8084 |
| [school-service](#_school-service)  spring.application.name  = school-service | A service that parses the necessary page and fills the atzarat shave for the school for a particular pupil. Once the document is completed, the documentis taken and sent to ee-mailmail service | 8085 |
| [tsofim-service](#_tsofim-service)  spring.application.name  = tsofim-service | A service that parses the necessary page and fills the atzarat bristly for scouts for a particular student. Once the document is completed, the service takes a screenshot of the document and sends it to ee-mailmail service | 8086 |
| [gymnast-service](#_gymnast-service)  spring.application.name  = gymnast-service | A service that parses the necessary page and fills the attic shave for gymnastics for a particular student. Once the document is completed, the service takes a screenshot of the document and sends it to ee-mailmail service | 8087 |
| email-service  spring.application.name  = email-service | A service that constantly "listens" to where to send a letter, and if a letter appears in the queue - the service sends it to the specified address. | 8093 |

# How to download

1. Config-service
2. Eureka-service (берет данные для загрузки из config-service)
3. Zuul-service
4. Other services

# config-service

keeps a link to the git repository, which stores application-properites for all services (https://github.com/smilyk/briut.git

)

# eureka-service

To connect is used:

|  |  |
| --- | --- |
| spring.security.user.name | user |
| spring.security.user.password | password |

# zuul-service

## Security

Requires authentication. Without authentication, methods work

|  |  |
| --- | --- |
| Confirmation of e-mail | /user-service/users/v1/email-verification |
| User registration | /user-service/users/v1 |
| Login | /user-service/users/login |
| updating your user's password | /user-service/users/v1/password-reset-request |
| new user password | /user-service/users/v1/password-reset |

All other methods require authentication.

# user-service

## Description

service that stores all data about the user (the child's parent and/or the responsible person - who fills the atzarat brie)

## database

**a\_b\_user-service**

таблицы:

### users

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | User ID | Auto-generated.  Used as a key |
| uuidUser | Unique user ID | It is generated in the conservation process.  Mandatory field |
| firstName | Username | Mandatory field |
| secondName | User's surname |  |
| mainEmail | Home email | It is on this email will come a letter to confirm registration and/or change the password.  Mandatory field |
| altEmail | Alternative email | It is necessary if the user wants to register from one address, and the atzarat is shaved to receive to another address (screenshots) |
| password | Password | Mandatory field.  In the process of preservation, the database is encrypted (bCryptPasswordEncoder  ) |
| tz | Theaida zeut No. | This field is not necessary to register, but when planning to fill the atzarat shaving application will require you to fill this box, as the atzarat brie requires filling the tz of the parent/responsible person |
| confirmEmailToken | Email confirmation token | The token is created when the user registers. This token is sent to the user's email address, and the token is removed when the mail is confirmed. Need only to confirm e-mail |
| deleted | Removing the user | User data is not physically removed from the database. The flag is put |
| confirmEmail | The default is false. | Once the user confirms their e-mail the field becomes true and the user can login to the app. For users with unconfirmed e-mail, the login is prohibited. |

### resp\_person

The table in which the data of the responsible persons is stored. Accordingly, all screenshots will come to the email address of this person in charge

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | User ID | Auto-generated.  Used as a key |
| uuidRespPerson | Unique user ID | It is generated in the conservation process.  Mandatory field |
| firstName | Username | Mandatory field |
| secondName | User's surname |  |
| emailRespPerson | Home email | Mandatory field. |
| tz | Theaida zeut No. | This field is not necessary to register, but when planning to fill the atzarat shaving application will require you to fill this box, as the atzarat brie requires filling the tz of the parent/responsible person |
| deleted | Removing the user | User data is not physically removed from the database. The flag is put |

### password\_reset\_tokens

table that stores the token needed to change a user's password

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | User ID | Auto-generated.  Used as a key |
| Token | Token | You need to change the user's password. Sent to the user's email address, specified at registration (or addition of the person in charge) |

## функциональность(API)

### Mapping - "/users/v1"

#### User registration

Метод POST

API

When you sign up:

* Создается userUuid
* The password of the userI
* ConfirmEmailToken is created !!! It has a duration of 10 days
* Deleted = false
* A request is sent to ee-mailmail service in order to send an email to the user who will store the token to confirm the email address.

When you create a user, you're going to Send EmailVerificationDto(

String tokenValue; String userName; String userLastName; String email;

) которое передается в e-mail-service для отправки сообщения.

#### передача данных в e-mail-service при регистрации пользователя

Для передачи данных в email-service используются очереди (RabbitMQ)

RabbitMQ настройки –

|  |  |
| --- | --- |
| email.key | сonfemail |
| email.queue | CONFEMAIL |
| email.exchange | Email |

Once registered, the user must confirm their email address.

#### Confirmation of e-mail

In order to confirm your email, the user must open the email and click on the link that is listed in the letter.

The image looks like text

Automatic description

When you go

Метод GET

API /email-verification

Response:

The image looks like text

Automatic description

Поле confirm-email становится true, confirm\_email\_token становится null

#### смена пароля шаг 1

Метод POST

API /password-reset-request

что бы This generates a request in the email-service.

#### In the database in the password\_reset\_tokens table, a token is created and saved to change the password (it has a validity period of 1 day)

#### Change the password step 1

In order to change the password, the user must open the letter and follow the link specified in it.

The image looks like text

Automatic description

When you go

Метод POST

API /email-verification

There's a new password in the users database

#### In the database in the table, the password\_reset\_tokens token is removed.

#### Login user

Метод GET

API -> /{uuidUser}

Localhost:0000/users/login

В Headers получаем token

#### Get a user on his Uuid

Метод GET

API -> /{uuidUser}

#### Update the user on his Uuid

Метод PUT

API -> /{uuidUser}

#### Удалить пользователя по его Uuid

Метод DELETE

API -> /{uuidUser}

/f5279f6b-7eef-4593-adce-ef19a7158e33

#### Get all users

Метод GET

### Mapping - /resp\_pers/v1

Обычный CRUD

Methods - create, get by Uuid,delete, update, get a list of all responsible persons

### Mapping - /users/v1/admin"

Contains one method - the creation of a super administrator (needed to get an administrator's token, which is necessary for the work of the parsers)

## Security

Access to the service is made by checking token

Without authorization, methods are available:

/\*\*create admin

/\*\*проверка email

/ Password change

/Enter the link to change the password from the letter

# children-service

## Description

service that stores all data about the child for whom the atzarat is filled with briat

## database

#### a\_b\_user-service

таблицы:

### children

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | Child ID | Auto-generated.  Used as a key |
| uuidChild; | Unique child ID | It is generated in the conservation process.  Mandatory field |
| firstName | Child's name | Mandatory field |
| secondName | Child's surname |  |
| uuidParent | Uuid родителя ребенка |  |
| uuidRespPers | Uuid ответственного лица |  |
| tz | Child's theudat number | This field is not necessary to register, but when planning to fill the atzarat shaving application will require you to fill this box, as the atzarate brie requires filling the baby's ts |
| deleted | Removing the user | The child's data is not physically removed from the database. |
|  |  |  |

## функциональность (API)

Mapping - "children"

Обычный CRUD

Methods - create, get by Uuid,delete, update, get a list of all children

## Security

Access to the service is made by checking token

# scheduler-service

## Description

a service that checks whether to fill the atzarat shave or not. If you need to, send a request for a certainservice.

## database

#### a\_b\_user-service

таблицы:

#### plan

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | Record ID | Auto-generated.  Used as a key |
| uuidPlan; | Unique record ID | It is generated in the conservation process.  Mandatory field |
| uuidChild; | Unique child ID | Received while maintaining a planning record.  Mandatory field |
| sendEmail | Do you send an email? | If the flag is true then a letter with a message that will be filled with the atzarat brie will be sent to the responsible person/parent (will be implemented in the second version) |
| service | The name of the service. | The name of the service, which will be used to fill the atzarat briat |
| date | Date | The date when you need to fill the atzarat brie |
| Monday  Tuesday  Wednesday  Thursday  Friday  Saturday  Sunday | Day of the week | Day of the week is the day when the atzarat is filled with bristly, provided that it needs to be filled regularly. For example - every Tuesday |
| deleted | Removing the record | Entries are not physically removed from the database. |
|  |  |  |

## функциональность (API)

Mapping - /plan/v1

Обычный CRUD

Методы – создать, получить по uuidChild, получить по uuidPlan, удалить, обновить, получить список всех запланированных записей

## Sending messages to microservices

### school-service

отправка сообщения на API …/{childUuid}

Sending a message about the need to fill out is with RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| school.key | sc |
| school.queue | SCHOOL |
| rabbitmq.exchange | atBriut |

### tsofim-service

отправка сообщения на API …/{childUuid}

Sending a message about the need to fill out is with RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| tsofim.key | ts |
| tsofim.queue | TSOFIM |
| rabbitmq.exchange | atBriut |

### gymnast-service

отправка сообщения на API …/{childUuid}

Sending a message about the need to fill out is with RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| gymnsat.key | gym |
| gymnsat.queue | GYMNAST |
| rabbitmq.exchange | atBriut |

# school-service

## Description and functionality

Сервис ждет от scheduler-service сообщение о том, что необходимо заполнить атцарат бриют.

1. Receiving a message about the need to fill out is via RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| school.key | sc |
| school.queue | SCHOOL |
| rabbitmq.exchange | atBriut |

1. After the service fills the atzarat brie, it takes a screenshot of the screen, and sends the data that you need to send a screenshot to the responsible person/parent of the child

D-lings are sent to the email-service with the help of RabbitM

RabbitMQ настройки –

|  |  |
| --- | --- |
| email.key | email |
| email.queue | EMAIL |
| email.exchange | Email |

В запросе email-service содержатся:

EmailDto

|  |  |
| --- | --- |
| **Field** | **Description** |
| email | электронная поста, на которую нужно будет отправить письмо |
| picture | Screenshot (Base64) |
| lastName | Parent/responsible surname |
| firstName | Parent/responsible name |
| childFirstName | Child's name |
| childSecondNam | Child's surname |
| service | School-service (название сервиса, от которого исходит запрос об отправке письма) |

1. In order to fill out the atzarat briat the service must obtain the data of the child and his responsible person/parent.

Для этого используется HystrixApi:

|  |  |
| --- | --- |
| Name | Method |
| Получение данных о ребенке – запрос в child-service | @GetMapping("/child/v1/{uuidChild}")@Headers("Authorization: {token}")public Response getChildByChildUuid(@PathVariable String uuidChild, @RequestHeader("Authorization") String token); |
| Получение данных о ответственном лице – запрос в user-service | @GetMapping("/resp\_pers/v1/{uuidRespPerson}}") @Headers("Authorization: {token}") public Response getResponsePersonByUserUuid(@PathVariable String uuidRespPerson, @RequestHeader("Authorization") String token); |
| Получение данных о родителе– запрос в user-service | @GetMapping("/users/v1/{uuidUser}")@Headers("Authorization: {token}")public Response getUserByUserUuid(@PathVariable String uuidUser, @RequestHeader("Authorization") String token); |

## Database

**a\_b\_school-service**

### таблицы:

### school\_details

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | User ID | Auto-generated.  Used as a key |
| uuidChildDetails | Unique record ID for a child | Mandatory field |
| uuidChild | Unique child ID | Mandatory field |
| schoolUserName | Login for school | Issued at school |
| schoolPassword | Password for school | Issued at school |

## функциональность (API)

Mapping – “/school/v1”

Обычный CRUD

Методы – создать, получить по childUuid, получить по planUuid, удалить, обновить, получить список всех записей

## Security

Access to the service is made by checking token

# tsofim-service

## Description and functionality

Сервис ждет от scheduler-service сообщение о том, что необходимо заполнить атцарат бриют.

1. Receiving a message about the need to fill out is via RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| tsofim.key | ts |
| tsofim.queue | TSOFIM |
| rabbitmq.exchange | atBriut |

1. After the service fills the atzarat brie, it takes a screenshot of the screen, and sends the data that you need to send a screenshot to the responsible person/parent of the child

The data is sent to the email-service with the help of RabbitM

RabbitMQ настройки –

|  |  |
| --- | --- |
| email.key | email |
| email.queue | EMAIL |
| email.exchange | Email |

В запросе email-service содержатся:

EmailDto

|  |  |
| --- | --- |
| **Field** | **Description** |
| email | электронная поста, на которую нужно будет отправить письмо |
| picture | Screenshot (Base64) |
| lastName | Parent/responsible surname |
| firstName | Parent/responsible name |
| childFirstName | Child's name |
| childSecondNam | Child's surname |
| service | sofim--service название сервиса, от которого исходит запрос об отправке письма(' |

1. In order to fill the atzarat shave service you need to get the data of the child and his responsible person/parent.

Для этого используется HystrixApi:

|  |  |
| --- | --- |
| Name | Method |
| Получение данных о ребенке – запрос в child-service | @GetMapping("/child/v1/{uuidChild}")@Headers("Authorization: {token}")public Response getChildByChildUuid(@PathVariable String uuidChild, @RequestHeader("Authorization") String token); |
| Получение данных о ответственном лице – запрос в user-service | @GetMapping("/resp\_pers/v1/{uuidRespPerson}}")@Headers("Authorization: {token}")public Response getResponsePersonByUserUuid(@PathVariable String uuidRespPerson, @RequestHeader("Authorization") String token); |
| Получение данных о родителе– запрос в user-service | public interface UserServiceClient { @GetMapping("/users/v1/{uuidUser}") @Headers("Authorization: {token}") public Response getUserByUserUuid(@PathVariable String uuidUser, @RequestHeader("Authorization") String token); |

## Database

### **a\_b\_tsofim-service**

### таблицы:

### tsofim\_details

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Notes** |
| id | User ID | Auto-generated.  Used as a key |
| uuidTsofimDetails | Unique record ID for a child | Mandatory field |
| uuidChild | Unique child ID | Mandatory field |
| place | The place where the Scouts are stationed | Issued at school |
| groupTs | Band name | Issued at school |
| school | School name |  |
| childClass | The class where the child is studying |  |
| deleted | Removing the record | The data is not physically removed from the database. The flag is put |

## функциональность(API)

Mapping – “/tsofim/v1”

Обычный CRUD

Методы – создать, получить по childUuid, получить по planUuid, удалить, обновить, получить список всех записей

## Security

Access to the service is made by checking token

# gymnast-service

## Description and functionality

Сервис ждет от scheduler-service сообщение о том, что необходимо заполнить атцарат бриют.

1. Receiving a message about the need to fill out is via RabbitM..

RabbitMQ настройки –

|  |  |
| --- | --- |
| gymnsat.key | gym |
| gymnsat.queue | GYMNAST |
| rabbitmq.exchange | atBriut |

1. After the service fills the atzarat brie, it takes a screenshot of the screen, and sends the data that you need to send a screenshot to the responsible person/parent of the child

The data is sent to the email-service with the help of RabbitM

RabbitMQ настройки –

|  |  |
| --- | --- |
| email.key | email |
| email.queue | EMAIL |
| email.exchange | Email |

В запросе email-service содержатся:

EmailDto

|  |  |
| --- | --- |
| **Field** | **Description** |
| email | электронная поста, на которую нужно будет отправить письмо |
| picture | Screenshot (Base64) |
| lastName | Parent/responsible surname |
| firstName | Parent/responsible name |
| childFirstName | Child's name |
| childSecondNam | Child's surname |
| service | gymnast-service (название сервиса, от которого исходит запрос об отправке письма) |

1. In order to fill the atzarat shave service you need to get the data of the child and his responsible person/parent.

Для этого используется HystrixApi:

|  |  |
| --- | --- |
| Name | Method |
| Получение данных о ребенке – запрос в child-service | @GetMapping("/child/v1/{uuidChild}")@Headers("Authorization: {token}")public Response getChildByChildUuid(@PathVariable String uuidChild, @RequestHeader("Authorization") String token); |
| Получение данных о ответственном лице – запрос в user-service | @GetMapping("/resp\_pers/v1/{uuidRespPerson}}")@Headers("Authorization: {token}")public Response getResponsePersonByUserUuid(@PathVariable String uuidRespPerson, @RequestHeader("Authorization") String token); |
| Получение данных о родителе– запрос в user-service | public interface UserServiceClient { @GetMapping("/users/v1/{uuidUser}") @Headers("Authorization: {token}") public Response getUserByUserUuid(@PathVariable String uuidUser, @RequestHeader("Authorization") String token); |

## Database

No

## Security

Access to the service is made by checking token

# email-service

## Description and functionality

The service is waiting for messages from other services in order to send letters with the necessary content.

The service "listens" to messages with RabbitM,if a message fails when you send a letter to a specified address, the service sends letters to the administrator informing that a failure has occurred, and which