Тестовая страница

Тестовая страница доступна только в "отладочном режиме" и позволяет проверить работу методов регистрации новых пользователей, восстановления пароля, авторизации в системе. Может быть использована в процессе отладки "фронтэнда", а также для просмотра списка пользователей зарегистрированных в системе (для случая если у разработчика нет доступа к БД).

Тестовая страница доступна по пути

> http://host:XXXX/test/

где ***host*** - адрес на котором запущен сервис, ***XXXX*** - порт на котором запущен сервис (управляется переменной окружения ***SERVER_PORT***=XXXX).

На тестовой странице можно:

- залогиниться в систему с заданными логином и паролем (получить пару *token* и *refreshToken*);
- обновить пару *token* и *refreshToken*;
- перейти на страницу регистрации в системе нового пользователя;
- перейти на страницу восстановления пароля пользователя по email, указанному при регистрации пользователя;
- перейти нра страницу для расчета хеша *SHA256* строки символов (сервисная функция для прописывания пароля пользователя напрямую через БД);
- получить список всех пользователей системы (доступно если зарегистрироваться под администратором - пользователь ***enforce_dba***);
- удалить пользователя по его логину (доступно если зарегистрироваться под администратором - пользователь ***enforce_dba***).

Процесс регистрации пользователя в системе

Для регистрации в системе необходимы API методы "***/login/auto_register_user***" и "***/login/confirm_requisition***" (подробное описание можно посмотреть в файле ***REST_API_server.yaml*** в формате swagger). Процесс регистрации состоит их трех этапов:

1. Создание заявки на регистрацию пользователя (метод "***/login/auto_register_user***").
2. Подтверждение созданной ранее заявки на регистрацию администратором (или подтверждение может осуществляться автоматически).
3. Верификация адреса email, указанного при создании заявки (этап 1) посредством отправки на этот адрес уникальной ссылки, при переходе по которой завершается процесс регистрации и создается пользователь. Метод "***/login/confirm_requisition***".

Создание заявки на регистрацию нового пользователя

Для регистрации в системе нового пользователя должна быть предусмотрена отдельная страница. Которая должна содержать следующие поля:

- **Лицевой счет.** Указанный в данном поле набор символов является логином при последующем входе в систему.
- **Пароль.** Указанный в данном поле набор символов является паролем при последующем входе в систему.
- **Адрес электронной почты.** Указанный в данном поле адрес электронной почты будет использоваться для отправки уведомлений пользователю, в том числе при процедуре восстановления пароля пользователя. Для всей системы адрес электронной почты является уникальным, т.е. не может существовать нескольких пользователей с одинаковыми адресами email.
- **Номер телефона.** Данное поле является не обязательным.
- **Тип абонента.** Задает один из двух возможных типов абонентов: "Физ. лицо" или "Юр. лицо". От значения данного поля зависит алгоритм создания нового пользователя в системе.

После заполнения всех перечисленных полей и нажатия кнопки подтверждения вызывается метод API "***/login/auto_register_user***". При передаче пароля введенного пользователем необходимо произвести следующие действия:

- Получить RSA 2048 битный открытый ключ с помощью метода "***/get_open_key***";
- Вычислить хеш SHA256 (функция вычисления SHA256 на java script находится в файле ***sh256.hash.js***);
- Полученный на предыдущем шаге хеш пароля зашифровать открытым RSA ключом.

В метод "***/login/auto_register_user***" передается закодированный хеш пароля, введенного пользователе и UID ключа, полученный в методе "***/get_open_key***". Все остальные поля передаются в открытом виде.

Также в метод "***/login/auto_register_user***" передается url страницы подтверждения электронной почты. Данный url будет использоваться при формировании ссылки, передаваемой пользователю в письме. Ссылка в письме будет выглядеть следующим образом:

> ?uid=<xxxxx-xxxxxx-xxxxxx-xxxxxxx>

где - строка текста переданная в метод "***/login/auto_register_user***" в параметре url, а <xxxxx-xxxxxx-xxxxxx-xxxxxxx> сгенерированный случайным образом uid заявки на регистрацию нового пользователя.

После вызова метода "***/login/auto_register_user***" система проверяет возможность создания заявки на регистрацию нового пользователя в системе. По результату выполнения проверки будет прислан ответ о создании заявки или отказ в создании заявки на регистрацию пользователя с указанием причин отказа (см. документацию на метод "***/login/auto_register_user***" в файле ***REST_API_server.yaml***).

Подтверждение созданной ранее заявки на регистрацию администратором

Процесс подтверждения заявки на регистрацию может осуществляться администратором или в автоматическом режиме (зависит от пожелания заказчика). Автоматический режим настраивается через таблицу в БД BP_VARIABLE.

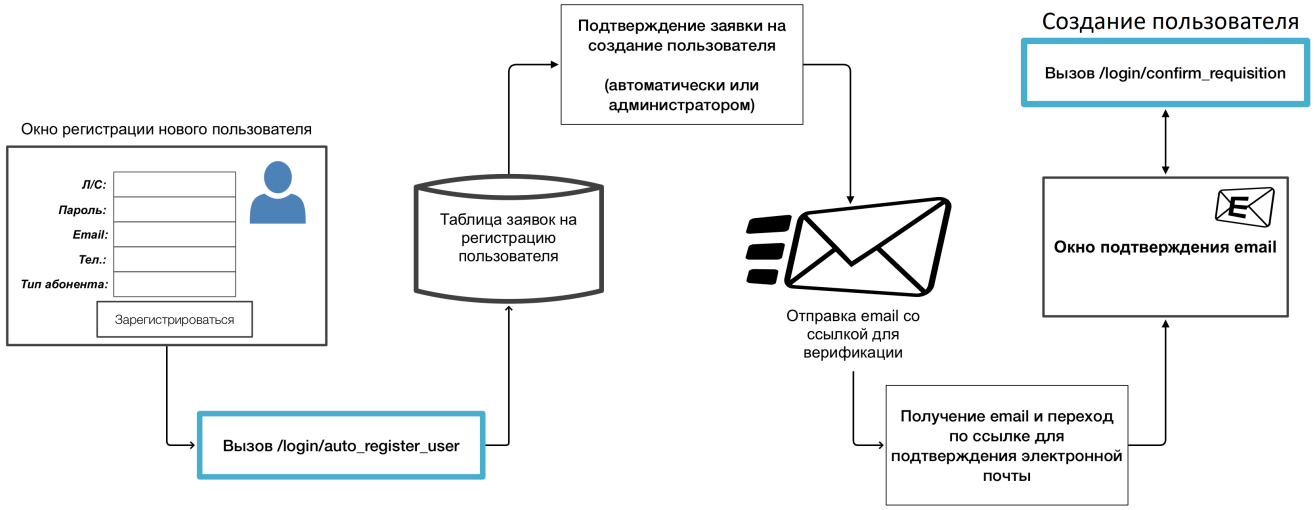
На данном этапе пользователю отправляется email на указанный при регистрации заявки адрес электронной почты со ссылкой для активации процесса создания пользователя. При переходе по ссылке в письме должен будет вызываться API метод "***/login/confirm_requisition***".

Верификация адреса email и создание пользователя

Когда пользователь переходит по ссылке из присланного ему электронного письма, то в браузере открывается страница на которой вызывается метод "***/login/confirm_requisition***". В данный метод передается UID заявки на создание нового пользователя (передается через соответствующий параметр в ссылке). На данном этапе происходит создание нового пользователя с параметрами сохраненными ранее в заявке на регистрацию нового пользователя.

В процессе регистрации нового пользователя создается новый WEB пользователь с логином идентичными лицевому счету (таблица БД SERVER_AUTH). Также создается и привязывается к WEB пользователю десктоп пользователь (таблица ACC_USER). Создается и привязывается к десктоп пользователю абонент (таблица BP2_ABONENT). Для физ. лица ищутся все ПУ (по полю PERSONAL_ACCOUNT таблицы SCHET_FR) и привязываются к абоненту. Для юр. лица также ищутся все ОУ (по полю DOG_N таблицы SREZ) и привязываются к десктоп пользователю, при этом все подчиненные для наденных ОУ ПУ также привязываются к абоненту.

Иллюстрация процесса регистрации в системе нового пользователя:



Процесс авторизации пользователя в системе

В окне авторизации пользователь вводит логин и пароль. Авторизация в системе осуществляется с помощью метода "***/login/user***". Введенный пользователем логин передается в открытом виде, а вместо пароля передается его хеш по алгоритму SHA256 (функция вычисления SHA256 на java script находится в файле ***sh256.hash.js***). При этом сам хеш пароля должен быть зашифрован с помощью 2048 битного открытого ключа RSA, полученного в API методе "***/get_open_key***".

В случае валидной пары логина и хеша пароля метод "***/login/user***" возвращает пару ***token*** и ***refreshToken***. ***token*** - имеет короткое время жизни (10 минут) и используется для вызова других методов REST API. "***/login/user***" имеет больший срок жизни (1 час) и используется для получения новой пары ***token*** и ***refreshToken*** с помощью метода "***/login/refresh***". Более подробное описание можно посмотреть в файле ***REST_API_server.yaml*** в формате swagger.

Процесс восстановления забытого пароля

Пароль можно восстановить с помощью адреса электронной указанного при регистрации нового пользователя. Для восстановления пароля должна быть предусмотрена отдельная страница на которую можно перейти со страницы авторизации.

Процесс восстановления пароля состоит из трех этапов:

1. Создание заявки на восстановление пароля. На этом этапе пользователь вводит адрес email указанный им при регистрации в системе и создается заявка на восстановление пароля.
2. Подтверждение заявки на восстановление пароля (администратором или в автоматическом режиме). На этом этапе пользователю отправляется письмо на адрес email указанный на предыдущем шаге со ссылкой на страницу ввода нового пароля.
3. Ввод нового пароля и завершение процесса обработки заявки на восстановление пароля.

Создание заявки на восстановление пароля

На странице восстановления пароля пользователь вводит адрес электронной почты и жмет кнопку восстановить пароль (также на данной странице желательно добавить CAPTCHA). После чего вызывается метод "**/login/forgot_password**" в который передаются введенный пользователем адрес электронной почты и url. url будет использоваться при формировании ссылки, передаваемой пользователю в письме для восстановления пароля. После чего создается заявка на восстановление пароля пользователя.

Ссылка в письме будет выглядеть следующим образом:

> ?uid=<xxxxx-xxxxxx-xxxxxx-xxxxxxx>

где - строка текста переданная в метод "***/login/forgot_password***" в параметре url, а <xxxxx-xxxxxx-xxxxxx-xxxxxxx> сгенерированный случайным образом uid заявки на восстановление пароля пользователя.

Подтверждение заявки на восстановление пароля

Процесс подтверждения заявки на восстановление пароля может осуществляться администратором или в автоматическом режиме (зависит от пожелания заказчика). Автоматический режим настраивается через таблицу в БД BP_VARIABLE.

На данном этапе пользователю отправляется email на указанный при регистрации адрес электронной почты со ссылкой для активации процесса восстановления пароля. При переходе по ссылке в письме пользователь попадает на страницу ввода нового пароля.

Ввод нового пароля и завершение процесса обработки заявки на восстановление пароля

На странице ввода нового пароля пользователь дважды вводит новый пароль в соответствующие поля "новый пароль" и "подтверждение пароля" и нажимает кнопку "задать новый пароль". После чего вызывается метод "***/login/confirm_forgot_password***" в который передается uid заявки на восстановление пароля полученный через ссылку в письме и зашифрованный хеш пароля введенного пользователем. При передаче пароля введенного пользователем необходимо произвести следующие действия:

- Получить RSA 2048 битный открытый ключ с помощью метода "***/get_open_key***";
- Вычислить хеш SHA256 (функция вычисления SHA256 на java script находится в файле ***sh256.hash.js***);
- Полученный на предыдущем шаге хеш пароля зашифровать открытым RSA ключом. По результату возвращаемому методом "***/login/confirm_forgot_password***" в случае успешного обновления пароля пользователя перекидывает на страницу авторизации, а в случае если не удалось обновить пароль пользователю выдается сообщение об ошибке с пояснением причины (см. описание метода "***/login/confirm_forgot_password***" в файле ***REST_API_server.yaml*** в формате swagger.).

Иллюстрация процесса восстановления пароля: