### Документация к сервису “База данных”

#### **Авторизация в приложении.**

Для того, чтобы попасть в приложение, нужно зарегистрироваться.

Сохраните аутентификационные данные, они пригодятся при работе с API.

#### **Рабочие вкладки**

Приложение “База данных” состоит из двух вкладок: “Создание клиента” и “Просмотр клиентов”.

Первая вкладка “Создание клиента” позволяет сохранить на сервер карточку клиента.

Вторая вкладка “Просмотр клиентов” представляет собой список клиентов, с возможностью открыть карточку каждого конкретного клиента. Также предусмотрена возможность отфильтровать список по параметрам ФИО, дата рождения и место проживания, но графический интерфейс страницы находится в разработке, а серверная логика уже реализована (подробнее про то какая серверная логика организована можно прочитать в разделе “Методы API”)

Карточка клиента состоит из 5 полей:

* Имя (обязательное поле)
* Фамилия (обязательное поле)
* Отчество (необязательное поле)
* Дата рождения (необязательное поле)
* Место проживания (необязательное поле)

#### **Валидации графического интерфейса и сервера**

В графическом интерфейсе предусмотрена валидация на обязательность полей “Имя” и “Фамилия” - незаполненные поля выделяются красной рамкой.

На второй вкладке предусмотрен просмотр пользователей. Но функциональность еще не реализована, поэтому там стоит заглушка.

#### 

#### **Методы API**

Карточка клиента сохраняется в базу данных

| Поле в графическом интерфейсе | Поле в в таблице на сервере |
| --- | --- |
| Имя | Name |
| Фамилия | Surname |
| Отчество | Patronymic |
| Дата рождения | DateOfBirth |
| Место проживания | Location |

Для работы с сервером реализованы следующие методы API

| **Метод** | **Название** | **Запрос** |
| --- | --- | --- |
| GET | Получение списка пользователей | [http://](http://database.ru/backendApi/usersCatalog)[{url}](http://database.ru/backendApi/usersCatalog/create)[/api/users](http://database.ru/backendApi/usersCatalog)/ |
| GET | Получение конкретного пользователя | [http://](http://database.ru/backendApi/usersCatalog)[{url}](http://database.ru/backendApi/usersCatalog/create)[/api/users](http://database.ru/backendApi/usersCatalog)/{userId} |
| POST | Создание пользователя | [http://{url}/api/users/create](http://database.ru/backendApi/usersCatalog/create)  Пример body в формате json:  {  "itemsOfClients": [  {  "id": int (число),  "person": {  "name": "string (строка)",  "surname": "string (строка)",  "patronymic": "string (строка)",  "dateOfBirth": "формат даты dd.mm.YYYY",  "location”: "string (строка)"  }  }  ]  } |
| PUT | Обновление пользователя | [http://](http://database.ru/backendApi/usersCatalog/update)[{url}](http://database.ru/backendApi/usersCatalog/create)[/api/users/{userId}/update](http://database.ru/backendApi/usersCatalog/update)  {  "update": {  "name": "string (строка)",  "surname": "string (строка)",  "dateOfBirth": "формат даты dd.mm.YYYY",  "location": "string (строка)"  }  } |
| DELETE | Удаление конкретного пользователя | [http://](http://database.ru/backendApi/usersCatalog/%7BuserId%7D/delete)[{url}](http://database.ru/backendApi/usersCatalog/create)[/api/users/{userId}/delete](http://database.ru/backendApi/usersCatalog/%7BuserId%7D/delete) |
| DELETE | Удаление всех пользователей | [http://](http://database.ru/backendApi/usersCatalog/%7BuserId%7D/delete)[{url}](http://database.ru/backendApi/usersCatalog/create)[/api/users/allUsers/delete](http://database.ru/backendApi/usersCatalog/%7BuserId%7D/delete) |

#### 

#### **Шаблон для заведения багов:**

| ID | B1 |
| --- | --- |
| Название | Маска ввода в поле “Дата рождения” формы содержит 6 знаков для ввода года рождения |
| Сценарий воспроизведения | 1. Переходим по ссылке <http://api-qa.skillbox.ru/practice2/>  2. Авторизуемся  3. Вносим значение “01.01.111111” в поле “Дата рождения” |
| Воспроизводимость | Всегда |
| Важность | Ниже средней |
| Приоритет | Средний |
| Ожидаемый результат | Поле “Дата рождения” позволяет внести сведения в формате ДД.ММ.ГГГГ |
| Фактический результат | Поле “Дата рождения” позволяет внести сведения в формате ДД.ММ.ГГГГГГ |
| Дополнительная информация | При вводе некорректных данных форма выдает ошибку о том что пользователь Null |

| ID | B2 |
| --- | --- |
| Название | При направлении формы с некорректной датой рождения пользователь получает сообщение непонятного содержания |
| Сценарий воспроизведения | 1. Переходим по ссылке <http://api-qa.skillbox.ru/practice2/>  2. Авторизуемся  3. Вносим значение “01.01.111111”  4. Нажимаем создать |
| Воспроизводимость | Всегда |
| Важность | Средняя |
| Приоритет | Средняя |
| Ожидаемый результат | Пользователь получает уведомление, содержащее ясную информацию о поле, в которое некорректно введены данные |
| Фактический результат | Пользователь получает уведомление “Users is null” |
| Дополнительная информация |  |

| ID | B3 |
| --- | --- |
| Название | Поля “Фамилия” и “Имя” не отмечены как обязательные к заполнению |
| Сценарий воспроизведения | 1. Переходим по ссылке <http://api-qa.skillbox.ru/practice2/>  2. Авторизуемся  3. Смотрим на форму |
| Воспроизводимость | Всегда |
| Важность | Средняя |
| Приоритет | Средняя |
| Ожидаемый результат | Поля “Фамилия” и “Имя” помечены как обязательные для заполнения |
| Фактический результат | Форма не содержит отметок об обязательных для заполнения полях |
| Дополнительная информация |  |

| ID | B4 |
| --- | --- |
| Название | В базе данных значение поля “Фамилия” помещается в поле “Имя”, а значение поля “Имя” в поле “Фамилия” |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “Александров”  3. В поле “Имя” вносим значение “Сергей”  4. Нажимаем кнопку создать  5. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Критическая |
| Приоритет | Высокий |
| Ожидаемый результат | Поле “Фамилия” содержит значение “Александров”, а поле “Имя” значение “Сергей” |
| Фактический результат | Поле “Фамилия” содержит значение “Сергей”, а поле “Имя” значение “Александров” |
| Дополнительная информация |  |

| ID | B5 |
| --- | --- |
| Название | Поля “Фамилия”, “Имя”, “Отчество”, “Место проживания” могут быть заполнены только пробелами |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим 5 пробелов  3. В поле “Имя” вносим 5 пробелов  4. В поле “Отчество” вносим 5 пробелов  5. В поле “Место проживания” вносим 5 пробелов  6. Нажимаем кнопку создать  7. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Высокая |
| Приоритет | Высокий |
| Ожидаемый результат | Пользователь получает сообщение об ошибке “Ввод некорректных данных” |
| Фактический результат | Форма воспринимает пробелы как валидные значения |
| Дополнительная информация | Форма допускает использование любых спецсимволов, а также цифр или только цифр для заполнения полей. |

| ID | B6 |
| --- | --- |
| Название | Форма/База данных допускает использование пробелов в начале, в середине и в конце строки |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “ Иванов Иван “  3. В поле “Имя” вносим значение “ Иванович чич “  4. Нажимаем кнопку создать  5. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Высокая |
| Приоритет | Высокий |
| Ожидаемый результат | Пользователь получает сообщение об ошибке “Ввод некорректных данных” |
| Фактический результат | Поле “Фамилия” содержит значение “ Иванов Иван “, а поле “Имя” содержит значение “ Иванович чич “ |
| Дополнительная информация | Таким образом, форма допускает ввод ФИО в одном поле. В последующем может быть затруднительно использовать данные из базы данных |

| ID | B7 |
| --- | --- |
| Название | Допускается использование спецсимвола “/” при заполнении полей формы, текст после спецсимвола становится ссылкой |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “Сидо/ров“  3. В поле “Имя” вносим значение “Николай“  4. Нажимаем кнопку создать  5. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Высокая |
| Приоритет | Высокий |
| Ожидаемый результат | Пользователь получает сообщение об ошибке “Ввод некорректных данных” |
| Фактический результат | Поле “Фамилия” содержит значение “Сидо/[ров](#_mk26bvje9m6g)“ |
| Дополнительная информация | Актуально для полей “Фамилия” и “Имя” текущей версии билда. |

| ID | B8 |
| --- | --- |
| Название | Сведения о клиенте с id = 1 не отображаются при запросе сведений о всех клиентах |
| Сценарий воспроизведения | 1. Создаем клиентов с id=1 и id=2 в приложении (заполняем для них поля “Фамилия” и “Имя”)  2. Запрашиваем сведения о всех клиентах |
| Воспроизводимость | Всегда |
| Важность | Высокая |
| Приоритет | Высокий |
| Ожидаемый результат | Отображаются сведения о двух клиентах |
| Фактический результат | Отображаются сведения только о клиенте с id=2 |
| Дополнительная информация | При регистрации сведения о пользователе попадают в базу данных клиентов с id=1. После удаления из базы всех сведений о клиентах и внесении новых клиентов первому клиенту присваивается id = 1 (если id не указан), а сведения о нем отображаются только при запросе сведений о клиенте с таким id |

| ID | B9 |
| --- | --- |
| Название | Сведения, внесенные в поле “Отчество”, не сохраняются в базе данных |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “Иванов”  3. В поле “Имя” вносим значение “Иван”  4. В поле “Отчество” вносим значение “Иванович”  5. Нажимаем кнопку создать  6. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Выше средней |
| Приоритет | Высокая |
| Ожидаемый результат | В поле “Отчество” сохраняется значение “Иванович” |
| Фактический результат | Поле “Отчество” содержит значение “Null” |
| Дополнительная информация |  |

| ID | B10 |
| --- | --- |
| Название | Сведения, внесенные в поле “Дата рождения”, не сохраняются в базе данных в случае, если дата рождения менее 1 января 1991 г. |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “И”  3. В поле “Имя” вносим значение “И”  4. В поле “Дата рождения” вносим значение “01.01.1990”  5. Нажимаем кнопку создать  6. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Выше средней |
| Приоритет | Высокая |
| Ожидаемый результат | В поле “Дата рождения” сохраняется значение “01.01.1990” |
| Фактический результат | Поле “Дата рождения” содержит значение “Null” |
| Дополнительная информация |  |

| ID | B11 |
| --- | --- |
| Название | Сведения, внесенные в поле “Дата рождения”, сохраняются в базе данных с добавлением к году 1 единицы в случае, если дата рождения более 1 января 1991 г. |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “И”  3. В поле “Имя” вносим значение “И”  4. В поле “Дата рождения” вносим значение “09.11.1991”  5. Нажимаем кнопку создать  6. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Выше средней |
| Приоритет | Высокая |
| Ожидаемый результат | В поле “Дата рождения” сохраняется значение “09.11.1991” |
| Фактический результат | Поле “Дата рождения” содержит значение “09.11.1992” |
| Дополнительная информация |  |

| ID | B12 |
| --- | --- |
| Название | Поле “Дата рождения” позволяет вводить значения больше текущей даты |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “Иванов”  3. В поле “Имя” вносим значение “Иван”  4. В поле “Дата рождения” вносим значение “01.01.2030”  5. Нажимаем кнопку создать  6. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Выше средней |
| Приоритет | Высокая |
| Ожидаемый результат | Пользователь получает сообщение об ошибке “Введенная дата, превышает текущую” |
| Фактический результат | Поле “Дата рождения” содержит значение “01.01.2031” |
| Дополнительная информация |  |

| ID | B13 |
| --- | --- |
| Название | Сведения, внесенные в поле “Место проживания”, не сохраняются в базе данных, если поле “Дата рождения” имеет значение не null |
| Сценарий воспроизведения | 1. Входим в форму  2. В поле “Фамилия” вносим значение “Иванов”  3. В поле “Имя” вносим значение “Иван”  4. В поле “Дата рождения” вносим значение “31.12.2000”  5. В поле “Место проживания” вносим значение “Москва”  6. Нажимаем кнопку создать  7. Проверяем запись в базе данных |
| Воспроизводимость | Всегда |
| Важность | Выше средней |
| Приоритет | Высокая |
| Ожидаемый результат | В поле “Место проживания” сохраняется значение “Москва” |
| Фактический результат | Поле “Место рождения” содержит значение “” |
| Дополнительная информация | Баг нужно перепроверить после редактирования функционала полей “Отчество” и “Дата рождения” |