# Создать пользователя - createUser

|  |  |
| --- | --- |
| **Название сценария** | Создать пользователя |
| **Название метода** | createUser |
| **Эндпоинт** | {apiUrl}/user/create |
| **Глагол** | POST |
| **Описание** | Метод создает нового пользователя |
| **Ожидаемый результат** | Возвращена информация о созданном пользователе  ИЛИ  Возвращено сообщение об ошибке |

## Входные параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| firstName | string | 1 | Body | Имя пользователя |
| lastName | string | 1 | Body | Фамилия пользователя |
| patronymic | string | 0..1 | Body | Отчество пользователя |
| login | string | 1 | Body | Логин пользователя |

## Выходные параметры

В случае успешного выполнения метод возвращает информацию о пользователе или сообщение об ошибке, формат ответа приведен ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | Long | 1 | Body | Идентификатор пользователя |
| firstName | string | 1 | Body | Имя пользователя |
| lastName | string | 1 | Body | Фамилия пользователя |
| patronymic | string | 0..1 | Body | Отчество пользователя |
| login | string | 1 | Body | Логин пользователя |
| archive | bool | 1 | Body | Архивный пользователь (Да/Нет) |

### Сообщения об ошибках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код ошибки** | **Текст ошибки** | **http-код** | **Параметры** | **Комментарий** |
| validationError | В запросе отсутствуют обязательные параметры | 400 | key  value | key – название не заполненного поля  value=null |
| databaseError | *Текст ошибки, полученный при обращении к базе данных* | 500 |  |  |

## Описание алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действие** | **АС** | **Код ошибки** |
| **Базовый сценарий. Создать пользователя** | | | |
| 1 | Система извлекает данные из запроса метода и проверяет заполненность обязательных полей. | АС.1 | validationError |
| 2 | Система проверяет, что в таблице user не существует записи, для которой выполнено условие  user.login == *login*  И  user.archive == false | АС.2 | databaseError |
| 3 | Система создает новую запись в таблице user. Запись заполняется по следующему правилу:  user.id = автоматически сгенерированный идентификатор  user.firstname = *firstName*  user.lastname = *lastName*  user.patronymic = *patronymic*  user.login = *login*  user.archive = false | АС.2 | databaseError |
| 4 | Система заполняет ответ по следующему правилу:  **id** = идентификатор созданной записи  **firstName** = *firstName*  **lastName** = *lastName*  **patronymic** = *patronymic*  **login** = *login*  **archive** = false |  |  |
| 5 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |  |  |
|  | Сценарий завершен. |  |  |

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = validationError  **error.message** = В запросе отсутствуют обязательные параметры  **error.details** - коллекция объектов вида  **key** = <наименование не заполненного поля>  **value** = null  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.2 При обращении к базе данных произошла ошибка

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = databaseError  **error.message** = <текст ошибки, полученный при обращении к базе данных>  **error.details** – пустая коллекция  и возвращает ответ с http-кодом 500. |
|  | Сценарий завершен |

# Изменить данные пользователя - updateUser

|  |  |
| --- | --- |
| **Название сценария** | Изменить данные пользователя |
| **Название метода** | updateUser |
| **Эндпоинт** | {apiUrl}/user/update |
| **Глагол** | POST |
| **Описание** | Метод изменяет данные существующего пользователя |
| **Ожидаемый результат** | Возвращена информация о пользователе  ИЛИ  Возвращено сообщение об ошибке |

## Входные параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Body | Идентификатор пользователя |
| firstName | string | 0..1 | Body | Имя пользователя |
| lastName | string | 0..1 | Body | Фамилия пользователя |
| patronymic | string | 0..1 | Body | Отчество пользователя |
| login | string | 0..1 | Body | Логин пользователя |

## Выходные параметры

В случае успешного выполнения метод возвращает информацию о пользователе или сообщение об ошибке, формат ответа приведен ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Body | Идентификатор пользователя |
| firstName | string | 1 | Body | Имя пользователя |
| lastName | string | 1 | Body | Фамилия пользователя |
| patronymic | string | 0..1 | Body | Отчество пользователя |
| login | string | 1 | Body | Логин пользователя |
| archive | bool | 1 | Body | Архивный пользователь (Да/Нет) |

### Сообщения об ошибках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код ошибки** | **Текст ошибки** | **http-код** | **Параметры** | **Комментарий** |
| validationError | В запросе отсутствуют обязательные параметры | 400 | key  value | key – название не заполненного поля  value=null |
| databaseError | *Текст ошибки, полученный при обращении к базе данных* | 500 |  |  |
| userNotFoundError | Пользователь с указанным идентификатором не найден | 404 | key  value | key=id  value=идентификатор из входных параметров |
| userIsArchiveError | Пользователь с указанным идентификатором является архивным | 400 | key  value | key=id  value=идентификатор из входных параметров |
| userAlreadyExistError | Пользователь с указанным логином уже существует | 400 | key  value | key=login  value=идентификатор найденного пользователя |

## Описание алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действие** | **АС** | **Код ошибки** |
| **Базовый сценарий. Изменить данные пользователя** | | | |
| 1 | Система извлекает данные из запроса метода и проверяет заполненность обязательных полей. | АС.1 | validationError |
| 2 | Система осуществляет поиск записи в таблице user по переданному идентификатору пользователя и запоминает найденную запись. | АС.2  АС.3 | databaseError  userNotFoundError |
| 3 | Система проверяет, что для полученной на предыдущем шаге записи не выполнено условие  user.archive == true | АС.4 | userIsArchiveError |
| 4 | **Выполняется если *login* не пуст и передан.**  Система проверяет, что в таблице user не существует записи, для которой выполнено условие  user.id != *id*  И  user.login == *login*  И  user.archive == false | АС.2  АС.5 | databaseError  userAlreadyExistError |
| 5 | Система исключает из запроса те параметры, которые совпадают по значению с соответствующими полями записи с шага 2. |  |  |
| 6 | Система проверяет, что в результате выполнения предыдущего шага остались параметры для изменения. | АС.6 |  |
| 7 | Система изменяет запись в таблице user. Запись заполняется по следующему правилу:  user.firstname = *firstName*  user.lastname = *lastName*  user.patronymic = *patronymic*  user.login = *login*  ***Примечание:*** *изменяются только те поля, которые не были исключены по результатам шага 5.* | АС.2 | databaseError |
| 8 | Система заполняет ответ по следующему правилу:  **id** = идентификатор записи  **firstName** = значение поляuser.firstname  **lastName** = значение поляuser. lastname  **patronymic** =значение поляuser.patronymic  **login** =значение поляuser.login  **archive** = значение поляuser.archive |  |  |
| 9 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |  |  |
|  | Сценарий завершен. |  |  |

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = validationError  **error.message** = В запросе отсутствуют обязательные параметры  **error.details** - коллекция объектов вида  **key** = <наименование не заполненного поля>  **value** = null  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.2 При обращении к базе данных произошла ошибка

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = databaseError  **error.message** = <текст ошибки, полученный при обращении к базе данных>  **error.details** – пустая коллекция  и возвращает ответ с http-кодом 500. |
|  | Сценарий завершен |

### АС.3 Пользователь по переданному идентификатору не найден

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = userNotFoundError  **error.message** = Пользователь с указанным идентификатором не найден  **error.details** - коллекция, содержащая объект вида  **key**=id  **value**=идентификатор из входных параметров  и возвращает ответ с http-кодом 404. |
|  | Сценарий завершен |

### АС.4 Пользователь по переданному идентификатору находится в архиве

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = userIsArchiveError  **error.message** = Пользователь с указанным идентификатором является архивным  **error.details** - коллекция, содержащая объект вида  **key**=id  **value**=идентификатор из входных параметров  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.5 Пользователь с переданным логином уже существует

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = userAlreadyExistError  **error.message** = Пользователь с указанным логином уже существует  **error.details** - коллекция, содержащая объект вида  key=login  value=идентификатор найденного пользователя  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.6 Переданные в запрос параметры совпадают с установленными у записи

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ по следующему правилу:  **id** = идентификатор записи  **firstName** = значение поляuser.firstname  **lastName** = значение поляuser. lastname  **patronymic** =значение поляuser.patronymic  **login** =значение поляuser.login  **archive** = значение поляuser.archive |
| 2 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |
|  | Сценарий завершен |

# Архивировать пользователя - archiveUser

|  |  |
| --- | --- |
| **Название сценария** | Архивировать пользователя |
| **Название метода** | archiveUser |
| **Эндпоинт** | {apiUrl}/user/archive |
| **Глагол** | POST |
| **Описание** | Метод архивирует запись о пользователе |
| **Ожидаемый результат** | Возвращен идентификатор записи о пользователе  ИЛИ  Возвращено сообщение об ошибке |

## Входные параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Body | Идентификатор пользователя |

## Выходные параметры

В случае успешного выполнения метод возвращает иденификатор пользователе или сообщение об ошибке, формат ответа приведен ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Body | Идентификатор пользователя |

### Сообщения об ошибках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код ошибки** | **Текст ошибки** | **http-код** | **Параметры** | **Комментарий** |
| validationError | В запросе отсутствуют обязательные параметры | 400 | key  value | key – название не заполненного поля  value=null |
| databaseError | *Текст ошибки, полученный при обращении к базе данных* | 500 |  |  |
| userNotFoundError | Пользователь с указанным идентификатором не найден | 404 | key  value | key=id  value=идентификатор из входных параметров |

## Описание алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действие** | **АС** | **Код ошибки** |
| **Базовый сценарий. Создать пользователя** | | | |
| 1 | Система извлекает данные из запроса метода и проверяет заполненность обязательных полей. | АС.1 | validationError |
| 2 | Система осуществляет поиск записи в таблице user по следующему правилу:  user.id = *id*  и запоминает найденную запись. | АС.2  АС.3 | databaseError  userNotFoundError |
| 3 | Система проверяет, что полученная запись не является архивной -- archive == false. | АС.2  АС.4 | databaseError |
| 4 | Система устанавливает записи с шага 2 признак архивности.  User.archive=true. |  |  |
| 5 | Система заполняет ответ по следующему правилу:  **id** = *id* |  |  |
| 6 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |  |  |
|  | Сценарий завершен. |  |  |

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = validationError  **error.message** = В запросе отсутствуют обязательные параметры  **error.details** - коллекция объектов вида  **key** = <наименование не заполненного поля>  **value** = null  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.2 При обращении к базе данных произошла ошибка

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = databaseError  **error.message** = <текст ошибки, полученный при обращении к базе данных>  **error.details** – пустая коллекция  и возвращает ответ с http-кодом 500. |
|  | Сценарий завершен |

### АС.3 Пользователь по переданному идентификатору не найден

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = userNotFoundError  **error.message** = Пользователь с указанным идентификатором не найден  **error.details** - коллекция, содержащая объект вида  **key**=id  **value**=идентификатор из входных параметров  и возвращает ответ с http-кодом 404. |
|  | Сценарий завершен |

### АС.4 Запись о пользователе по переданному идентификатору уже является архивной

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ по следующему правилу:  **id** = *id* |
| 2 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |
|  | Сценарий завершен |

# Получить пользователя по идентификатору - getUserById

|  |  |
| --- | --- |
| **Название сценария** | Получить пользователя по идентификатору |
| **Название метода** | getUserById |
| **Эндпоинт** | {apiUrl}/user/{id} |
| **Глагол** | GET |
| **Описание** | Метод возвращает информацию о пользователе |
| **Ожидаемый результат** | Возвращена информация о пользователе  ИЛИ  Возвращено сообщение об ошибке |

## Входные параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Path | Идентификатор пользователя |

## Выходные параметры

В случае успешного выполнения метод возвращает информацию о пользователе или сообщение об ошибке, формат ответа приведен ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| id | long | 1 | Body | Идентификатор пользователя |
| firstName | string | 1 | Body | Имя пользователя |
| lastName | string | 1 | Body | Фамилия пользователя |
| patronymic | string | 0..1 | Body | Отчество пользователя |
| login | string | 1 | Body | Логин пользователя |
| archive | bool | 1 | Body | Архивный пользователь (Да/Нет) |

### Сообщения об ошибках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код ошибки** | **Текст ошибки** | **http-код** | **Параметры** | **Комментарий** |
| validationError | В запросе отсутствуют обязательные параметры | 400 | key  value | key – название не заполненного поля  value=null |
| databaseError | *Текст ошибки, полученный при обращении к базе данных* | 500 |  |  |
| userNotFoundError | Пользователь с указанным идентификатором не найден | 404 | key  value | key=id  value=идентификатор из входных параметров |

## Описание алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действие** | **АС** | **Код ошибки** |
| **Базовый сценарий. Создать пользователя** | | | |
| 1 | Система извлекает данные из запроса метода и проверяет заполненность обязательных полей и их тип. | АС.1 | validationError |
| 2 | Система осуществляет поиск записи в таблице user по следующему правилу:  user.id = *id*  и запоминает найденную запись. | АС.2  АС.3 | databaseError  userNotFoundError |
| 3 | Система заполняет ответ по следующему правилу:  **id** = идентификатор записи  **firstName** = значение поляuser.firstname  **lastName** = значение поляuser. lastname  **patronymic** =значение поляuser.patronymic  **login** =значение поляuser.login  **archive** = значение поляuser.archive |  |  |
| 4 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |  |  |
|  | Сценарий завершен. |  |  |

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = validationError  **error.message** = В запросе отсутствуют обязательные параметры  **error.details** - коллекция объектов вида  **key** = <наименование не заполненного поля>  **value** = null  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.2 При обращении к базе данных произошла ошибка

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = databaseError  **error.message** = <текст ошибки, полученный при обращении к базе данных>  **error.details** – пустая коллекция  и возвращает ответ с http-кодом 500. |
|  | Сценарий завершен |

### АС.3 Пользователь по переданному идентификатору не найден

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = userNotFoundError  **error.message** = Пользователь с указанным идентификатором не найден  **error.details** - коллекция, содержащая объект вида  **key**=id  **value**=идентификатор из входных параметров  и возвращает ответ с http-кодом 404. |
|  | Сценарий завершен |

# Поиск пользователей по параметрам - findUsers

|  |  |
| --- | --- |
| **Название сценария** | Поиск пользователей по параметрам |
| **Название метода** | findUsers |
| **Эндпоинт** | {apiUrl}/user/find |
| **Глагол** | POST |
| **Описание** | Метод возвращает информацию о найденных пользователях |
| **Ожидаемый результат** | Возвращена коллекция записей о пользователях и результаты пагинации  ИЛИ  Возвращено сообщение об ошибке |

## Входные параметры

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название параметра** | | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| firstName | | string | 0..1 | Body | Подстрока поиска пользователей по имени без учета регистра |
| lastName | | string | 0..1 | Body | Подстрока поиска пользователей по фамилии без учета регистра |
| patronymic | | string | 0..1 | Body | Подстрока поиска пользователей по отчеству без учета регистра |
| login | | string | 0..1 | Body | Подстрока поиска пользователей по логину без учета регистра |
| archive | | bool | 0..1 | Body | Фильтр поиска по архивности записи |
| pagingOptions | | Группирующая сущность | 0..1 | Body | Параметры пагинации и сортировки результатов.  Если узел не заполнен, используются следующие значения:  pageNumber = 0  pageSize = 20  sortField = id  descending = false |
|  | pageNumber | Long | 1 | Body | Номер запрашиваемой страницы. Нумерация страниц начинается с 0. |
|  | pageSize | Long | 1 | Body | Размер запрашиваемой страницы |
|  | sortField | String | 1 | Body | Наименование поля для сортировки. Обрабатываются следующие значения:  id – идентификатор записи  firstName – фамилия пользователя  lastName – имя пользователя  patronymic – отчество пользователя  login – логин пользователя  archive – запись является архивной (Да/Нет) |
|  | descending | Bool | 1 | Body | Сортировка по убыванию (Да/Нет) |

## Выходные параметры

В случае успешного выполнения метод возвращает коллекцию записей о пользователях и результаты пагинации или сообщение об ошибке, формат ответа приведен ниже.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название параметра** | | **Тип данных** | **Обяз./Множ.** | **Размещение** | **Описание** |
| users | | User[] | 1..N | Body | Коллекция записей о пользователях, удовлетворяющих условиям фильтрации |
|  | id | long | 1 | Body | Идентификатор пользователя |
|  | firstName | string | 1 | Body | Имя пользователя |
|  | lastName | string | 1 | Body | Фамилия пользователя |
|  | patronymic | string | 0..1 | Body | Отчество пользователя |
|  | login | string | 1 | Body | Логин пользователя |
|  | archive | bool | 1 | Body | Архивный пользователь (Да/Нет) |
| pegingResult | | Группирующая сущность | 1 | Body | Результаты пагинации |
|  | pageNumber | long | 1 | Body | Номер текущей страницы |
|  | pageSize | long | 1 | Body | Количество элементов на странице |
|  | pageTotal | long | 1 | Body | Общее количество страниц выборки |
|  | itemsCount | long | 1 | Body | Общее количество элементов выборки |
|  | hasNextPage | bool | 1 | Body | Доступна следующая страница (Да/Нет) |

### Сообщения об ошибках

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код ошибки** | **Текст ошибки** | **http-код** | **Параметры** | **Комментарий** |
| validationError | Переданы некорректные параметры запроса | 400 | key  value | key – название поля  value – переданное значение поля |
| databaseError | *Текст ошибки, полученный при обращении к базе данных* | 500 |  |  |

## Описание алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действие** | **АС** | **Код ошибки** |
| **Базовый сценарий. Поиск пользователей по параметрам** | | | |
| 1 | Система извлекает данные из запроса метода, проверяет заполненность обязательных параметров и следующие условия:   * pagingOptions.pageNumber >= 0 * pagingOptions.pageSize >= 1 * значение pagingOptions.sortField входит в список допустимых | АС.1 | validationError |
| 2 | Система осуществляет поиск записей в таблице user по следующему правилу:  upper(user.firstname) like ‘%upper(*firstName*)%’  *И*  upper(user.lastname) like ‘%upper(*lastName*)%’  И  upper(user.patronymic) like ‘%upper(*patronymic*)%’  И  upper(user.login) like ‘%upper(*login*)%’  И  user.archive = *archive*  Дополнительно Система накладывает критерий сортировки выборки по переданным значениям *pagingOptions.sortField* и *pagingOptions.descending* и параметры пагинации  LIMIT *pagingOptions*.pageSize  OFFSET (*pagingOptions*.*pageNumber*) \* *pagingOptions*.pageSize  Система запоминает найденные записи.  ***Примечание:*** *в запросе участвуют только не пустые и переданные поля фильтрации. Если никакой фильтр не заполнен, ограничений на выборку не накладывается.* | АС.2 | databaseError |
| 3 | Система запрашивает общее количество записей в таблице user по критерию с шага 2 и запоминает количество записей. | АС.2 | databaseError |
| 4 | Система заполняет ответ по следующему правилу:  **users** – коллекция записей о пользователях с шага 2  **pаgingResult.pageNumber** = *pagingOptions*.*pageNumber*  **pаgingResult.pageSize** – количество элементов, полученных на шаге 2  **pаgingResult.pageTota**l - количество элементов, полученных на шаге 3, деленное на *pagingOptions*.pageSize  **pаgingResult.itemsCount** – количество элементов, полученных на шаге 3  **pаgingResult.hasNextPage** – результат проверки условия **pаgingResult.pageTota**l – 1 > *pagingOptions*.*pageNumber* |  |  |
| 5 | Ответ метода возвращается запрашивающей системе с http-кодом 200. |  |  |
|  | Сценарий завершен. |  |  |

## Альтернативные сценарии

### АС.1 Переданы некорректные параметры запроса

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = validationError  **error.message** = Переданы некорректные параметры запроса  **error.details** - коллекция объектов вида  **key** = <наименование поля>  **value** = <переданное значение>  и возвращает ответ с http-кодом 400. |
|  | Сценарий завершен |

### АС.2 При обращении к базе данных произошла ошибка

|  |  |
| --- | --- |
| **№** | **Действие системы** |
| 1 | Система заполняет ответ метода по следующему правилу:  **error.code** = databaseError  **error.message** = <текст ошибки, полученный при обращении к базе данных>  **error.details** – пустая коллекция  и возвращает ответ с http-кодом 500. |
|  | Сценарий завершен |

# Объекты

## User

Объект содержит информацию о пользователе.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Описание** |
| id | long | 1 | Идентификатор пользователя |
| firstName | string | 1 | Имя пользователя |
| lastName | string | 1 | Фамилия пользователя |
| patronymic | string | 0..1 | Отчество пользователя |
| login | string | 1 | Логин пользователя |
| archive | bool | 1 | Архивный пользователь (Да/Нет) |

### Пример

{

"id": 1,

"firstName":"Иванов",

"lastName":"Иван",

"patronymic":"Иванович",

"login": "ivanov",

"archive": false

}

## Error

Объект содержит информацию о произошедшей ошибке.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название параметра** | | **Тип данных** | **Обяз./Множ.** | **Описание** |
| code | | string | 1 | Код возникшей ошибки |
| message | | string | 1 | Сообщение об ошибке |
| details | | Коллекция | 1 | Коллекция пар ключ-значение |
|  | key | string | 1 | Ключ |
|  | value | string | 1 | Значение |

### Пример

{

"code": "validationError",

"message": "В запросе отсутствуют обязательные параметры",

"details": [

{

"key": "login"

"value": null

}

]

}

## PagingOptions

Объект содержит информацию о параметрах пагинации.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Описание** |
| pageNumber | long | 1 | Номер запрашиваемой страницы. Нумерация страниц начинается с 0. |
| pageSize | long | 1 | Размер запрашиваемой страницы |
| sortField | string | 1 | Наименование поля для сортировки |
| descending | bool | 1 | Сортировка по убыванию (Да/Нет) |

### Пример

{

"pageNumber": 1,

"pageSize ": 50,

"sortField ":"id",

"descending ": true

}

## PagingResults

Объект содержит информацию о результатах пагинации.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название параметра** | **Тип данных** | **Обяз./Множ.** | **Описание** |
| pageNumber | long | 1 | Номер текущей страницы |
| pageSize | long | 1 | Количество элементов на странице |
| pageTotal | long | 1 | Общее количество страниц выборки |
| itemsCount | long | 1 | Общее количество элементов выборки |
| hasNextPage | bool | 1 | Доступна следующая страница (Да/Нет) |

### Пример

{

"pageNumber": 1,

"pageSize ": 35,

"pageTotal ":7,

"itemsCount": 240,

"hasNextPage": true

}