Редакция 7 от 26 августа, 21:30

Финальный раунд

Задача “Проверяющая система (Чекер)”

Введение

Вам предстоит разработать с использованием Spring Framework бэкенд сервис, реализующего функционал проверяющей системы. Разработка фронтенда системы в задачу не входит.
От системы требуется реализация API, описанного ниже.

Формулировка задачи

В этом раунде Вы должны создать проверяющую системы и протестить решения всех остальных участников. Api будет похожим, как и в первом раунде, но с небольшими изменениями:

- Сначала, Вам нужно будет создать результат ответа на тот или иной запрос
- Постом отправить его
- При запросе указать id ранее созданного результата в параметрах

Редакция 7 от 26 августа, 21:30

Описание API, с которым придется работать

Процесс тестирования

Тестирование происходит циклически, каждая итерация инициируется обращением к базовому эндпоинту Вашей проверяющей системы:

Создание итерации запроса GET
/api/start

Описание

При обращении к описанному выше эндпоинту все Ваши запросы, включая как и стандартные, так и запросы на добавление результатов, должны будут выполниться и попасть в тестирующий Proxy-сервер. При недоступности данного эндпоинта Ваше решение не считается валидным.

Универсальные запросы

Создание результата запроса POST

/api/single-query/add-new-query-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата создания sql-запроса.

Особенности обработки

По результатам должен быть добавлен sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

Редакция 7 от 26 августа, 21:30

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия. Возможные коды для добавления: 400, 201

При возникновении ошибок должен прийти код результата 403.
Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
    "resultId": 1,
    "code":400
}
```

Примеры неправильного JSON:

```
{
  "resultId": 1,
  "code":406
}
```

Создание нового запроса таблицы POST /api/single-query/add-new-query?resultId=id

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	queryId	integer	ИСТИНА	id запроса
2	query	string(120)	ИСТИНА	Запрос
3	resultId	integer	ИСТИНА	resultId
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Редакция 7 от 26 августа, 21:30

Описание

Метод предназначен для создания sql-запроса.

Особенности обработки

По результатам должен быть добавлен sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат id
- Не существует заданного resultId

При возникновении ошибок должен прийти код результата 403.

Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
    "queryId": 1,
    "query": "select * from Cars"
}
```

```
{
    "queryId": 2,
    "query": "select * from Motos"
}
```

```
{
  "queryId": 3,
  "query": "select * from Customer"
}
```

Примеры неправильного JSON:

```
{
  "queryId": abc,
  "query": "раз dva"
}
```

Редакция 7 от 26 августа, 21:30

Пример ответа на неправильный JSON:


```
{
  "timestamp": "2022-08-12T01:32:09.533+00:00",
  "path": "/api/single-query/add-new-query",
  "status": 400,
  "error": "Bad Request",
  "requestId": "c493e828-7"
}
```


Создание результата изменения POST /api/single-query/add-modify-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата изменения sql-запроса.

Редакция 7 от 26 августа, 21:30

Особенности обработки

По результатам должен быть добавлен sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 400.
Если ошибок нет, то 202.

Примеры правильного JSON:

```
{
    "resultId": 1,
    "code":406
}
```

Примеры неправильного JSON:

```
{
  "resultId": 1,
  "code":400
}
```

Редакция 7 от 26 августа, 21:30

Изменение запроса PUT

/api/single-query/modify-single-query?resultId=id

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	queryId	integer	ИСТИНА	id запроса
2	query	string(120)	ИСТИНА	Запрос
3	resultId	integer	ИСТИНА	resultId
Выходные параметры				
3	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен модификации sql-запроса.

Особенности обработки

По результатам должен быть изменен существующий запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Запроса с таким id не существует
- Недопустимый формат id
- Не существует заданного resultId

При возникновении ошибок должен прийти код результата 406.
При внутренних ошибках на нашей стороне придет ответ 400.
Если ошибок нет, то 200.

Редакция 7 от 26 августа, 21:30

Примеры правильного JSON:

```
{
    "queryId": 1,
    "query": "select * from Planes"
}
```

Примеры неправильного JSON:

```
{
    "queryId": 1a,
    "query": "select * from Planes"
}
```

Редакция 7 от 26 августа, 21:30

Создание результата удаления POST /api/single-query/add-delete-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата удаления sql-запроса.

Особенности обработки

По результатам должен быть удален sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 400.

Если ошибок нет, то 202.

Примеры правильного JSON:

```
{
    "resultId": 1,
    "code":406
}
```

Примеры неправильного JSON:

```
{
  "resultId": 1,
```

Редакция 7 от 26 августа, 21:30


```
  "code":200
}
```


Удаление запроса DELETE /api/single-query/delete-single-query-by-id/{id}?resultId=id

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	id	integer	ИСТИНА	id запроса
2	resultId	integer	ИСТИНА	resultId
Выходные параметры				
3	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для удаления sql-запроса по id.

Особенности обработки

По результатам должен быть удален существующий запрос. При невозможности удаления должны быть обработаны следующие ошибки:

Редакция 7 от 26 августа, 21:30

- Запроса с таким id не существует
- Не существует заданного resultId

При возникновении ошибок должен прийти код результата 406.
Если resultId не существует, то должен прийти код 400
Если ошибок нет, то 202.

Пример правильного запроса:

DELETE /api/single-query/delete-single-query-by-id/1?resultId=1

Пример неправильного запроса:

DELETE /api/single-query/delete-single-query-by-id/sting

Создание результата запуска запроса POST /api/single-query/add-execute-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата удаления sql-запроса.

Особенности обработки

По результатам должен запущен sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId

Редакция 7 от 26 августа, 21:30

- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 400.
Если ошибок нет, то 202.

Примеры правильного JSON:

```
{
    "resultId": 1,
    "code":406
}
```

Примеры неправильного JSON:

```
{
  "resultId": 1,
  "code":200
}
```

Запуск запроса GET /api/single-query/execute-single-query-by-id/{id}?resultId=id

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	id	integer	ИСТИНА	id запроса
2	resultId	integer	ИСТИНА	resultId
Выходные параметры				
3	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для запуска sql-запроса по id.

Редакция 7 от 26 августа, 21:30

Особенности обработки

По результатам должен быть запущен существующий запрос. При невозможности запуска должны быть обработаны следующие ошибки:

- Запроса с таким id не существует
- Синтаксис запроса неверный
- Не существует заданного resultId

При возникновении ошибок должен прийти код результата 400.
При возникновении ошибок с выполнением запроса должен прийти код 406.
Если ошибок нет, то 201.

Пример правильного запроса:

GET /api/single-query/execute-single-query-by-id/1?resultId=1

Пример неправильного запроса:

GET /api/single-query/execute-single-query-by-id/sting

Редакция 7 от 26 августа, 21:30

Создание результата получения запроса POST /api/single-query/add-get-single-query-by-id-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
3	queryId	integer	ИСТИНА	queryId
4	query	string	ИСТИНА	Запрос
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата получения sql-запроса.

Особенности обработки

По результатам должен запущен sql-запрос. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 400.
Если ошибок нет, то 202.

Примеры правильного JSON:

```
{
    "resultId": 1,
    "code":500,
    “queryId”: 1,
    “query”: “select * from Tokens”
}
```

Редакция 7 от 26 августа, 21:30

Примеры неправильного JSON:

```
{
   "resultId": 1,
   "code":201
}
```

Получение запроса по id GET /api/single-query/get-single-query-by-id/{id}?resultId=id

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	id	integer	ИСТИНА	id таблицы
2	resultId	integer	Истина	id результата
Выходные параметры				
1	queryId	integer	ИСТИНА	id запроса
3	query	string(120)	ИСТИНА	Запрос

Описание

Метод предназначен для получения sql-запроса по id.

Особенности обработки

По результатам должен быть получен JSON с запросом по id. При невозможности запуска должны быть обработаны следующие ошибки:

- Запроса с таким id не существует
- Не существует заданного resultId

Если запроса не существует, ответ с кодом 400.

Редакция 7 от 26 августа, 21:30

Если ошибок нет, то код 200 с запросом.

При ошибках запрос вернет код ошибки 500.

Пример правильного запроса:

GET /api/single-query/get-single-query-by-id/1?resultId=1

Пример правильного ответа:


```
{
  "queryId": 1,
  "query": "select * from Cars"
}
```


Пример неправильного запроса:

GET /api/single-query/get-single-query-by-id/10

Пример ответа на неправильный запрос:


```
{
  "timestamp": "2022-08-12T01:47:20.416+00:00",
  "path": "/api/single-query/get-single-query-by-id/10",
  "status": 500,
  "error": "Internal Server Error",
  "requestId": "c493e828-11"
}
```

Редакция 7 от 26 августа, 21:30

Получение всех запросов GET /api/single-query/get-all-single-queries

№	Название	Тип	Обязательность	Описание
Выходные параметры				
1	querysList	list	ИСТИНА	Все запросы

Описание

Метод предназначен для получения всех sql-запросов.

Особенности обработки

По результатам должен быть получен JSON со всеми запросами.

Пример правильного запроса:

GET /api/single-query/get-all-single-queries

Примеры правильного ответа:

Редакция 7 от 26 августа, 21:30

```
[
  {
    "queryId": 1,
    "query": "select * from Cars"
  },
  {
    "queryId": 2,
    "query": "select * from Motos"
  }
]
```

Если запросов нет:

```
[]
```

Запросы таблиц

Создание нового запроса таблицы POST /api/table-query/add-new-query-to-table?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	queryId	integer	ИСТИНА	id запроса
2	tableName	string(50)	ИСТИНА	Имя таблицы
3	query	string(120)	ИСТИНА	Запрос
4	resultId	integer	ИСТИНА	resultId
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Редакция 7 от 26 августа, 21:30

Описание

Метод предназначен для создания sql-запроса в рамках выбранной таблицы.

Особенности обработки

По результатам должен быть добавлен sql-запрос для заданной таблице. При невозможности добавления должны быть обработаны следующие ошибки:

- Таблица с таким именем не существует
- Недопустимый формат id

При возникновении ошибок должен прийти код результата 406.

Если resultId не существует или он невозможен, код 400

Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
  "queryId": 2,
  "tableName": "Artists",
  "query": "select * from Artists"
}
```

```
{
  "queryId": 1,
  "tableName": "Artists",
  "query": "раз dva"
}
```

Редакция 7 от 26 августа, 21:30

Создание результата получения запроса POST /api/table-query/add-new-query-to-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
3	tableQuery	TableQuery	ИСТИНА	Модель запроса
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата добавления табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат добавления sql-запроса. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.
- Допустимые статусы: 406, 201

При возникновении ошибок должен прийти код результата 400.
Если ошибок нет, то 202.

Примеры правильного JSON:

```
{
  "resultId": 2,
  "code": 201,
  "tableQuery" : {
     "queryId": 1,
     "tableName": "Artist",
     "query": "select * from Artist"
```

Редакция 7 от 26 августа, 21:30

```
  }
}
```

Получение всех запросов таблицы GET /api/table-query/get-all-queries-by-table-name/{name}?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	name	string	ИСТИНА	Имя таблицы
Выходные параметры				
3	tableQueries	list	ИСТИНА	Список запросов

Описание

Метод предназначен для получения всех sql-запросов в рамках выбранной таблицы имени таблицы.

Особенности обработки

По результатам должен быть получен JSON с запросами по конкретной таблице. Привязка говорит о целевой таблице запроса. При невозможности запуска должны быть обработаны следующие ошибки:

- Таблицы с таким именем не существует

Если таблицы не существует, должен вернуться полностью пустой ответ с кодом 200. Если ошибок нет, то код 200 со всеми запросами.

Пример правильного запроса:

Редакция 7 от 26 августа, 21:30

GET /api/table-query/get-all-queries-by-table-name/Customer?resultId=1

Создание результата получения ответа GET /api/table-query/add-new-query-to-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
3	tableQuery	TableQuery	ИСТИНА	Модель запроса
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата добавления табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат добавления sql-запроса. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия. Возможные коды ответа: 406, 201

При возникновении ошибок должен прийти код результата 400.
Если ошибок нет, то 202.

Редакция 7 от 26 августа, 21:30

Примеры правильного JSON:

```
{
  "resultId": 2,
  "code": 201,
  "tableQuery" : {
     "queryId": 1,
     "tableName": "Artist",
     "query": "select * from Artist"
  }
}
```

Получение запроса таблицы по id GET /api/table-query/get-table-query-by-id/{id}?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	id	integer	ИСТИНА	id таблицы
Выходные параметры				
1	queryId	integer	ИСТИНА	id запроса
2	tableName	string(50)	ИСТИНА	Имя таблицы
3	query	string(120)	ИСТИНА	Запрос

Описание

Метод предназначен для получения sql-запроса по id.

Особенности обработки

По результатам должен быть получен JSON с запросом по id. При невозможности запуска должны быть обработаны следующие ошибки:

- Запроса с таким id не существует

Если запроса не существует, ответ с кодом 500.

Редакция 7 от 26 августа, 21:30

Если ошибок нет, то код 200 с запросом.

Пример правильного запроса:

GET /api/table-query/get-table-query-by-id/2?resultId=1

Пример правильного ответа:


```
{
  "queryId": 2,
  "tableName": "Customer",
  "query": "select * from Customer"
}
```


Пример неправильного запроса:

GET /api/table-query/get-table-query-by-id/20?resultId1

Пример ответа на неправильный запрос:


```
{
  "timestamp": "2022-08-12T01:12:22.322+00:00",
  "path": "/api/table-query/get-table-query-by-id/20",
  "status": 500,
  "error": "Internal Server Error",
  "requestId": "f240e5ad-53"
}
```


Создание ответа запроса таблицы POST /api/table-query/get-table-query-by-id-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
3	tableQuery	TableQuery	ИСТИНА	Модель запроса
Выходные параметры				

Редакция 7 от 26 августа, 21:30

1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата получения табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат добавления sql-запроса. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 500.
Если ошибок нет, то 200.

Примеры правильного JSON:

```
{
  "resultId": 1,
  "code": 200,
  "tableQuery" : {
     "queryId": 1,
     "tableName": "Artist",
     "query": "select * from Artist"
  }
}
```

Создание ответа выполнения запроса таблицы POST /api/table-query/execute-table-query-by-id-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата

Редакция 7 от 26 августа, 21:30

Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата выполнения табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат выполнения sql-запроса. При невозможности выполнения должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 406.
Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
  "resultId": 1,
  "code": 201
}
```

Создание ответа удаления таблицы POST /api/table-query/delete-table-query-by-id-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Редакция 7 от 26 августа, 21:30

Описание

Метод предназначен для создания результата удаления табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат выполнения sql-запроса. При невозможности выполнения должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 406.
Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
  "resultId": 1,
  "code": 201
}
```

Создание ответа добавления таблицы POST /api/table-query/add-new-query-to-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
3	tableQuery	TableQuery	ИСТИНА	Модель запроса

Редакция 7 от 26 августа, 21:30

Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата добавления табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат добавления sql-запроса. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 406.
Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
  "resultId": 1,
  "code": 200,
  "tableQuery" : {
     "queryId": 1,
     "tableName": "Artist",
     "query": "select * from Artist"
  }
}
```

Создание результата изменения запроса таблицы POST /api/table-query/modify-query-in-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				

Редакция 7 от 26 августа, 21:30

1	resultId	integer	ИСТИНА	id результата
2	code	integer	ИСТИНА	Код результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата выполнения табличного sql-запроса.

Особенности обработки

По результатам должен быть сформирован результат выполнения sql-запроса. При невозможности выполнения должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия.

При возникновении ошибок должен прийти код результата 406.
Если ошибок нет, то 201.

Примеры правильного JSON:

```
{
  "resultId": 1,
  "code": 201
}
```

Создание нового запроса таблицы PUT /api/table-query/modify-query-in-table?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	queryId	integer	ИСТИНА	id запроса
2	tableName	string(50)	ИСТИНА	Имя таблицы
3	query	string(120)	ИСТИНА	Запрос

Редакция 7 от 26 августа, 21:30

Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для изменения sql-запроса в рамках выбранной таблицы.

Особенности обработки

По результатам должен быть изменен sql-запрос по заданной таблице. При невозможности изменения должны быть обработаны следующие ошибки:

- Таблица с таким именем не существует
- Недопустимый формат id

При возникновении ошибок должен прийти код результата 406.

Если ошибок нет, то 200.

Примеры правильного JSON:

```
{
  "queryId": 2,
  "tableName": "Artists",
  "query": "select * from Artists"
}
```

```
{
  "queryId": 1,
  "tableName": "Artists",
  "query": "раз dva"
}
```

Удаление запроса таблицы DELETE /api/table-query/delete-table-query-by-id/{id}?resultId={id}

№	Название	Тип	Обязательность	Описание

Редакция 7 от 26 августа, 21:30

Входные параметры				
1	id	integer	ИСТИНА	id запроса
2	resultId	integer	ИСТИНА	id результата
Выходные параметры				
1	status	string	ИСТИНА	Код результата

Описание

Метод предназначен для изменения sql-запроса в рамках выбранной таблицы.

Особенности обработки

По результатам должен быть изменен sql-запрос по заданной таблице. При невозможности изменения должны быть обработаны следующие ошибки:

- Таблица с таким именем не существует
- Недопустимый формат id

При возникновении ошибок должен прийти код результата 406.

Если ошибок нет, то 200.

Примеры правильного JSON:

```
{
  "queryId": 2,
  "tableName": "Artists",
  "query": "select * from Artists"
}
```

```
{
  "queryId": 1,
  "tableName": "Artists",
  "query": "раз dva"
}
```

Редакция 7 от 26 августа, 21:30

Запрос запуска таблицы по id GET /api/table-query/execute-table-query-by-id/{id}?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	id	integer	ИСТИНА	id таблицы
Выходные параметры				
1	status	integer	ИСТИНА	id запроса

Описание

Метод предназначен для запуска sql-запроса по id.

Особенности обработки

По результатам должен быть получен JSON с запросом по id. При невозможности запуска должны быть обработаны следующие ошибки:

- Запроса с таким id не существует

Если запроса не существует, ответ с кодом 406.
Если ошибок нет, то код 201 с запросом.

Пример правильного запроса: GET /api/table-query/execute-table-query-by-id/2?resultId=1

Запросы на таблицы

Создание таблицы POST /table/create-table?resultId={id}

Редакция 7 от 26 августа, 21:30

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	tableName	string	ИСТИНА	Имя таблицы
2	columnsAmount	integer	ИСТИНА	Количество полей
3	primaryKey	string	ИСТИНА	Ключ
4	columnInfos	list	ИСТИНА	Список столбцов
Выходные параметры				
1	result code	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания новой таблицы в рамках выбранной ранее базы данных. В параметре columnInfos должен быть передан список полей. Как минимум должна быть передана колонка, в которой находится первичный ключ, остальные колонки могут быть заведены уже отдельным запросом.

Особенности обработки

По результатам должна быть добавлена таблица с заданным именем. При невозможности добавления должны быть обработаны следующие ошибки:

- Таблица с таким именем уже существует
- Недопустимое имя таблицы
- Недопустимое название типа данных
- Список полей содержит недопустимые имена полей
- База данных недоступна
- Недостаточно прав на редактирование БД

При возникновении таких ошибок должен вернуться код ошибки 406.
Если ошибок нет, то должен вернуться код 201.

Примеры правильных JSON:

```
{
  "tableName": "Customer",
  "columnsAmount": 12,
  "columnInfos": [
    {
      "title": "CustomerId",
      "type": "int4"
    },
```

Редакция 7 от 26 августа, 21:30


```
    {
      "title": "FirstName",
      "type": "VARCHAR(40)"
    },
    {
      "title": "LastName",
      "type": "VARCHAR(20)"
    },
    {
      "title": "Company",
      "type": "VARCHAR(80)"
    },
    {
      "title": "Address",
      "type": "VARCHAR(70)"
    },
    {
      "title": "City",
      "type": "VARCHAR(40)"
    },
    {
      "title": "Country",
      "type": "VARCHAR(40)"
    },
    {
      "title": "PostalCode",
      "type": "VARCHAR(10)"
    },
    {
      "title": "Phone",
      "type": "VARCHAR(24)"
    },
    {
      "title": "Fax",
      "type": "VARCHAR(24)"
    },
    {
      "title": "Email",
      "type": "VARCHAR(60)"
    },
    {
      "title": "SupportRepId",
      "type": "int4"
    }
  ],
```

Редакция 7 от 26 августа, 21:30

```
  "primaryKey": "CustomerId"
}
```

```
{
  "tableName": "Artists",
  "columnsAmount": 3,
  "columnInfos": [
    {
      "title": "id",
      "type": "int4"
    },
    {
      "title": "name",
      "type": "varchar"
    },
    {
      "title": "age",
      "type": "int4"
    }
  ],
  "primaryKey": "id"
}
```

Создание результата создания таблицы POST /table/add-create-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	code	integer	ИСТИНА	Code
Выходные параметры				

Редакция 7 от 26 августа, 21:30

1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата создания новой таблицы.

Особенности обработки

По результатам должен быть добавлен результат создания таблицы. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия: 406, 201

При возникновении таких ошибок должен вернуться код ошибки 406.
Если все хорошо, то 201

Получение таблицы по имени GET /api/table/get-table-by-name/{name}?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	name	string(50)	ИСТИНА	Имя таблицы
Выходные параметры				

Редакция 7 от 26 августа, 21:30

1	tableName	string	ИСТИНА	Имя таблицы
2	columnsAmount	integer	ИСТИНА	Количество полей
3	primaryKey	string	ИСТИНА	Ключ
4	columnInfos	list	ИСТИНА	Список столбцов

Описание

Метод предназначен для чтения структуры таблицы в рамках выбранной ранее базы данных. В параметре fields должен вернуться список полей.

Особенности обработки

По результатам должна быть добавлена таблица с заданным именем. При невозможности добавления должны быть обработаны следующие ошибки:

- Таблицы с таким именем не существует

Если таблицы с таким именем нет, то должен вернуться код 200, но тело ответа должно быть полностью пустым.
Если таблица с таким именем существует, должен прийти код 200 и выходной JSON,в котором значения полей title и type будут написаны в верхнем регистре, а primaryKey в нижнем. Значение type должно быть приведено к стандартному виду, смотрите в примерах ниже.
При недопустимом resultId вернется код 400.

Пример правильного запроса на Artists:

```
{
  "tableName": "Artists",
  "columnsAmount": 3,
  "primaryKey": "id",
  "columnInfos": [
    {
      "title": "ID",
      "type": "INTEGER"
    },
```

Редакция 7 от 26 августа, 21:30

```
    {
      "title": "NAME",
      "type": "CHARACTER VARYING"
    },
    {
      "title": "AGE",
      "type": "INTEGER"
    }
  ]
}
```

Пример правильного запроса на Customer:

```
{
  "tableName": "Customer",
  "columnsAmount": 12,
  "primaryKey": "customerid",
  "columnInfos": [
    {
      "title": "CUSTOMERID",
      "type": "INTEGER"
    },
    {
      "title": "FIRSTNAME",
      "type": "CHARACTER VARYING"
    },
    {
      "title": "LASTNAME",
      "type": "CHARACTER VARYING"
    },
    {
      "title": "COMPANY",
      "type": "CHARACTER VARYING"
    },
    {
      "title": "ADDRESS",
      "type": "CHARACTER VARYING"
    },
    {
      "title": "CITY",
      "type": "CHARACTER VARYING"
    },
    {
```

Редакция 7 от 26 августа, 21:30


```
        "title": "COUNTRY",
        "type": "CHARACTER VARYING"
      },
      {
        "title": "POSTALCODE",
        "type": "CHARACTER VARYING"
      },
      {
        "title": "PHONE",
        "type": "CHARACTER VARYING"
      },
      {
        "title": "FAX",
        "type": "CHARACTER VARYING"
      },
      {
        "title": "EMAIL",
        "type": "CHARACTER VARYING"
      },
      {
        "title": "SUPPORTREPID",
        "type": "INTEGER"
      }
    ]
}
```


Создание результата создания таблицы POST /table/add-get-table-by-name-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	code	integer	ИСТИНА	Code

Редакция 7 от 26 августа, 21:30

3	table	Table	Истина	таблица
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата получения таблицы по имени.

Особенности обработки

По результатам должен быть добавлен результат создания таблицы. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия: 200

При возникновении таких ошибок должен вернуться код ошибки 400.
Если все хорошо, то 202

Пример правильного Json:

```
{
  "resultId": 1,
  "code": 202,
  "table": {
    "tableName": "Artists",
    "columnsAmount": 3,
    "columnInfos": [
      {
        "title": "id",
        "type": "INTEGER"
      },
      {
        "title": "height",
        "type": "INTEGER"
      },
      {
        "title": "name",
        "type": "VARCHAR"
      }
    ],
    "primaryKey": "id"
```

Редакция 7 от 26 августа, 21:30

```
  }
}
```

Удаление таблицы DELETE /api/table/drop-table/{name}?resultId={id}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	name	string	ИСТИНА	имя таблицы
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для удаления таблицы по имени.

Особенности обработки

По результатам должна быть удалена таблица. При невозможности выполнения должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- resultId не существет
- Таблицы с таким именем не существует

Редакция 7 от 26 августа, 21:30

Если resultId не существует, должен вернуться код 400

Если таблицы не существует, то 406.

Если ошибок нет, то 201

Создание результата удаления таблицы POST /table/add-drop-table-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	code	integer	ИСТИНА	Code
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата удаления таблицы по имени.

Особенности обработки

По результатам должен быть добавлен результат удаления таблицы. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId

Редакция 7 от 26 августа, 21:30

- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия: 201 или 406

При возникновении таких ошибок должен вернуться код ошибки 400.

Если все хорошо, то 202

Создание репортов

Создание результата получения репорта POST /api/report/add-get-report-by-id-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	code	integer	ИСТИНА	Code
3	getReport	getReport	ИСТИНА	getReport
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата получения репорта.

Редакция 7 от 26 августа, 21:30

Особенности обработки

По результатам должен быть добавлен результат получения репорта. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия: 201 или 406

При возникновении таких ошибок должен вернуться код ошибки 400.

Если все хорошо, то 202

Пример правильного JSON:

```
{
  "resultId":1,
  "code":201,
  "getReport":{
    "reportId":1,
    "tableAmount":1,
    "tables":[
      {
        "tableName":"Artists",
        "columns":[
          {
            "title":"id",
            "type":"int4",
             "size": "0"
          },
          {
            "title":"name",
            "type":"varchar",
             "size": "0"
          },
          {
            "title":"age",
            "type":"int4",
             "size": "0"
          }
        ]
      }
    ]
  }
}
```

Редакция 7 от 26 августа, 21:30

Получение репорта GET

/api/report/get-report-by-id/{id}?resultId={resultId}

№	Название	Тип	Обязательность	Описание
Выходные параметры				
1	reportId	integer	ИСТИНА	reportId
2	tableAmount	integer	ИСТИНА	Кол-во таблиц
3	tables	list	ИСТИНА	Список таблиц
Входные параметры				
1	resultId	integer	ИСТИНА	id результата
2	id	integer	ИСТИНА	Код результата

Описание

Метод предназначен для получения репорта.

Редакция 7 от 26 августа, 21:30

Особенности обработки

По результатам должен получен репорт. Должны быть обработаны следующие ошибки:

- resultId не существует
- Запроса с таким id не существует

При возникновении таких ошибок должен вернуться код ошибки 400.

Пример правильного запроса:

/api/report/get-report-by-id/1?resultId=1

Пример правильного JSON:


```
{
  "reportId": 3,
  "tableAmount": 2,
  "tables": [
    {
      "tableName": "Artists",
      "columns": [
        {
          "title": "id",
          "type": "int4",
          "size": "0"
        },
        {
          "title": "name",
          "type": "varchar",
          "size": "0"
        },
        {
          "title": "age",
          "type": "int4",
          "size": "0"
        }
      ]
    },
    {
      "tableName": "Artist",
      "columns": [
        {
          "title": "artistId",
```

Редакция 7 от 26 августа, 21:30


```
            "type": "int4",
            "size": "0"
          },
          {
            "title": "name",
            "type": "varchar",
            "size": "0"
          }
        ]
      }
  ]
}
```


Создание результата создания репорта POST /api/report/add-create-report-result

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	resultId	integer	ИСТИНА	resultId
2	code	integer	ИСТИНА	Code
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания результата создания репорта.

Особенности обработки

По результатам должен быть добавлен результат создания репорта. При невозможности добавления должны быть обработаны следующие ошибки:

Редакция 7 от 26 августа, 21:30

- Недопустимый формат resultId
- Недопустимый формат code. Код должен соответствовать с возможным кодом результата целевого действия: 201 или 406

При возникновении таких ошибок должен вернуться код ошибки 400.

Если все хорошо, то 202

Пример правильного JSON:

```
{
  "resultId": 1,
  "code": 201
}
```

Создание репорта POST /api/report/create-report?resultId={resultId}

№	Название	Тип	Обязательность	Описание
Входные параметры				
1	reportId	integer	ИСТИНА	reportId
2	tableAmount	integer	ИСТИНА	tableAmount
3	tables	list	ИСТИНА	Список таблиц
Выходные параметры				
1	status	integer	ИСТИНА	Код результата

Описание

Метод предназначен для создания репорта.

Редакция 7 от 26 августа, 21:30

Особенности обработки

По результатам должен быть добавлен репорт. При невозможности добавления должны быть обработаны следующие ошибки:

- Недопустимый формат resultId(400)
- resultId не существует.(400)
- Таблица не существует(406)
- Неверный тип данных с колонке(406)
- Неверное имя колонки(406)
- Репорт с таким reportId уже существует(406)

При возникновении таких ошибок должен вернуться код ошибки 406 или 400.
Если все хорошо, то 201

Пример правильного запроса:

/api/report/create-report?resultId=1

Пример правильного JSON:

```
{
  "reportId": 1,
  "tableAmount": 1,
  "tables": [
    {
      "tableName": "Cars",
      "columns": [
        {
          "title": "id",
          "type": "int4"
        },
        {
          "title": "price",
          "type": "int4"
        }
      ]
    }
  ]
```

Редакция 7 от 26 августа, 21:30

```
}
```

Редакция 7 от 26 августа, 21:30

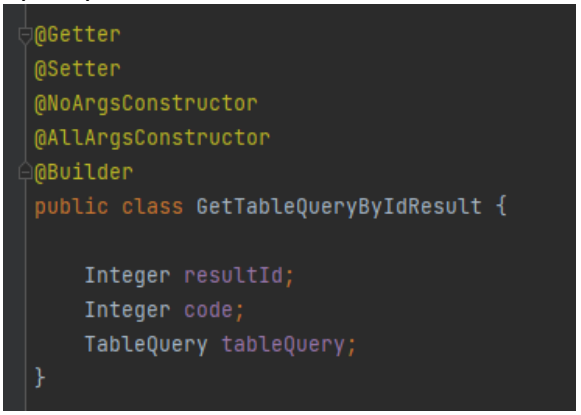
Примечание

- При изменении имени имени таблицы с помощью sql-запросов или их модификации, запросы, привязанные к данной таблице, также должны менять привязку к таблице на новую.
- При удалении таблицы все связанные вопросы должны удаляться
- Репорты не должны изменяться при создании, модификации, удалении таблиц
- В репортах нужно делать проверку, совпадают ли столбцы и типы с реальными
- Две таблицы с одинаковым именем не могут существовать
- Таблицу, добавленную любым способом можно использовать в репортах
- На SQL-запросы ограничений нет, блоки операторов могут быть.
- Customerлала = валидное имя :)
- Если вы получаете код, не описанный в документации, значит ваш тест признан невалидным(например, 500)
- Обработка заголовков с кастомными типами данными не предусмотрена(см. пункт выше).
-

Вопрос - ответ

Вопрос	Ответ
Стр.13 результат getQuery - только код ответа, тела нет, значит и проверить нельзя.	По этому запросу проверяются коды и тело
Стр.16 getAllQueries - вообще ничего не проверяет	Все верно
Стр.19 в результат addQuery для чего-то передаётся тело, хотя оно по апи квалы не возвращается	Тело передается
Стр.19 все-запросы-по-таблице просят ид результата, но апи не описано (и тело в этот результат не принимается)	Для каждого запроса должен быть результат его выполнения согласно доке

Редакция 7 от 26 августа, 21:30

Стр.24 запрос-по-ид опечатка в теле (и судя по тестам, проверки нет)	Проверка есть, опечатки нет
	 
	@Getter
	@Setter
	@NoArgsConstructor
	@AllArgsConstructor
	@Builder
	public class GetTableQueryByIdResult {
	
	Integer resultId;
	Integer code;
	TableQuery tableQuery;
	}

Стр.39 результат getTable без тела	Опечатка, спасибо!