**TelegramBot: Technical**

**Specification**

**1. Introduction**

TelegramBot is looking are there any unreaded messages or posts in users selected telegram channels and sending him/her notifications.

**2.UserController**

**2.1. allows us make basic operations with user profiles**

Base path /users

**2.1.1. Contents**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Resource path** |
| **Users list** | GET | /users |
| **Create user** | POST | /users |
| **Delete user** | DELETE | /users/{id} |
| **Get user** | GET | /users/{id} |
| **Update user** | PUT | /users/{id} |

**2.1.1.1. Users list**

Returns users list

Output List<User>

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | *User inner identifier* |
| name | String | User name |
| status | UserRole | Users role, ADMIN or User |

**2.1.1.2. Create user**

Creates user in database

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| name | String | User name |

Output User

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | *User inner identifier* |
| status | UserRole | User role |
| name | String | User name |

*Output execution result*

{

    "id": 30,

    "name": "qwerty",

    "status": "USER"

}

**2.1.1.3. Delete user**

Delete user from database

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | *User inner identifier* |

OkHttpClient client = new OkHttpClient().newBuilder()

.build();

MediaType mediaType = MediaType.parse("text/plain");

RequestBody body = RequestBody.create(mediaType, "");

Request request = new Request.Builder()

.url("localhost:8080/users/1")

.method("DELETE", body)

.build();

Response response = client.newCall(request).execute();

**2.1.1.4. Get user**

Get user by identifier

*Input parameters*

|  |  |  |
| --- | --- | --- |
| *Name* | *Type* | *Description* |
| *id* | *int* | *User inner identifier* |

*Output User*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | User inner identifier |
| status | UserRole | User role |
| name | String | User name |

**2.1.1.5. Update user**

Method can update users field name

*Input parameters*

|  |  |  |
| --- | --- | --- |
| *Name* | *Type* | *Description* |
| *id* | *int* | *User inner identifier* |

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| name | String | User name |

Output User

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | *User inner identifier* |
| status | UserRole | User role |
| name | String | User name |

*Output execution result*

{

    "id": 30,

    "name": "qwerty",

    "status": "user"

}

{

    "id": 30,

    "name": "qwerty",

    "status":”USER”

}

{

    "id": 30,

    "name": "qwerty",

    "status": "user"

}

**3. ChannelController**

**3.1. allows us make basic operations with channels**

Base path /channels

**3.1.1. Contents**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Resource path** |
| **Channel list** | GET | /channels |
| **Get channel** | GET | /channels/{id} |
| **Create channel** | POST | /channels |
| **Update channel** | PUT | /channels/{id} |
| **Delete channel** | DELETE | /channels/{id} |

**3.1.1.1. Channel list**

Returns all channel listfrom database

Output List<Channel>

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | *Channel inner identifier* |
| name | String | Channel name |
| isBotEnabled | boolean | Shows is channel open for reading message and generating events for it |

**3.1.1.2. Get channel**

Gets channel from database by identifier

*Input parameters*

|  |  |  |
| --- | --- | --- |
| *Name* | *Type* | *Description* |
| *id* | *int* | *Channel inner identifier* |

*Output Channel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | *Channel inner identifier* |
| name | String | Channel name |
| isBotEnabled | boolean | Shows is channel open for reading message and generating events for it |

**3.1.1.3. Create channel**

Add channel to database

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| name | String | Channel name |

*Output Channel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | *Channel inner identifier* |
| name | String | Channel name |
| isBotEnabled | boolean | Shows is channel open for reading message and generating events for it |

*Output execution result*

{

    "id": 40,

    "name": "TestGroupTelranTest",

    "botEnabled": **true**

}

**3.1.1.4. Update channel**

Method can update channelName

*Input parameters*

|  |  |  |
| --- | --- | --- |
| *Name* | *Type* | *Description* |
| *id* | *int* | *Channel inner identifier* |

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| name | String | Channel name |

*Output Channel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | Channel inner identifier |
| name | String | Channel name |
| isBotEnabled | boolean | Shows is channel open for reading message and generating events for it |

**3.1.1.5. Delete channel**

Delete channel from database

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | Channel inner identifier |

**4. EventController**

**4.1. allowing get event list for specific user by id**

Base path /events

**4.1.1. Contents**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Resource path** |
| **Event list** | GET | /events/{userid} |

**4.1.1.1. Event list**

Gets all new events from database for specific by id user

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | User inner identifier |

*Output List<Event>*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | Event inner identifier |
| text | String | Notification text |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |
| isNew | boolean | Shows is event new or were sent before |

**5. StatisticController**

**51. allowing get event list for specific user by id**

Base path /statistic

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Resource path** |
| **Statistic list** | GET | /statistic/{userid} |
| **Create statistic** | POST | /statistic/{userid}/new |

**5.1.1.1. List Statistic**

Gets all statistic list from database, but user must be admin to get it

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | User inner identifier |

*Output List<Statistic>*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | Statistic inner identifier |
| statisticMessage | String | Basic statistic message how much new events were sent from last statistic |
| lastIndexOfLastEvent | int | Event inner identifier, used in creating new statistic |

**5.1.1.2. Create Statistic**

Prepare new Statistic and displays it, but user must be admin to get it

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | User inner identifier |

*Output Statistic*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | Statistic inner identifier |
| statisticMessage | String | Basic statistic message how much new events were sent from last statistic |
| *lastIndexOfLastEvent* | int | Event inner identifier, used in creating new statistic |

*Output execution*

{

"id": 8,

"statisticMessage": "From last time bot sent 0 new events!",

"lastIndexOfLastEvent": 106

}

**5. UserChannelController**

**5.1. Controller allows to make basic operations with subscriptions and 2 more options for activating or deactivating subscription without deleting**

Base path /subscriptions

|  |  |  |
| --- | --- | --- |
| **Operation** | **Method** | **Resource path** |
| **Subscription list** | GET | /subscriptions/{userid} |
| **Create subscription** | POST | /subscriptions/{userid} |
| **Delete subscription** | DELETE | /subscriptions/{userid}/{channelid} |
| **Activate subscription** | PUT | /subscriptions/{userid}/activate |
| **Deactivate subscription** | PUT | /subscriptions/{userid}/deactivate |

**5.1.1.1. Subscription list**

Gets all user subscriptions by user inner identifier

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| id | int | User inner identifier |

*Output List<UserChannel>*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | UserChannel inner identifier |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |
| isSubscribed | boolean | Defines if subscriptions is active or not, by default is active |

**5.1.1.2. Create Subscription**

Adds to database new UserChannel if user and channel exists, and if subscription already doesn’t exist

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |

*Output UserChannel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | UserChannel inner identifier |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |
| isSubscribed | boolean | Defines if subscriptions is active or not, by default is active |

*Output execution*

{

"id": 27,

"userId": 18,

"channelId": 36,

"subscribed": true

}

**5.1.1.3. Delete Subscription**

*Deletes from database UserChannel if subscription exist and user exist*

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userid | int | User inner identifier |
| channelid | int | Channel inner identifier |

**5.1.1.4. Activate Subscription**

Activates existing subscription if user and subscription exist

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |

*Output UserChannel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | UserChannel inner identifier |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |
| isSubscribed | boolean | Defines if subscriptions is active or not, by default is active |

*Output execution*

{

"id": 27,

"userId": 18,

"channelId": 36,

"subscribed": true

}

**5.1.1.5. Activate Subscription**

Activates existing subscription if user and subscription exist

*Input parameters*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |

*Input parameters in the request body*

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |

*Output UserChannel*

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | int | UserChannel inner identifier |
| userId | int | User inner identifier |
| channelId | int | Channel inner identifier |
| isSubscribed | boolean | Defines if subscriptions is active or not, by default is active |

*Output execution*

{

"id": 27,

"userId": 18,

"channelId": 36,

"subscribed": false

}