

StreaMoor API documentation

Getting started

This API is designed to be implemented with the StreaMoor streaming service for any frontend basically. All operations are done through routes, every route has their specific method that they can reply. Each basic operation has their own route that are described below.

Routes

login

Method = 'post'

Gets the data from the login form and authorizes the input

Returns: JSON with auth result info that can be one of two:

- result='fail' if the authorization was unsuccessful due to incorrect email or password
- result = user_nick returns the nickname to display if the auth was successful

Register

Method = 'post'

Inserts the information about new user unto the database

Returns JSON with the result that can be one of two:

- result = 'bad' if the user already exists
- result = 'good' if the user gets successfully registered and he's been put in the database

subscribe

Subscribes the user to the channel

unsubscribe

Unsubscribes the user from the channel

channels

method = 'get'

Gets the channels that are in the hub selected

Returns JSON with the channels

channel

method = 'get'

Gets the channel info from the database and returns info about it

Returns JSON with the channel info: nickname, subs, description, curr_hub

video_feed

Generates the video image from the webcam and sends it to the channel

Returns the next frame to be put in the 'img' tag

search

method = 'get'

Searches the database for anything that is similar to the search input

Returns: channels that might fit the search input

Database methods

Class SQLModel contains methods to work with the Database, in this case it's POSTGRES

connect

Private class method

Connects to the database

Returns connection to the database

query

Performs a query than us not specified in the other methods

Arguments:

- query: the SQL query to be executed
- attrs: any attributes to be added to the query during the execute() method Returns: the result of the query

normalize_cols

Class method Changes columns presentation to be correct for the query

Arguments:

- cols: columns to be normalized

insert

Class method

Insert the values in the database

Arguments:

- values: values to be inserted to the database

get_by_attrs

Class method

Gets the values from the database that correspond to the values given

Arguments:

- cols: columns to be selected from the database
- attr_cols: columns that are used in the WHERE statement
- attr_values: values that are used in the WHERE statement in corresponding columns
- group_by: None by default, GROUP BY statement
- order_by: None by default, ORDER BY statement

Has 4 overloads depending on the group_by and order_by, these overloads contain defferent queries

Returns: tuple with the query result

update_by_attrs

Class method

Updates the values in the database that correspond to the values given

Arguments:

- columns: columns to be updated
- values: values to be put in the database
- attr_cols: columns that are used in the WHERE statement
- attr_values: values that are used in the WHERE statement

delete_by_attrs

Class method

Deletes the rows from the database that correspond to the values given

Arguments:

- attr_cols: columns that are used in the where statement
- attr_values: columns that are used in the statement