POST - запросы

Регистрация пользователя

Method : registration

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obligatory): login, password, nickname; ( 3 < lenLogin < 30; 5 < lenPassword; 3 < nickname) (only English language);

Return: true

Авторизация пользователя

Method : login

API: api/index.php

Type: get

Params(obligatory): login, password;

Return: token (hash);

Выход пользователя с учетной записи

Method : logout

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obiligatory): token;

Return: true

Создание стола

Method : createtable

API: api/index.php

Type: post

Params(obiligatory): token, name;

Params(optional): quantplayers, rates, password;( По умолчанию кол-во игроков: 4; Min:4 ; Max:7)

(Ставки по умолчанию : 20; Min : 20; Max : 10000)

(Password по умолчанию: \*без пароля\*)

Return: true

Удаление стола по ID

Method: deletetablebyid

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obiligatory): id

Return: true

Присоединение к столу

Method : connecttotable

API: api/index.php

Type: post

Params(obilitory): token, id

Return: true

Выход игрока со стола

Method : disconnectfromtable

API : [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): token, id

Return: true

Перевод денег из «активных» в банк

Method: transfertobank

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): token, (int)money;

Return: true;

Перевод денег из банка в «активные»

Method: transfertomoney

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): token, (int)money;

Return: true;

Запуск игры(со стола)

Method: startgame

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): (int)idtable

Return: id игры, которую мы создали;

Повышение Ставки

Method: raise

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): (int)id(айди игры, на которой все происходит), (int)sum(Сумма, на которую повышаем ставку)

Return: true;

Уравнивание ставки(call)

Method: call

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): (int)id(айди игры, на которой все происходит)

Return: true;

Пропуск хода(Check)

Method: check

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): (int)id(айди игры, на которой все происходит)

Return: true;

Пас(fold)

Method: fold

API: [api/index.php](http://localhost/api/index.php)

Type: post

Params(obilitory): (int)id(айди игры, на которой все происходит)

Return: true;

GET- запросы

Получение пользователя по токену

Method : getuserbytoken

API: http://localhost/api/index.php?method=getuserbytoken

Type: get

Params(obilitory): token

Return: Ассоциотивный массив:

{id: int, login: string, password: string(hash), token(Если онлайн, то прилетает hash, если оффлайн, то поле пустое), nickname: string, money: int}

Получение пользователя по ID

Method : getuserbyid

API: http://localhost/api/index.php?method=getuserbyid

Type: get

Params(obilitory): token

Return: Ассоциотивный массив:

{id: int, login: string, password: string(hash), token(Если онлайн, то прилетает hash, если оффлайн, то поле пустое), nickname: string, money: int}

Получение списка всех столов

Method: getalltables

API: http://localhost/api/index.php?method=getalltables

Type: get

Params: ---;

Return:

Массив ассоциотивных массивов;

Array[0]: { id: int, name: string, quantity\_players: int, active\_players\_id: string(Ай ди игроков через пробел), rates: string, password: string };

Получение стола по ID

Method : gettablebyid

API: http://localhost/api/index.php?method=gettablebyid

Type: get

Params(obilitory): id

Return: Ассоциотивный массив: { id: int, name: string, quantity\_players: int, active\_players\_id: string(Ай ди игроков через пробел), rates: string, password: string };

Получение количества активных игроков за столом

Method: getquantplayersontable

API: API: http://localhost/api/index.php?method= getquantplayersontable

Type: get

Params(obilitory): id стола

Return: int(Количество активных игроков за столом)

Получение статистики по ID игрока

Method: getstatsbyid

API: <http://localhost/api/index.php?method=getstatsbyid>

Type: get

Params(obilitory): id игрока

Return:

{ “user\_id”: int, "win":int,"loss": int, "biggest\_win":int, "biggest\_loss":int, "date\_registration": YYYY-MM-DD }

Получение игры по id игры(Возможно, мы уберем некоторые вещи, которые будут прилетать на фронт. Например, закрытые карты)

Method: getgame

API: <http://localhost/api/index.php?method=getgame>

Type: get

Params(obilitory): id игры;

Return:

{active: (int). Айди игрока, чей сейчас ход

all\_rates: (int). Все ставки за столом

board\_cards: [""]. Карты на столе

circle: (int). Номер круга

close\_cards: (5) [{…}, {…}, {…}, {…}, {…}] Массив закрытых карт.

id: (int). id игры

player1: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player2: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player3: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player4: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player5: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player6: {id: (int), card1: {…}, card2: {…}, comb: (int), rates: (int)}

player7: false(если место пустое);

start\_circle: (int). Айди игрока, с которого начался круг(повышение ставок)

table\_id: (int)

}

КОГДА ИГРА ЗАВЕРШАЕТСЯ, нам прилетает массив, где на 0 месте победитель игры, а на 1ом айди новой игры