**Серверный API игровой системы «Колесо инвестиций»**

**Описание**

Система состоит из двух основных частей: серверного приложения и клиентского приложения. Серверное приложение реализует в себе логику самой игры, веб интерфейс для настройки серверного приложения (он доступен администратору системы) и API для работы с ним клиентских приложений.

В качестве базового прокола был выбран JSON over HTTP, являющийся стандартом коммуникации с веб-приложениями и широко используемый в мобильных устройствах.

**Взаимодействие**

Работа с API происходит следующим образом. Клиент API делает http запрос к серверу, при этом, передавая данные в качестве параметров GET или POST запроса. В ответ серверный API возвращает JSON документ, со всеми необходимыми данными. Все методы API могут быть опробованы при помощи браузера, так как статическая информация (такая как id сессии), может быть передана как в cookie так и простым параметром HTTP запроса.

Запросы к API могу быть сделаны по адресу <domain>/api/<method\_name>. Ко всем запросам необходимо добавлять параметр api\_token (являющийся идентификатором пользователя API). *В тестовой версии доступ происходит без api\_token*.

**Методы**

* Registration – регистрация пользователя
* Login – вход пользователя
* Games – получение списка созданных игр
* Add\_game – создать новую игру
* Join\_game – присоединиться к игре
* Setup\_game – задать настройки для игры
* Start\_game – начать игру
* Turn\_game – завершить ход

**Registration**

Запрос:

[http://<domain>/registration?login=user\_vova&password=123456&role=PLAYER](http://localhost:8080/registration?login=user_vova&password=123456&role=PLAYER)

Ответ:

{  
   **"result"**:"REGISTRATION\_SUCCESSFUL",  
   **"success"**:true  
}

или

{  
   **"result"**:"REGISTRATION\_FAILED\_ALREADY\_EXISTS",  
   **"success"**:false  
}

Пояснение

Параметр role может быть PLAYER или ADMIN

**Login**

Запрос:

[http://<domain>/login?login= user\_vova &password=123456](http://localhost:8080/login?login=dima&password=1111)

либо

[http://<domain>/login?session\_id=eeafee721b28089bcdb0119c26668816f0705a94aa74a71d0807b998dc108d3a](http://localhost:8080/login?session_id=eeafee721b28089bcdb0119c26668816f0705a94aa74a71d0807b998dc108d3a)

Ответ:

{  
   **"result"**:"AUTHORIZATION\_SUCCESSFUL",  
   **"session\_id"**:"13023e0472470fb87099bd177579fd19dd5cac650b97bace5640d61ae9173396",  
   **"success"**:true  
}

***(Ответ на запрос авторизации устанавливает cookie, в которой записан session\_id. Если используется механизм, который не устанавливает cookie автоматически, session\_id можно передавать параметром http запроса (session\_id). Наличие session\_id в cookie или в качестве параметра запроса обязательно для вызова большинства методов API.)***

**Add\_game**

Запрос:

[http://<domain>/api/add\_game?name=my+name](http://localhost:8080/api/add_game?name=my+name)

Ответ:

{

"status\_code":"SUCCESS",

"games":{

"3":{

"id":3,

"turn":0,

"status":"NEW",

"players":{

},

"name":"my name",

"mode":"DEFAULT"

}

},

"game\_id":3

}

**Games**

Запрос:

[http://<domain>/api/games](http://localhost:8080/api/games)

Ответ:

{

"games":{

"3":{

"id":3,

"turn":0,

"status":"NEW",

"players":{

},

"name":"my name",

"mode":"DEFAULT"

},

"4":{

"id":4,

"turn":0,

"status":"NEW",

"players":{

},

"name":"mega game",

"mode":"DEFAULT"

}

}

}

**Setup\_game**

Запрос:

[http://<domain>/api/setup\_game?id=4](http://localhost:8080/api/setup_game?id=4)

Ответ:

{

"status\_code":"SUCCESS",

"game":{

"id":"4",

"turn":0,

"status":"READY",

"players":{

},

"name":"mega game",

"mode":"DEFAULT"

}

}

**Join\_game**

Запрос:

[http://<domain>/api/join\_game?id=4](http://localhost:8080/api/join_game?id=4)

Ответ:

{

"status\_code":"SUCCESS",

"game":{

"id":"6",

"turn":0,

"status":"READY",

"players":{

"1":{

"id":1,

"login":"AlThar"

}

},

"name":"mega game",

"mode":"DEFAULT"

}

}

**Start\_game**

Запрос:

[http://<domain>/api/start\_game?id=4](http://localhost:8080/api/start_game?id=4)

Ответ:

{

"status\_code":"SUCCESS",

"game":{

"id":"4",

"turn":0,

"status":"IN\_PROGRESS",

"players":{

"1":{

"id":1,

"login":"AlThar"

}

},

"name":"mega game",

"mode":"DEFAULT"

}

}

**Turn\_game**

Запрос:

[http://<domain>/api/turn\_game?id=4](http://localhost:8080/api/turn_game?id=4)

Ответ:

{

"status\_code":"SUCCESS",

"game":{

"id":"4",

"turn":1,

"status":"IN\_PROGRESS",

"players":{

"1":{

"id":1,

"login":"AlThar"

}

},

"name":"mega game",

"mode":"DEFAULT"

}

}

В ответе на запрос метода API, возвращается JSON документ, в котором есть поле status\_code, показывающее результат запроса. Возможные варианты этого поля:

* SUCCESS("Успех")
* GAME\_NOT\_READY("Игра не готова")
* GAME\_NOT\_IN\_PROGRESS("Игра не в процессе")
* GAME\_ALREADY\_STARTED("Игра уже в процессе")
* NO\_SUCH\_USER("Нет такого игрока")
* NO\_SUCH\_GAME("Нет такой игры")
* NO\_PERMISSIONS("Нет доступа")
* FAILURE\_PARAMETERS\_WRONG("Провал. Не верные данные")