**Раздел комнат**

**Метод + наименование endpoint-а:** POST /rooms/create

**Описание:** Создает комнату, добавляет её в переменную

**Что принимает**:

name – string

password – int

Пример: {name: “Комната”, password: “123”}

**Ответ** code: 201

При уже существовании комнаты с таким именем, нужно ответить status code: 403 – Bad request и json что-то типо {error: “Комната с таким именем уже существует!”}.

**Метод + наименование endpoint-а:** GET /rooms

**Описание:** Возвращает список всех активных комнат с количеством игроков.

**Ответ:** [ { “name”: “Комната 1”, “players”: 1 }, { “name”: “Комната 2”, “players”: 2 } ]

**WebSocket – Подключение к комнате**

**Описание:** При подключении к WebSocket-клиент отправляет имя комнаты и имя игрока. Если имя игрока уникально в рамках комнаты — подключение успешно. При выходе игрока (закрытие вкладки, отключение) – то есть разрыв соединения — он удаляется из комнаты.

**Событие подключения**

{  
 "event": "join",  
 "data": {  
 "room": "Комната 1",  
 "username": "Игрок123"  
 }  
}

**Ответ при успешном подключении**

{  
 "event": "joined",  
 "data": {  
 "room": "Комната 1",  
 "username": "Игрок123",

"objects": [object, object, …],  
 }  
}

**WebSocket – События игры**

**1. Движение мыши**

{  
 "event": "mousemove",  
 "data": {  
 "x": 150,  
 "y": 300  
 }  
}

**2. Изменение выбранного предмета**

{  
 "event": "changeItem",  
 "data": {  
 "item": object  
 }  
}

**3. Установка блока**

{  
 "event": "placeBlock",  
 "data": {  
 "item": object  
 }  
}

**4. Удаление блока**

{  
 "event": "removeBlock",  
 "data": {  
 "x": 10,  
 "y": 15  
 }  
}