

Snake API ^{v1}

Player

<div>GET</div> <div>/api/Player/gameboard</div> <div>Возвращает состояние игрового поля.</div>	
<div>Parameters</div> <div>Try it out</div>	
<div>Name</div>	<div>Description</div>
<div>token</div> <div>string</div> <div>(query)</div>	<div>Токен игрока.</div>
<div>Responses</div> <div>Response content type</div> <div>text/plain</div>	
<div>Code</div>	<div>Description</div>
<div>200</div>	<div>Success</div>
<div>Example Value Model</div>	
<pre>{ "isStarted": true, "isPaused": true, "roundNumber": 0, "turnNumber": 0, "turnTimeMilliseconds": 0, "timeUntilNextTurnMilliseconds": 0, "gameBoardSize": { "width": 0, "height": 0 }, "maxFood": 0, "players": [{ "name": "string", "isSpawnProtected": true, "snake": [{ "x": 0, "y": 0 }] }], "snake": [{ "x": 0, "y": 0 }] }</pre>	

GET /api/Player/name Возвращает имя игрока.	
Parameters Try it out	
Name	Description
token string (query)	Токен игрока.
Responses Response content type text/plain ▾	
Code	Description
200	<div>Success</div> <div>Example Value Model</div> <div><pre>{ "name": "string"}</pre></div>
401	<div>Некорректный токен.</div>

POST /api/Player/direction		Задаёт поворот змейки, который будет использован во время следующего хода.
Parameters		Try it out
Name	Description	
request (body)	Данные авторизации и направление змейки.	
	Example Value Model	
	<pre>{ "direction": "Left", "token": "string" }</pre>	
	Parameter content type	
	application/json-patch+json	
Responses		Response content type application/json
Code	Description	
200	Success	
401	Некорректный токен.	

Models

GameStateResponse ▾ {	
description:	Содержит состояние игрового поля.
isStarted	boolean Запущена ли игра администратором.
isPaused	boolean Поставлена ли игра на паузу администратором.
roundNumber	integer(\$int32) Текущий номер раунда.
turnNumber	integer(\$int32) Текущий номер хода.
turnTimeMilliseconds	integer(\$int32) Время хода в миллисекундах.
timeUntilNextTurnMilliseconds	integer(\$int32) Время до следующего хода в миллисекундах.
gameBoardSize	Size2D > {...}
maxFood	integer(\$int32) Максимальное количество еды генерируемое на поле.
players	> [...]
snake	> [...]
food	> [...]
walls	> [...]

```
Size2D {
    description: Описывает размер двумерной области.
    width: integer($int32)
        Ширина.
    height: integer($int32)
        Высота.
}
```

```
PlayerState ▾ {
  description:
    Содержит состояния игрока и его змейки.

  name
    string
    Имя игрока.

  isSpawnProtected
    boolean
    Является ли игрок защищенным после появления на игровой доске.
    В это время все кто в него врежется - умрут, а сам игрок не может двигаться какое-то время.

  snake
    > [...]
}
```

```
Point2D ~ {
    description:
        Описывает точку на двумерной области.

    x
        integer($int32)
        Координата X.

    y
        integer($int32)
        Координата Y.
}
```

Rectangle2D ▾ {	
description:	Описывает прямоугольную область.
x	integer(\$int32) Координата X верхнего левого угла области.
y	integer(\$int32) Координата Y верхнего левого угла области.
width	integer(\$int32) Ширина области.
height	integer(\$int32) Высота области.

```
NameResponse {
  description: Содержит имя игрока.
  name: string
  Имя игрока.
}
```

```
DirectionRequest {
  description: Содержит данные авторизации и направление змейки.
  direction: string
               Направление змейки.
  Enum:
    > Array [ 4 ]
    string
    Уникальный секретный токен для авторизации игрока
}
}
```