**Roulette API Guide**

The Roulette API is a REST based API which accepts and returns JSON responses. There endpoints which can be called are described below.

**Authentication**

There is currently no authentication for this API

**Endpoints**

**New Game**

***Method:*** *POST*

**URL**: *api/roulette/new*

**Returns**:

*200 OK*: Successful response containing the *gameId* of the new game created

*500 Internal Server Error*: Error response

**Add Bet**

***Method:*** *POST*

**URL**: *api/roulette/{gameId}/bet*

**Headers:** *Content-Type: application/json*

**Request Body Fields:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Values** | **Description** |
| CustomerId | GUID | Any GUID | Id for the customer placing the bet |
| BetType | BetType Enum | Single, Black, Red, SplitHorizontal, Split Vertical, Corner, Column | The type of bet being placed |
| Position | Int | Any position between 0 and 36, however see *BetType Position Rules* section below | The position for the bet |
| Amount | Double | Any value between 1 and 10,000 | The amount being placed |

**Request Body Example:**

{

"CustomerId": "6fb77cfc-e8d5-48f8-b9bd-4d7ddac292d4",

"BetType": "Single", -- this can also be an integer

"Position": 1,

"Amount": 50.0

}

**Returns**:

*200 OK*: Successful response containing the *bet* that has been placed, which contains the *betId* to use for other bet requests

*400 Bad Request*: Bad request response indicating what is wrong with the request e.g. Invalid BetType

*401 Not Found*: Game with *gameId* not found

*500 Internal Server Error*: Error response

**Successful Response Fields:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Id | GUID | The id of the bet |
| GameId | GUID | The game id that the bet has been placed in |
| CustomerId | GUID | The customer id |
| BetType | BetType Enum | The type of bet been placed |
| Position | Int | The position of the bet been placed |
| Amount | Double | The amount bet |

**Successful Response Example:**

{

"bet": {

"gameId": "93b02763-1449-4a6d-a318-9c40608a4769",

"customerId": "6fb77cfc-e8d5-48f8-b9bd-4d7ddac292d4",

"amount": 50,

"id": "57302080-84e1-4c24-947b-5a0c301f7561",

"betType": "Single", -- I have set this to string for clarity but it can easily be changed to an integer

"position": 1

}

}

**Update Bet**

***Method:*** *PUT*

**URL**: *api/roulette/{gameId}/bet/{betId}*

**Headers:** *Content-Type: application/json*

**Request Body Fields:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Values** | **Amount** |
| Amount | Double | Any value between 1 and 10,000 | The amount to be updated to |

**Request Body Example:**

{

"Amount": 50.0

}

**Returns**:

*200 OK*: Successful response containing the *bet* that has been updated

*400 Bad Request*: Bad request response indicating what is wrong with the request e.g. Invalid amount

*401 Not Found*: Game with *gameId* not found or Bet with *betId* not found

*500 Internal Server Error*: Error response

**Successful Response Fields:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| Id | GUID | The id of the bet |
| GameId | GUID | The game id that the bet has been placed in |
| CustomerId | GUID | The customer id |
| BetType | BetType Enum | The type of bet been placed |
| Position | Int | The position of the bet been placed |
| Amount | Double | The amount bet |

**Successful Response Example:**

{

"bet": {

"gameId": "93b02763-1449-4a6d-a318-9c40608a4769",

"customerId": "6fb77cfc-e8d5-48f8-b9bd-4d7ddac292d4",

"amount": 50,

"id": "57302080-84e1-4c24-947b-5a0c301f7561",

"betType": "Single", -- I have set this to string for clarity but it can easily be changed to an integer

"position": 1

}

}

**Delete Bet**

***Method:*** *DELETE*

**URL**: *api/roulette/{gameId}/bet/{betId}*

**Returns**:

*204 No Content*: Successful response with no content

*400 Bad Request*: Bad request response indicating what is wrong with the request e.g. betting closed

*401 Not Found*: Game with *gameId* not found or Bet with *betId* not found

*500 Internal Server Error*: Error response

**Close Betting**

***Method:*** *PUT*

**URL**: *api/roulette/{gameId}/close-betting*

**Returns**:

*204 No Content*: Successful response with no content

*400 Bad Request*: Bad request response indicating what is wrong with the request e.g. game is closed

*401 Not Found*: Game with *gameId* not found

*500 Internal Server Error*: Error response

**Play Game**

***Method:*** *POST*

**URL**: *api/roulette/{gameId}/play*

**Returns**:

*200 OK*: Successful response containing the results

*400 Bad Request*: Bad request response indicating what is wrong with the request e.g. game is closed

*401 Not Found*: Game with *gameId* not found

*500 Internal Server Error*: Error response

**Successful Response Fields:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Field Type** | **Description** |
| GameId | GUID | The game id |
| Winning number | Int | The winning number |
| WinningBets | Dictionary<string, ICollection<WinningBet>> | A dictionary containing each customers winning bets |
| LosingBets | Dictionary<string, ICollection<LosingBet>> | A dictionary containing each customers losing bets |
| CustomerTotalWinnings | Dictionary<string, double> | A dictionary containing each customers total winnings |

**Successful Response Example:**

{

"gameId": "543a31ec-1a37-4b67-91af-9579d846f896",

"winningNumber": 1,

"winningBets": {

"6fb77cfc-e8d5-48f8-b9bd-4d7ddac292d4": [

{

"amountBet": 50,

"amountWon": 1800,

"id": "bbb74306-4f5b-40fa-bc19-a9c21a0ee951",

"betType": "Single", -- I have set this to string for clarity but it can easily be changed to an integer

"position": 1

}

],

"4cfba7e1-1112-4166-88ff-1f9aadb044e3": [

{

"amountBet": 50,

"amountWon": 1800,

"id": "bfcf99ff-4c11-4ebb-997f-2c5f456593eb",

"betType": "Single", -- I have set this to string for clarity but it can easily be changed to an integer

"position": 1

}

]

},

"losingBets": {

"e8b2280b-a4a7-4664-913a-1326551469d7": [

{

"amountBet": 25,

"amountWon": 900,

"id": "c0dd40c6-6440-42de-8fc5-e46aafa48e85",

"betType": "Single", -- I have set this to string for clarity but it can easily be changed to an integer

"position": 2

}

],

},

"customerTotalWinnings": {

"6fb77cfc-e8d5-48f8-b9bd-4d7ddac292d4": 1800,

"4cfba7e1-1112-4166-88ff-1f9aadb044e3": 1800

}

}

**API Rules**

There are certain rules to be aware of:

* A bet can only be placed when both the game and game betting is open.
* A bet can only be updated when both the game and game betting is open.
* A bet can only be deleted when both the game and game betting is open.
* Betting can only be closed when both the game is open and betting is open.
* A game can only be played when both the game is open and the betting is closed.
* A bet with the same gameId, customerId, betType and position cannot be created more than once. The bet must be updated instead via the Update endpoint.

**Bet Type Position Rules**

Certain bet types only allow certain positions. These are explained below:

**Single Bet**

Any position is allowed from 0 –36.

**Black Bet**

Position is irrelevant and is ignored if one is passed through on the request.

**Red Bet**

Position is irrelevant and is ignored if one is passed through on the request.

**Split Horizontal Bet**

Position must be set to the highest number of the Split Vertical bet e.g. for 25 and 26, the position must be set to 26 to indicate these two numbers have been chosen.

Position can therefore be any of 2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33, 35 and 36.

**Split Vertical Bet**

Position must be set to the highest number of the Split Vertical bet e.g. for 15 and 18, the position must be set to 18 to indicate these two numbers have been chosen.

Position can therefore be any number except 0.

**Corner Bet**

Position must be set to the highest number of the Corner bet e.g. for 8, 9, 11 and 12, the position must be set to 12 to indicate these four numbers have been chosen.

Position can therefore be any of 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33, 35, and 36.

Any other position will return bad request.

**Column Bet**

Position can be 1, 2 or 3 to indicate which column the bet has been placed on.

Any other position will return bad request.