PHP task - make bet

Notes:

- We follow PSR-2 style guide.
- You may use any technology/software to achieve result: PHP + anything (Laravel preferable).
- Simple and lightweight solutions are always an advantage.

Goal

Create API which follows provided documentation.

Part 1

Validate input bet for provided rules and add error codes and messages to global or selection levels.

- (global) input data structure should be valid
- (global) `stake_amount` should be in interval between [0.3, 10000]
- (global) min number of selections is 1
- (global) max number of selections is 20
- (global) maximum `max win amount` is 20000
- (selection) odds should be in interval between [1, 10000]
- (selection) player can not bet on selections with same ID

```
max win amount = stake_amount * (multiply of all selection odds)
```

Avaliable errors:

Code	Type	Message
0	global	Unknown error
1	global	Betslip structure mismatch
2	global	Minimum stake amount is :min_amount
3	global	Maximum stake amount is :max_amount
4	global	Minimum number of selections is :min_selections
5	global	Maximum number of selections is :max_selections
6	selection	Minimum odds are :min_odds
7	selection	Maximum odds are :max_odds
8	selection	Duplicate selection found
9	global	Maximum win amount is :max_win_amount
10	global	Your previous action is not finished yet
11	global	Insufficient balance

Request:

- **URL**

  /api/bet

- **Method:**

  ```
  POST
  ```

- **Data:**

  ```
  {
      // type: int
      // (mandatory) unique player id in the system
      "player_id": 1,

      // type: string
      // (mandatory) amount of money player wants to bet, max number of numbers after dot is 2
      "stake_amount": "99.99",

      // type: array
      // (mandatory) selection (events) on which player wants to bet
      "selections": [
          {
              // type: int
              // (mandatory) selection (event) ID on which player want to bet
              "id": 1,

              // type: string, max number of numbers after dot is 3
              // (mandatory) odds (coefficient) of our selection,
              "odds": "1.601",
          },
      ]
  }
  ```

Response:

- **Successful response:**
  - **HTTP Code:** 201
    Content: `{}`

- **Failed response:**
  - **HTTP Code:** 400
    Content:

    ```
    {
        // GLOBAL errors
        // type: array
        // (on error) will be added if global level errors will occur
        "errors": [],

        // type: array
        // (on error) selection (events) on which player wants to bet
        "selections": [
            {
                // type: int
                // (on error) selection (event) ID on which player want to bet
                "id": 1,

                // SELECTION errors
                // type: array
                // (on error) will be added if selection level errors will occur
                "errors": [],
            },
        ]
    }
    ```

Samples:

Successful

Request:

```
{
    "player_id": 1,
    "stake_amount": "5",
    "selections": [
        {
            "id": 1,
            "odds": "1.601"
        }
    ]
}
```

Response: `{}`

Failed

Request:

```
{
    "player_id": 1,
    "stake_amount": "10000.01",

    "selections": [
        {
            "id": 1,
            "odds": "2.001"
        },

        {
            "id": 1,
            "odds": "2.001"
        }
    ]
}
```

Max win amount: `10000.01 * 2.001 * 2.001 = 40040,05`

Response:

```
{
    "errors": [
        {
            "code": 3,
            "message": "Maximum stake amount is 10000"
        },
        {
            "code": 9,
            "message": "Maximum win amount is 20000"
        }
    ],
    "selections": [
        {
            "id": 1,
            "errors": [
                {
                    "code": 8,
                    "message": "Duplicate selection found"
                }
            ]
        },
        {
            "id": 1,
            "errors": [
                {
                    "code": 8,
                    "message": "Duplicate selection found"
                }
            ]
        }
    ]
}
```

Part 2

Create at least this tables: `player` , `balance_transaction` , `bet` , `bet_selections` . You can create new tables or add new columns if needed. Column types are up to you.

player:

- id - player_id
- balance - current player balance, default 1000

balance_transaction:

- id
- player_id
- amount
- amount_before

bet:

- id
- stake_amount
- created_at

bet_selections:

- id
- bet_id
- selection_id
- odds

If validation did not return any errors and player does not exist in database, create player with provided player_id and default balance amount (1000).

Fill tables `bet` , `bet_selections` , `balance_transaction` and `player` according to input data. If balance is not enough return appropriate `error_code` .

Do not forget to update `player.balance` after operation.