# Тестовое задание

*Задача*: реализовать класс обработки данных (receive/response) между сервером и устройством.

---

## Имеются таблицы и данные

```
CREATE TABLE machines
(
  id SERIAL PRIMARY KEY NOT NULL,
  serial bigint NOT NULL
);
CREATE UNIQUE INDEX machines_serial_index ON machines (serial);

CREATE TABLE machines_options
(
  machine_id int NOT NULL,
  firmware VARCHAR(32),
  connect_freq int NOT NULL
);
CREATE INDEX machines_options_machine_id_index ON machines_options (machine_id);
ALTER TABLE machines_options
ADD FOREIGN KEY (machine_id) REFERENCES machines (id);

CREATE TABLE machines_options_set
(
  machine_id int NOT NULL,
  connect_freq int
);
CREATE INDEX machines_options_set_mahcine_id_index ON machines_options_set
(machine_id);
ALTER TABLE machines_options_set
ADD FOREIGN KEY (machine_id) REFERENCES machines (id);

INSERT INTO machines (serial) VALUES (123456789012345);
INSERT INTO machines_options (machine_id, firmware, connect_freq) VALUES (1, '1.01a',
5);
INSERT INTO machines_options_set (machine_id, connect_freq) VALUES (1, 10);
```

## Описание таблиц

- machines - справочник устройств
- machines_options - параметры устройств
- machines_options_set - задания для устройств по параметрам

## Описание параметров

- serial - идентификатор устройства
- firmware - версия прошивки
- connect_freq - частота подключения к серверу (в минутах)

## Логика

Устройство подключаются к серверу с интервалом 1 раз в connect_freq минут, POST-запросом и отправляют идентификационные данные. Данные могут содержать параметры устройства. Сервер получает данные, валидирует required атрибуты, проверяет наличие устройства в справочнике (serial), обновляет параметры, удаляет задания (machines_set)

## Формат данных

Прием и отправка данных происходит в формате HTTP POST data type:

```
parameter=value&also=another
```

Все задания из machines_options_set для устройств должны иметь префикс у каждого параметра [set_]

```
set_parameter=value&set_also=value_another
```

## POST-данные от устройства

- serial [int] (length: 15) REQUIRED
- time, [DateTime] () REQUIRED
- connect_freq [int] (min-length: 1, max-length: 2)
- firmware [string] (max-length: 32)

## Sample POST/receive

### Соединение с сервером по интервалу

```
[serial] => 123456789012345
[time] => 2014-08-13 11:43:10
```

### Все параметры

```
[serial] => 123456789012345
[time] => 2014-08-13 11:43:10
[connect_freq] => 5
[firmware] => 1.00c
```

### Ответ на задание параметра

```
[serial] => 123456789012345
[time] => 2014-08-13 11:43:10
[sets] => array(
    connect_freq,
    ...,
    <param_name>
)
```

## Sample POST/response

### Стандартный ответ от сервера

```
[time] => 2014-08-13 11:43:10
[status] => ok
```

### Ответ с заданием параметров

```
[time] => 2014-08-13 11:43:10
[set_connect_freq] => 5
[status] => ok
```

## Требования

- Для прерывания обработки данных или вывода ошибок использовать прерывания throw exceptions
- POST-данные, полученные от устройства будут передавать в доступный для вызова метод класса:

  ```
  $response = (new NameClass())
          ->receive($_POST)
          ->response();
  ```

- Для работы с базой использовать PDO драйвер

## Дополнительным преимуществом при оценке будет, если Вы сможете:

1. Реализовать GUI - front-end решения (верстка и дизайн на Ваше усмотрение), наглядно демонстрирующее решение данной задачи на Вашем Shared hosting / VDS / VPS.