

# Тестовое задание для веб-разработчика

## Обзор

Даны 2 кадастровых номера:

- 69:27:0000022:1306
- 69:27:0000022:1307

Нужно создать приложение, в котором будет присутствовать:

1. Миграция и модель для таблицы с данными по участкам с РосРеестра
2. Компонент для парсинга РосРеестра
3. Бизнес логика
4. Unit-тесты
5. Web-контроллер
6. Rest-контроллер
7. Console-контроллер
8. Документация (readme.md)

Оцениваться будет общий подход к разработке - понимание паттернов, понятность кода и его композиция, соблюдение стандартов, работа с использованием DI.

## Компонент

Создать компонент, который на вход будет получать один или несколько кадастровых номеров и получать данные с сервера по API.

Перед запросом к API, компонент должен проверить, есть ли данные по кадастровому номеру в базе данных.

1. Если есть
  - Вернуть модель.
2. Если данных в базе нет
  - Получить данные через API
  - Сохранить их в базу данных
  - Вернуть модель.

## API

Endpoint - GET <http://pkk.bigland.ru/api/test/plots>

Пример запроса:

```
GET http://pkk.bigland.ru/api/test/plots
Content-Type: application/json
Accept: application/json

{
  "collection": {
    "plots": [
      "69:27:0000022:1306",
      "69:27:0000022:1307"
    ]
  }
}
```

## Тесты

Написать один PHPUnit или Codeception тест для проверки работы парсера.

## Документация

Написать документацию по следующим пунктам:

- Развертывание приложения,
- Использование компонента
- Скриншоты результата работы приложения

## Приложение

Создать следующие контроллеры, которые будут использовать компонент.

1. Web-контроллер
2. Rest-контроллер
3. Console-контроллер

## Послесловие

Приложение должно быть создано на любом из шаблонов (в зависимости от вашего стека):

- [yii/yii2-app-advanced](#)
- [laravel/laravel](#)
- [symfony/symfony-demo](#)

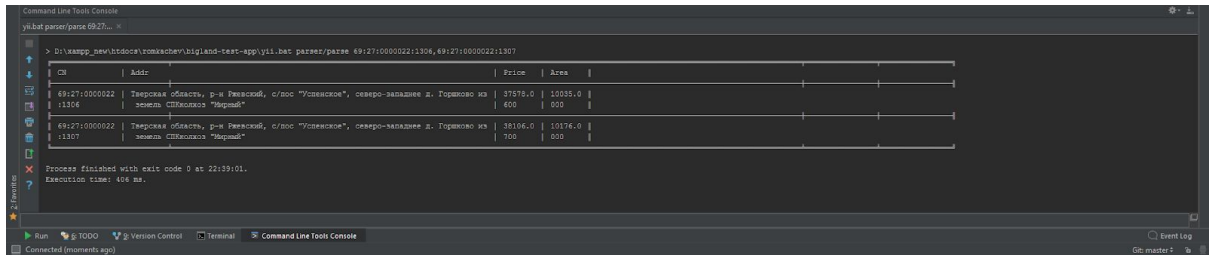
При разработке, должны использоваться возможности фреймворка.

Запросы должны строится на основе встроенного, используемого по умолчанию http-клиента.

- Для Yii - [yiisoft/yii2-httpclient](#)
- Для Laravel - [guzzle/guzzle](#)
- Для Symfony - [symfony/http-client](#) или [guzzle/guzzle](#)

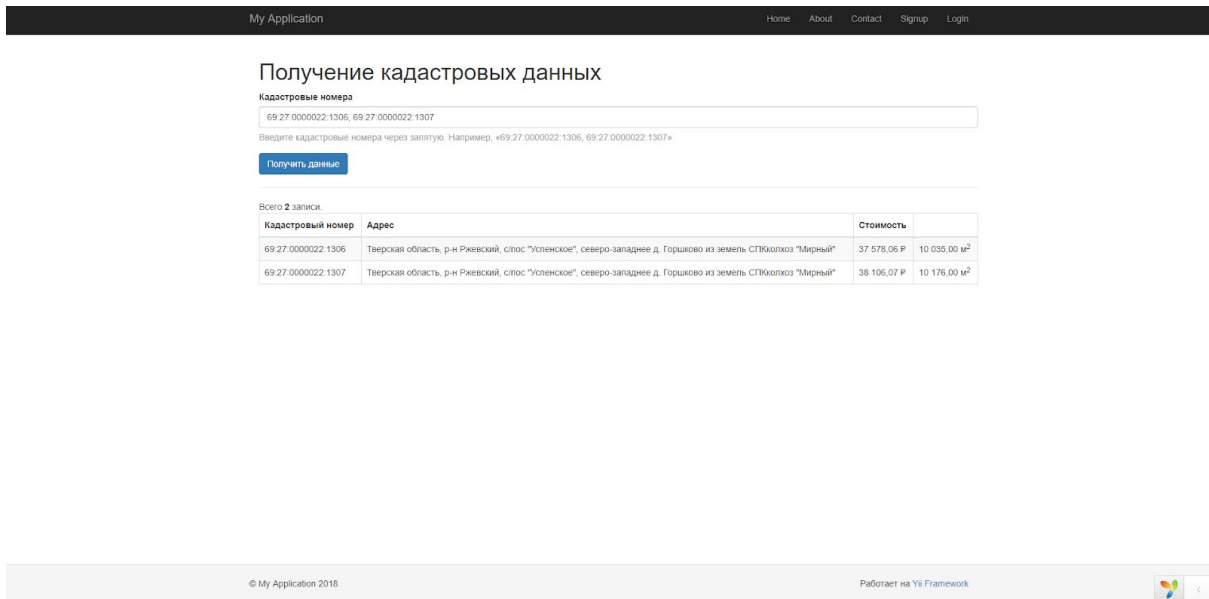
## Скриншоты ожидаемого результата

### Console



[Оригинал изображения](#)

### Web



[Оригинал изображения](#)

## Rest

```
GET http://pkk.bigland.loc/api/test/plots

HTTP/1.1 200 OK
Server: nginx/1.16.1 (Ubuntu)
Date: Tue, 03 Mar 2020 09:52:04 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Ray-Id: 6559208
Vary: Accept
X-Pagination-Total-Count: 0
X-Pagination-Page-Count: 0
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://pkk.bigland.loc/api/test/plots/index?page=1>; rel=self
X-Debug-Tag: 5e5e28c41b800
X-Debug-Duration: 53
X-Debug-Link: /debug/default/view?tag=5e5e28c41b800

[
  {
    "cadastral_number": "69:27:0000022:1306",
    "address": "Тверская область, р-н Ржевский, с/пос \"Успенское\", северо-западнее д. Горшково из земель СПКколхоз \"Мирный\"",
    "price": 36126,
    "area": 10035,
    "_links": []
  },
  {
    "cadastral_number": "69:27:0000022:1307",
    "address": "Тверская область, р-н Ржевский, с/пос \"Успенское\", северо-западнее д. Горшково из земель СПКколхоз \"Мирный\"",
    "price": 36633.6,
    "area": 10176,
    "_links": []
  }
]

Response code: 200 (OK); Time: 71ms; Content length: 413 bytes
```

[Оригинал изображения](#)