

MARKETPRICE – сервис на основе ИИ для оценки рыночной стоимости

Информация по эксплуатации экземпляра ПО

Содержание:

1. Описание API.....	3
1.1. Базовый URL.....	3
1.2. Методы API.....	3
1.2.1. Метод: POST "/api/token/".....	3
1.2.2. Метод: POST "/api/token/refresh/".....	3
1.2.3. Метод: POST "/car/estimate".....	4
1.2.4. Метод: POST "/home/estimate".....	5
1.2.5. Метод: GET "/laptops/priceAllByParams/{manufacturer_name}/{display_size}/{resolution}/{cpu}/{ram}/{storage_type}/{storage_size}".....	6
1.2.6. Метод: GET "/laptops/priceAllByParams/Dell/15.6/1920x1080/Intel%20Core%20i7/16/SSD/512.....	7
1.2.7. Метод: GET "/laptops/priceAppleByParams/{model_name}/{year}/{display_size}/{resolution}/{cpu}/{ram}/{storage_type}/{storage_size}".....	7
1.2.8. Метод: GET "/phones/allPrices".....	8
1.2.9. Метод: GET "/phones/search/".....	8
1.2.10. Метод: GET "/phones/search/{id}/".....	9
1.2.11. Метод: GET "/tablet/allPrices".....	10
1.2.12. Метод: GET "/tablet/manufactures".....	10
1.2.13. Метод: GET "/tablet/modelsByManufacturer/{manufacturer_name}".....	11
1.2.14. Метод: GET "/tablet/paramsByManufacturer/{model_name}".....	12
1.2.15. Метод: GET "/tablet/priceByParams/{model_name}/{ram}/{builtin_memory}/{has_sim}/".....	13
1.3. Коды ответов.....	13
1.4. Аутентификация.....	13
2. Инструкции по интеграции.....	14
3. Требования к инфраструктуре.....	14
Приложение А. (Список данных для запросов).....	16

1. Описание API

1.1. Базовый URL

<https://api.estimator.marketprice.pro>

1.2. Методы API

1.2.1. Метод: POST "/api/token/"

Описание: Получение пары токенов (access и refresh) для аутентификации пользователя.

Тело запроса:

```
{
  "username": "string",
  "password": "string"
}
```

Пример запроса:

POST /api/token/

```
{
  "username": "user123",
  "password": "password123"
}
```

Пример ответа:

```
{
  "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

1.2.2. Метод: POST "/api/token/refresh/"

Описание: Обновление access-токена с использованием refresh-токена.

Тело запроса:

```
{
  "refresh": "string"
}
```

Пример запроса:

POST /api/token/refresh/

```
{
  "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Пример ответа:

```
{
  "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

1.2.3. Метод: POST "/car/estimate"

Описание: Оценка стоимости автомобиля на основе предоставленных параметров.

Тело запроса:

```
{
  "year": 2020,
  "distance": 50000,
  "engine_volume": 2.0,
  "horsepower": 150,
  "brand": "BMW",
  "model": "3 series",
  "generation": "F30/F31",
  "owners": "1",
  "engine_type": "Benzin",
  "transmission": "Automat",
  "drive": "Front",
  "complectation": "Base",
  "body_type": "Sedan"
}
```

Пример запроса:

POST /car/estimate

```
{
  "year": 2020,
  "distance": 50000,
  "engine_volume": 2.0,
  "horsepower": 150,
  "brand": "BMW",
  "model": "3 series",
  "generation": "F30/F31",

```

```
"owners": "1",  
"engine_type": "Benzin",  
"transmission": "Automat",  
"drive": "Front",  
"complectation": "Base",  
"body_type": "Sedan"  
}
```

Пример ответа:

```
{  
  "predicted_price": "25000.00"  
}
```

Примечание: данные для некоторых полей указаны в Приложении А.

1.2.4. Метод: POST "/home/estimate"

Описание: Оценка стоимости недвижимости на основе предоставленных параметров.

Тело запроса:

```
{  
  "lat": 55.7558,  
  "lng": 37.6176,  
  "area": 50.0,  
  "rooms": 2,  
  "floor": 5,  
  "floors_in_house": 10,  
  "ceiling_height": 2.7,  
  "kitchen_area": 10.0,  
  "living_space": 30.0,  
  "year_construction": 2000,  
  "type_house": "type_house_Кирпичный",  
  "repair": "repair_Косметический",  
  "bathroom": ["bathroom_Совмещённый"],  
  "parking": ["parking_underground_True"],  
  "yard": ["yard_playground_True"],  
  "window": ["window_inside_True"]  
}
```

Пример запроса:

POST /home/estimate

```
{
  "lat": 55.7558,
  "lng": 37.6176,
  "area": 50.0,
  "rooms": 2,
  "floor": 5,
  "floors_in_house": 10,
  "ceiling_height": 2.7,
  "kitchen_area": 10.0,
  "living_space": 30.0,
  "year_construction": 2000,
  "type_house": "type_house_Кирпичный",
  "repair": "repair_Косметический",
  "bathroom": ["bathroom_Совмещённый"],
  "parking": ["parking_underground_True"],
  "yard": ["yard_playground_True"],
  "window": ["window_inside_True"]
}
```

Пример ответа:

```
{
  "predicted_price": "5000000.00"
}
```

Примечание: данные для некоторых полей указаны в Приложении А.

1.2.5. Метод: GET

/laptops/priceAllByParams/{manufacturer_name}/{display_size}/{resolution}/{cpu}/{ram}/{storage_type}/{storage_size}"

Описание: Получение стоимости ноутбука любого производителя (кроме Apple) на основе параметров.

Параметры пути:

- "manufacturer_name": Название производителя (например, "Dell").
- "display_size": Размер дисплея (например, "15.6").
- "resolution": Разрешение экрана (например, "1920x1080").

- "cpu": Процессор (например, "Intel Core i7").
- "ram": Объем оперативной памяти (например, 16).
- "storage_type": Тип накопителя (например, "SSD").
- "storage_size": Объем накопителя (например, 512).

Пример запроса:

1.2.6. Метод: GET

/laptops/priceAllByParams/Dell/15.6/1920x1080/Intel%20Core%20i7/16/SSD/512

Пример ответа:

```
{  
  "predicted_price": "1200.00"  
}
```

1.2.7. Метод: GET

/laptops/priceAppleByParams/{model_name}/{year}/{display_size}/{resolution}/{cpu}/{ram}/{storage_type}/{storage_size}"

Описание: Получение стоимости ноутбука Apple на основе указанных параметров.

Параметры пути:

- "model_name": Название модели (например, "MacBook Pro").
- "year": Год выпуска (например, 2021).
- "display_size": Размер дисплея (например, "13.3").
- "resolution": Разрешение экрана (например, "2560x1600").
- "cpu": Процессор (например, "Apple M1").
- "ram": Объем оперативной памяти (например, 16).
- "storage_type": Тип накопителя (например, "SSD").
- "storage_size": Объем накопителя (например, 512).

Пример запроса:

GET

/laptops/priceAppleByParams/MacBook%20Pro/2021/13.3/2560x1600/Apple%20M1/16/SSD/512/

Пример ответа:

```
{  
  "predicted_price": "1800.00"  
}
```

1.2.8. Метод: GET "/phones/allPrices"

Описание: Получение списка всех телефонов с ценами.

Параметры:

- "page" (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

GET /phones/allPrices?page=2

Пример ответа:

```
{
  "count": 100,
  "next":
  "https://api.estimator.marketprice.pro/phones/allPrices?page=3",
  "previous":
  "https://api.estimator.marketprice.pro/phones/allPrices?page=1",
  "results": [
    {
      "id": 1,
      "ram": 8,
      "builtin_memory": 128,
      "predicted_price": "500.00"
    },
    {
      "id": 2,
      "ram": 6,
      "builtin_memory": 64,
      "predicted_price": "400.00"
    }
  ]
}
```

1.2.9. Метод: GET "/phones/search/"

Описание: Быстрый поиск телефонов по названию, производителю или модели.

Параметры:

- "search": Поисковый запрос.
- "only_category" (опционально): Поиск только в категориях (по умолчанию false).
- "page" (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

GET /phones/search/?search=iPhone&only_category=true&page=1

Пример ответа:

```
{
  "results": [
    {
      "id": 1,
      "title": "iPhone 13",
      "description": "Смартфон Apple iPhone 13",
      "type": "phone"
    },
    {
      "id": 2,
      "title": "iPhone 12",
      "description": "Смартфон Apple iPhone 12",
      "type": "phone"
    }
  ]
}
```

1.2.10. Метод: GET "/phones/search/{id}/"

Описание: Возвращает детали элемента поиска по его ID.

Параметры пути:

- "id": Уникальный идентификатор элемента поиска (обязательный).

Пример запроса:

GET /phones/search/123/

Пример ответа:

```
{
  "id": 123,
  "title": "iPhone 13",
  "description": "Смартфон Apple iPhone 13 с 128 ГБ памяти",
  "type": "phone",
  "param": {
    "ram": 6,
    "builtin_memory": 128
  }
}
```

```
}
```

1.2.11. Метод: GET "/tablet/allPrices"

Описание: Получение списка всех планшетов с ценами.

Параметры:

- "page" (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

```
GET /tablet/allPrices?page=2
```

Пример ответа:

```
{
  "count": 50,
  "next":
  "https://api.estimator.marketprice.pro/tablet/allPrices?page=3",
  "previous":
  "https://api.estimator.marketprice.pro/tablet/allPrices?page=1",
  "results": [
    {
      "id": 1,
      "ram": 4,
      "builtin_memory": 64,
      "predicted_price": "300.00"
    },
    {
      "id": 2,
      "ram": 6,
      "builtin_memory": 128,
      "predicted_price": "450.00"
    }
  ]
}
```

1.2.12. Метод: GET "/tablet/manufactures"

Описание: Получение списка всех производителей планшетов.

Параметры:

- "page" (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

```
GET /tablet/manufactures?page=1
```

Пример ответа:

```
{
  "count": 10,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "manufacturer": "Apple"
    },
    {
      "id": 2,
      "manufacturer": "Samsung"
    }
  ]
}
```

1.2.13. Метод: GET

`"/tablet/modelsByManufacturer/{manufacturer_name}"`

Описание: Получение списка моделей планшетов по производителю.

Параметры пути:

- `"manufacturer_name"`: Название производителя (например, "Apple").

Параметры:

- `"page"` (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

`GET /tablet/modelsByManufacturer/Apple?page=1`

Пример ответа:

```
{
  "count": 5,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
```

```
    "model": "iPad Pro"
  },
  {
    "id": 2,
    "model": "iPad Air"
  }
]
```

1.2.14. Метод: GET "/tablet/paramsByManufacturer/{model_name}"

Описание: Получение списка параметров планшета по модели.

Параметры пути:

- "model_name": Название модели планшета (например, "iPad Pro").

Параметры запроса:

- "page" (опционально): Номер страницы (по умолчанию 1).

Пример запроса:

GET /tablet/paramsByManufacturer/iPad%20Pro?page=1

Пример ответа:

```
{
  "count": 3,
  "next": null,
  "previous": null,
  "results": [
    {
      "model_id": 1,
      "ram": 8,
      "builtin_memory": 128,
      "has_sim": 1
    },
    {
      "model_id": 2,
      "ram": 8,
      "builtin_memory": 256,
      "has_sim": 1
    },
    {
```

```
"model_id": 3,  
"ram": 16,  
"builtin_memory": 512,  
"has_sim": 0  
}  
]  
}
```

1.2.15. Метод: GET

"/tablet/priceByParams/{model_name}/{ram}/{builtin_memory}/{has_sim}"/

Описание: Получение цен на планшеты по параметрам.

Параметры пути:

- "model_name": Название модели (например, "iPad Pro").
- "ram": Объем оперативной памяти (например, 8).
- "builtin_memory": Объем встроенной памяти (например, 256).
- "has_sim": Наличие SIM-карты (1 — есть, 0 — нет).

Пример запроса:

GET /tablet/priceByParams/iPad%20Pro/8/256/1/

Пример ответа:

```
{  
  "predicted_price": "900.00"  
}
```

1.3. Коды ответов

- "200 OK": Успешный запрос;
- "400 Bad Request": Неверные параметры запроса;
- "401 Unauthorized": Требуется авторизация;
- "404 Not Found": Ресурс не найден;
- "500 Internal Server Error": Ошибка на сервере.

1.4. Аутентификация

Логин: estimator_adm

Пароль: Estimator0805

Шаг 3: Обработка ответов

- Проверяйте коды ответов и обрабатывайте ошибки;
- Используйте данные из ответа для интеграции с вашей системой.

3. Требования к инфраструктуре

Требования к клиенту

- Операционная система: любая поддерживающая актуальные версии браузеров;
- Браузер: актуальные версии Google Chrome, Mozilla Firefox, Safari или аналогичные;
- Сеть: Стабильное интернет-соединение (рекомендуемая скорость 5 Мбит/с).

Требования к серверу (для интеграции)

- Протоколы: HTTPS (TLS 1.2 или выше);
- Частота запросов: Не более 10 запросов в секунду;
- Формат данных: JSON.

Требования к безопасности

- Используйте HTTPS для всех запросов;
- Храните API-ключи в защищенном месте;
- Регулярно обновляйте библиотеки и зависимости.

Приложение А. (Список данных для запросов)

Справочные данные для API `"/car/estimate"`:

"BrandEnum":

- "BMW",
- "Mercedes-Benz"

"ModelEnum":

- "3 series",
- "5 series",
- "E-Class",
- "C-Class"

"EngineTypeEnum":

- "Benzin",
- "Diesel",
- "Hybrid"

"TransmissionEnum":

- "Automat",
- "Manual",
- "Robot"

"DriveEnum":

- "Front",
- "Back",
- "Full"

"BodyTypeEnum":

- "Sedan",
- "Hatchback",
- "SUV"

Справочные данные для API `"/home/estimate"`:

"TypeHouseEnum":

- "type_house_Кирпичный",
- "type_house_Панельный",

- "type_house_Монолитный"

"RepairEnum":

- "repair_Косметический",
- "repair_Евро",
- "repair_Дизайнерский"

"BathroomEnum":

- "bathroom_Совмещённый",
- "bathroom_Раздельный"

"ParkingEnum":

- "parking_underground_True",
- "parking_open_True"

"YardEnum":

- "yard_playground_True",
- "yard_closed_True"

"WindowEnum":

- "window_inside_True",
- "window_outside_True"