

**Задача**

**Данные о товаре, которые необходимо получать**

**Общее техническое описание решения**

**Схема взаимодействия с решением**

**Технический стек**

**Конфигурация сервиса**

**Формат объекта для сериализации и отправки в RabbitMQ**

**Endpoint'ы для запросов к сервису со стороны внешних систем**

**Парсинг категории товаров и подкатегорий в ней**

**Парсинг конкретной карточки товара**

**Дополнительные требования к соискателю**

**Стоимость работ**

**Требования к отклику**

## Задача

Необходимо реализовать парсинг товаров МегаМаркет в указанной категории товаров. Полученный результат сериализовать в json и отправлять в RabbitMQ

## Данные о товаре, которые необходимо получить

The screenshot shows the product page for a Babycare Venga stroller. Red arrows with numbers 1 through 5 point to specific data points required for the task:

- 1. **Наименование товара:** Коляска прогулочная Babycare, Venga, надувные колеса, Тёмно-серый
- 2. **Стоимость товара со скидкой:** 11 123 Р
- 3. **Стоимость товара без скидки:** 15 899 Р
- 4. **Размер скидки при оплате Сбером:** -30%
- 5. **Дополнительное вознаграждение с подпиской СберПрайм:** +223 Р

Additional visible details include a 4.8 rating, 540 reviews, 118 questions, and a 'Купить' button.

1. наименование товара
2. стоимость товара со скидкой (если есть скидка)
3. стоимость товара без скидки
4. размер скидки при оплате Сбером:
  - размер скидки в %
  - размер скидки в “Спасибо”
5. Дополнительное вознаграждение с подпиской СберПрайм

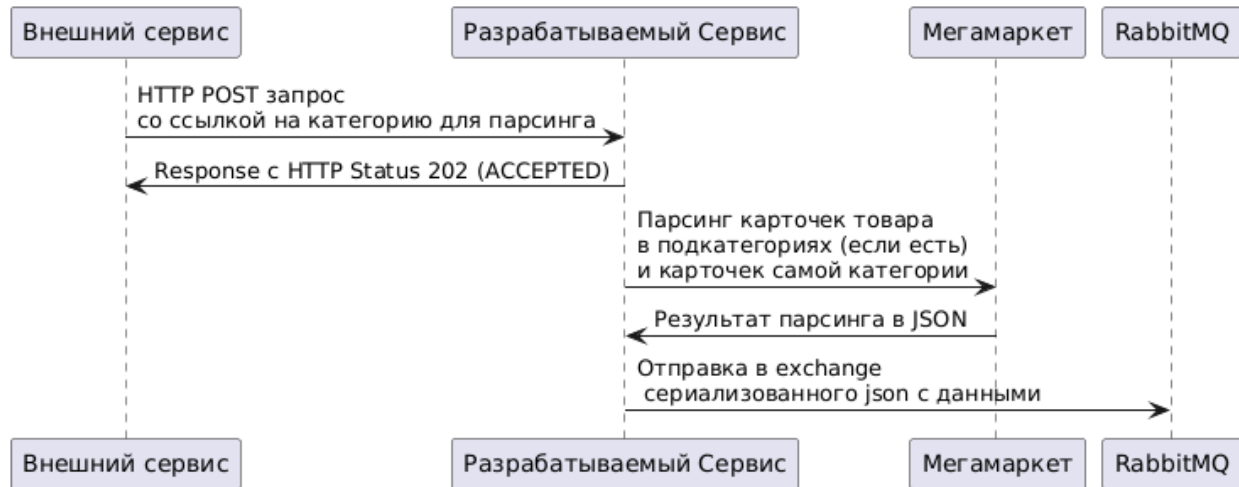
## Общее техническое описание решения

Итоговый вариант реализации должен представлять собой Spring Boot Web приложение.

Приложение получает от внешнего сервиса пост-запросы: - на парсинг категории - на парсинг карточки

Результат парсинга карточек в виде сериализованного json отправляются в exchange в RabbitMQ

## Схема взаимодействия с решением



## Технический стек

- Java 21
- Spring Boot 3.3+
- RabbitMQ
- Дополнительные библиотеки/фреймворки - на усмотрение Исполнителя

## Конфигурация сервиса

Конфигурирование сервиса производится с помощью стандартного `application.yaml`, в котором должно быть указано:

- `application.name`
- подключение к RabbitMQ, а именно:
  - сервер
  - порт
  - virtual host
- название очереди для отправки данных `destination-exchange`

## Формат объекта для сериализации и отправки в RabbitMQ

```
public record Product(  
    @JsonProperty("url")  
    String url, // ссылка на товар  
  
    @JsonProperty("name")  
    String name, // наименование товара  
  
    @JsonProperty("sale_price")  
    BigDecimal salePrice, // стоимость товара со скидкой
```

```

@JsonProperty("price")
BigDecimal price, // стоимость товара без скидки

@JsonProperty("sber_thnx_percent")
Double sberThnxPercent, // размер скидки при оплате Сбером в %

@JsonProperty("sber_thnx_amount")
BigDecimal sberThnxAmount, // размер скидки при оплате Сбером в "спасибо"

@JsonProperty("sber_prime_amount")
BigDecimal sberPrimeAmount, // дополнительное вознаграждение с подпиской
СберПрайм
){}

```

## Endpoint'ы для запросов к сервису со стороны внешних систем

### Парсинг категории товаров и подкатегорий в ней

POST /api/v1/category  
Content-Type: application/json

```

{
  "url": "Ссылка на категорию товаров в Мегамаркет"
}

```

### Парсинг конкретной карточки товара

POST /api/v1/product  
Content-Type: application/json

```

{
  "url": "Ссылка на категорию товаров в Мегамаркет"
}

```

## Дополнительные требования к соискателю

- наличие статуса ИП или самозанятого будет большим плюсом

## Стоимость работ

- Стоимость работ обсуждается с Исполнителем