

## Общее описание

Ваша задача – разработать агента, который будет предоставлять информацию об Университете ИТМО. Этот бот должен совмещать общие сведения, полученные из языковой модели (например, DeepSeek, GPT, LLaMA и т.д.), с фактами и дополнительным контекстом, полученным из внешних источников.

Главное требование: формат JSON-ответа

Все ответы формируются и возвращаются в формате JSON со следующими ключами:

- *id* — числовое значение, соответствующее идентификатору запроса (передаётся во входном запросе).
- *answer* — числовое значение, содержащее правильный ответ на вопрос (если вопрос подразумевает выбор из вариантов). Если вопрос не предполагает выбор из вариантов, значение должно быть `null`.
- *reasoning* — текстовое поле, содержащее объяснение или дополнительную информацию по запросу.
- *sources* — список ссылок на источники информации (если используются). Если источники не требуются, значение должно быть пустым списком `[]`.

Итоговый ответ формируется как валидный JSON-объект, а не строка с сериализованным JSON. Убедитесь, что возвращаемый объект всегда является валидным JSON.

Пример запроса:

POST /api/request

```
{
  "query": "В каком городе находится главный кампус Университета ИТМО?\n1. Москва\n2. Санкт-Петербург\n3. Екатеринбург\n4. Нижний Новгород",
  "id": 1
}
```

Ответ

```
{
  "id": 1,
  "answer": 1,
  "reasoning": "Из информации на сайте",
  "sources": [
    "https://itmo.ru/ru/",
    "https://abit.itmo.ru/"
  ]
}
```

---

## Дополнительные требования

### 1. Поиск ссылок в интернете

- Бот должен уметь выполнять поиск по запросу пользователя, связанному с Университетом ИТМО.
- В результате запроса возвращается не более **трёх** релевантных ссылок. Для получения актуальной информации используйте доступный API (например, Bing Search API, Google Custom Search API и т. п.).

### 2. Получение последних новостей об Университете ИТМО

- Бот должен выводить последние новости, связанные с ИТМО, из официальных источников (например, [ITMO News](#) или других).
- Для получения новостей можно использовать web scraping или RSS-ленты.
- Убедитесь, что вы соблюдаете политику использования данных (Terms of Service) сайта.

### 3. Контейнеризация (Docker)

Шаблон сервиса будет передан организаторами.

- Подготовьте **Dockerfile** для сборки проекта в контейнер.
- Установите все зависимости и скопируйте код бота в контейнер.
- Весь сервис должен подниматься по команде:  
**docker-compose up -d**
- Доступ к решению должен осуществляться через API по следующему эндпоинту:  
**POST <ip>:<port>/api/request**

**Обратите внимание:** Предполагается, что бот получает вопрос и возвращает JSON.

---

## Требования к реализации

### 1. Разработка чат-бота

- Напишите серверное или консольное приложение на Python, которое будет принимать запросы по REST API.
- При получении вопроса бот должен обратиться к языковой модели для генерации ответа.
- Итоговый ответ формируется **исключительно** как строка с сериализованным JSON-объектом.

### 2. Ответ в формате JSON

- В **каждом** ответе должны быть описанные в общем описании поля.
- Убедитесь, что возвращаемая строка всегда является валидным JSON.

### 3. Сборка в Docker

- Создайте **Dockerfile**, который:
  1. Использует базовый образ (например, **python:3.9-slim**).
  2. Устанавливает все необходимые зависимости (включая языковую модель и библиотеки для парсинга/поиска).

3. Запускает ваше приложение.
  - Подготовьте `docker-compose.yml`, в котором:
    1. Описана сборка образа из `Dockerfile`.
    2. Указан запуск сервиса/контейнера.
    3. Настроен проброс порта `<port>` наружу, чтобы можно было выполнять запросы вида `POST <ip>:<port>/api/request`.
    4. Запуск всего проекта должен осуществляться командой:  
**`docker-compose up`**
  4. **Запуск и предоставление**
    - Предоставьте ссылку на репозиторий с кодом вашего проекта
    - Сервис должен быть доступен в течение как минимум 24 часов с момента окончания разработки
- 

## Пример взаимодействия с пользователем и пример JSON-ответов

### Формат вопроса с вариантами ответов (цифры от 1 до 10)

- Вопросы, которые задаются боту, всегда содержат варианты ответов, пронумерованные цифрами от 1 до 10.
- Каждый вариант ответа соответствует определённому утверждению или факту. Бот должен определить правильный вариант ответа и вернуть его в поле `answer` JSON-ответа. Если вопрос не предполагает выбор из вариантов, поле `answer` должно содержать `null`.

#### Структура вопроса

Формат вопроса:

- Вопрос всегда начинается с текстового описания.
- После описания перечисляются варианты ответов, каждый из которых пронумерован цифрой от 1 до 10.
- Варианты ответов разделяются символом новой строки (`\n`).

#### Ответ бота:

- Бот должен определить правильный вариант ответа.
- Правильный вариант указывается в поле `answer` (например, 1, 2, ..., 10).
- Если вопрос не предполагает выбор из вариантов, поле `answer` должно быть `null`.

### Запрос от пользователя:

```
{
  "query": "В каком году Университет ИТМО был включён в число Национальных исследовательских университетов России?\n1. 2007\n2. 2009\n3. 2011\n4. 2015",
  "id": 2
}
```

### Пример ответа (JSON):

```
{
  "id": 2,
  "answer": 2,
  "reasoning": "Университет ИТМО был включён в число Национальных исследовательских университетов России в 2009 году. Это подтверждается официальными данными Министерства образования и науки РФ.",
  "sources": ["https://www.itmo.ru", "https://минобрнауки.рф"]
}
```

---

### Запрос от пользователя:

```
{
  "query": "В каком рейтинге (по состоянию на 2021 год) ИТМО впервые вошёл в топ-400 мировых университетов?\n1. ARWU (Shanghai Ranking)\n2. Times Higher Education (THE) World University Rankings\n3. QS World University Rankings\n4. U.S. News & World Report Best Global Universities",
  "id": 3
}
```

### Пример ответа (JSON):

```
{
  "id": 3,
  "answer": 3,
  "reasoning": "В 2023 году Университет ИТМО впервые вошёл в топ-400 мировых университетов согласно рейтингу ARWU (Shanghai Ranking). Это достижение подтверждается официальным сайтом рейтинга.",
  "sources": ["https://news.itmo.ru/ru/university_live/ratings/news/10389/"]
}
```

## Итоговые критерии оценки

- 1. Использование моделей DeepSeek/GPT/LLaMA и др**
  - агент должен сообщать, какой моделью сгенерирован ответ
- 2. Корректное формирование JSON-ответов**
  - Все ответы строго в формате JSON: ответ формируется как валидный JSON-объект
- 3. Поиск ссылок и новостей**
  - При необходимости бот запрашивает свежую информацию из поисковых систем или RSS-ленты / сайта новостей, добавляя не более трёх ссылок

#### **4. Инференс решения и предоставление URL**

- Участники должны предоставить работающий сервис, доступный через REST API.
- Сервис должен быть размещён на публично доступном URL (например, с использованием облачных сервисов, таких как Yandex Cloud, Heroku и т. д.).
- URL должен быть предоставлен организаторам для тестирования и оценки работы бота.
- Предоставьте ссылку на репозиторий с кодом вашего проекта
- Сервис должен быть доступен в течение как минимум 24 часов с момента окончания разработки

#### **5. Тестирование**

- Бот будет протестирован на примерно 100 вопросах, охватывающих различные темы, связанные с Университетом ИТМО (история, факультеты, рейтинги, новости и т. д.).
- На ответы на все вопросы будет выделено 7 минут.
- Бот должен корректно обрабатывать запросы в рамках этого временного ограничения.