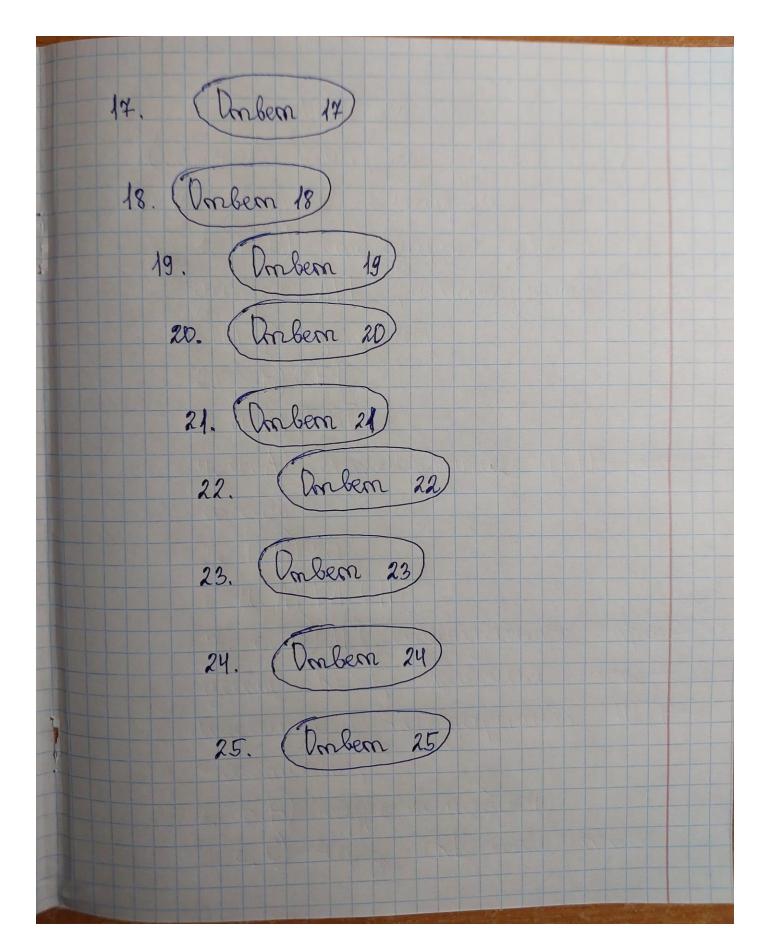
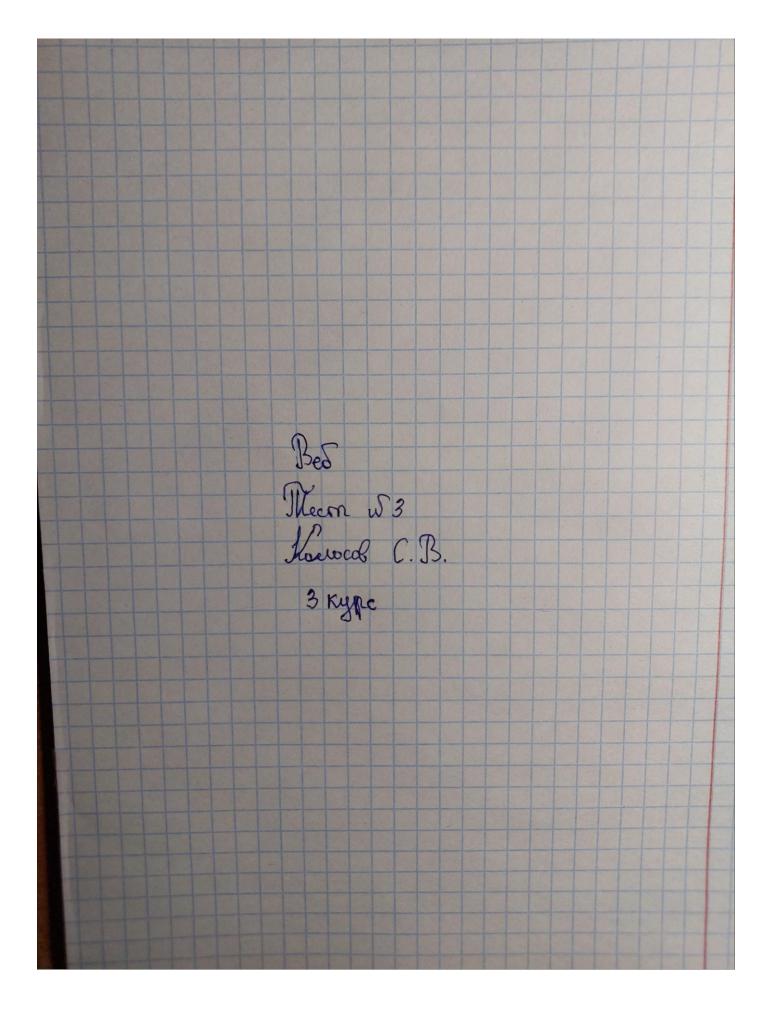
Web-noorpauleurepobature
Therm w3
Nouvcob C.B. 09.03.01 213M - 3, 3 Kype

Mean w3 Mecmober Koumpoul , Donobe Rest API 1-15. Bonpoese c bredopoul 1. b. B) Conaregapor gua cozgarna API, u crontzy sousen HTTP zampocon; 2. b. B) GET 3. a.A) POST y. a. A) Ucrochozobarre HTTP- elemogobo Du 1 5. B.B) 200 6. L. A) PATCH 7. c. C) APT, coontagaragee nouryearse REST, Excursioned ucrouszobatul concugarontesex HTTP-- memogol; 8. b.B) Peagre the transgett 9. c.C) Mepez URL ment

zanpocd; 10. B) JSON 11. B) 21 RL, no komopouly burner warman zampoc K API 12. C) Obrobusion cycycombyroyour pecype

13. C) DELETE 14. D) GET 15. B) 201 Bonpock na conocinabelliere (26.) 48 16. -25. Donapsense bongock 16. Omben 16)

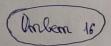




Что такое REST API и какие основные принципы лежат в его основе?

REST API (Representational State Transfer Application Programming Interface) — это набор правил и принципов для создания веб-сервисов, которые позволяют взаимодействовать с ресурсами через HTTP-протокол. Основные принципы REST:

- **Клиент-серверная архитектура**: Клиент и сервер работают независимо друг от друга.
- **Безсостояние (Stateless)**: Каждый запрос от клиента к серверу должен содержать всю необходимую информацию для его обработки.
- Кэширование: Ответы сервера могут быть кэшированы клиентом для улучшения производительности.
- **Единообразие интерфейса**: Использование стандартных HTTPметодов (GET, POST, PUT, DELETE и т.д.).
- **Cлои (Layered System)**: Архитектура может включать промежуточные серверы, такие как прокси или шлюзы.
- **Код по требованию (Code on Demand, опционально)**: Сервер может отправлять исполняемый код клиенту для выполнения.

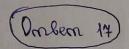


Опишите, как работает метод GET в контексте REST API и как его использовать для получения данных.

Метод **GET** используется для получения данных с сервера. Он не изменяет состояние ресурса на сервере. Параметры запроса передаются в URL. Пример:

GET /users?id=123

Этот запрос вернет данные пользователя с ID 123.



Как в REST API происходит обработка запросов с использованием метода POST? Приведите пример его применения.

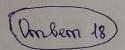
Метод **POST** используется для отправки данных на сервер, обычно для создания нового ресурса. Данные передаются в теле запроса. Пример:

POST /users

Content-Type: application/json

```
"name": "John Doe",
"email": "john.doe@example.com"
}
```

Этот запрос создаст нового пользователя с указанными данными.



Объясните, чем отличаются методы PUT и PATCH в REST API. В каких случаях следует использовать каждый из них?

- **PUT**: Используется для полного обновления ресурса. Клиент отправляет полное представление ресурса.
- **PATCH**: Используется для частичного обновления ресурса. Клиент отправляет только измененные данные.

Примеры:

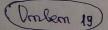
• PUT:

PUT /users/123

Content-Type: application/json

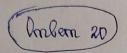
```
{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "age": 30
}
    PATCH:
PATCH /users/123
Content-Type: application/json

{
    "email": "new.email@example.com"
```



Что такое статусный код 500, и в каких случаях он может быть возвращен сервером в ответ на запрос?

Статусный код 500 Internal Server Error указывает на то, что сервер столкнулся с непредвиденной ошибкой, которая помешала выполнению запроса. Это может быть вызвано различными причинами, такими как ошибки в коде сервера, проблемы с базой данных и т.д.



Как происходит передача параметров в REST API, если они находятся в теле запроса? Приведите пример.

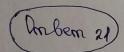
Параметры в теле запроса обычно передаются в формате JSON или XML. Пример:

POST /users

Content-Type: application/json

```
"name": "John Doe",

"email": "john.doe@example.com"
```



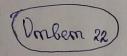
Какие преимущества и ограничения имеет использование JSON для обмена данными в REST API?

Преимущества JSON:

- Легко читаем и пишется.
- Поддерживается большинством языков программирования.
- Меньше объем данных по сравнению с ХМL.

Ограничения JSON:

- Нет встроенной поддержки схем данных.
- Менее строгий формат, что может привести к ошибкам.



Объясните, что такое аутентификация в REST API и как она может быть реализована с помощью токенов.

Аутентификация — это процесс проверки подлинности пользователя. В REST API часто используется токен-аутентификация, где клиент получает токен после успешной аутентификации и использует его для доступа к защищенным ресурсам. Пример:

POST /login

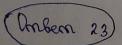
Content-Type: application/json

"username": "user",
"password": "pass"

Сервер возвращает токен, который клиент использует в заголовке Authorization для последующих запросов:

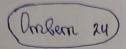
GET /users

Authorization: Bearer <token>



Каковы основные отличия между REST API и SOAP API?

- **REST**: Легковесный, использует стандартные HTTP-методы, данные обычно передаются в формате JSON или XML.
- SOAP: Более строгий протокол, использует XML для передачи данных, поддерживает WSDL для описания сервисов.



Какие средства для тестирования REST API вы бы порекомендовали и почему?

- Postman: Удобный интерфейс, поддержка автоматических тестов, коллекции запросов.
- Insomnia: Легковесный, поддержка работы с окружениями, автоматические тесты.
- Curl: Командная строка, гибкость, подходит для автоматизации.
- **JMeter**: Для нагрузочного тестирования, поддержка сложных сценариев.

Эти инструменты помогают упростить процесс тестирования и отладки REST API.

