**Документация API для сервера**

**Цель**: Эта документация описывает эндпоинты, которые сервер должен реализовать для поддержки функциональности веб-приложения. Приложение использует REST API, отправляет запросы с заголовком Authorization: Bearer <token> для защищённых маршрутов и ожидает ответы в формате JSON.

**Базовый URL**: Предполагается, что сервер будет доступен по адресу https://api.yourserver.com. Замените на реальный URL.

**Аутентификация**:

* Большинство эндпоинтов требуют JWT-токен в заголовке Authorization: Bearer <token>.
* Токен выдаётся при успешной авторизации через /api/login.
* Если токен недействителен или отсутствует, сервер должен возвращать 401 Unauthorized с JSON { "message": "Требуется авторизация" }.

**Общий формат ошибок**:

* При ошибке сервер возвращает JSON: { "message": "<описание ошибки>" }.
* Коды ошибок: 400 (неверные данные), 401 (неавторизован), 403 (доступ запрещён), 404 (не найдено), 500 (внутренняя ошибка).

**Эндпоинты**

**1. Авторизация пользователя**

* **Эндпоинт**: POST /api/login
* **Описание**: Аутентифицирует пользователя и возвращает JWT-токен.
* **Входные данные** (JSON):

json

{

"email": "string",

"password": "string"

}

* **Выходные данные** (успех, 200 OK):

json

{

"token": "string",

"role": "string", // "admin" или "user"

"user\_id": "number"

}

* **Ошибки**:
  + 400: {"message": "Неверные email или пароль"}
  + 500: {"message": "Ошибка сервера"}

**2. Получение текущего пользователя**

* **Эндпоинт**: GET /api/users/me
* **Описание**: Возвращает данные авторизованного пользователя.
* **Заголовки**: Authorization: Bearer <token}
* **Входные данные**: Нет
* **Выходные данные**:

json

{

"id": "number",

"name": "string",

"email": "string",

"role": "string",

"photo": "string | null", // base64-encoded image или null

"lastName": "string | null",

"firstName": "string | null",

"middleName": "string | null",

"completedTasks": ["number"] // массив ID заданий

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 500: {"message": "Ошибка сервера"}

**3. Получение списка пользователей**

* **Эндпоинт**: GET /api/users
* **Описание**: Возвращает список всех пользователей (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: Нет
* **Выходные данные**:

json

[

{

"id": "number",

"name": "string",

"email": "string",

"role": "string",

"photo": "string | null",

"completedTasks": ["number"]

}

]

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"} (если не админ)
  + 500: {"message": "Ошибка сервера"}

**4. Получение пользователя по ID**

* **Эндпоинт**: GET /api/users/:id
* **Описание**: Возвращает данные пользователя по его ID.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: :id (число в URL)
* **Выходные данные**:

json

{

"id": "number",

"name": "string",

"email": "string",

"role": "string",

"photo": "string | null",

"lastName": "string | null",

"firstName": "string | null",

"middleName": "string | null",

"completedTasks": ["number"]

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Пользователь не найден"}
  + 500: {"message": "Ошибка сервера"}

**5. Создание пользователя**

* **Эндпоинт**: POST /api/users
* **Описание**: Создаёт нового пользователя (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные** (JSON):

json

{

"lastName": "string",

"firstName": "string",

"middleName": "string | null",

"email": "string",

"password": "string",

"role": "string", // "admin" или "user"

"photo": "string | null", // base64-encoded image

"name": "string" // объединённое ФИО

}

* **Выходные данные** (201 Created):

json

{

"id": "number",

"email": "string",

"role": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 500: {"message": "Ошибка сервера"}

**6. Обновление пользователя**

* **Эндпоинт**: PUT /api/users/:id
* **Описание**: Обновляет данные пользователя (для админов или самого пользователя).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:
  + :id (число в URL)
  + JSON:

json

{

"lastName": "string | null",

"firstName": "string | null",

"middleName": "string | null",

"email": "string | null",

"password": "string | null",

"role": "string | null",

"photo": "string | null",

"name": "string | null"

}

* **Выходные данные** (200 OK):

json

{

"id": "number",

"email": "string",

"role": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Пользователь не найден"}
  + 500: {"message": "Ошибка сервера"}

**7. Добавление задания пользователю**

* **Эндпоинт**: POST /api/users/:id/tasks
* **Описание**: Добавляет задание в список выполненных пользователем (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:
  + :id (ID пользователя)
  + JSON: { "taskId": "number" }
* **Выходные данные**:

json

{

"completedTasks": ["number"]

}

* **Ошибки**:
  + 400: {"message": "Неверный ID задания"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Пользователь не найден"}
  + 500: {"message": "Ошибка сервера"}

**8. Удаление задания у пользователя**

* **Эндпоинт**: DELETE /api/users/:id/tasks/:taskId
* **Описание**: Удаляет задание из списка выполненных пользователем.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:
  + :id (ID пользователя)
  + :taskId (ID задания)
* **Выходные данные**:

json

{

"completedTasks": ["number"]

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Пользователь или задание не найдены"}
  + 500: {"message": "Ошибка сервера"}

**9. Получение списка преподавателей**

* **Эндпоинт**: GET /api/teachers
* **Описание**: Возвращает список всех преподавателей.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: Нет
* **Выходные данные**:

json

[

{

"id": "number",

"lastName": "string",

"firstName": "string",

"middleName": "string | null",

"department": "string",

"description": "string | null",

"photo": "string | null",

"topics": ["number"] // массив ID тем

}

]

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 500: {"message": "Ошибка сервера"}

**10. Получение преподавателя по ID**

* **Эндпоинт**: GET /api/teachers/:id
* **Описание**: Возвращает данные преподавателя.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: :id (число)
* **Выходные данные**:

json

{

"id": "number",

"lastName": "string",

"firstName": "string",

"middleName": "string | null",

"department": "string",

"description": "string | null",

"photo": "string | null",

"topics": ["number"]

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 404: {"message": "Преподаватель не найден"}
  + 500: {"message": "Ошибка сервера"}

**11. Создание преподавателя**

* **Эндпоинт**: POST /api/teachers
* **Описание**: Создаёт нового преподавателя (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:

json

{

"lastName": "string",

"firstName": "string",

"middleName": "string | null",

"department": "string",

"description": "string",

"photo": "string | null"

}

* **Выходные данные** (201 Created):

json

{

"id": "number",

"lastName": "string",

"firstName": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 500: {"message": "Ошибка сервера"}

**12. Обновление преподавателя**

* **Эндпоинт**: PUT /api/teachers/:id
* **Описание**: Обновляет данные преподавателя (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:
  + :id (число)
  + JSON:

json

{

"lastName": "string | null",

"firstName": "string | null",

"middleName": "string | null",

"department": "string | null",

"description": "string | null",

"photo": "string | null"

}

* **Выходные данные**:

json

{

"id": "number",

"lastName": "string",

"firstName": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Преподаватель не найден"}
  + 500: {"message": "Ошибка сервера"}

**13. Получение списка тем**

* **Эндпоинт**: GET /api/topics
* **Описание**: Возвращает список всех тем.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: Нет
* **Выходные данные**:

json

[

{

"id": "number",

"title": "string",

"template": "string",

"parameters": {

"difficulty": "string", // "easy", "medium", "hard"

"questions": "number"

},

"teacherId": "number"

}

]

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 500: {"message": "Ошибка сервера"}

**14. Получение темы по ID**

* **Эндпоинт**: GET /api/topics/:id
* **Описание**: Возвращает данные темы.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: :id (число)
* **Выходные данные**:

json

{

"id": "number",

"title": "string",

"template": "string",

"parameters": {

"difficulty": "string",

"questions": "number"

},

"teacherId": "number"

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 404: {"message": "Тема не найдена"}
  + 500: {"message": "Ошибка сервера"}

**15. Создание темы**

* **Эндпоинт**: POST /api/topics
* **Описание**: Создаёт новую тему (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:

json

{

"title": "string",

"template": "string",

"parameters": {

"difficulty": "string",

"questions": "number"

},

"teacherId": "number"

}

* **Выходные данные** (201 Created):

json

{

"id": "number",

"title": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 500: {"message": "Ошибка сервера"}

**16. Обновление темы**

* **Эндпоинт**: PUT /api/topics/:id
* **Описание**: Обновляет данные темы (для админов).
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:
  + :id (число)
  + JSON:

json

{

"title": "string | null",

"template": "string | null",

"parameters": {

"difficulty": "string | null",

"questions": null

}

}

* undefined
* **Выходные данные**:

json

{

"id": "number",

"title": "string"

}

* **Ошибки**:
  + 400: {"message": "Неверные данные"}
  + 401: {"message": "Требуется авторизация"}
  + 403: {"message": "Доступ запрещён"}
  + 404: {"message": "Тема не найдена"}
  + 500: {"message": "Ошибка сервера"}

**17. Получение задачи по теме**

* **Эндпоинт**: GET /api/topics/:id/task
* **Описание**: Возвращает задачу, связанную с темой.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: :id (ID темы)
* **Выходные данные**:

json

{

"id": "number",

"title": "string",

"content": "string",

"teacherId": "number"

}

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 404: {"message": "Задача не найдена"}
  + 500: {"message": "Ошибка сервера"}

**18. Отправка ответа на задачу**

* **Эндпоинт**: POST /api/topics/:id/submit
* **Описание**: Проверяет ответ пользователя на задачу.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**:

json

{

"answer": "string"

}

* **Выходные данные**:

json

{

"isCorrect": true

}

* **Ошибки**:
  + 400: {"message": "Неверный ответ"}
  + 401: {"message": "Требуется авторизация"}
  + 404: {"message": "Задача не найдена"}
  + 500: {"message": "Ошибка сервера"}

**19. Получение списка заданий**

* **Эндпоинт**: GET /api/tasks
* **Описание**: Возвращает список всех заданий.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: Нет
* **Выходные данные**:

json

[

{

"id": "number",

"title": "string",

"content": "string"

}

]

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 500: {"message": "Ошибка сервера"}

**20. Получение списка кафедр**

* **Эндпоинт**: GET /api/departments
* **Описание**: Возвращает список всех кафедр.
* **Заголовки**: Authorization: Bearer <token>
* **Входные данные**: Нет
* **Выходные данные**:

json

["string"]

* **Ошибки**:
  + 401: {"message": "Требуется авторизация"}
  + 500: {"message": "Ошибка сервера"}

**Примечания для разработчика сервера**

1. **Формат данных**:
   * Все ответы должны быть в формате JSON.
   * Поля, помеченные как null, могут отсутствовать или быть null.
   * Поле photo ожидается в формате base64 (например, data:image/jpeg;base64,...).
2. **Аутентификация и авторизация**:
   * Реализуйте проверку JWT-токена для всех защищённых эндпоинтов.
   * Для админских действий (POST /api/users, PUT /api/teachers, и т.д.) проверяйте, что пользователь имеет роль admin.
3. **Валидация**:
   * Проверяйте входные данные (например, email должен быть валидным, questions должно быть положительным числом).
   * Возвращайте понятные сообщения об ошибках.