

# REST API системы управления учебными планами

В разделе представлены сценарии REST-методов API для управления учебными процессами.

Для работы с сервисами зарегистрированы следующие контроллеры и эндполиты:

Контроллер	Базовый эндполит	Описание
AuthController	{host}/api/auth	Аутентификация и авторизация
FieldOfStudyController	{host}/api/fields	Публичные операции с направлениями подготовки
StudyPlansController	{host}/api/curricula	Публичные операции с учебными планами
AdminFieldController	{host}/api/admin/fields	Администрирование направлений подготовки
AdminStudyPlanController	{host}/api/admin/fields/{fieldId}/study-plans	Администрирование учебных планов
TestController	{host}/api/test	Тестовые эндпоинты

## Получить токен аутентификации - login

Название сценария	Получить токен аутентификации
Контроллер	AuthController
Название метода	login
Эндполит	{host}/api/auth/login
Глагол	POST
Описание	Метод выполняет аутентификацию пользователя по логину и паролю, возвращает JWT токен для доступа к защищенным эндпоинтам
Ожидаемый результат	Возвращен JWT токен с информацией о пользователе или Возвращено сообщение об ошибке аутентификации
Ограничения	Отсутствуют

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
username	string	Да	Body	Имя пользователя администратора/менеджера
password	string	Да	Body	Пароль пользователя

### Описания сценария

#### Базовый сценарий. Успешная аутентификация

№	Действие	АС	Класс бизнес-исключения
1	Система извлекает данные из запроса метода и проверяет, что заполнены все обязательные поля.	АС.1	ValidationException
2	Система осуществляет поиск пользователя в базе данных при условии user.username = username	АС.2	DatabaseException

№	Действие	АС	Класс бизнес-исключения
	и запоминает результат.		
3	Система проверяет, что результат поиска с шага 2 не пуст.		
4	Система проверяет соответствие хеша пароля из базы данных переданному паролю.		
5	Система генерирует JWT токен по следующему правилу:		
	токен содержит username и role пользователя		
	срок действия токена = 24 часа		
6	Система заполняет результат работы метода по следующему правилу:		
	token = сгенерированный токен		
	username = имя аутентифицированного пользователя		
	role = роль пользователя (ADMIN, MANAGER)		
7	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
8	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

№	Действие системы
1	Система возбуждает исключение ValidationException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Ошибка валидации"
	Content = "Неверные данные"
	и возвращает ответ с http-кодом 400.
2	Сценарий завершен.

### АС.2 Неверные учетные данные

№	Действие системы
1	Система возбуждает исключение RuntimeException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Ошибка аутентификации"
	Content = "Неверное имя пользователя или пароль"
	и возвращает ответ с http-кодом 401.
2	Сценарий завершен.

## Проверить валидность токена - validateToken

Название сценария	Проверить валидность токена
Контроллер	AuthController
Название метода	validateToken

Название сценария	Проверить валидность токена
Эндполит	{host}/api/auth/validate
Глагол	GET
Описание	Метод проверяет валидность JWT токена и возвращает информацию о пользователе
Ожидаемый результат	Возвращена информация о валидности токена и данные пользователя или Возвращено сообщение об ошибке
Ограничения	Требуется заголовок Authorization с Bearer токеном

Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
Authorization	string	Да	Header	Заголовок формата "Bearer {token}"

Описания сценария

Базовый сценарий. Успешная проверка токена

№	Действие	АС	Класс бизнес-исключения
1	Система извлекает токен из заголовка Authorization.	АС.1	ValidationException
2	Система проверяет формат заголовка "Bearer {token}".	АС.1	ValidationException
3	Система извлекает JWT токен из заголовка.		
4	Система проверяет валидность токена (подпись, срок действия).	АС.1	ValidationException
5	Система извлекает данные из токена (username, role, exp).	АС.2	
6	Система вычисляет оставшееся время действия токена в минутах.		
7	Система заполняет результат работы метода по следующему правилу:		
	valid = true		
	username = имя пользователя из токена		
	role = роль пользователя из токена		
	remainingMinutes = оставшееся время в минутах		
8	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
9	Сценарий завершен.		

Альтернативные сценарии

АС.1 Невалидный или просроченный токен

№	Действие системы
1	Система возбуждает исключение RuntimeException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Ошибка аутентификации"
	Content = "Токен недействителен или просрочен"
	и возвращает ответ с http-кодом 401.
2	Сценарий завершен.

АС.2 Токен валиден, но пользователь не аутентифицирован как администратор

№	Действие системы
1	Система успешно проверяет токен и извлекает данные пользователя.
2	Система проверяет наличие специального claim "isAdmin" или роль пользователя.
3	Если пользователь не является администратором, система заполняет результат работы метода по следующему правилу:
	<b>valid</b> = false
4	Тело ответа возвращается запрашивающей системе в формате JSON с http-кодом 200.
5	Сценарий завершен.

Получить учебный план с дисциплинами - **getStudyPlanDisciplines**

Название сценария	Получить учебный план с дисциплинами
Контроллер	StudyPlansController
Название метода	getStudyPlanDisciplines
Эндполит	{host}/api/curricula/{studyPlanId}
Глагол	GET
Описание	Метод возвращает полную информацию об учебном плане, включая дисциплины с преподавателями
Ожидаемый результат	Возвращена информация об учебном плане с дисциплинами или Возвращена пустая коллекция или Возвращено сообщение об ошибке
Ограничения	Отсутствуют

Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
studyPlanId	string	Да	Path	Идентификатор учебного плана

Описания сценария

Базовый сценарий. Получить учебный план по существующему ID

№	Действие	АС	Класс бизнес-исключения
1	Система извлекает идентификатор учебного плана из запроса метода.		
2	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = studyPlanId		
	и запоминает результат.		
3	Система проверяет, что результат поиска с шага 2 не пуст.	АС.1	StudyPlanNotFoundException
4	Система осуществляет поиск информации о направлении подготовки по ID учебного плана.		
	и запоминает результат.		
5	Система осуществляет поиск дисциплин, связанных с учебным планом, через таблицу subsidiary_discip_curriculum.		

№	Действие	АС	Класс бизнес-исключения
	и запоминает результат.		
6	Система осуществляет поиск преподавателей для каждой дисциплины через таблицу subsidiary_discip_teacher.		
	и запоминает результат.		
7	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = объединенные данные с шагов 2-6		
	metadata.objectsCount = количество дисциплин		
	metadata.responseTime = текущая дата и время		
8	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
9	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Учебный план по переданному ID не найден

№	Действие системы
1	Система возбуждает исключение StudyPlanNotFoundException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Учебный план не найден"
	Content = "Учебный план с ID {studyPlanId} не найден"
	metadata = null
	и возвращает ответ с http-кодом 404.
2	Сценарий завершен.

## Создать направление подготовки - createField

Название сценария	Создать направление подготовки
Контроллер	AdminFieldController
Название метода	createField
Эндполит	{host}/api/admin/fields
Глагол	POST
Описание	Метод создает новое направление подготовки
Ожидаемый результат	Возвращена информация о созданном направлении ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldCode	string	Да	Body	Код направления (например "03.03.01")

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldName	string	Да	Body	Название направления
degreeLevel	string	Да	Body	Уровень образования
studyLength	integer	Да	Body	Срок обучения (лет)
profileName	string	Да	Body	Название профиля/специализации
qualification	string	Нет	Body	Квалификация

Описания сценария

Базовый сценарий. Создать направление подготовки

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.3	AccessDeniedException
3	Система извлекает данные из запроса метода и проверяет, что заполнены все обязательные поля.	АС.1	ValidationException
4	Система проверяет, что profileName не существует в базе данных.	АС.2	DuplicateResourceException
5	Система создает новую запись в таблице field_of_study.		
	Поля заполняются по следующему правилу:		
	field_id = сгенерированный базой данных идентификатор		
	field_code = fieldCode		
	field_name = fieldName		
	degree_level = degreeLevel		
	study_length = studyLength		
	profile_name = profileName		
	qualification = qualification		
	created_date = текущая дата и время		
6	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = информация о созданном направлении		
	metadata.responseTime = текущая дата и время		
7	Результат работы метода возвращается запрашивающей системе с http-кодом 201.		
8	Сценарий завершен.		

Альтернативные сценарии

АС.1 Не заполнены обязательные поля

№	Действие системы
1	Система возбуждает исключение ValidationException, заполняет ответ метода по следующему правилу:
	status = "error"

№	Действие системы
	error = "Не заполнены обязательные поля"
	Content = "Не заполнены обязательные поля"
	и возвращает ответ с http-кодом 400.
2	Сценарий завершен.

АС.2 Профиль с наименованием уже существует

№	Действие системы
1	Система возбуждает исключение DuplicateResourceException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Профиль с наименованием уже существует"
	Content = "Профиль с наименованием '{profileName}' уже существует"
	и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

АС.3 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Доступ запрещен"
	Content = "Требуется роль ADMIN для архивации учебных планов"
	и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

Создать учебный план - createStudyPlan

Название сценария	Создать учебный план
Контроллер	AdminStudyPlanController
Название метода	createStudyPlan
Эндполит	{host}/api/admin/fields/{fieldId}/study-plans
Глагол	POST
Описание	Метод создает новый учебный план для указанного направления подготовки
Ожидаемый результат	Возвращена информация о созданном учебном плане ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER

Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
studyPlanName	string	Да	Body	Название учебного плана

Название параметра	Тип данных	Обязательное	Размещение	Описание
yearStart	integer	Да	Body	Год начала действия
yearEnd	integer	Да	Body	Год окончания действия
course	integer	Да	Body	Курс (год обучения)
status	string	Нет	Body	Статус учебного плана (по умолчанию "1.00")
filePath	string	Нет	Body	Путь к файлу

Описания сценария

Базовый сценарий. Создать учебный план

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	AC.2	AccessDeniedException
3	Система извлекает данные из запроса метода и проверяет, что заполнены все обязательные поля.	AC.1	ValidationException
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система проверяет, что комбинация параметров учебного плана уникальна для данного направления.	AC.2	DuplicateResourceException
7	Система создает новую запись в таблице curriculum.		
	Поля заполняются по следующему правилу:		
	curriculum_id = сгенерированный базой данных идентификатор		
	field_id = fieldId		
	curriculum_name = studyPlanName		
	year_start = yearStart		
	year_end = yearEnd		
	archive_status = false		
	created_date = текущая дата и время		
	last_modified = null		
	file_path = filePath		
	course = course		
	status = status (если не передан, то "1.00")		
8	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = информация о созданном учебном плане		
	metadata.responseTime = текущая дата и время		



№	Действие	АС	Класс бизнес-исключения
9	Результат работы метода возвращается запрашивающей системе с http-кодом 201.		
10	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Не заполнены обязательные поля

№	Действие системы
1	Система возбуждает исключение ValidationException, заполняет ответ метода по следующему правилу: status = "error" error = "Не заполнены обязательные поля" Content = "Не заполнены обязательные поля" и возвращает ответ с http-кодом 400.
2	Сценарий завершен.

### АС.2 Учебный план с такими параметрами уже существует

№	Действие системы
1	Система возбуждает исключение DuplicateResourceException, заполняет ответ метода по следующему правилу: status = "error" error = "Учебный план уже существует" Content = "Учебный план с такими параметрами уже существует" и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

### АС.3 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу: status = "error" error = "Доступ запрещен" Content = "Требуется роль ADMIN для архивации учебных планов" и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

## Удалить направление подготовки - deleteField

Название сценария	Удалить направление подготовки
Контроллер	AdminFieldController
Название метода	deleteField
Эндпоинт	{host}/api/admin/fields/{id}
Глагол	DELETE

Название сценария	Удалить направление подготовки
Описание	Метод удаляет направление подготовки по идентификатору
Ожидаемый результат	Возвращено подтверждение удаления ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; не должно быть связанных учебных планов

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
id	string	Да	Path	Идентификатор направления подготовки

### Описания сценария

#### Базовый сценарий. Удалить направление подготовки

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.2	AccessDeniedException
3	Система извлекает идентификатор направления из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = id		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система проверяет, что у направления подготовки нет связанных учебных планов.	АС.1	ValidationException
7	Система удаляет запись из таблицы field_of_study.		
8	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = "Направление с ID {id} удалено"		
	metadata.responseTime = текущая дата и время		
9	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
10	Сценарий завершен.		

## Альтернативные сценарии

#### АС.1 Существуют связанные учебные планы

№	Действие системы
1	Система возбуждает исключение ValidationException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Невозможно удалить направление"
	Content = "Невозможно удалить направление, к которому привязаны учебные планы"
	и возвращает ответ с http-кодом 400.
2	Сценарий завершен.

# Архивировать учебный план - toggleArchiveStatus

Название сценария	Архивировать учебный план
Контроллер	AdminStudyPlanController
Название метода	toggleArchiveStatus
Эндполит	{host}/api/admin/fields/{fieldId}/study-plans/{planId}/archive
Глагол	PATCH
Описание	Метод переводит учебный план в архивный статус или возвращает из архива
Ожидаемый результат	Возвращена информация об архивном учебном плане ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана

### Описания сценария

#### Базовый сценарий. Архивировать учебный план

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN.	АС.1	AccessDeniedException
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = planId и curriculum.field_id = fieldId		
	и запоминает результат.		
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система изменяет запись в таблице curriculum.		
	Поля изменяются по следующему правилу:		
	archive_status = NOT archive_status (инвертируется текущее значение)		
	last_modified = текущая дата и время		
9	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = информация об учебном плане с обновленным архивным статусом		
	metadata.responseTime = текущая дата и время		

№	Действие	АС	Класс бизнес-исключения
10	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
11	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Доступ запрещен"  Content = "Требуется роль ADMIN для архивации учебных планов"  и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

## Удалить учебный план - deleteStudyPlan

Название сценария	Удалить учебный план
Контроллер	AdminStudyPlanController
Название метода	deleteStudyPlan
Эндполит	{host}/api/admin/fields/{fieldId}/study-plans/{planId}
Глагол	DELETE
Описание	Метод удаляет учебный план
Ожидаемый результат	Возвращено подтверждение удаления ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; план не должен быть в архиве и не должен содержать контент

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана

### Описания сценария

#### Базовый сценарий. Удалить учебный план

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.1	AccessDeniedException

№	Действие	АС	Класс бизнес-исключения
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии <code>field_of_study.field_id = fieldId</code>		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система осуществляет поиск учебного плана в базе данных при условии <code>curriculum.curriculum_id = planId</code> и <code>curriculum.field_id = fieldId</code>		
	и запоминает результат.		
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система проверяет, что учебный план не находится в архиве ( <code>archive_status = false</code> ).	АС.2	ArchiveStudyPlanException
9	Система проверяет, что учебный план не содержит дисциплины.		
10	Система удаляет запись из таблицы <code>curriculum</code> .		
11	Система заполняет результат работы метода по следующему правилу:		
	<code>status = "success"</code>		
	<code>data = "Учебный план с ID {planId} удален из направления с ID {fieldId}"</code>		
	<code>metadata.responseTime =</code> текущая дата и время		
12	Результат работы метода возвращается запрашивающей системе с <code>http-кодом 200</code> .		
13	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение <code>AccessDeniedException</code> , заполняет ответ метода по следующему правилу:
	<code>status = "error"</code>
	<code>error = "Доступ запрещен"</code>
	<code>Content = "Недостаточно прав для выполнения операции"</code>
	и возвращает ответ с <code>http-кодом 403</code> .
2	Сценарий завершен.

### АС.2 Учебный план находится в архиве

№	Действие системы
1	Система возбуждает исключение <code>ArchiveStudyPlanException</code> , заполняет ответ метода по следующему правилу:
	<code>status = "error"</code>
	<code>error = "Невозможно удалить учебный план"</code>
	<code>Content = "Невозможно удалить учебный план, находящийся в архиве"</code>
	и возвращает ответ с <code>http-кодом 409</code> .

№	Действие системы
2	Сценарий завершен.

## Изменить учебный план - updateStudyPlan

Название сценария	Изменить учебный план
Контроллер	AdminStudyPlanController
Название метода	updateStudyPlan
Эндпоинт	{host}/api/admin/fields/{fieldId}/study-plans/{planId}
Глагол	PUT
Описание	Метод изменяет параметры существующего учебного плана
Ожидаемый результат	Возвращена информация об измененном учебном плане ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; план не должен быть в архиве

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана
studyPlanName	string	Нет	Body	Название учебного плана
yearStart	integer	Нет	Body	Год начала действия
yearEnd	integer	Нет	Body	Год окончания действия
filePath	string	Нет	Body	Путь к файлу
course	integer	Нет	Body	Курс
status	string	Нет	Body	Статус/версия

### Описания сценария

#### Базовый сценарий. Изменить учебный план

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.1	AccessDeniedException
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = planId и curriculum.field_id = fieldId		
	и запоминает результат.		

№	Действие	АС	Класс бизнес-исключения
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система проверяет, что учебный план не находится в архиве (archive_status = false).	AC.2	ArchiveStudyPlanException
9	Система проверяет, что переданные для изменения значения не совпадают с сохраненными в записи с шага 6.		
10	Система изменяет запись в таблице curriculum.		
	Поля заполняются по следующему правилу:		
	Для каждого переданного параметра: если значение не null, то обновляется поле		
	last_modified = текущая дата и время		
11	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = информация об измененном учебном плане		
	metadata.responseTime = текущая дата и время		
12	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
13	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Доступ запрещен"
	Content = "Недостаточно прав для выполнения операции"
	и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

### АС.2 Учебный план находится в архиве

№	Действие системы
1	Система возбуждает исключение ArchiveStudyPlanException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Невозможно изменить учебный план"
	Content = "Невозможно изменить учебный план, находящийся в архиве"
	и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

## Удалить контент из учебного плана - removeContentFromPlan

Название сценария	Удалить контент из учебного плана
Контроллер	AdminStudyPlanController
Название метода	removeContentFromPlan
Эндполит	{host}/api/admin/fields/{fieldId}/study-plans/{planId}/content
Глагол	DELETE
Описание	Метод удаляет дисциплины из учебного плана
Ожидаемый результат	Возвращено подтверждение удаления контента ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; план не должен быть в архиве

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана
blocks	List	Да	Body	Массив блоков для удаления

## Структура DisciplineTermDeleteDto

Поле	Тип данных	Обязательное	Описание
disciplineId	Long	Да	Идентификатор дисциплины
term	Integer	Да	Семестр (1 или 2)

## Выходные параметры

В случае успешного выполнения возвращается подтверждение удаления контента.

Название параметра	Тип данных	Обязательное	Размещение	Описание
status	string	Да	Body	Статус выполнения ("success")
data	string	Да	Body	Сообщение об успешном удалении
metadata	ResponseMetadata	Да	Body	Метаданные ответа

### Описания сценария

#### Базовый сценарий. Удалить контент из учебного плана

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.1	AccessDeniedException
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		



№	Действие	АС	Класс бизнес-исключения
6	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = planId и curriculum.field_id = fieldId		
	и запоминает результат.		
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система проверяет, что учебный план не находится в архиве (archive_status = false).	AC.2	ArchiveStudyPlanException
9	Система извлекает данные о блоках для удаления из запроса метода.		
10	Для каждого блока система выполняет:	AC.3	ResourceNotFoundException
	- проверяет существование дисциплины в указанном семестре учебного плана		
11	Система удаляет записи из таблиц subsidiary_discip_teacher и subsidiary_discip_curriculum.		
12	Система обновляет дату последнего изменения учебного плана.		
13	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = "Контент успешно удален из учебного плана"		
	metadata.responseTime = текущая дата и время		
14	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
15	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Доступ запрещен"
	Content = "Недостаточно прав для выполнения операции"
	и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

### АС.2 Учебный план находится в архиве

№	Действие системы
1	Система возбуждает исключение ArchiveStudyPlanException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Невозможно удалить контент"
	Content = "Невозможно удалить контент из учебного плана, находящегося в архиве"
	и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

АС.3 Дисциплина не найдена в учебном плане

№	Действие системы
1	Система возбуждает исключение ResourceNotFoundException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Ресурс не найден"  Content = "Дисциплина с ID {disciplinelid} не найдена в учебном плане в семестре {term}."  и возвращает ответ с http-кодом 404.
2	Сценарий завершен.

Сгенерировать PDF учебного плана - getStudyPlanPdf

Название сценария	Сгенерировать PDF учебного плана
Контроллер	StudyPlansController
Название метода	getStudyPlanPdf
Эндполит	{host}/api/curricula/{studyPlanId}/pdf
Глагол	GET
Описание	Метод генерирует и возвращает PDF-документ с информацией об учебном плане
Ожидаемый результат	Возвращен PDF-документ ИЛИ Возвращено сообщение об ошибке
Ограничения	Отсутствуют

Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
studyPlanId	string	Да	Path	Идентификатор учебного плана

Выходные параметры

В случае успешного выполнения возвращается PDF-документ.

Название параметра	Тип данных	Обязательное	Размещение	Описание
Content-Type	string	Да	Header	application/pdf
Content-Disposition	string	Да	Header	attachment; filename="study-plan-{studyPlanId}.pdf"
body	byte[]	Да	Body	Байты PDF-документа

Описания сценария

Базовый сценарий. Сгенерировать PDF учебного плана

№	Действие	АС	Класс бизнес-исключения
1	Система извлекает идентификатор учебного плана из запроса метода.		
2	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = studyPlanId		
	и запоминает результат.		

№	Действие	АС	Класс бизнес-исключения
3	Система проверяет, что результат поиска с шага 2 не пуст.		
4	Система осуществляет поиск информации о направлении подготовки по ID учебного плана.	АС.1	DatabaseException
	и запоминает результат.		
5	Система осуществляет поиск дисциплин, связанных с учебным планом, через таблицу subsidiary_discip_curriculum.		
	и запоминает результат.		
6	Система осуществляет поиск преподавателей для каждой дисциплины через таблицу subsidiary_discip_teacher.		
	и запоминает результат.		
7	Система использует PdfGenerateService для создания PDF документа.		
	Документ формируется по следующему шаблону:		
	- Заголовок: "Учебный план: {studyPlanName}"		
	- Информация о направлении подготовки		
	- Список дисциплин по семестрам с преподавателями		
	- Часы и формы контроля		
8	Система формирует заголовки ответа:		
	Content-Type: application/pdf		
	Content-Disposition: attachment; filename="study-plan-{studyPlanId}.pdf"		
9	Система возвращает PDF документ в теле ответа с http-кодом 200.		
10	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Учебный план по переданному ID не найден

№	Действие системы
1	Система возбуждает исключение StudyPlanNotFoundException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Учебный план не найден"
	Content = "Учебный план с ID {studyPlanId} не найден"
	и возвращает ответ с http-кодом 404.
2	Сценарий завершен.

## Добавить контент в учебный план - addContentToPlan

Название сценария	Добавить контент в учебный план
Контроллер	AdminStudyPlanController
Название метода	addContentToPlan
Эндпоинт	{host}/api/admin/fields/{fieldId}/study-plans/{planId}/content
Глагол	POST

Название сценария	Добавить контент в учебный план
Описание	Метод добавляет дисциплины с преподавателями в учебный план
Ожидаемый результат	Возвращено подтверждение добавления контента ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; план не должен быть в архиве

## Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана
blocks	List	Да	Body	Массив блоков дисциплин с преподавателями

## Описания сценария

### Базовый сценарий. Добавить контент в учебный план

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	AC.1	AccessDeniedException
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId		
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.		
6	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = planId и curriculum.field_id = fieldId		
	и запоминает результат.		
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система проверяет, что учебный план не находится в архиве (archive_status = false).	AC.3	ArchiveStudyPlanException
9	Система извлекает данные о блоках дисциплин из запроса метода.		
10	Для каждого блока дисциплины система выполняет:	AC.2	DuplicateDisciplineException
	- проверяет существование дисциплины		
	- проверяет существование преподавателей		
	- проверяет уникальность комбинации дисциплина+семестр в плане		
11	Система создает записи в таблицах subsidiary_discip_curriculum и subsidiary_discip_teacher.		
12	Система обновляет дату последнего изменения учебного плана.		
13	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		

№	Действие	АС	Класс бизнес-исключения
	data = "Контент успешно добавлен в учебный план"		
	metadata.responseTime = текущая дата и время		
14	Результат работы метода возвращается запрашивающей системе с http-кодом 201.		
15	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Доступ запрещен"  Content = "Недостаточно прав для выполнения операции"  и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

### АС.2 Дисциплина уже существует в учебном плане

№	Действие системы
1	Система возбуждает исключение DuplicateDisciplineException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Конфликт при добавлении дисциплины"  Content = "Дисциплина с ID {disciplineId} уже существует в учебном плане в семестре {term}"  и возвращает ответ с http-кодом 400.
2	Сценарий завершен.

### АС.3 Учебный план находится в архиве

№	Действие системы
1	Система возбуждает исключение ArchiveStudyPlanException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Невозможно добавить контент"  Content = "Невозможно добавить контент в учебный план, находящийся в архиве"  и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

## Изменить контент учебного плана - updateContentInPlan

Название сценария	Изменить контент учебного плана
Контроллер	AdminStudyPlanController
Название метода	updateContentInPlan

Название сценария	Изменить контент учебного плана
Эндполит	{host}/api/admin/fields/{fieldId}/study-plans/{planId}/content
Глагол	PUT
Описание	Метод частично обновляет контент учебного плана (смена семестра, часов, преподавателей)
Ожидаемый результат	Возвращено подтверждение обновления контента ИЛИ Возвращено сообщение об ошибке
Ограничения	Требуется роль ADMIN или MANAGER; план не должен быть в архиве

Входные параметры

Название параметра	Тип данных	Обязательное	Размещение	Описание
fieldId	string	Да	Path	Идентификатор направления подготовки
planId	string	Да	Path	Идентификатор учебного плана
updates	List	Да	Body	Массив обновлений дисциплин

Структура DisciplineUpdateDto

Поле	Тип данных	Обязательное	Описание
disciplineId	string	Да	Идентификатор дисциплины
currentTerm	Integer	Да	Текущий семестр (1 или 2)
newTerm	Integer	Да	Новый семестр (1 или 2)
newTermHours	Integer	Нет	Новое количество часов (0-1000)
newTeachers	List	Нет	Новый список преподавателей

Выходные параметры

В случае успешного выполнения возвращается подтверждение обновления контента.

Название параметра	Тип данных	Обязательное	Размещение	Описание
status	string	Да	Body	Статус выполнения ("success")
data	string	Да	Body	Сообщение об успешном обновлении
metadata	ResponseMetadata	Да	Body	Метаданные ответа

Описания сценария

Базовый сценарий. Изменить контент учебного плана

№	Действие	АС	Класс бизнес-исключения
1	Система проверяет наличие и валидность JWT токена в заголовке Authorization.		
2	Система проверяет, что пользователь имеет роль ADMIN или MANAGER.	АС.1	AccessDeniedException
3	Система извлекает идентификаторы направления и учебного плана из запроса метода.		
4	Система осуществляет поиск направления подготовки в базе данных при условии field_of_study.field_id = fieldId	АС.2	FieldNotFoundException
	и запоминает результат.		
5	Система проверяет, что результат поиска с шага 4 не пуст.	АС.2	FieldNotFoundException

№	Действие	АС	Класс бизнес-исключения
6	Система осуществляет поиск учебного плана в базе данных при условии curriculum.curriculum_id = planId и curriculum.field_id = fieldId		
	и запоминает результат.		
7	Система проверяет, что результат поиска с шага 6 не пуст.		
8	Система проверяет, что учебный план не находится в архиве (archive_status = false).	АС.3	ArchiveStudyPlanException
9	Система извлекает данные об обновлениях из запроса метода.		
10	Для каждого обновления система выполняет:		
	- проверяет существование дисциплины в текущем семестре		
	- проверяет, что новый семестр отличается от текущего или переданы новые параметры		
	- проверяет существование новых преподавателей (если переданы)		
11	Если переданы новые преподаватели, система проверяет уникальность комбинации преподаватель+роль.		
12	Система обновляет записи в таблицах subsidiary_discip_curriculum и subsidiary_discip_teacher.		
	При обновлении учитываются следующие правила:		
	- Если newTerm отличается от currentTerm, дисциплина перемещается в новый семестр		
	- Если передан newTermHours, обновляется количество часов		
	- Если переданы newTeachers, полностью заменяется список преподавателей		
13	Система обновляет дату последнего изменения учебного плана.		
14	Система заполняет результат работы метода по следующему правилу:		
	status = "success"		
	data = "Контент учебного плана успешно обновлен"		
	metadata.responseTime = текущая дата и время		
15	Результат работы метода возвращается запрашивающей системе с http-кодом 200.		
16	Сценарий завершен.		

## Альтернативные сценарии

### АС.1 Отсутствие или недостаточность прав доступа

№	Действие системы
1	Система возбуждает исключение AccessDeniedException, заполняет ответ метода по следующему правилу:
	status = "error"
	error = "Доступ запрещен"
	Content = "Недостаточно прав для выполнения операции"
	и возвращает ответ с http-кодом 403.
2	Сценарий завершен.

АС.2 Направление подготовки не найдено

№	Действие системы
1	Система возбуждает исключение FieldNotFoundException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Направление подготовки не найдено"  Content = "Направление с ID {fieldId} не найдено"  и возвращает ответ с http-кодом 404.
2	Сценарий завершен.

АС.3 Учебный план находится в архиве

№	Действие системы
1	Система возбуждает исключение ArchiveStudyPlanException, заполняет ответ метода по следующему правилу:  status = "error"  error = "Невозможно изменить контент"  Content = "Невозможно изменить контент учебного плана, находящегося в архиве"  и возвращает ответ с http-кодом 409.
2	Сценарий завершен.

Дополнительный объект сервиса

DisciplineUpdateDto

Объект содержит информацию об обновлении дисциплины в учебном плане.

Поле	Тип данных	Обязательное	Описание
disciplineId	Long	Да	Идентификатор дисциплины
currentTerm	Integer	Да	Текущий семестр (1 или 2)
newTerm	Integer	Да	Новый семестр (1 или 2)
newTermHours	Integer	Нет	Новое количество часов (0-1000)
newTeachers	List	Нет	Новый список преподавателей

Объекты сервиса

ResponseMetadata

Объект содержит метаданные ответа для всех успешных операций.

Поле	Тип данных	Обязательное	Описание
objectsCount	Integer	Нет	Количество объектов в ответе
responseTime	LocalDateTime	Да	Время формирования ответа в формате yyyy-MM-dd'T'HH:mm:ss'Z'

DisciplineTermBlockDto

Объект содержит информацию о блоке дисциплины с преподавателями для добавления в учебный план.



Поле	Тип данных	Обязательное	Описание
disciplineId	Long	Да	Идентификатор дисциплины
term	Integer	Да	Семестр (1 или 2)
termHours	Integer	Нет	Часы в семестре (0-1000)
teachers	List	Нет	Список преподавателей

## TeacherAssignmentDto

Объект содержит информацию о назначении преподавателя на дисциплину.

Поле	Тип данных	Обязательное	Описание
teacherId	Long	Да	Идентификатор преподавателя
teacherRole	String	Нет	Роль преподавателя (макс. 50 символов)
hoursAssigned	Integer	Нет	Назначенные часы (0-200)

## StudyPlansDisciplinesResponseDto

Объект содержит полную информацию об учебном плане с дисциплинами.

Поле	Тип данных	Обязательное	Описание
studyPlanId	Long	Да	Идентификатор учебного плана
studyPlanName	String	Да	Название учебного плана
course	Integer	Да	Курс
yearStart	Integer	Да	Год начала действия
yearEnd	Integer	Да	Год окончания действия
archiveStatus	Boolean	Да	Статус архивации
filePath	String	Нет	Путь к файлу
status	String	Да	Статус/версия
fieldOfStudy	FieldOfStudyBriefDto	Да	Информация о направлении подготовки
disciplines	List	Да	Список дисциплин с преподавателями

## FieldOfStudyBriefDto

Объект содержит краткую информацию о направлении подготовки.

Поле	Тип данных	Обязательное	Описание
fieldId	Long	Да	Идентификатор направления
fieldCode	String	Да	Код направления
fieldName	String	Да	Название направления
degreeLevel	String	Да	Уровень образования
studyLength	Integer	Да	Срок обучения
profileName	String	Да	Название профиля
qualification	String	Нет	Квалификация

## DisciplineDto

Объект содержит информацию о дисциплине с преподавателями.

Поле	Тип данных	Обязательное	Описание
disciplineId	Long	Да	Идентификатор дисциплины
disciplineCode	String	Нет	Код дисциплины
disciplineName	String	Да	Название дисциплины
term	Integer	Да	Семестр (1 или 2)
termHours	Integer	Нет	Часы в семестре
totalHours	Integer	Да	Общее количество часов
independentHours	Integer	Нет	Часы самостоятельной работы
report	String	Да	Форма контроля
description	String	Нет	Описание дисциплины
competences	String	Нет	Компетенции
teachers	List	Да	Список преподавателей

## TeacherInDisciplineDto

Объект содержит информацию о преподавателе в дисциплине.

Поле	Тип данных	Обязательное	Описание
teacherId	Long	Да	Идентификатор преподавателя
fio	String	Да	ФИО преподавателя
role	String	Да	Роль преподавателя
hoursAssigned	Integer	Да	Назначенные часы
department	String	Да	Кафедра
post	String	Нет	Должность
academicStatus	String	Нет	Ученое звание
academicDegree	String	Нет	Ученая степень
email	String	Нет	Электронная почта
phone	String	Нет	Телефон
information	String	Нет	Дополнительная информация

## Исключения и обработка ошибок

### ErrorCode

Перечисление содержит информацию о кодах ошибок.

Значение	Описание
DatabaseException	Ошибка при обращении к базе данных
DuplicateResourceException	Сущность уже существует
FieldNotFoundException	Направление подготовки не найдено
StudyPlanNotFoundException	Учебный план не найдено
ResourceNotFoundException	Ресурс не найден (дисциплина, преподаватель)

Значение	Описание
ArchiveStudyPlanException	Ошибка операции с архивным учебным планом
ValidationException	Ошибка валидации (не заполнены обязательные поля)
AccessDeniedException	Ошибка доступа (недостаточно прав)
RuntimeException	Общая ошибка выполнения
DuplicateDisciplineException	Дубликация данных

## BusinessFault

Объект описывает формат ошибок, возвращаемых в сценариях работы методов.

Параметр	Тип	Описание
status	String	Статус выполнения (всегда "error")
error	String	Общее описание ошибки
Content	String	Детализированное описание ошибки
metadata	ResponseMetadata	Метаданные ответа (время запроса)

## DatabaseException

Возвращается в случае получения ошибок при обращении к базе данных.

Поле	Тип	Обязательное	Описание
message	String	Да	Сообщение об ошибке

## ValidationException

Возвращается в случае получения ошибок при валидации входных данных.

Поле	Тип	Обязательное	Описание
message	String	Да	Сообщение об ошибке
details	Map	Да	Карта деталей ошибки

## EntityNotFoundException

Базовый класс для исключений "сущность не найдена".

Исключение	HTTP-код	Описание
FieldNotFoundException	404	Направление подготовки не найдено
StudyPlanNotFoundException	404	Учебный план не найден
ResourceNotFoundException	404	Ресурс не найден (дисциплина, преподаватель)

## BusinessLogicException

Исключения бизнес-логики.

Исключение	HTTP-код	Описание
DuplicateResourceException	409	Нарушение уникальности ресурса
ArchiveStudyPlanException	409	Операция с архивным учебным планом

# AccessDeniedException

Возвращается при отсутствии необходимых прав доступа.

Поле	Тип	Обязательное	Описание
message	String	Да	Сообщение об ошибке

## Конфигурация безопасности

### Роли пользователей

Роль	Доступные эндполиты	Описание
ADMIN	Все эндполиты /api/admin/	Полный доступ к администрированию, включая архивацию
MANAGER	Большинство эндполитов /api/admin/	Доступ к созданию/редактированию, кроме архивации
Публичный	/api/auth/, /api/curricula/, /api/fields/	Доступ без аутентификации

### Требования к заголовкам

Для доступа к защищенным эндпоинтам требуется заголовок Authorization:  
Authorization: Bearer {JWT\_TOKEN}

Пример:  
Key: Authorization:  
Value: Bearer  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwR

### Срок действия токена

Параметр	Значение
Время жизни токена	24 часа
Алгоритм подписи	HS256
Тип токена	JWT (JSON Web Token)

## Примечания и ограничения

### Ограничения валидации

#### 1. Направления подготовки:

- Название профиля ( profileName ) должно быть уникальным
- Срок обучения ( studyLength ) от 1 до 10 лет

#### 2. Учебные планы:

- Год начала ( yearStart ) от 2015 до 2025
- Год окончания ( yearEnd ) от 2025 до 2035
- Курс ( course ) от 1 до 10
- Нельзя создать дубликат плана с одинаковыми параметрами для одного направления

#### 3. Дисциплины:

- Общее количество часов ( hoursTotal ) от 0 до 400
- Часы самостоятельной работы ( hoursIndependent ) от 0 до 20
- Нельзя добавить одну дисциплину в один семестр дважды

#### 4. Преподаватели:

- Назначенные часы ( `hoursAssigned` ) от 0 до 200
- Нельзя назначить одного преподавателя в одну роль на одну дисциплину дважды

## Бизнес-правила

### 1. Архивация учебных планов:

- Только пользователи с ролью ADMIN могут архивировать/разархивировать планы
- Нельзя изменять или удалять архивные учебные планы
- Архивные планы остаются доступными для чтения

### 2. Удаление ресурсов:

- Нельзя удалить направление, к которому привязаны учебные планы
- Нельзя удалить учебный план, содержащий дисциплины
- Удаление дисциплины из плана каскадно удаляет всех назначенных преподавателей

### 3. Уникальность:

- Название профиля направления должно быть уникальным
- Комбинация параметров учебного плана уникальна в рамках направления
- Комбинация преподаватель+роль уникальна в рамках дисциплины

## Форматы данных

1. **Дата и время:** ISO 8601 ( `yyyy-MM-dd'T'HH:mm:ss'Z'` )
2. **JWT токен:** Bearer token в заголовке Authorization
3. **Идентификаторы:** целые положительные числа
4. **Коды направлений:** формат "XX.XX.XX"
5. **Статус учебного плана:** строковый формат (например "1.00", "2.01")

## Рекомендации по использованию

1. Всегда проверяйте ответ на наличие поля `status` ("success" или "error")
2. Для массовых операций используйте соответствующие batch-методы
3. Перед удалением ресурсов проверяйте зависимости
4. Используйте фильтрацию и пагинацию при работе с большими коллекциями
5. Регулярно обновляйте JWT токены (срок действия 24 часа)

## Дополнительные сервисы и компоненты

### JwtUtil

Утилита для работы с JWT токенами.

Метод	Описание
<code>generateToken(username, role)</code>	Генерирует JWT токен
<code>validateToken(token)</code>	Проверяет валидность токена
<code>extractUsername(token)</code>	Извлекает имя пользователя из токена
<code>extractRole(token)</code>	Извлекает роль из токена

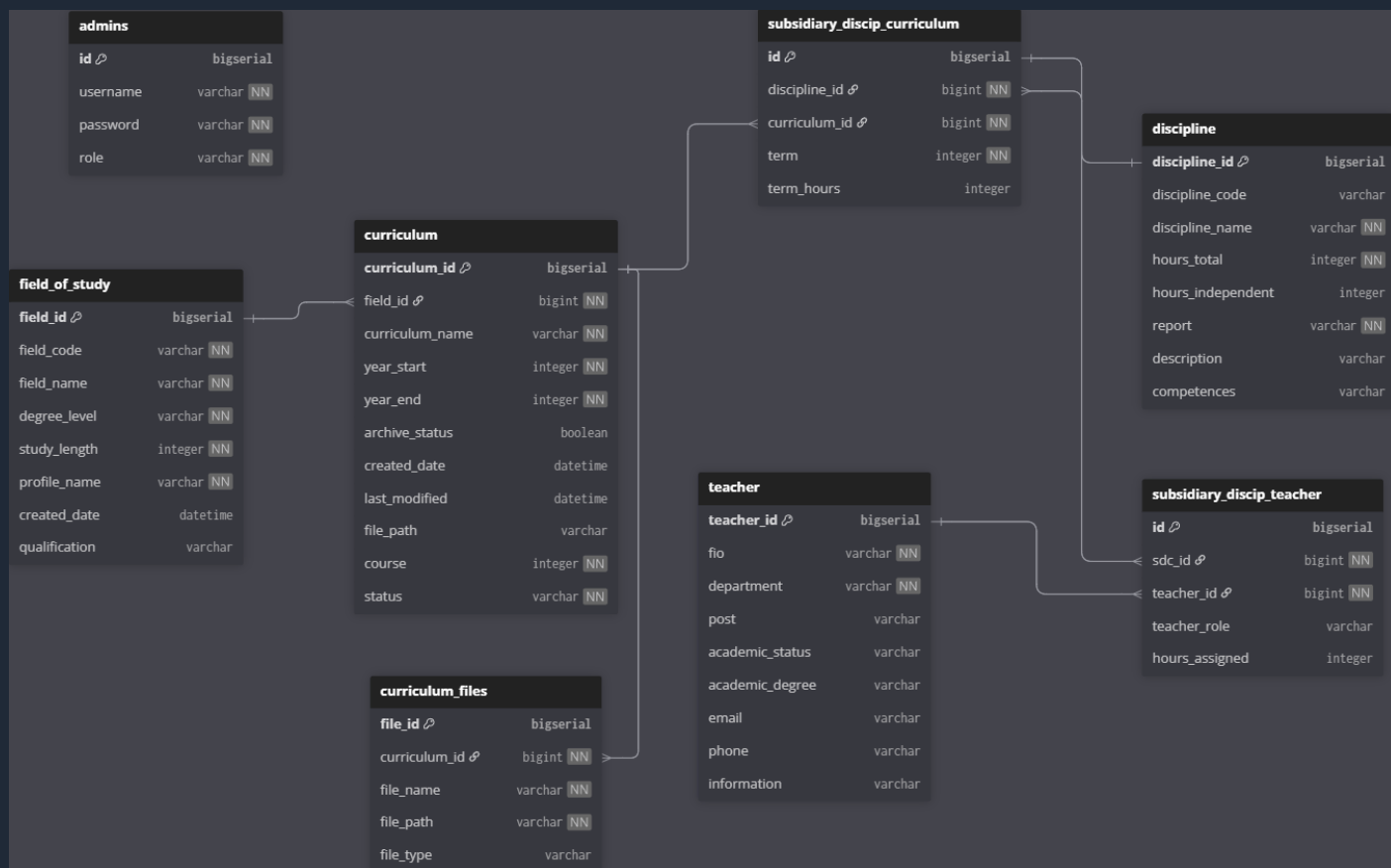
### GlobalExceptionHandler

Обработчик исключений для всего приложения.

Обрабатываемые исключения	HTTP код
<code>ValidationException</code>	400
<code>EntityNotFoundException</code>	404
<code>DuplicateResourceException</code>	409

Обрабатываемые исключения	HTTP код
ArchiveStudyPlanException	409
AccessDeniedException	403
Exception	500

## Физическая модель



## Инструкция по сборке и разворачиванию

### Установка и настройка системного ПО

1. **Java JDK 17+** - скачайте с <https://adoptium.net/>
2. **Apache Maven 3.8+** - скачайте с <https://maven.apache.org/download.cgi>
3. **PostgreSQL 15+** - скачайте с <https://www.postgresql.org/download/windows/>

### Конфигурация приложения

Файл: `src/main/resources/application.properties`

```

````properties
server.port=8080
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.username=sa
spring.jpa.hibernate.ddl-auto=update
jwt.secret=education-system-secret-key-2025
jwt.expiration=86400000

# Клонирование
git clone https://github.com/SAFE9HOUSE/Curriculum_Web.git
cd Curriculum_Web
  
```

Сборка

## Клонирование

```
git clone https://github.com/SAFE9HOUSE/Curriculum_Web.git
cd Curriculum_Web
```

## Сборка

```
./mvnw clean package
```

Остановка

## Остановка JAR процесса (Ctrl+C в терминале)

### Основные точки доступа приложения

Endpoint	Описание	Доступ
http://localhost:8080/api/auth/login	Аутентификация	Публичный
http://localhost:8080/api/curricula/{id}	Получить учебный план	Публичный
http://localhost:8080/api/fields	Получить направления	Публичный
http://localhost:8080/api/admin/fields	Администрирование направлений	ADMIN/MANAGER
http://localhost:8080/h2-console	Консоль БД (dev)	Публичный