

Самостоятельная групповая работа

Medical Startup Prototype — PatientCare

Дополнение к заданию: расширенный набор REST API

Это дополнение **обязательно** и является частью итогового прототипа.

REST API (РАСШИРЕННЫЙ НАБОР)

Минимальный API из предыдущей версии остается обязательным.

К нему добавляются **ещё 3 API**, которые делают систему более реалистичной и полезной.

Базовые API (обязательные)

1. **GET /api/patients**

Получить список всех пациентов

2. **GET /api/patients/{id}**

Получить пациента по id

3. **POST /api/patients**

Создать нового пациента

- используется `@RequestBody`

- используется `@Valid`

- при ошибке → `400 Bad Request`

4. **DELETE /api/patients/{id}**

Удалить пациента

- допускается soft-delete (`deleted = true`)

Дополнительные API (ОБЯЗАТЕЛЬНЫЕ)

Обновление данных пациента

PUT /api/patients/{id}

Назначение:

- обновить данные существующего пациента

Требования:

- использовать `@RequestBody`
- использовать `@Valid`
- если пациента не существует → `404 Not Found`
- если данные невалидны → `400 Bad Request`
- логировать:

- успешное обновление (`INFO`)
- невалидную попытку (`WARN`)

Пример бизнес-логики:

Пациент сменил страховую или была ошибка в фамилии.

Фильтрация пациентов (Query Parameters)

`GET /api/patients/search`

Поддерживаемые query-параметры (минимум два):

- `gender`
- `bloodType`
- `ageFrom`
- `ageTo`

Примеры запросов:

```
GET /api/patients/search?gender=FEMALE
GET /api/patients/search?bloodType=O_POS
GET /api/patients/search?ageFrom=18&ageTo=65
GET /api/patients/search?gender=MALE&ageFrom=30
```

Требования:

- параметры **опциональны**
- если параметры не переданы → вернуть всех пациентов
- логировать факт применения фильтра (INFO)

Получение статистики (агрегирующий API)

`GET /api/patients/statistics`

Назначение:

- вернуть обобщённую информацию, а не список пациентов

Пример возможного ответа:

```
{  
    "totalPatients": 42,  
    "maleCount": 20,  
    "femaleCount": 21,  
    "otherCount": 1  
}
```

Или (альтернатива):

- количество пациентов по группам крови
- количество пациентов старше 60 лет

Требования:

- отдельный DTO для ответа
 - логировать запрос статистики (INFO)
-

Тестирование (обновлённые требования)

Помимо предыдущих тестов:

Дополнительно протестировать:

- успешный PUT → 200 OK
- PUT с невалидными данными → 400
- PUT для несуществующего id → 404
- GET /search с параметрами
- GET /statistics возвращает корректные данные

⚠ Достаточно **1-2 интеграционных тестов**, покрывающих новые API.

Что дополнительно показать на демо

На финальной демонстрации обязательно показать:

- PUT /api/patients/{id}
 - GET /api/patients/search
 - GET /api/patients/statistics
-

Итоговый минимум API

В проекте должно быть **не менее 7 endpoint'ов**:

1. GET all
2. GET by id
3. POST

4. DELETE
5. PUT
6. SEARCH (query params)
7. STATISTICS