

TÀI LIỆU TÍCH HỢP API ĐẶT KHÁM THEO BÁC SĨ

Hướng dẫn tích hợp API HIS, API Web đặt khám và deep-link điều hướng người dùng.

Mã tài liệu DHS-6433	Phiên bản 1.0	Cập nhật 24/07/2026
-------------------------	------------------	------------------------

TỔNG QUAN ENDPOINT

HỆ THỐNG	ENDPOINT	MỤC ĐÍCH
HIS	GET /api/portal/his/doctors/work-schedules	Danh sách bác sĩ có lịch
HIS	GET /api/app/his/doctors/{doctorCode}/work-schedule-days	Ngày làm việc theo bác sĩ
WEB ĐẶT KHÁM	GET /api/v1/doctors/work-schedules?doctorCode={code}	Resolve bác sĩ cho deep-link

Luồng deep-link

Website bên thứ ba → Trang đặt khám → Resolve doctorCode → Chọn bác sĩ → Ngày khám → Thời gian → Hồ sơ → Xác nhận

1. API HIS: lấy bác sĩ có lịch làm việc

GET /api/portal/his/doctors/work-schedules?fromDate=2026-07-25&toDate=2026-07-31
Authorization: Bearer <access-token>

Response:

```
{
  "success": true,
  "statusCode": 200,
  "data": {
    "fromDate": "2026-07-25",
    "toDate": "2026-07-31",
    "doctors": [
      {
        "doctorId": "655a3bf5-6462-4e76-a4bc-76275193944f",
        "doctorCode": "BS001",
        "fullName": "Nguyễn Văn A",
        "birthYear": 1985,
        "gender": null,
        "title": "Bác sĩ",
        "departments": [
          {
            "departmentId": "47c81965-0a80-4164-8ec2-a158759b234b",
            "departmentCode": "PK001",
            "departmentName": "Phòng khám Ngoại"
          }
        ]
      }
    ]
  }
}
```

TypeScript:

```
interface DoctorDepartment {
  departmentId: string;
  departmentCode: string;
  departmentName: string;
}

interface HisDoctorWorkSchedule {
  doctorId: string;
  doctorCode: string;
  fullName: string;
  birthYear?: number | null;
  gender?: string | null;
  title?: string | null;
  departments: DoctorDepartment[];
}
```

2. API HIS: lấy ngày làm việc của bác sĩ

GET /api/app/his/doctors/BS001/work-schedule-days?fromDate=2026-07-25&toDate=2026-08-25
Authorization: Bearer <access-token>

Query:

Tên	Bắt buộc	Định dạng
doctorCode	Có	Path parameter
fromDate	Không	yyyy-MM-dd
toDate	Không	yyyy-MM-dd

Response:

```
{
  "success": true,
  "statusCode": 200,
  "data": {
    "doctorId": "655a3bf5-6462-4e76-a4bc-76275193944f",
    "doctorCode": "BS001",
    "fromDate": "2026-07-25",
    "toDate": "2026-08-25",
    "days": [
      {
        "date": "2026-07-28",
        "dayOfWeek": 3
      },
      {
        "date": "2026-07-30",
        "dayOfWeek": 5
      }
    ]
  }
}
```

TypeScript:

```
interface DoctorWorkScheduleDay {
  date: string;
  dayOfWeek: number;
}

interface DoctorWorkScheduleDayList {
  doctorId: string;
  doctorCode: string;
  fromDate: string;
  toDate: string;
  days: DoctorWorkScheduleDay[];
}
```

Ví dụ gọi API:

```
async function getDoctorWorkScheduleDays(
  doctorCode: string,
  fromDate: string,
  toDate: string,
  accessToken: string,
): Promise<DoctorWorkScheduleDayList> {
  const query = new URLSearchParams({ fromDate, toDate });
  const response = await fetch(
    `/api/app/his/doctors/${encodeURIComponent(doctorCode)}/work-schedule-days?${query.toString()}`,
    {
      headers: {
        Authorization: `Bearer ${accessToken}`,
      },
    },
  );
  const result =
    (await response.json()) as ApiResponse<DoctorWorkScheduleDayList>;

  if (!response.ok || !result.success || !result.data) {
    throw new Error(result.error?.code ?? "WORK_SCHEDULE_DAYS_FAILED");
  }

  return result.data;
}
```

Sử dụng **days** để hiển thị các ngày có thể chọn tại bước Ngày khám.

3. API Web đặt khám: tìm bác sĩ theo mã

```
GET /api/v1/doctors/work-schedules?doctorCode=BS001
```

Response thành công:

```
{
  "success": true,
  "statusCode": 200,
  "data": {
    "doctorId": "655a3bf5-6462-4e76-a4bc-76275193944f",
    "doctorCode": "BS001",
    "fullName": "Nguyễn Văn A",
    "birthYear": 1985,
    "title": "Bác sĩ"
  }
}
```

Không tìm thấy bác sĩ:

```
{
  "success": false,
  "statusCode": 404,
  "error": {
    "code": "DOCTOR_NOT_FOUND",
    "message": "Doctor was not found."
  }
}
```

TypeScript:

```
interface DoctorByCode {
  doctorId: string;
  doctorCode: string;
  fullName: string;
  birthYear?: number | null;
  title?: string | null;
}

interface ApiResponse<T> {
  success: boolean;
  statusCode: number;
  data?: T;
  error?: {
    code: string;
    message: string;
    details?: string[];
  };
}
```

Ví dụ gọi API:

```
async function getDoctorByCode(doctorCode: string): Promise<DoctorByCode> {
  const query = new URLSearchParams({
    doctorCode: doctorCode.trim(),
  });

  const response = await fetch(
    `/api/v1/doctors/work-schedules?${query.toString()}`,
  );
  const result = (await response.json()) as ApiResponse<DoctorByCode>;

  if (!response.ok || !result.success || !result.data) {
    throw new Error(result.error?.code ?? "DOCTOR_LOOKUP_FAILED");
  }

  return result.data;
}
```

4. Tích hợp deep-link đặt khám

/book-appointment?doctorCode=BS001

Frontend đọc `doctorCode`, gọi API tìm bác sĩ và cập nhật state:

```
const doctor = await getDoctorByCode(doctorCode);

setSelectedBookingMethod("doctor");
setSelectedDoctor({
  id: doctor.doctorId,
  name: doctor.fullName,
  appointmentCount: "",
});
setCurrentStepId(4);
```

Nếu API trả `DOCTOR_NOT_FOUND`:

```
setSelectedBookingMethod("doctor");
setSelectedDoctor(null);
setCurrentStepId(3);
showDoctorNotFoundWarning();
```

Frontend có thể đọc `mabs` làm alias cho `doctorCode`:

```
const doctorCode = searchParams.doctorCode ?? searchParams.mabs;
```

Request gửi xuống backend luôn dùng query `doctorCode`.

5. Flow đặt khám

Theo bác sĩ:

Hình thức → Bác sĩ → Ngày khám → Thời gian → Hồ sơ → Xác nhận

Theo chuyên khoa:

Hình thức → Chuyên khoa → Bác sĩ → Ngày khám → Thời gian → Hồ sơ → Xác nhận

ID	Bước
1	Hình thức
2	Chuyên khoa
3	Bác sĩ
4	Ngày khám
5	Thời gian
6	Hồ sơ
7	Xác nhận

6. Xử lý slot và đăng nhập

- Mở popup đăng nhập khi người dùng chưa đăng nhập.
- Giữ nguyên state deep-link sau khi đăng nhập.
- Khóa slot khi người dùng chọn thời gian.
- Chọn slot thành công chuyển tới bước Hồ sơ.
- Chọn hồ sơ chuyển tới bước Xác nhận.
- Quay lại từ Hồ sơ về Thời gian phải giải phóng slot.
- Slot hết hạn tại Hồ sơ hoặc Xác nhận phải đưa người dùng về bước Thời gian.