

```

HEADME.md
> api > routers > calendar_router.py > ...
    get_events_by_calendar_id
)
from api.services.courses import get_course_by_id
from api.services.students import get_students_active_courses

calendar_router = APIRouter(prefix="/calendars", tags=["Calendars"])

@calendar_router.get("/")
def get_all_calendars(current_user: User = Depends(get_current_user)):
    if current_user.role == constants.ADMIN_ROLE:
        return get_all_calendars_admin()
    elif current_user.role == constants.TEACHER_ROLE:
        return get_all_calendars_teacher(current_user.id)
    return get_all_calendars_students(current_user.id)

@calendar_router.post("/")
def create_calendars(
    calendar: CalendarCreate, current_user: User = Depends(get_current_user)
):
    if current_user.role == constants.STUDENT_ROLE:
        raise HTTPException(
            status_code=403, detail=constants.SECTION_ACCESS_DENIED_DETAIL
        )
    if get_course_by_id(calendar.course_id) is None:
        raise HTTPException(status_code=404, detail=constants.COURSE_NOT_FOUND_DETAIL)
    if (
        get_course_by_id(calendar.course_id).teacher.id != current_user.id
        and current_user.role == constants.TEACHER_ROLE
    ):
        raise HTTPException(
            status_code=403, detail=constants.SECTION_ACCESS_DENIED_DETAIL
        )
    create_calendar(calendar, current_user.id)
    return JSONResponse(
        status_code=201, content={"msg": constants.CALENDAR_CREATED}
    )

@calendar_router.get("/{calendar_id}")
def get_calendar_by_calendar_id(
    calendar_id: int, current_user: User = Depends(get_current_user)
):
    result = get_calendar_by_id(calendar_id)
    if result is None:
        raise HTTPException(status_code=404, detail=constants.CALENDAR_NOT_FOUND_DETAIL)
    if (
        current_user.role == constants.STUDENT_ROLE
        and result.course_id not in get_students_active_courses(current_user.id)
    ):
        raise HTTPException(
            status_code=403, detail=constants.SECTION_ACCESS_DENIED_DETAIL
        )
    if (
        current_user.role == constants.TEACHER_ROLE
        and result.owner_id != current_user.id
    ):
        raise HTTPException(
            status_code=403, detail={"msg": constants.SECTION_ACCESS_DENIED_DETAIL}
        )
    return result

@calendar_router.get("/{calendar_id}/events")
def get_events_for_calendar(calendar_id: int, current_user: User = Depends(get_current_user)):
    result = get_calendar_by_id(calendar_id)
    if result is None:
        raise HTTPException(status_code=404, detail=constants.CALENDAR_NOT_FOUND_DETAIL)
    if (
        current_user.role == constants.STUDENT_ROLE

```