```python
@event_router.get("/", response_model=List[Event])
async def retrieve_all_events() -> List[Event]:
    events = await event_database.get_all()
    return events


@event_router.get("/{id}", response_model=Event)
async def retrieve_event(id: PydanticObjectId) -> Event:
    event = await event_database.get(id)
    if not event:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Event with supplied ID does not exist"
        )
    return event
```

POST 라우트를 다음과 같이 변경한다.

```python
@event_router.post("/new")
async def create_event(body: Event) -> dict:
    await event_database.save(body)
    return {
        "message": "Event created successfully."
    }
```

UPDADE 라우트를 다음과 같이 추가한다.

```python
@event_router.put("/{id}", response_model=Event)
async def update_event(id: PydanticObjectId, body: EventUpdate)-> Event:
    updated_event = await event_database.update(id, body)
    if not updated_event:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Event with supplied ID does not exist"
        )
    return updated_event
```

마지막으로 DELETE 라우트를 다음과 같이 변경한다.