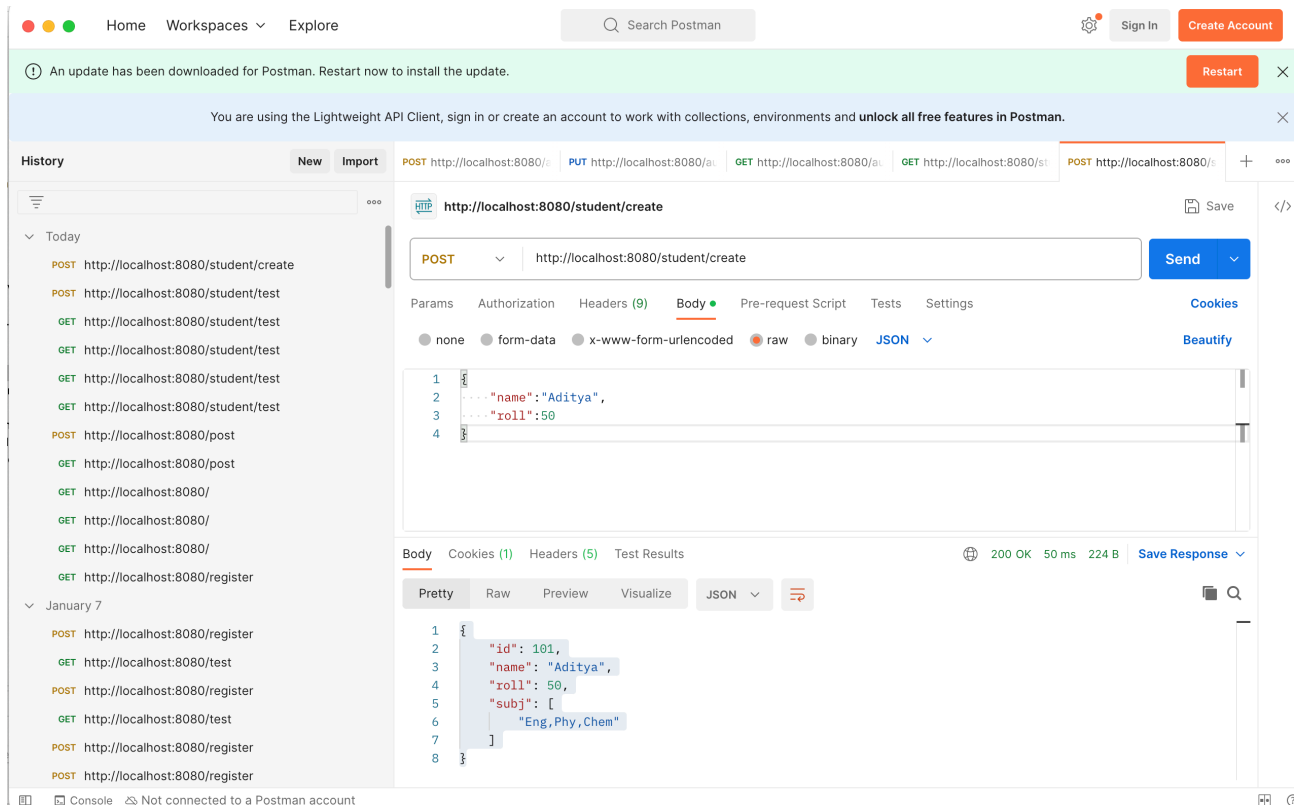


## Post Mapping Flow

1. The API is called from the Postman.
- 2.



2. The API calls is taken to the StudentController

3. Post Mapping:

@PostMapping("/create")

```
public StudentPostResponseModel createStudent(@RequestBody  
StudentPostRequestModel requestModel) {
```

```
    Student s = new Student(101, requestModel.getName(), requestModel.getRoll());  
    StudentPostResponseModel response = new  
StudentPostResponseModel(s.getId(), s.getName(), s.getRoll());
```

```
    return response;
```

```
}
```

Now the Requestbody from the postman is parsed into StudentPostRequestModel.  
(The purpose of StudentPostRequestModel to abstract or hide entity's variables which I don't want to expose.

Use cases:

1. Id is to be automatically
2. There are derived fields, for ex I just dob and age is automatically calculated).

The business logic then executes and rather than returning the entity itself, we return the StudentPostResponseModel.

In the above the Jackson is working twice, first it converts the JSON into StudentPostRequestModel object and then converts the StudentPostResponseModel into JSON.