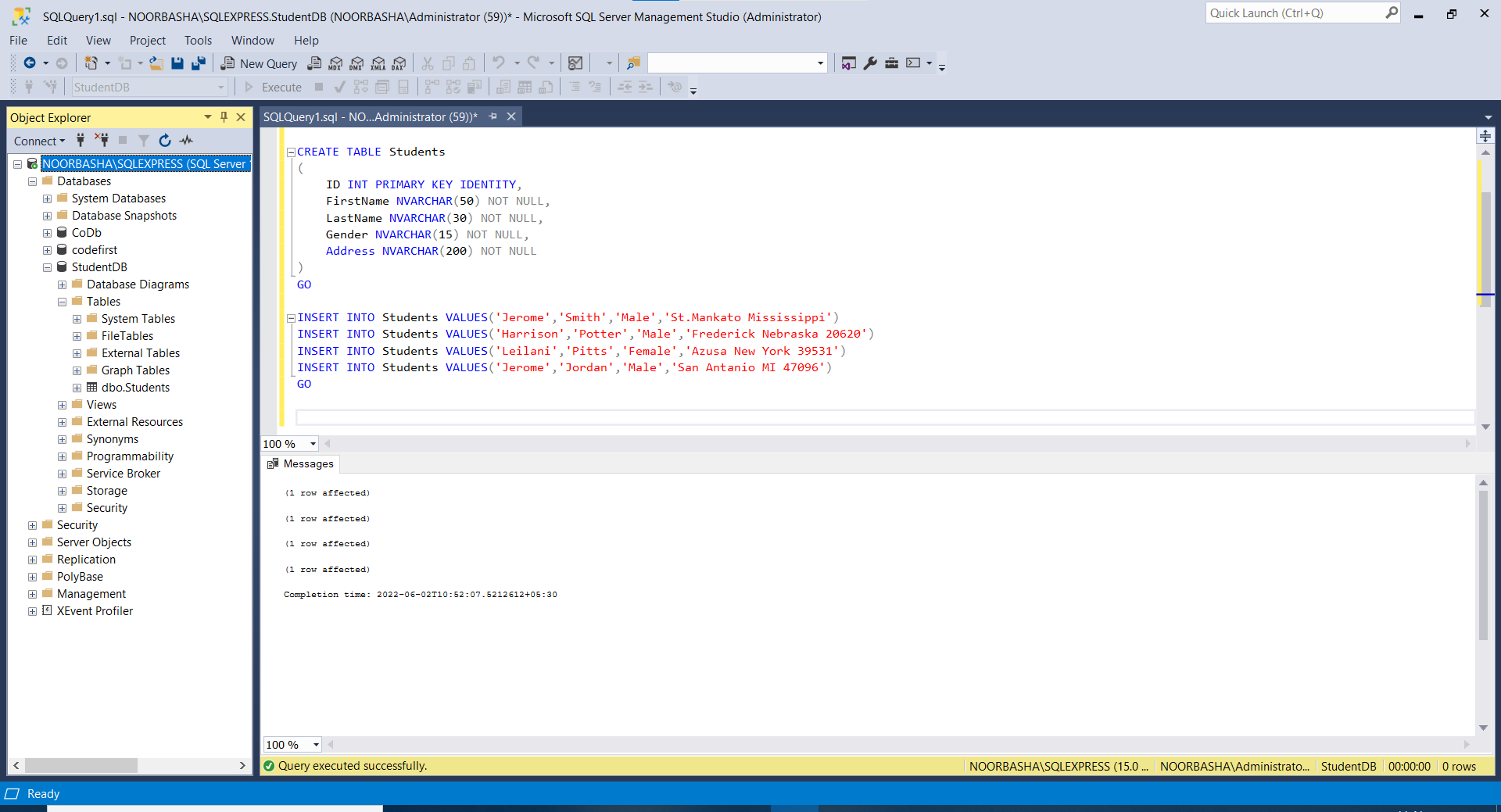
# DatabaseCreation:



# StudentsConroller:

using StudentService.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace StudentService.Controllers

{

public class StudentsController : ApiController

{

public HttpResponseMessage Put(int id, [FromBody] Student student)

{

using (StudentDBContext dbContext = new StudentDBContext())

{

var entity = dbContext.Students.FirstOrDefault(s => s.ID == id);

entity.FirstName=student.FirstName;

entity.LastName=student.LastName;

entity.Gender=student.Gender;

entity.Address=student.Address;

dbContext.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, entity);

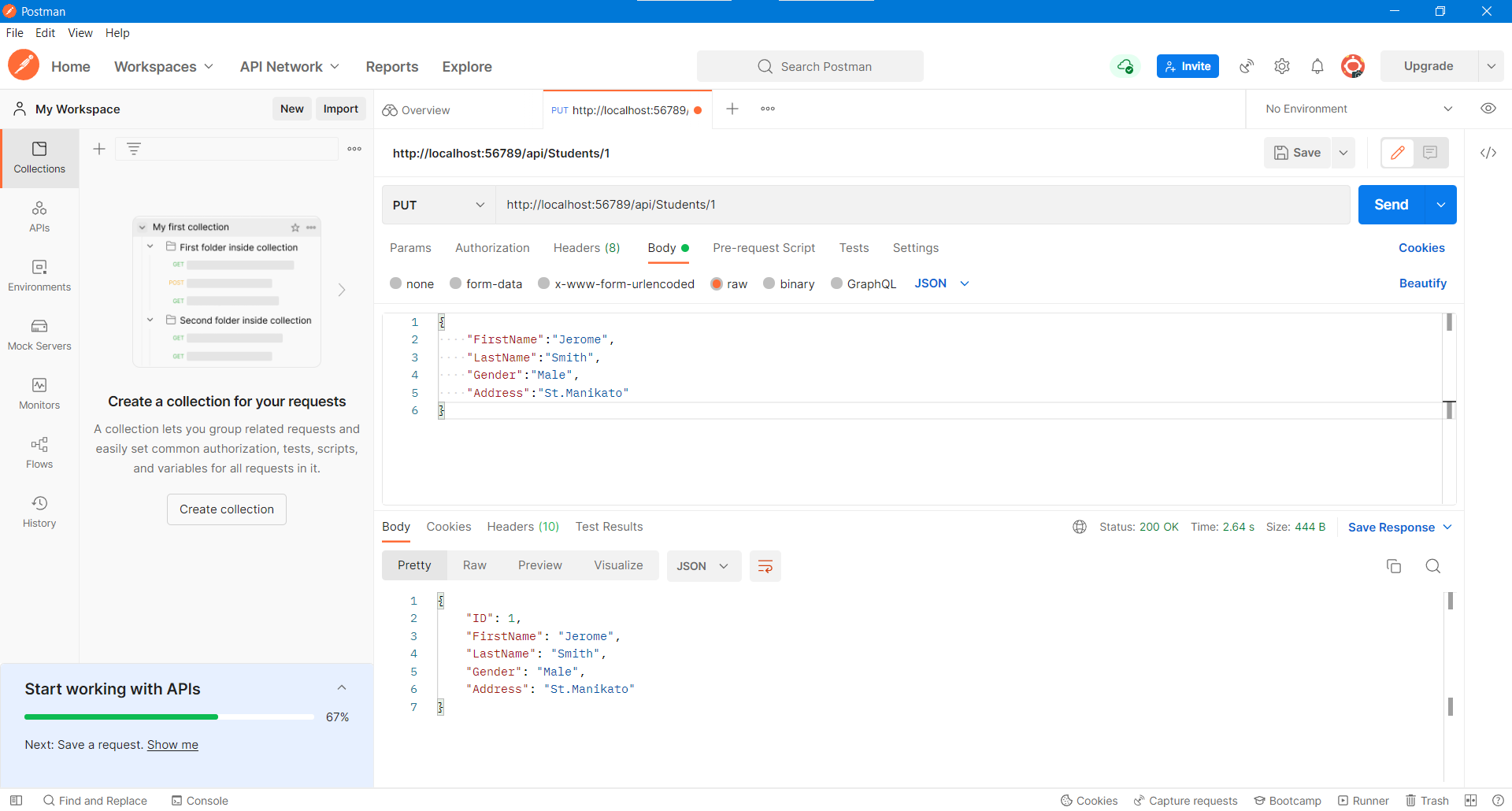
}

}

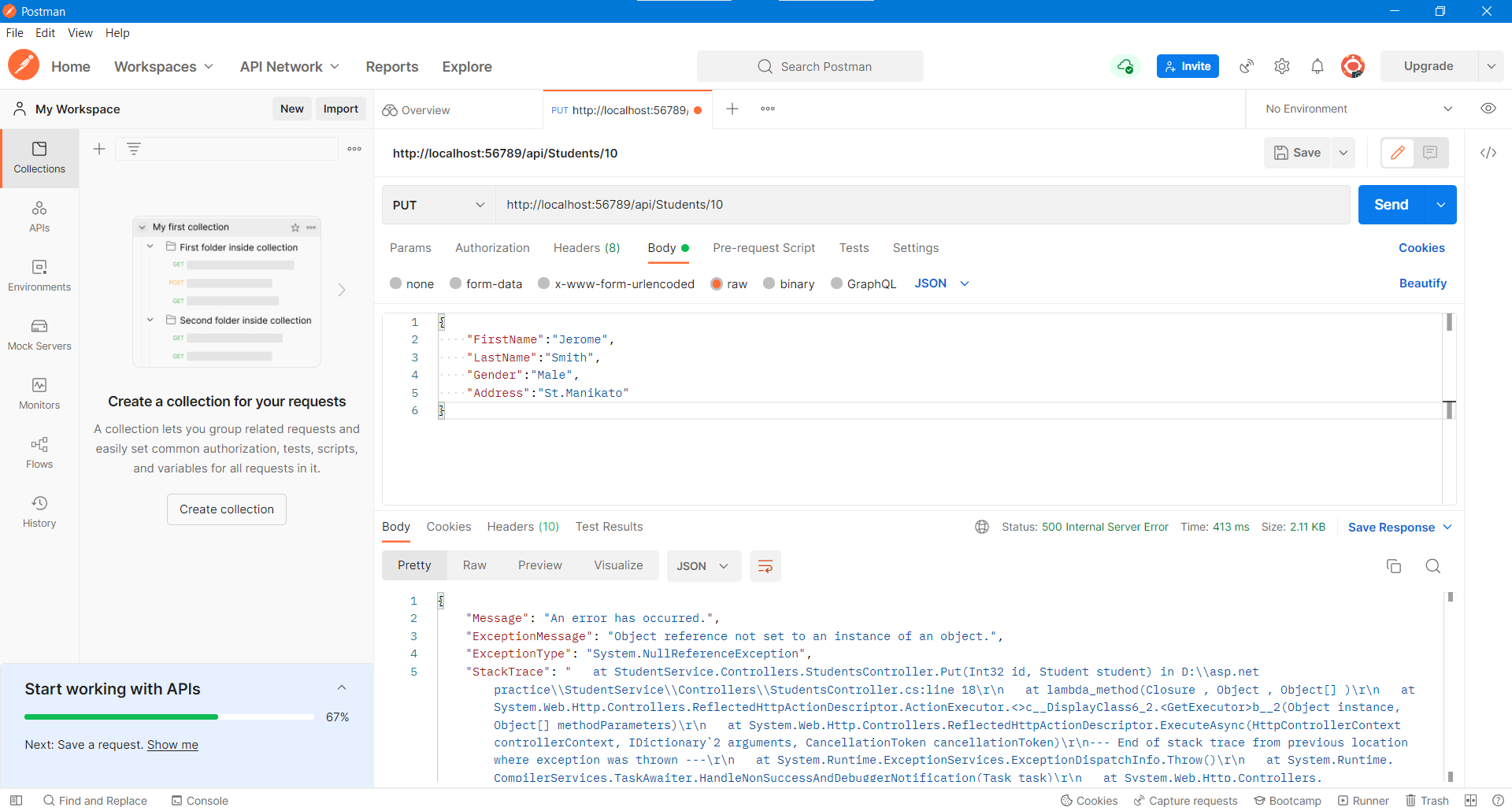
}

}

Url:/api/Students/1



[Url : /api/Students/10](Url:/api/Students/10)



# StudentsController:

using StudentService.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace StudentService.Controllers

{

public class StudentsController : ApiController

{

public HttpResponseMessage Put(int id, [FromBody] Student student)

{

try

{

using (StudentDBContext dbContext = new StudentDBContext())

{

var entity = dbContext.Students.FirstOrDefault(s => s.ID == id);

if (entity == null)

{

return Request.CreateErrorResponse(HttpStatusCode.NotFound, "Student with Id" + id.ToString() + "not found to update");

}

else

{

entity.FirstName = student.FirstName;

entity.LastName = student.LastName;

entity.Gender = student.Gender;

entity.Address = student.Address;

dbContext.SaveChanges();

return Request.CreateResponse(HttpStatusCode.OK, entity);

}

}

}

catch (Exception ex)

{

return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ex);

}

}

}

}

Url: /api/students/10

