|  |  |
| --- | --- |
|  | **Cognizant Academy**    **LearnToday**    **ASP.NET Web API Specification Document**    **Version 1.0** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** |  |  |  | | **Role** |  |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Contents

[**1.Introduction 3**](#_Toc62163183)

[**Purpose of this document 3**](#_Toc62163184)

[**Definitions & Acronyms 3**](#_Toc62163185)

[**Project Overview 3**](#_Toc62163186)

[**Scope 3**](#_Toc62163187)

[**Intended Audience 3**](#_Toc62163188)

[**Hardware and Software Requirement 3**](#_Toc62163189)

[**Use Case Diagram 3**](#_Toc62163190)

[**2.Create ASP.NET Web API Application in Visual Studio 4**](#_Toc62163191)

[**3.Table Specification and Solution Folder Structure 5**](#_Toc62163192)

[**4.Model Specification 7**](#_Toc62163193)

[**5.API Controller and Postman Test Specification 7**](#_Toc62163194)

[**5.1 Create a new AdminController which will contain 2 action methods: 7**](#_Toc62163195)

[**5.2 Create a new StudentController with the below 3 action methods: 9**](#_Toc62163196)

[**5.3 Create a new TrainerController with below 2 action methods: 11**](#_Toc62163197)

[**6.Code submission instructions 13**](#_Toc62163198)

[**7.Change Log 13**](#_Toc62163199)

# Introduction

## Purpose of this document

The purpose of this document is to specify a REST API for the LearnToday application.

## Definitions & Acronyms

|  |  |
| --- | --- |
| **Definition / Acronym** | **Description** |
| REST | Representational State Transfer |

## Project Overview

LearnToday is an online training application which allows students to view all the courses and allows them to enroll for the courses. It also features trainer registration and signin actions.

## Scope

* Develop LearnToday application using ASP.NET Web API.
* Test the actions using Postman.

## Intended Audience

* Application Architect
* Project Manager
* Test Manager
* Development Team
* Testing Team

## Hardware and Software Requirement

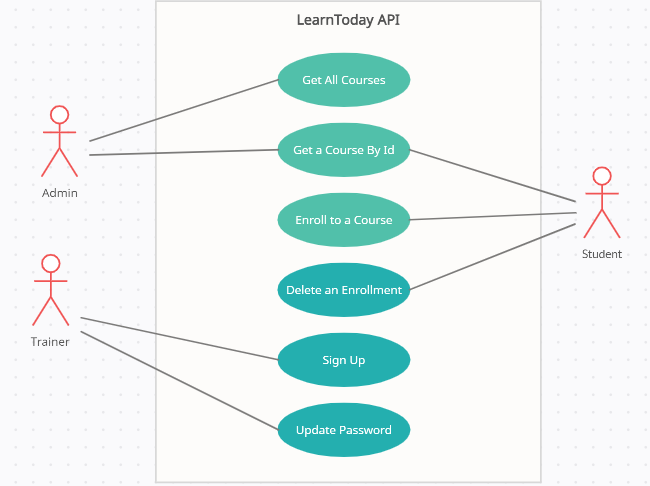
Hardware Requirement:

1. Developer Desktop PC with 8GB RAM

Software Requirement

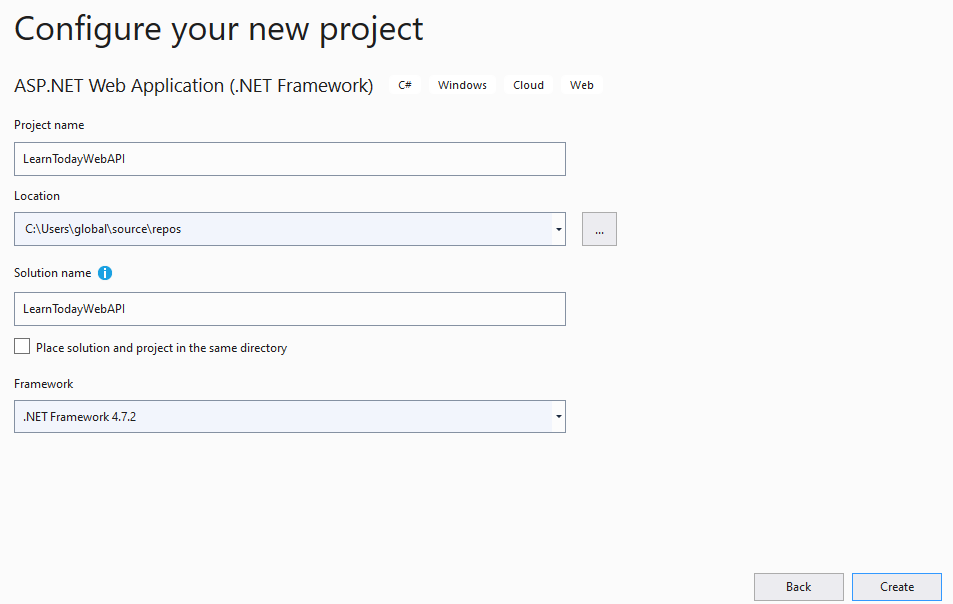
1. Visual studio 2017 Community Edition
2. Web API 2.x
3. Chrome Browser
4. Postman tool

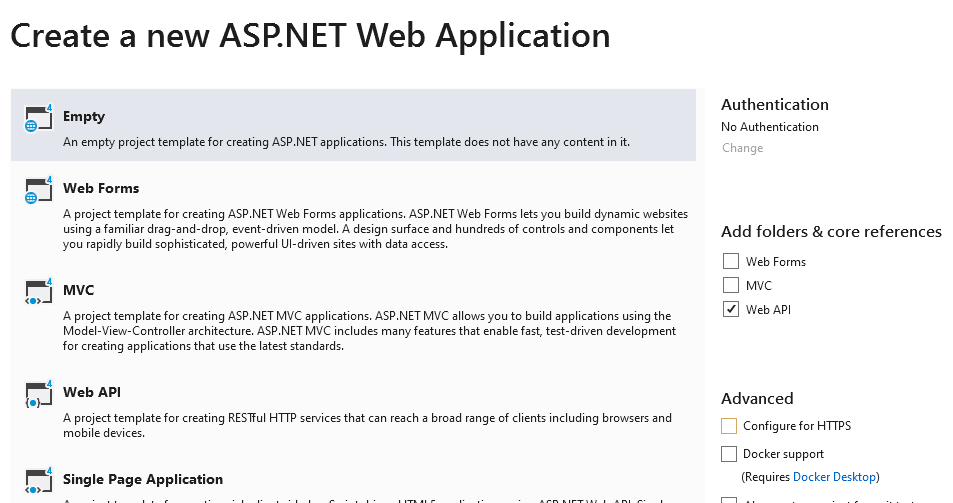
## Use Case Diagram



# Create ASP.NET Web API Application in Visual Studio

Create a Web Application project in Visual Studio. Select Web API template for creating RESTful service. Name it as **LearnTodayWebAPI**. [Sample image is given below for your reference]:





# Table Specification and Solution Folder Structure

**The sample Database and Table will be given for your reference:**

Database Name: **LearnTodayWebAPIDb**

Table: **Course**

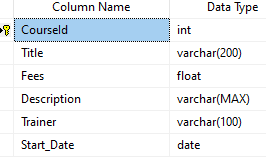
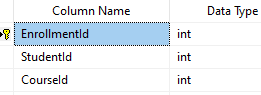
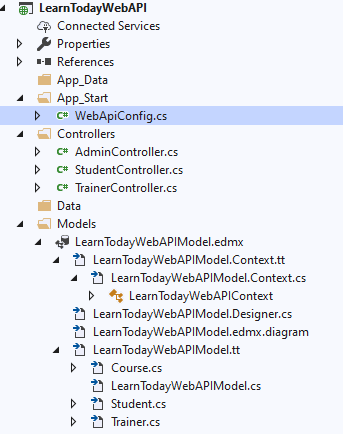


Table: **Student**

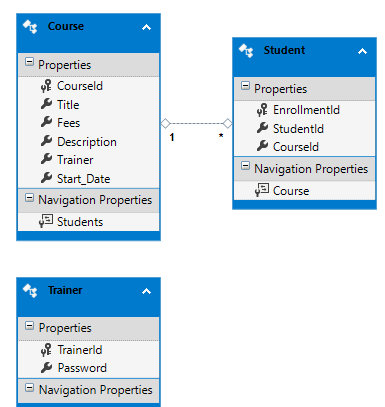


**Solution Folder Structure:**

****

# 4. Model Specification

You will use here **Data First Approach using Entity Framework 6** to generate your Model class and **DBContext** class. The sample .edmx file and screenshot is given for your reference:

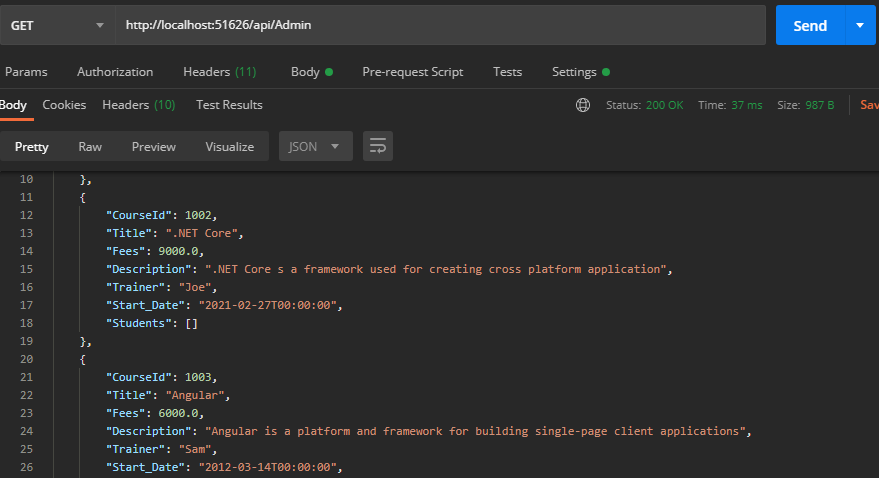


# API Controller and Postman Test Specification

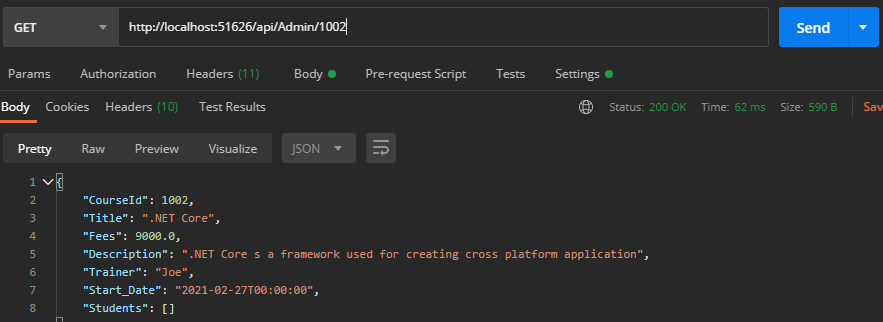
## 5.1 Create a new AdminController which will contain 2 action methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Type** | **Method Name** | **Parameter** | **Return Type** | **Description** |
| HttpGet | GetAllCourses | No Parameter | IEnumerable<Course> | This method will return all the courses. |
| HttpGet | GetCourseById | int id | HttpResponseMessage | Here the course will be searched by course id. If searched course is found it will return HttpStatusCode.Ok and the searched course data. Otherwise it will return HttpStatusCode.NotFound with a message "Searched Data not Found". |

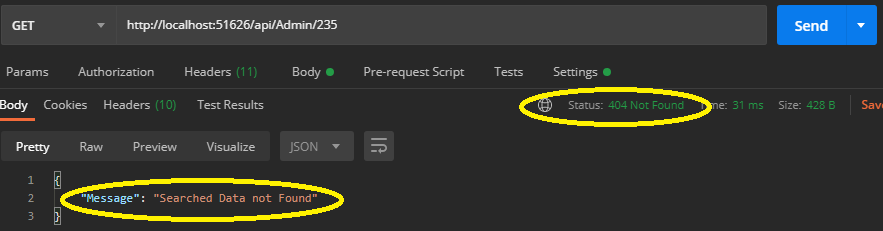
**Below is POSTMAN screenshot for GetAllCourses action:**



**Below is POSTMAN screenshot for GetCourseById action:**



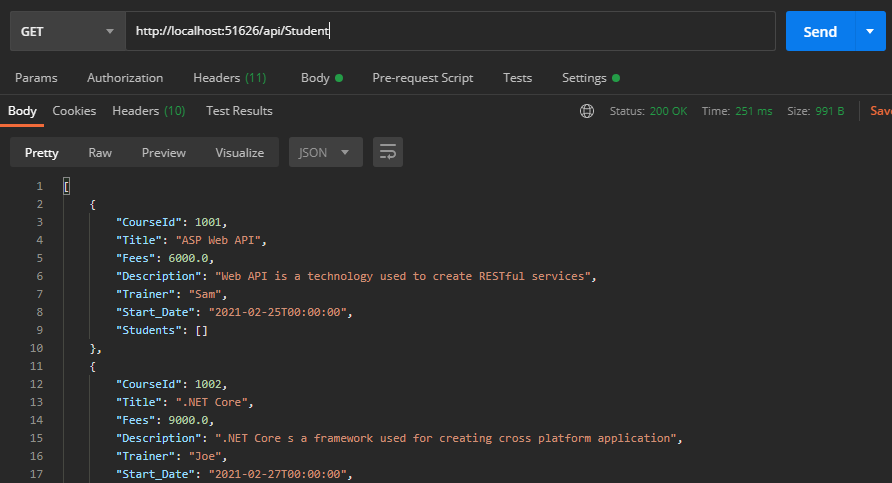
**If Course is not found by searched Course Id below is the Postman result:**



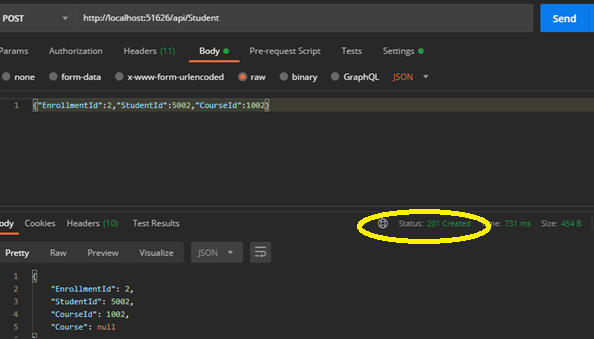
## 5.2 Create a new StudentController with the below 3 action methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Type** | **Method Name** | **Parameter** | **Return Type** | **Description** |
| HttpGet | GetAllCourses | No Parameter | IEnumerable<Course> | This method will return all the courses in ascending order by start date |
| HttpPost | PostStudent | [FromBody] Student model | HttpResponseMessage | This method is used to method to enroll student for the course. Input student information by using FromBody attributes. Use try-catch block. Return HttpStatusCode.Created with the model object if enrollment is successful; otherwise return HttpStatusCode.BadRequest with exception object in catch block. |
| HttpDelete | DeleteStudentEnrollment | int id | HttpResponseMessage | This method is used to remove the student information by the enrollment id Use a try catch block. If deletion successful then return HttpStatusCode.OK. If enrollement information is not found by searched id then return HttpStatusCode.NotFound with a message "No enrollement information found". Under catch block return HttpStatusCode.BadRequest with exception object if some exceptions occur in the program. |

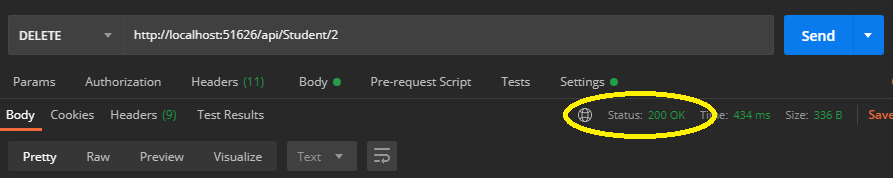
**Below is POSTMAN screenshot for GetAllCourses action:**



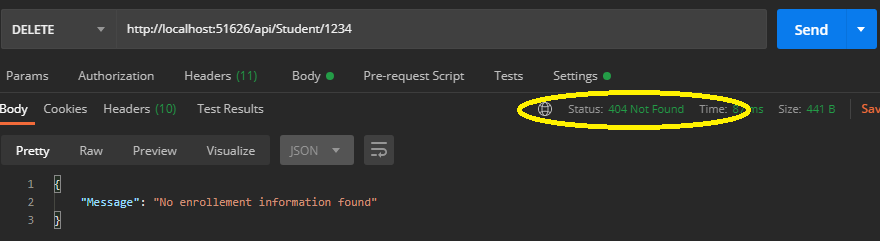
**Below is POSTMAN screenshot for PostStudent action:**

******

**Below is POSTMAN screenshot for DeleteStudentEnrollment action:**

******

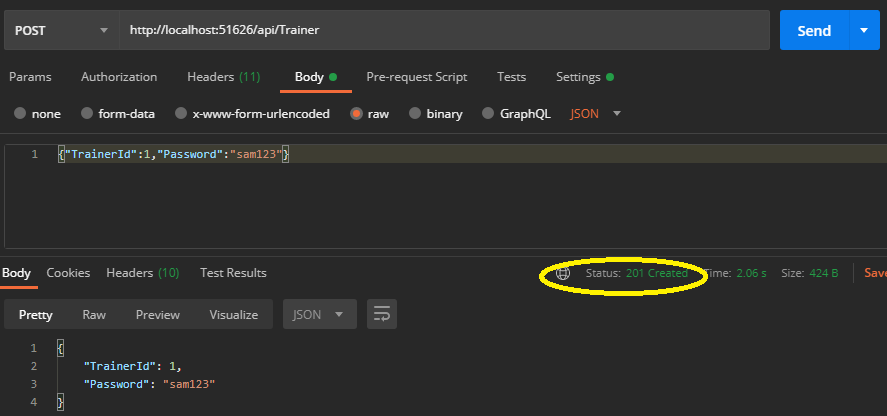
**Below is POSTMAN screenshot for DeleteStudentEnrollment action if searched enrolment id is not found:**

******

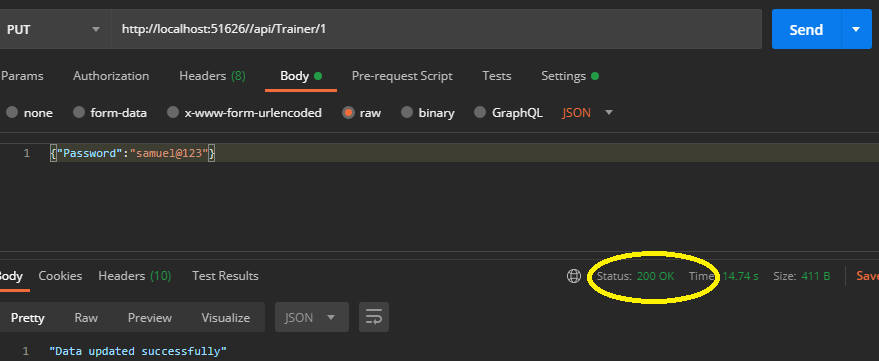
## 5.3 Create a new TrainerController with below 2 action methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Type** | **Method Name** | **Parameter** | **Return Type** | **Description** |
| HttpPost | TrainerSignUp | [FromBody] Trainer model | HttpResponseMessage | This method is used to method to register new trainer. Input trainer information by using FromBody attribute. Use try-catch block. Return HttpStatusCode.Created with the model object if sign up is successful; otherwise return HttpStatusCode.BadRequest with exception object in catch block. |
| HttpPut | UpdatePassword | int id, [FromBody] Trainer model) | HttpResponseMessage | This method will update trainer password by accepting trainer id from url and Trainer object from message body. If password update is successul then provide HttpStatusCode.OK response output with a message “Data updated successfully”; otherwise provide HttpStatusCode.NotFound response output with message "Searched Data not Found". Use try catch block. If there is any exception raised catch it and provide an HttpStatusCode.BadRequest response along with exception object. |

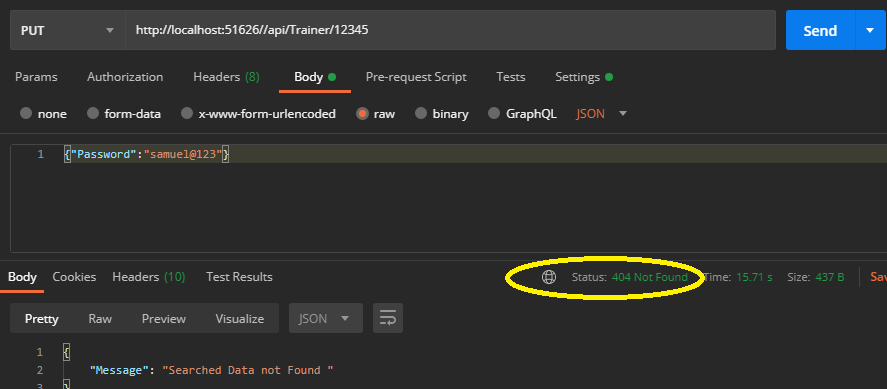
**Below is POSTMAN screenshot for TrainerSignUp action:**

******

**Below is POSTMAN screenshot for UpdatePassword action:**



**Below is POSTMAN screenshot for UpdatePassword action if trainer id not found:**

******

# Code submission instructions

Use git add, commit and push commands to upload your code into remote GltLab repository. During commit, give the commit message as “WebApi”.

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Changes Made** | | | |
| V1.0.0 | Initial baseline created on <dd-Mon-yy> by <Name of Author> | | | |
| Vx.y.z | <Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed> | | | |
| **Section No.** | **Changed By** | **Effective Date** | **Changes Effected** |
|  |  |  |  |