Harmony IT Solution

Web Application Programming Interface (API)

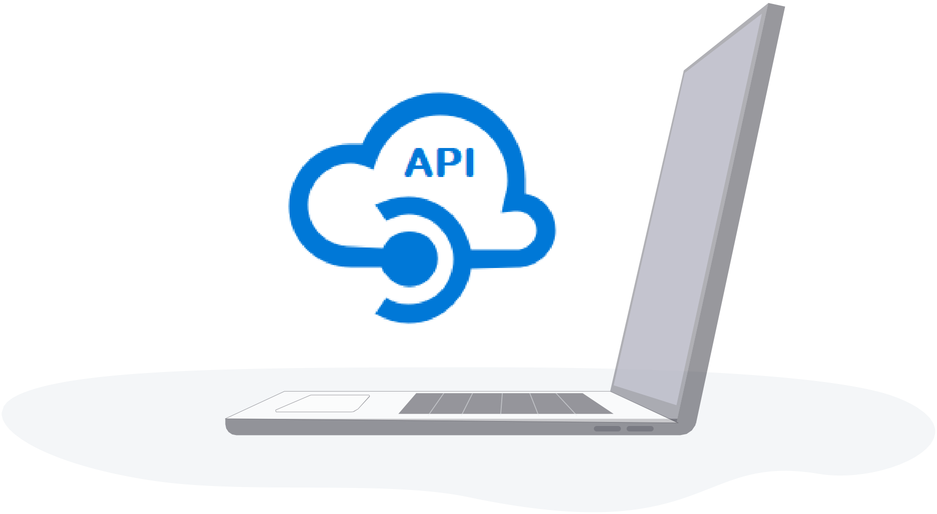
Tahaluf Training Center 2022

1 Overview of External API

2 The Differences Between external API and Internal API

3 Weather external API

4 Getting Data from Multiple Tables in Web API



Overview of External API

The main characteristics of external API activity are the Ability to fetch data in a JSON format file to a 3rd party restful API endpoint.

Ability to receive and save a JSON response back, map it to output tables, and pass it downstream to other workflow activities.

Limitations of External API:

1. 5MB HTTP response data size limit.
2. HTTP redirects are not allowed.
3. Request timeout is 1 minute.
4. Non-HTTPS URLs are rejected.

The Differences Between external API and Internal API

Internal APIs VS External APIs

One of the most important things that you should consider in both your interface architecture and API business strategy is the difference between internal and external API. An interface can be described as internal and external depending on whether its target is for in-house or external developers.

Internal APIs VS External APIs

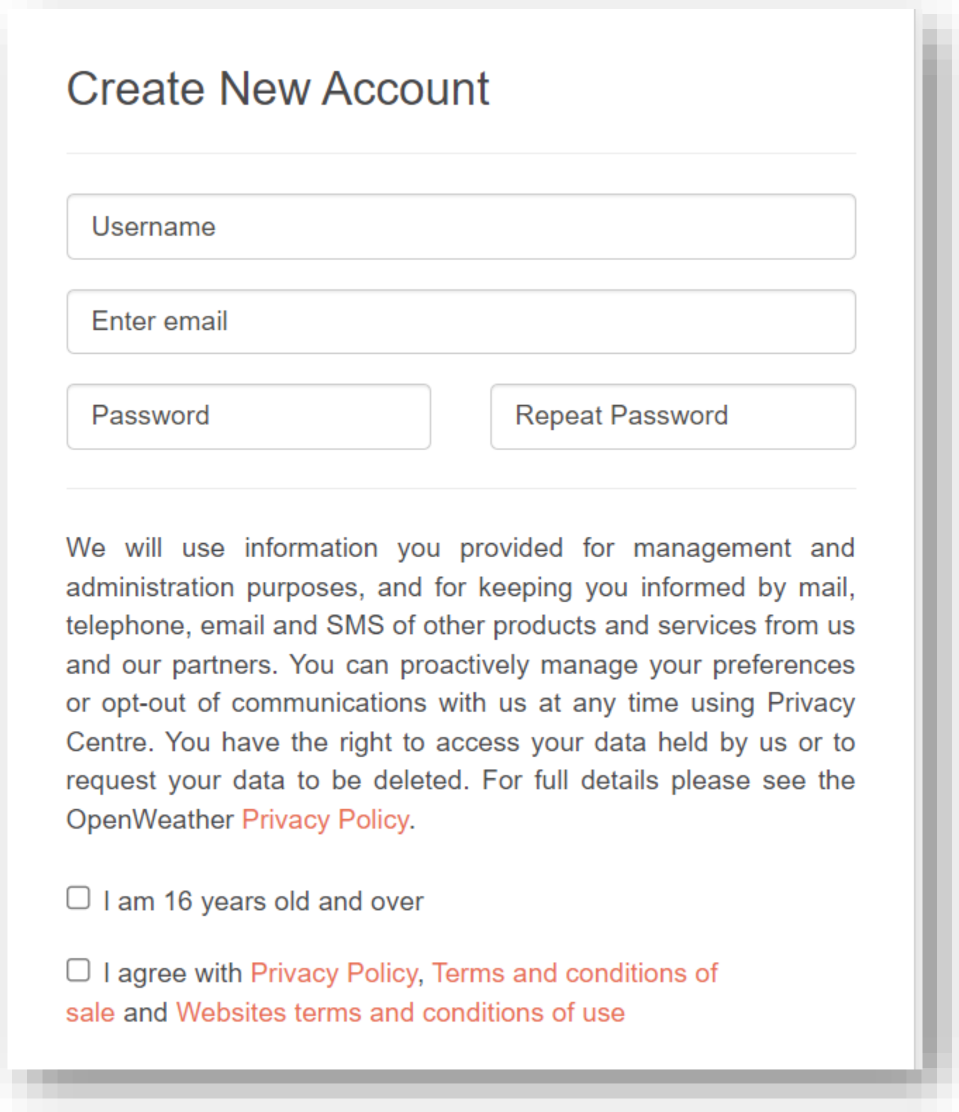
Is an API designed for access by a larger population as well as web developers. This implies that an external API can be easily used by developers inside the organization (that published the API) and any other developers from the outside who desires to register into the interface.

Weather external API

Overview of Weather external API

OpenWeather provides historical, current and forecasted weather data via light-speed APIs.

Before doing anything, you need an API key. Go to the **Signup page**.



Create New Account

Username

Enter email

Password

Repeat Password

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather Privacy Policy.

□ I am 16 years old and over

□ I agree with Privacy Policy, Terms and conditions of sale and Websites terms and conditions of use

Overview of Weather external API

The API key will send to you via email and may be found on the API keys page (under your account).

In order to display the current weather for any city using the weather API on openweathermap.org

Create Weather controller:

```
  [HttpGet("weather/{city}")]
    public async Task<Weather> City(string city)
        {
        using (var client = new HttpClient())
            {
                var response = await
client.GetAsync($"http://api.openweathermap.org/data/2.5/weather?q={city}&appid=511b
a00e6b1fdebcf7456541e7a16390");
                var stringResult = await response.Content.ReadAsStringAsync();
                var weatherResult =
JsonConvert.DeserializeObject<Weather>(stringResult);
                return weatherResult;
            }
        }
```

In TahalufLearn.core => DTO => Create a class for Weather:

```
namespace TahalufLearn.core.DTO
    {
    public class Main
        {
        public string Temp { get; set; }
        public string humidity { get; set; }
        }
    public class Wind
        {
        public string speed { get; set; }
        }
public class Weather
        {
        public Main main { get; set; }
        public Wind wind { get; set; }
        public string name { get; set; }
        public string timeZone { get; set; }
        }
    }
```

Getting Data from Multiple Tables

To retrieve each category and their courses:

In Course_Package Create GetAllCategoryCourse Proceadure:

```
create or replace PACKAGE Course_Package AS

PROCEDURE GetAllCategoryCourse;

END Course_Package;
```

```
create or replace PACKAGE Body Course_Package
as
PROCEDURE GetAllCategoryCourse
AS
c_all sys_refcursor;
BEGIN
OPEN c_all FOR
SELECT  cat.categoryid, cat.categoryName , C.CourseId , C.CourseName
FROM Course  C
INNER JOIN category  cat
ON c.categoryid = cat.categoryid;
DBMS_SQL.RETURN_RESULT(c_all);
END GetAllCategoryCourse;
END Course_Package;
```

In TahalufLearn.Core => Reopsitory => ICourseRepository => Create GetAllCategoryCourse:

```
Task<List<Category>> GetAllCategoryCourse();
```

In TahalufLearn.Infra => Reopsitory => CourseRepository => Create GetAllCategoryCourse:

```
public async Task<List<Category>> GetAllCategoryCourse()
            {
            var p = new DynamicParameters();
            var result = await dBContext.Connection.QueryAsync<Category, Course, 
Category>("Course_Package.GetAllCategoryCourse",
            (Category, course) =>
            {
                Category.Courses.Add(course);
                return Category;
            },
 splitOn: "Courseid",
      param: null,
      commandType: CommandType.StoredProcedure

      );
      var results = result.GroupBy(p => p.Categoryid).Select(g =>
      {
        var groupedPost = g.First();
        groupedPost.Courses = g.Select(p =>
p.Courses.Single()).ToList();
        return groupedPost;
      });
      return results.ToList();
      }
```

In TahalufLearn.Core => Service => ICourseService => Create GetAllCategoryCourse:

```
Task<List<Category>> GetAllCategoryCourse();
```

In TahalufLearn.Infra => Service => CourseService => Create GetAllCategoryCourse:

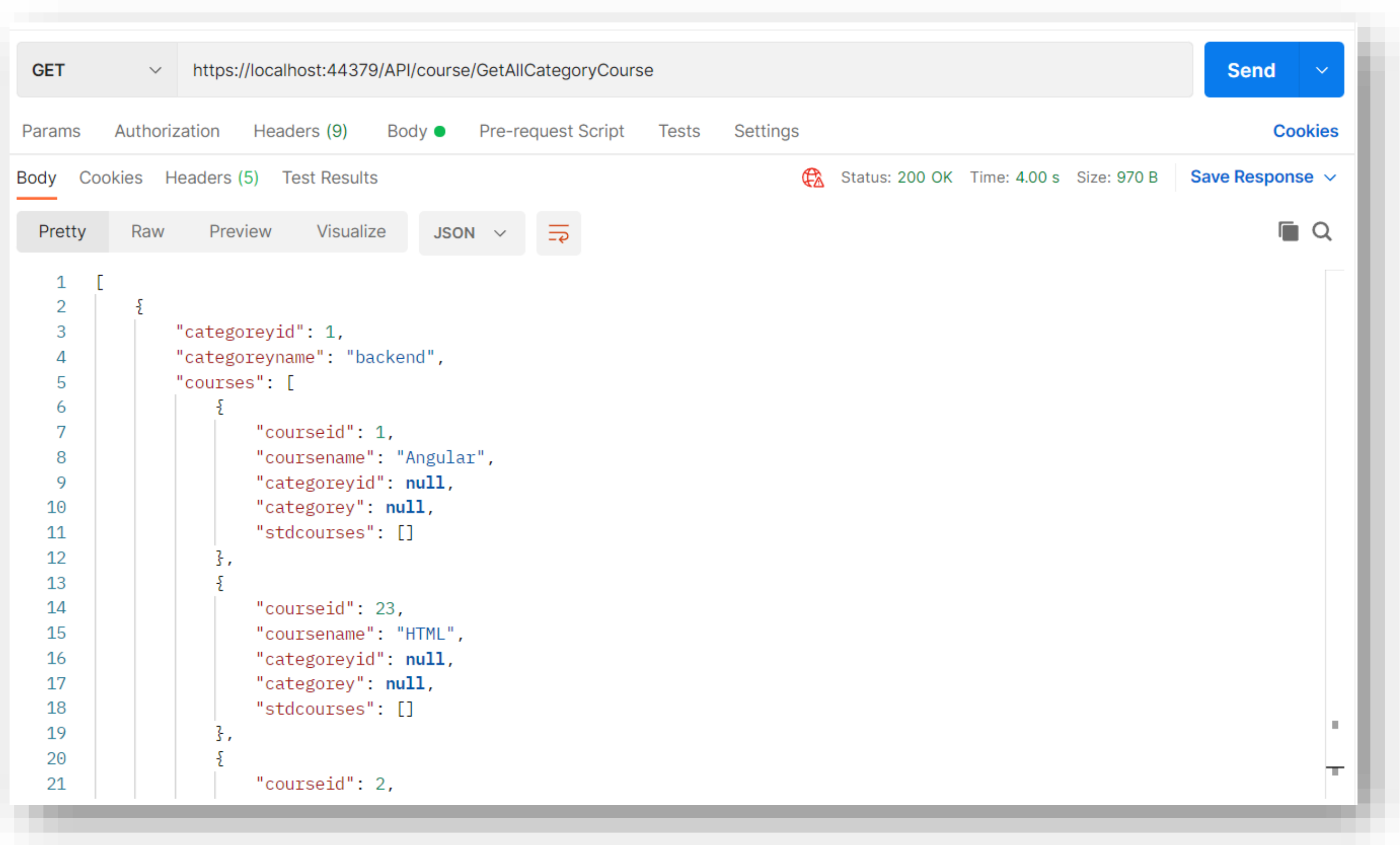
```
public Task<List<Category>> GetAllCategoryCourse()
           {
           return courseRepository.GetAllCategoryCourse();
           }
```

In TahalufLearn.API => Controller => CourseController => Create GetAllCategoryCourse:

```
[HttpGet]
       [Route("GetAllCategoryCourse")]
       public Task<List<Category>> GetAllCategoryCourse()
           {
           return courseService.GetAllCategoryCourse();

           }
```

The Result on postman:



GET https://localhost:44379/API/course/GetAllCategoryCourse

Status: 200 OK Time: 4.00 s Size: 970 B

```
[
    {
        "categoreyid": 1,
        "categoreyname": "backend",
        "courses": [
            {
                "courseid": 1,
                "coursename": "Angular",
                "categoreyid": null,
                "categorey": null,
                "stdcourses": []
            },
            {
                "courseid": 23,
                "coursename": "HTML",
                "categoreyid": null,
                "categorey": null,
                "stdcourses": []
            },
            {
                "courseid": 2,
```