# فهرست مطالب

[فهرست مطالب **Error! Bookmark not defined.**](#_Toc134386408)

[مقدمه: 2](#_Toc134386409)

[تعریف مسئله 3](#_Toc134386410)

[راهکارهای حل مسئله:(نیاز سنجی) 4](#_Toc134386411)

[ایجاد بصورت دستی و سرچ کردن تک به تک آنها در IEnumerable کانفیگ ها و استفاده از دیزاین پترن chain of responsibility 4](#_Toc134386412)

[استفاده از جست و جوی اول سطح و اول عمق در گراف: 4](#_Toc134386413)

[استفاده از گراف والگوریتم کوتاه ترین مسیر دایکسترا(Dijkstra) 5](#_Toc134386414)

[خلاصه راه حل: 5](#_Toc134386415)

[درباره پروژه: 6](#_Toc134386416)

[فناوری های مورد استفاده در این پروژه: 6](#_Toc134386417)

[توضیحات مربوط به لایه ها و انواع ساخته شده: 7](#_Toc134386418)

[تغییرات مورد نیاز در آینده برای بهتر کردن پروژه: 11](#_Toc134386419)

[نحوه ی استفاده از برنامه: 13](#_Toc134386420)

[حذف تنظیمات: 13](#_Toc134386421)

[ایجاد یا به روز رسانی تنظیمات: 13](#_Toc134386422)

[استفاده از تبدیلگر واحد پول: 14](#_Toc134386423)

# مقدمه:

با سلام و احترام خدمت خوانندگان عزیز این داکیومنت ،لازم به ذکر است ریز توضیحات کدها در خود سورس کد وجود دارد و دراینجا به کلیت نحوه ی انجام کار متدهای مهم و نوع های مورد استفاده اشاره شده است. و در کد نویسی سعی بر این بوده که خود کد و توضیحات آن تمام نیازهای خواننده کد را رفع نمائید یا به عبارتی Self Description باشد.

# تعریف مسئله

ایجاد یک Social network با ویژگی های زیر

* ایجاد یک وب اپلیکیشن MVC بر پایه Asp.net core
  + ایجاد امکان لاگین و ریجستر کردن برای یوزر ها
  + امکان مشاهده لیست همه ی یوزر ها و جست و جو براساس نام
  + امکان ایجاد موتال فرند و درخواست دوستی و ارسال پیام بعد از پذیرفتن یک نفر
* موارد مهمی که در این تسک مورد نظر است:
  + دیزاین دیتابیس و جلوگیری از افزونگی در آن
  + طراحی کوئری ها با بالاترین راندمان
  + جزئیات طراحی آن به عهده برنامه نویس است

# راهکارهای حل مسئله:(نیاز سنجی)

## 1-نحوه ی Authentication

با استفاده از سولوشن های ساده ی پیاده سازی Authenticate

### 2- Entity های مورد نیاز براساس مسئله

#### 1-User:

### فیلد های مورد نیاز برای یوزر به شرح ذیل خواهد بود:

### ID:long

### Name:string(255)

### LastName:string(255)

### Email:string(500)

* Birthday:DateTime
* Password:String
* About:String
* profilePicture:String
* CreationDate:DateTime

#### رفتارهای مورد نیاز برای یوزر:

* ایجاد یوزر
* تغییر پسورد
* تغییر مشخصات
* ارسال پیام
* امکان لایک

### 2-UserRelations:

این یک Value object برای user میباشد

#### فیلد های مورد نیاز به شرح ذیل است:

* UserA:User
* UserB:User
* Approve:bool

#### رفتارهای مورد نیاز برای دوستی ها(UserRelations)

* ارسال درخواست دوستی
* تایید درخواست
* رد درخواست

3-Message

این مورد نیز یک value object برای user میباشد

#### فیلد های مورد نیاز به شرح ذیل است:

* FromUser:User
* ToUser:User
* Message:string
* Like:bool
* IsRead:bool

#### رفتارهای مورد نیاز برای دوستی ها(Message)

* ارسال پیام
* لایک پیام
* آن لایک پیام
* تغییر به عنوان خوانده شده

# درباره پروژه:

## فناوری های مورد استفاده در این پروژه:



## توضیحات مربوط به لایه ها و انواع ساخته شده:









# تغییرات مورد نیاز در آینده برای بهتر کردن پروژه:

1-استفاده از معماری CleanArchitecture

2-استفاده از پکیج AutoMapper جهت مپ کردن های مختلف

3-استفاده از پکیج MediatR جهت پیاده سازی CQRS و هم چنین استفاده از دیزاین پترن Mediator برای حل این مسئله

4-استفاده از فایل آپلودر جهت آپلود کردن عکس کاربران

5-استفاده از Logger

تغییرات مهم ایجاد شده در حین توسعه پروژه:

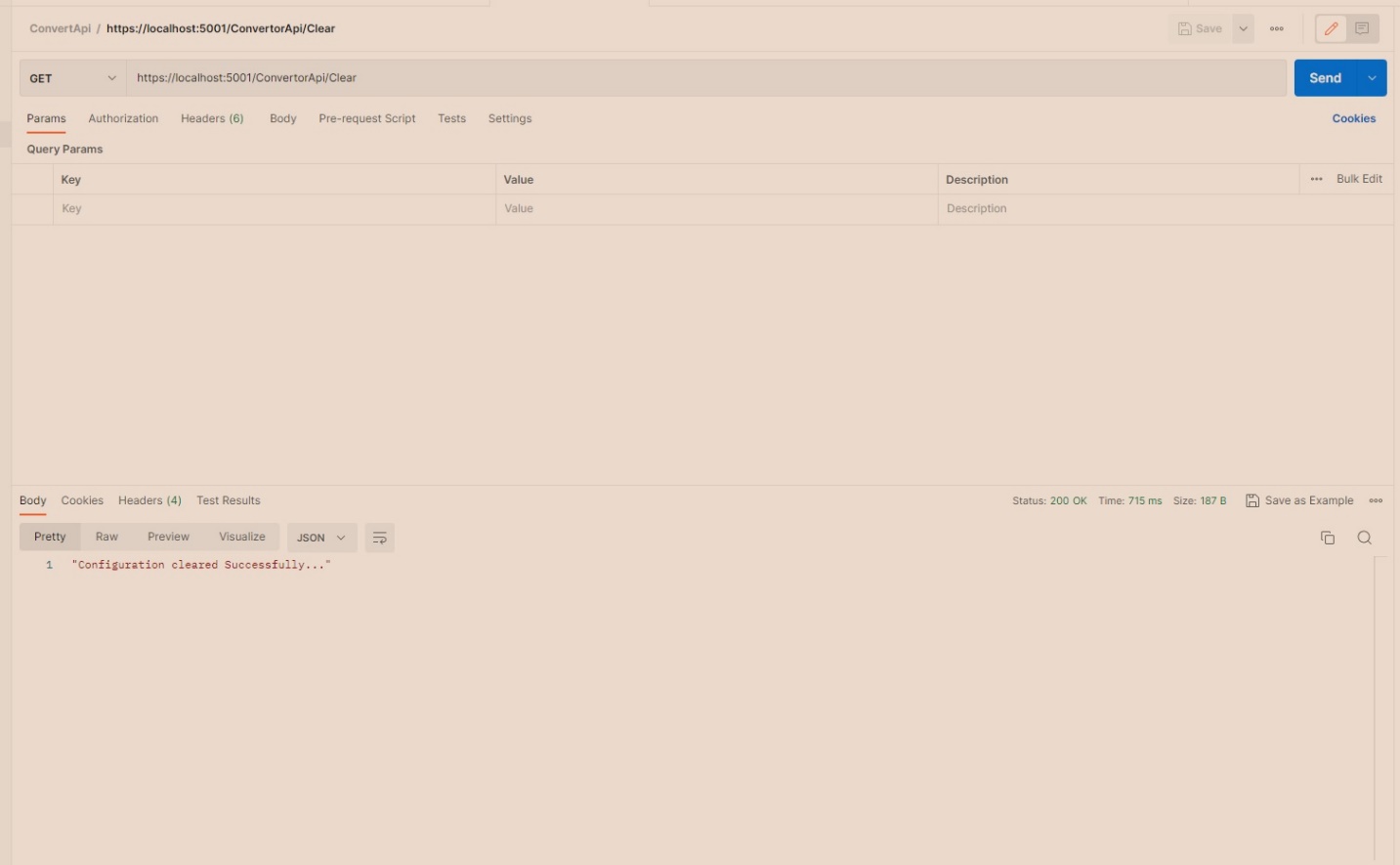
1-به علت عدم نیاز به دیتابیس جهت ذخیره سازی پیکربندی های تبدیل واحد این قسمت از متد UpdateConfiguration در کلاس CurrencyConvertor حذف گردید.و انجام ان بدون ذخیره سازی و واکشی از دیتابیس انجام شد.

# نحوه ی استفاده از برنامه:

## حذف تنظیمات:

با استفاده از API زیر میتوان تنظیمات فعلی را حذف کرد.

<https://Domain/ConvertorApi/Clear>



## ایجاد یا به روز رسانی تنظیمات:

با استفاده از درخواست Ajax زیر میتوانید تنظیمات را ایجاد یا تغییر دهید:

**var** settings = {

  "url": "https://Domain/ConvertorApi/Config",

  "method": "POST",

  "timeout": 0,

  "headers": {

    "Content-Type": "application/json"

  },

  "data": JSON.stringify([

    {

      "item1": "USD",

      "item2": "CAD",

      "item3": 1.34

    },

    {

      "item1": "CAD",

      "item2": "GPB",

      "item3": 0.58

    },

    {

      "item1": "USD",

      "item2": "EUR",

      "item3": 0.86

    }

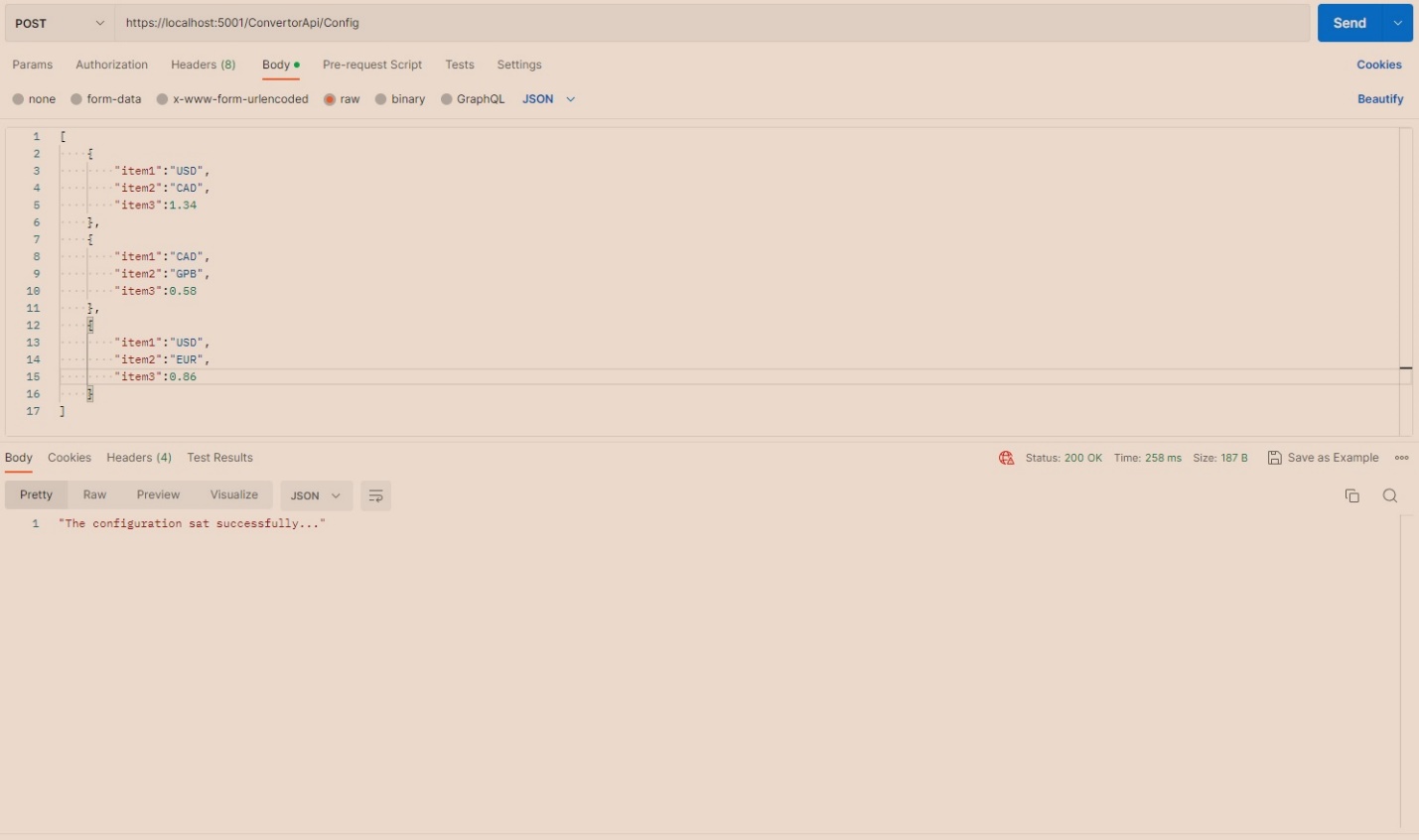
  ]),

};

$.ajax(settings).done(**function** (response) {

  console.log(response);

});



## استفاده از تبدیلگر واحد پول:

برای استفاده از تبدیلگر میتوانید از لینک زیر استفاده نمائید.یا...

<https://Domian/ConvertorApi/Convert?fromCurrency=CAD&toCurrency=IRR&amount=7>

هم چنین میتوانید از کد جاوااسکریپت زیر در پروژه هایتان نیز استفاده کنید.

**var** settings = {

  "url": "https://localhost:5001/ConvertorApi/Convert?fromCurrency=CAD&toCurrency=IRR&amount=7",

  "method": "GET",

  "timeout": 0,

};

$.ajax(settings).done(**function** (response) {

  console.log(response);

});

