

## سند API کنترل ترافیک

علی نجفی

### ۱-۱-۱ اپ ها:

سیستم دارای دو اپ به نام های mainapp و geo میباشد :

### ۱-۱-۱ mainapp :

این اپ منطق اصلی برنامه را بر عهده دارد ، از جمله مدیریت مدل‌هایی نظیر خودرو (CAR)، صاحب ماشین (Owner) و تردد ماشین ها (CarTraffic) .

### ۱-۱-۱ url ها:

آدرس اصلی این اپ با /system/ شروع میشود .

برای ادرس دهی از url router به واسطه modelviewset ها استفاده شده است.

### ۱-۱-۱-۱ صاحبان خودرو :

آدرس اولیه با /owners/ شروع میشود. که اگر متد get فرستاده شود لیست صاحبان نمایش داده میشود.

اگر متد post داده شود به منزله ایجاد صاحب خودرو خواهد بود. اگر کاربر میخواهد لیستی از صاحبان خودرو را ایجاد کند کافیسٹ فرمت json را به صورت لیست بفرستد تا لیستی از صاحبان ایجاد شوند.

برای مشاهده detail یک صاحب خودرو کافیسٹ به فرمت زیر آدرس را وارد کنید:

/owners/{national\_code}

با فرستادن درخواست get اطلاعات یک صاحب با کد ملی national\_code نمایان میشود.

برای مشاهده صاحبانی که تخلف عوارضی دارند از آدرس زیر استفاده میکنیم:

/owners/violated-list

تعریف صاحبان خودرو متخلف : صاحبانی که به واسطه خودروشان از عوارضی عبور کرده اند ولی کل مبلغ پرداختی آنها یا صفر باشد یا کمتر از مقدار مورد انتظار باشد.

با درخواست get لیست صاحبانی که تخلف داشته فرستاده میشود.

### ۱-۱-۱-۲ خودرو ها:

آدرس اولیه خودرو ها /cars/ میباشد :

درخواست get لیستی از تمامی خودرو هارا نمایش میدهد.

با درخواست post میتوان خودرو ایجاد کرد و یا میتوان لیستی از خودرو ها را نیز اضافه کرد.

برای مشاهده اطلاعات یک خودرو مانند ۱-۱-۱-۱ به فرمت زیر :

/cars/{id}

با دادن ایدی خودرو مورد نظر میتوان جزئیات آن را مشاهده کرد.

میتوان با دادن url parameter لیست خودرو هارا در آدرس /cars/ فیلتر کرد:

1- /cars?color=red,blue

2-/cars?color=RED,BLUE&age\_\_lte=70&age\_\_gt=40

۱- تمام خودرو هایی را میدهد که رنگشان ابی یا قرمز است .

۲- تمام خودرو هایی را میدهد که ابی و یا قرمز هستند و سن صاحبانشان بزرگتر از ۴۰ و کوچک تر از ۷۰ سال است.

نکته : همانطور که مشاهده میکنید مقدار رنگ case insensitive است.

برای مشاهده لیستی از خودرو های نزدیک به عوارضی و در محدوده مورد نظر از آدرس زیر استفاده میشود.

/cars/cars-near-tollStation/

این آدرس فقط از متد post پشتیبانی میکند زیرا کاربر باید یکسری اطلاعات نظیر اسم عوارضی ، شعاع محدوده (به کیلومتر) و تاریخ شروع و تاریخ پایان را به سرور ارسال کنند.

برای یافتن خودرو هایی که از جاده هایی با عرض کمتر از مقدار مورد نظر عبور کردند از آدرس زیر استفاده میکنیم:

/cars/cars-passage/

باید از متد post , داده هایی نظیر : عرض مورد نظر داده شود. تاریخ شروع و پایان و نوع خودرو اختیاری میباشد. نوع خودرو به صورت پیشفرض big است.

۱-۱-۳ عبور و مرور خودروها:

آدرس اولیه به صورت /traffics/ میباشد.

با ارسال درخواست get به این آدرس لیست تمامی عبور و مرور خودرو ها برگردانده میشود.

با ارسال post به همین آدرس میتوان عبور و مرور را در سیستم ثبت کرد. که میتوان لیستی از عبورمرور را برای ایجاد فرستاد.

۱-۲-۲ geo :

این اپ منطق ژئوگرافی برنامه را بر عهده دارد و دارای مدل هایی نظیر : راه (Road) و عوارضی (TollStation) است.

در ماژول utils در این اپ روش ها و ابزار های پردازش موقعیت و ابزار های موقعیت یابی وجود دارد.

۱-۲-۱ url ها:

آدرس اولیه این اپ با /geo/ آغاز میشود .

۱-۲-۱ جاده ها :

آدرس اصلی جاده ها /roads/ میباشد .

برای گرفتن لیست تمامی جاده ها از متد get استفاده میکنیم.

برای ایجاد یک یا چند جاده از همین آدرس با متد post جاده ها را ایجاد میکنیم.

۱-۲-۲ عوارضی ها:

آدرس عوارضی ها با /toll-stations/ شروع میشود .

درخواست متد get بر روی همین آدرس لیست عوارضی ها را نشان میدهد.

میتواند یک عوارضی یا لیستی از عوارضی ها را به واسطه درخواست post ایجاد کرد.

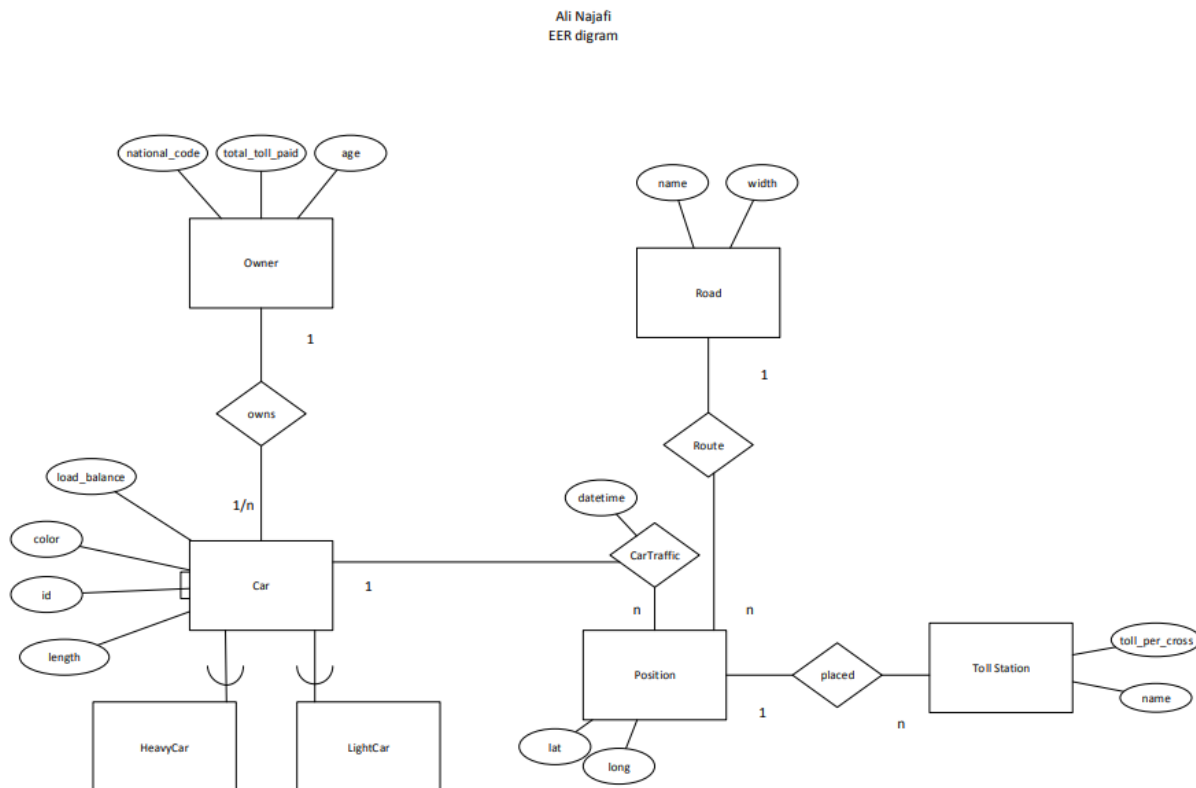
ادرس زیر به نشان میدهد که یک مختصات در محدوده یک عوارضی است یا نه:

/toll-stations/{id}/position-near/

کاربر باید مختصات مورد نظر (lat,lng) را به شعاع محدوده بدهد. مثلا مختصات ۱۲,۹۸ و ۵۶,۹۸ در محدوده ۳ کیلومتری یک عوارضی است یا نه. جواب true یا false است.

## ۲- پایگاه داده:

نمودار EER پایگاه داده به شکل زیر است:



از mysql به عنوان پایگاه داده استفاده شده.

## ۲-۱ محدودیت های پایگاه داده :

در سیستم هر صاحب خودرو فقط میتواند یک خودرو از نوع بزرگ داشته باشد. برای اعمال این محدودیت دوتریگر برای هم insert و update نوشته است. که در پوشه DB قابل مشاهده است.

## ۳ – ارسال داده های تست به سرور:

داده های تست قبلا به واسطه ماژول data\_sender.py به سرور ارسال شده اند. اگر میخواهید یک دیتا بیس جدید بسازید کافیست این ماژول را اجرا کنید سپس y یا yes را وارد کنید. به صورت اتوماتیک داده ها به سرور اضافه میشوند. پیش فرض اتوماتیک چهار فایل json موجود در دایرکتوری خود پروژه است. اگر میخواهید داده های تست دیگر را نیز اضافه کنید باید ابتدا no وارد کرده سپس طبق دستور العمل خود برنامه ادرس فایل های تست را وارد کنید. این برنامه قابلیت فرستادن داده را به صورت موازی برای هر فایل دارد تا سریع تر به داده ها به سرور فرستاده شوند. شما میتوانید تعداد thread ها را برای هر فایل نیز مشخص کنید.

