

# dashboardProject

---

to use api

`/api/(specify collection)/(specify category)/(sub category)`

## usage

API	Description
<code>/api/arrivedAt/type/visitors</code>	collection of visitors by arrival time
<code>/api/arrivedAt/type/customer</code>	collection of customers by arrival time
<code>/api/arrivedAt/gender/male</code>	collection of males by arrival time
<code>/api/arrivedAt/gender/female</code>	collection of females by arrival time
<code>/api/gender/type/customer</code>	collection of customers based on their gender
<code>/api/gender/type/visitors</code>	collection of visitors based on their gender
<code>/api/age/type/visitors</code>	collection of visitors by ages
<code>/api/age/type/customers</code>	collection of customers based on their gender
<code>/api/age/gender/male</code>	collection of males by ages
<code>/api/age/gender/female</code>	collection of females by ages
<code>/api/timeSpent/type/visitors</code>	collection of time spent by visitors
<code>/api/timeSpent/type/customers</code>	collection of time spent by customers
<code>/api/timeSpent/gender/male</code>	collection of time spent by males
<code>/api/timeSpent/gender/female</code>	collection of time spent by females

## API code

```
router.get("/api/:collection/:in/:of",function (req,res) {
  Record.aggregate([{$match:{$[req.params.in] : req.params.of}},
    {$group:{
      _id : "$"+req.params.collection,
      total : {$sum : 1}},
      {$sort: {_id: 1}}])
    .exec((err,data) => {
      if (err) {console.log(err)}
      res.json(data)
    })
  })
})
```

## Model Structure

```
var RecordSchema = new mongoose.Schema({  
  
  type:String,  
  gender:String,  
  age:Number,  
  timeSpent:Number,  
  arrivedAt:Number  
  
})
```