



Eloquent Cheat Sheet

[Cheat Sheet \(/explore?tag=Cheat Sheet\)](/explore?tag=Cheat Sheet) [Eloquent \(/explore?tag=Eloquent\)](/explore?tag=Eloquent) [Database \(/explore?tag=Database\)](/explore?tag=Database)

Dec 17, 2016, by Leonid Shumakov

Select

```
->select('col1','col2')
->select(array('col1','col2'))
->select(DB::raw('businesses.*, COUNT(reviews.id) as no_of_ratings, IFNULL(sum(reviews.
->addSelect('col3','col4')
->distinct() // distinct select
```

From

```
->from('table')
->from(DB::raw('table, (select @n :=0) dummy'))
->from(DB::raw("({$subQuery->toSql()}) T ")>mergeBindings($subQuery->getQuery()))
```

Where

```

->where('column','value')
->where('column','LIKE','%'.$value.'%')
->where(function ($query) {
    $query->where('a', '=', 1)
        ->orWhere('b', '=', 1);
})
->orWhere('column','!=', 'value')
->whereRaw('age > ? and votes = 100', array(25))

->whereRaw(DB::raw("id in (select city_id from addresses GROUP BY addresses.city_id)"));

->whereExists(function($query)
{
    $query->select(DB::raw(1))
        ->from('business_language')
        ->whereRaw('business_language.language_id = languages.id')
        ->groupBy('business_language.language_id')
        ->havingRaw("COUNT(*) > 0");
})
->orWhereExists()
->whereNotExists()
->orWhereNotExists()

->whereIn('column',[1,2,3])
->orWhereIn()
->whereNotIn('id', function($query){
    $query->select('city_id')
        ->from('addresses')
        ->groupBy('addresses.city_id');
})
->whereNotIn()
->orWhereNotIn

->whereNull('column') //where `column` is null
->orWhereNull('column') //or where `column` is null
->whereNotNull('column') //where `column` is not null
->orWhereNotNull('column') //or where `column` is not null

->whereDay()
->whereMonth('column', '=', 1) //
->whereYear('column', '>', 2000) //uses sql YEAR() function on 'column'
->whereDate('column', '>', '2000-01-01')

```

Joins

```

->join('business_category','business_category.business_id','=', 'businesses.id')
->leftJoin('reviews','reviews.business_id', '=', 'businesses.id')
->join('business_category',function($join) use($cats) {
    $join->on('business_category.business_id', '=', 'businesses.id')
    ->on('business_category.id', '=', $cats, 'and', true);
})
->join(DB::raw('(SELECT *, ROUND(AVG(rating),2) avg FROM reviews WHERE rating!=0 GROUP
    $join->on('genre_relation.movie_id', '=', 'T.id')
}))

```

Eager Loading

```

->with('table1','table2')
->with(array('table1','table2','table1.nestedtable3'))
->with(array('posts' => function($query) use($name){
    $query->where('title', 'like', '%'.$name.'%')
    ->orderBy('created_at', 'desc');
})))

```

Grouping

```

->groupBy('state_id','locality')
->havingRaw('count > 1 ')
->having('items.name','LIKE',"%$keyword%")
->orHavingRaw('brand LIKE ?',array("%$keyword%"))

```

Cache

```

->remember($minutes)
->rememberForever()

```

Offset & Limit

```

->take(10)
->limit(10)
->skip(10)
->offset(10)
->forPage($pageNo, $perPage)


```

Order

```
->orderBy('id','DESC')
->orderBy(DB::raw('RAND()'))
->orderByRaw('type = ? , type = ? ', array('published','draft'))
->latest() // on 'created_at' column
->latest('column')
->oldest() // on 'created_at' column
->oldest('column')
```

Create

```
->insert(array('email' => 'john@example.com', 'votes' => 0))
->insert(array(
    array('email' => 'taylor@example.com', 'votes' => 0),
    array('email' => 'dayle@example.com', 'votes' => 0)
)) //batch insert
->insertGetId(array('email' => 'john@example.com', 'votes' => 0)) //insert and return id
```



Update

```
->update(array('email' => 'john@example.com'))
->update(array('column' => DB::raw('NULL'))))
->increment('column')
->decrement('column')
->touch() //update timestamp
```

Delete

```
->delete()
->forceDelete() // when softdeletes enabled
->destroy($ids) // delete by array of primary keys
->roles()->detach() //delete from pivot table: associated by 'belongsToMany'
```

Getters

```

->find($id)
->find($id, array('col1','col2'))
->findOrFail($id)
->findMany($ids, $columns)
->first(array('col1','col2'))
->firstOrFail()
->all()
->get()
->get(array('col1','col2'))
->getFresh() // no caching
->get_cached() // get cached result
->chunk(1000, function($rows){
    $rows->each(function($row){

    });
})
->lists('column') // numeric index
->lists('column','id') // 'id' column as index
->lists('column')->implode('column', ',') // comma separated values of a column
->pluck('column') //Pluck a single column's value from the first result of a query.
->value('column') //Get a single column's value from the first result of a query.

```

Paginated results

```

->paginate(10)
->paginate(10, array('col1','col2'))
->simplePaginate(10)
->getPaginationCount() //get total no of records

```

Aggregate

```

->count()
->count('column')
->count(DB::raw('distinct column'))
->max('rating')
->min('rating')
->sum('rating')
->avg('rating')
->aggregate('sum', array('rating')) // use of aggregate functions

```

Others

```

->toSql() // output sql query
->exists() // check if any row exists
->fresh() // Return a fresh data for current model from database

```

Object methods

```
->toArray() //  
->toJson()  
->relationsToArray() //Get the model's relationships in array form.  
->implode('column', ',') // comma separated values of a column  
->isDirty()  
->getDirty() //Get the attributes that have been changed but not saved to DB
```

Debugging


```
DB::enableQueryLog();  
DB::getQueryLog();  
Model::where()->toSql() // output sql query
```

2 Comments

Laravel Gems

 Login ▾

 Recommend 5

 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

Jeff Page • 7 months ago

This is a great resource! Thank you very much

1 ^ | v • Reply • Share ›

Nixus • 8 months ago

Thank you

^ | v • Reply • Share ›

 Subscribe  Add Disqus to your siteAdd DisqusAdd  Disqus' Privacy PolicyPrivacy PolicyPrivacy Policy

Website built with Laravel (<http://laravel.com>)



(<https://www.facebook.com/laravelgems>)



(<https://twitter.com/laravelgems>)



(<https://github.com/laravelgems>)