CODE SCHOOL

Presents

TRY LARAVEL

AN INTERACTIVE AND EDUCATIONAL PRODUCTION

Level 1 – Section 1

Welcome to Laravel

What Are We Doing Here?

Prerequisites of Try Laravel

Basic & Some Object-oriented PHP

Try PHP & Close Encounters With PHP

Basic Database Knowledge

Try SQL

What Is Laravel?

Laravel is a web framework built in PHP.

Open source!

Initial release in 2011, currently in Version 5.4.

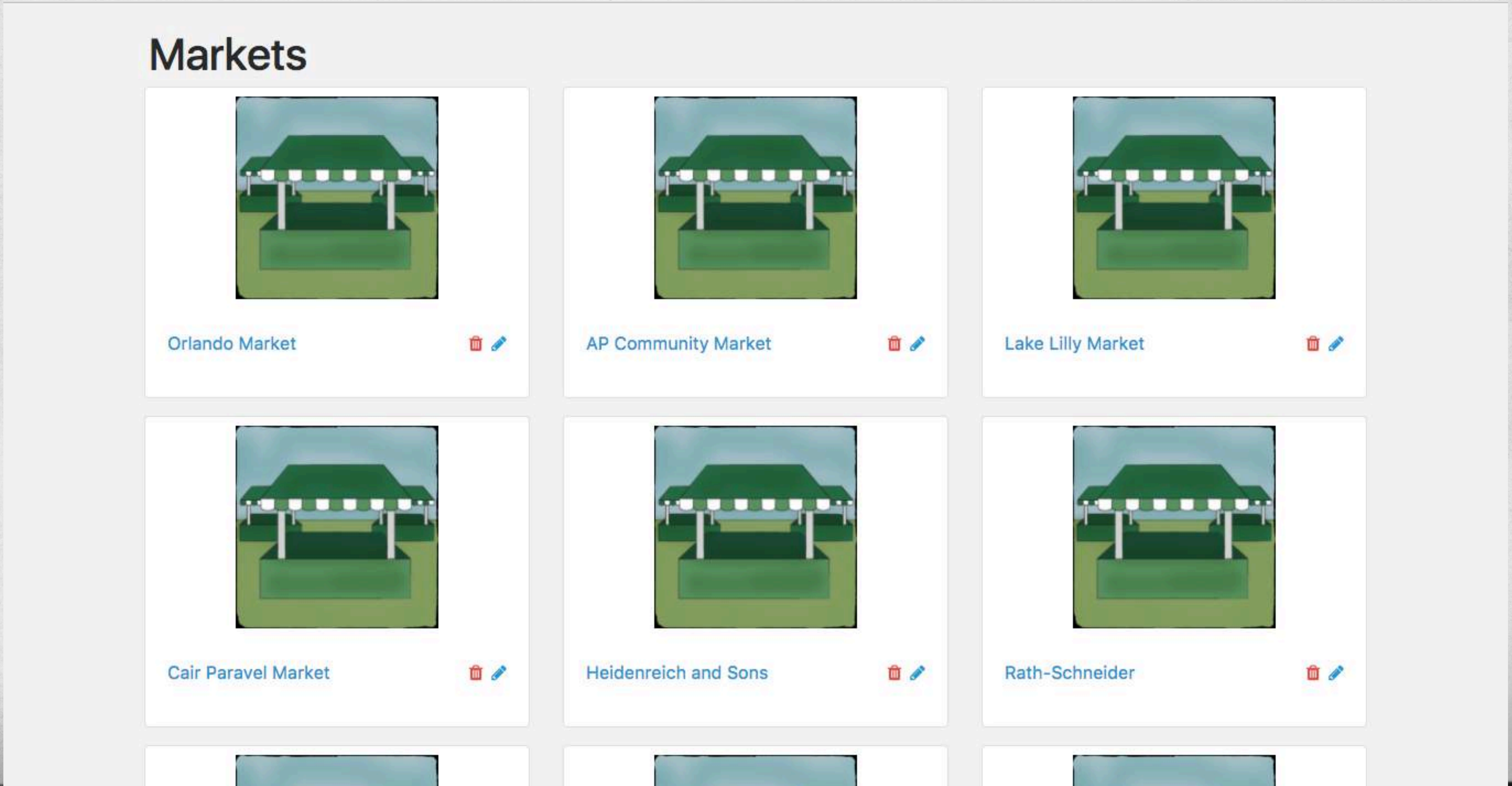
Built for developing applications using the MVC (model view controller) pattern.

Uses RESTful controllers for predictable URL patterns.

Built-in database ORM and migrations.

Far too many features to list!

Local Farmers Market Listing



Markets

Orlando Market

AP Community Market

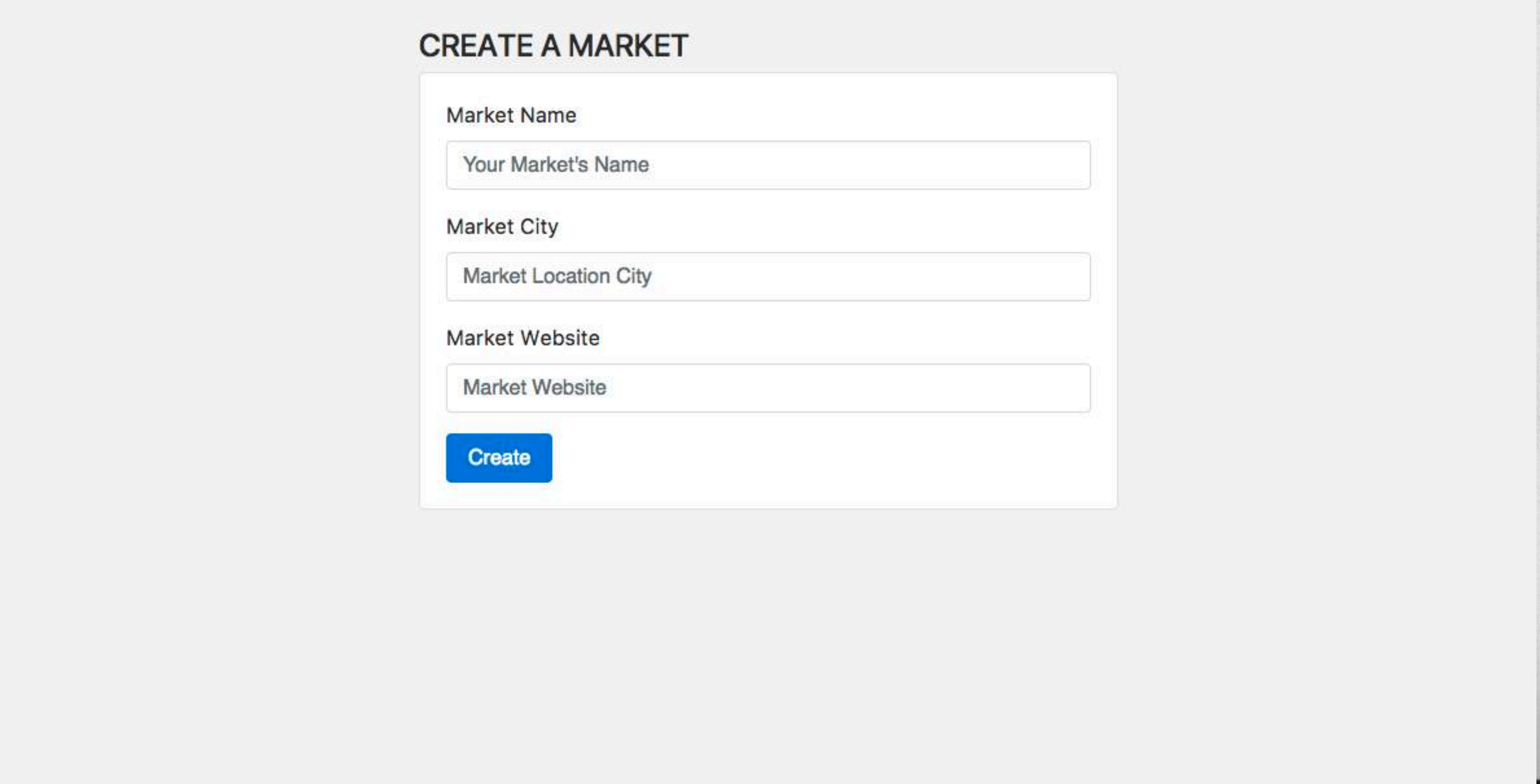
Lake Lilly Market

Cair Paravel Market

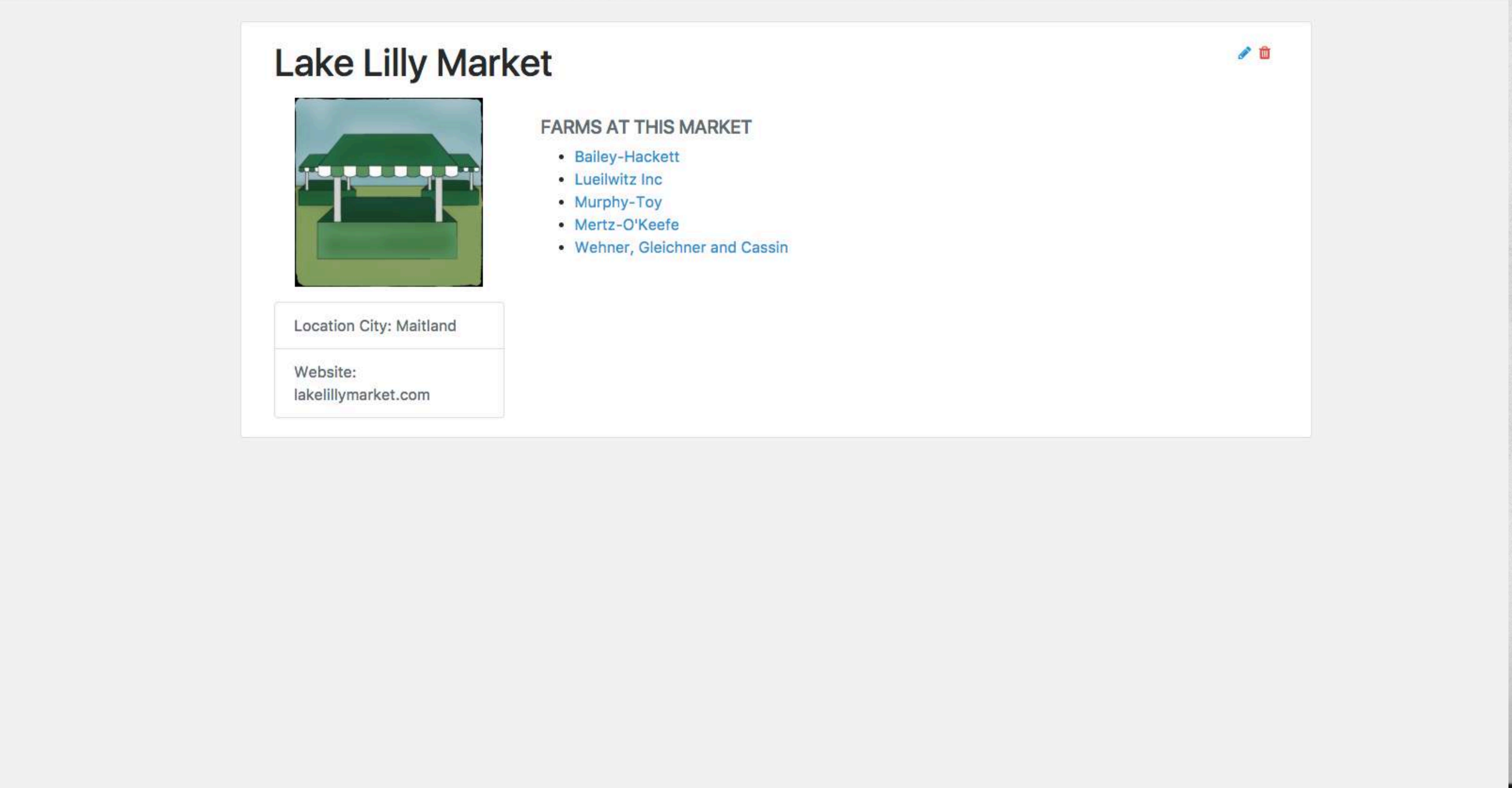
Heidenreich and Sons

Rath-Schneider

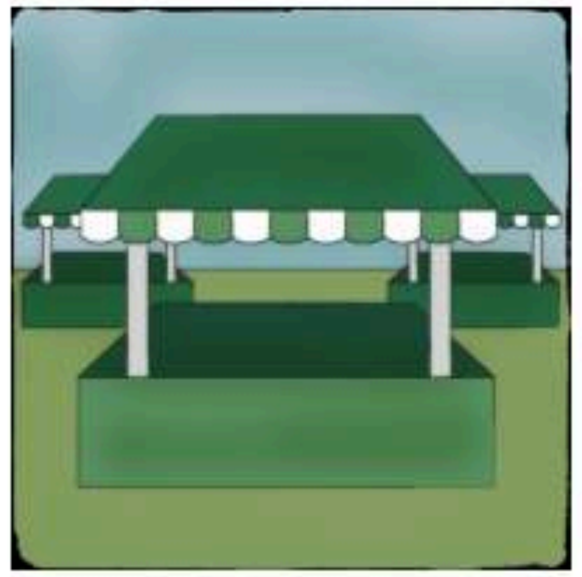
Creating a New Farmers Market



Viewing Each Market's Details



Lake Lilly Market



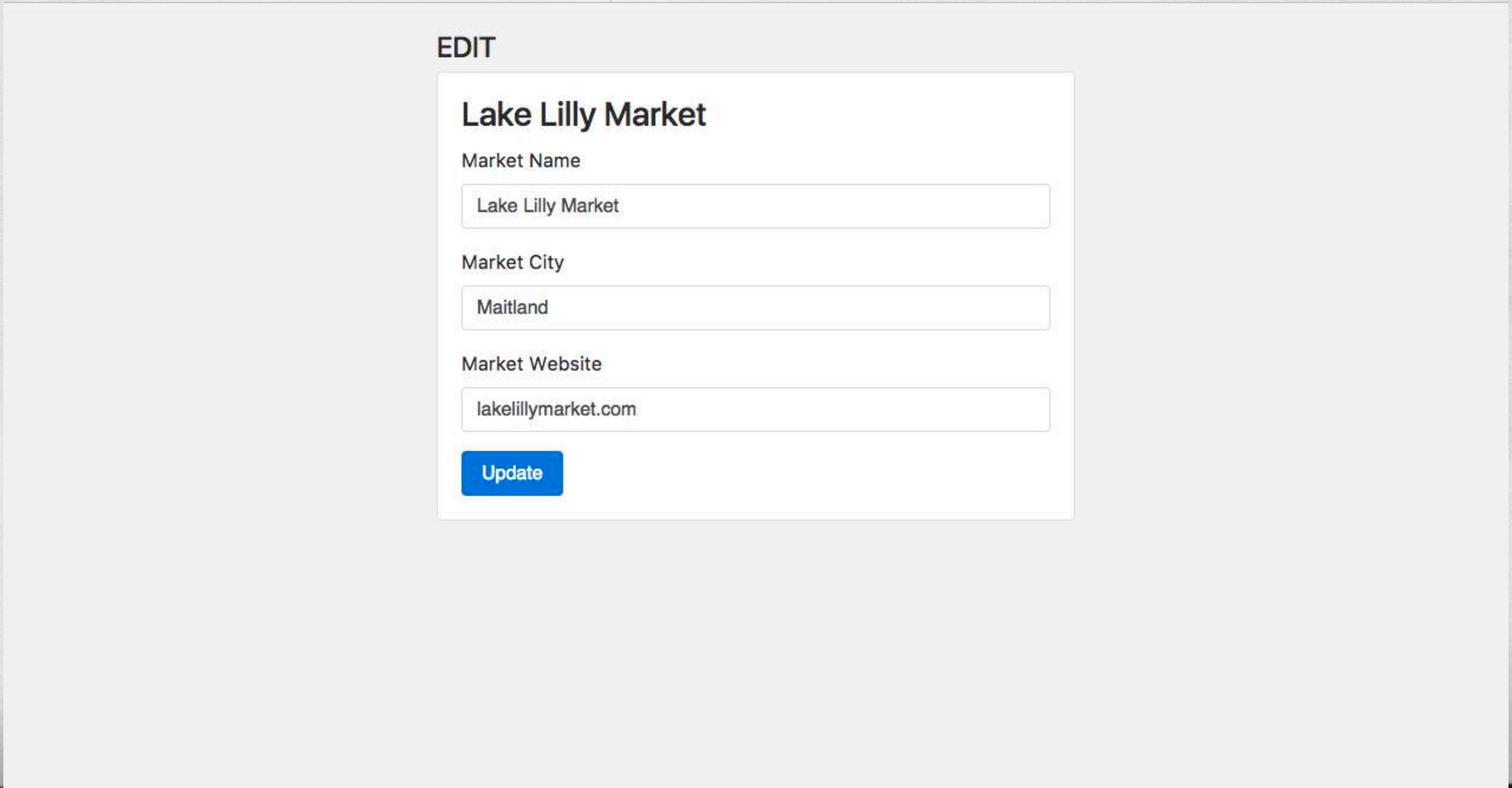
Location City: Maitland

Website: lakelillymarket.com

FARMS AT THIS MARKET

- Bailey-Hackett
- Lueilwitz Inc
- Murphy-Toy
- Mertz-O'Keefe
- Wehner, Gleichner and Cassin

Editing Existing Markets



EDIT

Lake Lilly Market

Market Name

Lake Lilly Market

Market City

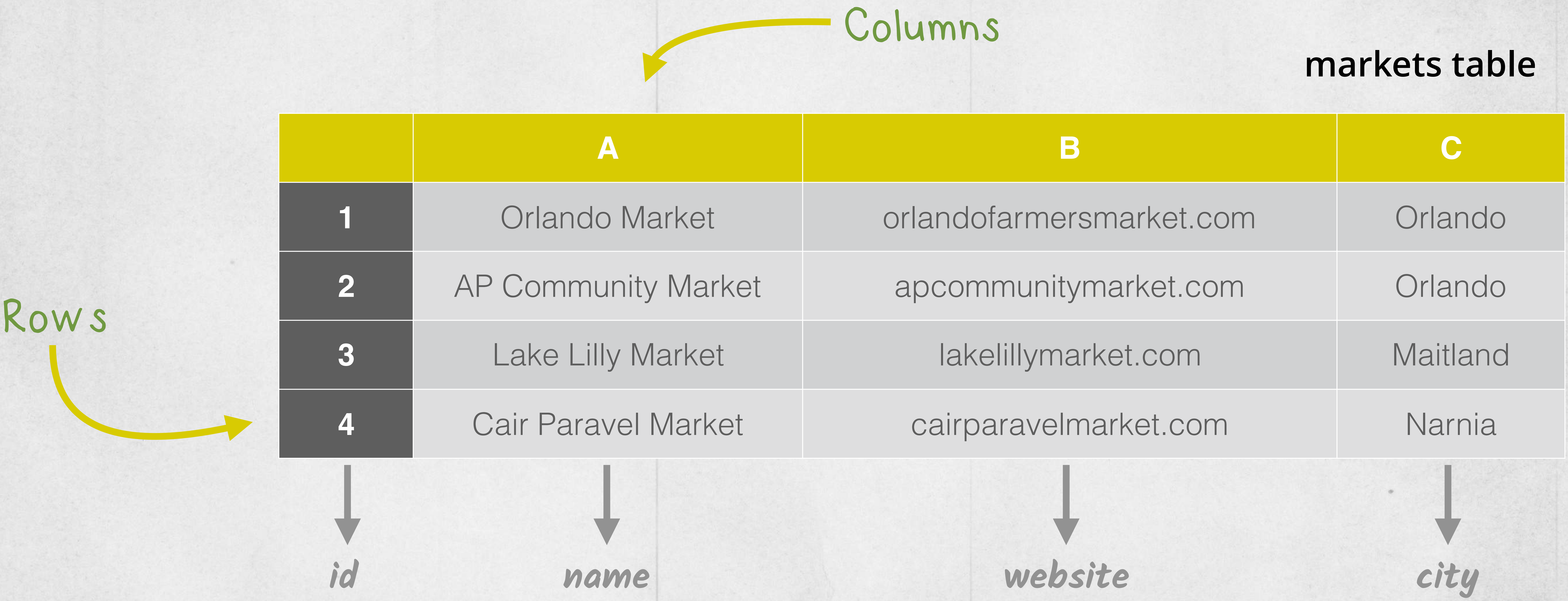
Maitland

Market Website

lakelillymarket.com

Databases, Like a Spreadsheet

Here’s a listing of farmers markets in a spreadsheet form.



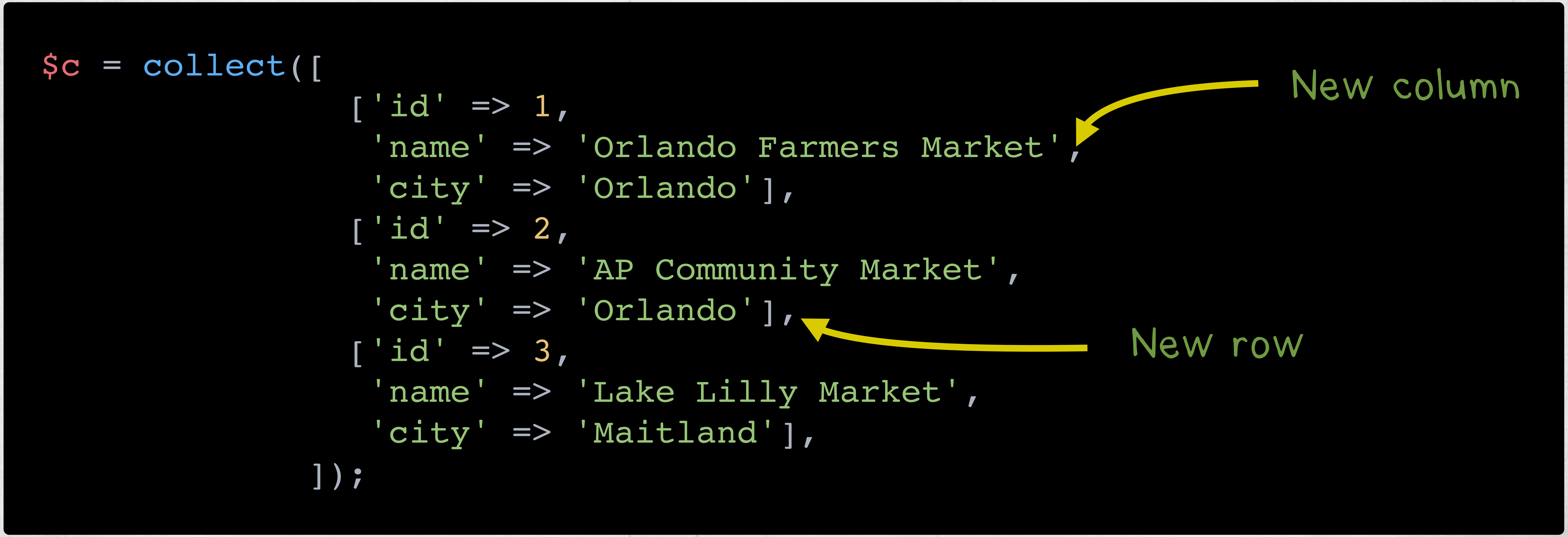
markets table

	A	B	C
1	Orlando Market	orlandofarmersmarket.com	Orlando
2	AP Community Market	apcommunitymarket.com	Orlando
3	Lake Lilly Market	lakelillymarket.com	Maitland
4	Cair Paravel Market	cairparavelmarket.com	Narnia

Collections, Data Organized

The collect **function is used to create a collection from an array.**

```
$c = collect(['Orlando Farmers Market', 'AP Community Market']);
```



```
$c = collect([
               ['id' => 1,
                'name' => 'Orlando Farmers Market',
                'city' => 'Orlando'],
               ['id' => 2,
                'name' => 'AP Community Market',
                'city' => 'Orlando'],
               ['id' => 3,
                'name' => 'Lake Lilly Market',
                'city' => 'Maitland'],
             ]);
```

New column

New row

Accessing Collection Data

There are many useful methods we can use with a collection.

```
$c = collect(['Orlando Farmers Market', 'AP Community Market']);
```

```
$c->first();
```

```
=> "Orlando Farmers Market"
```

```
$c->last();
```

```
=> "AP Community Market"
```

Accessing Larger Collection Data

Many methods allow us to get specific data from each item in the collection.

```
$c = collect([
               ['id' => 1,
                'name' => 'Orlando Farmers Market',
                'city' => 'Orlando'],
               ['id' => 2,
                'name' => 'AP Community Market',
                'city' => 'Orlando'],
               ['id' => 3,
                'name' => 'Lake Lilly Market',
                'city' => 'Maitland'],
             ]);
```

```
$c->fetch('name');
```

=> ["Orlando Farmers Market", "AP Community Market" , "Lake Lilly Market"]

Retrieving a Collection From the Database

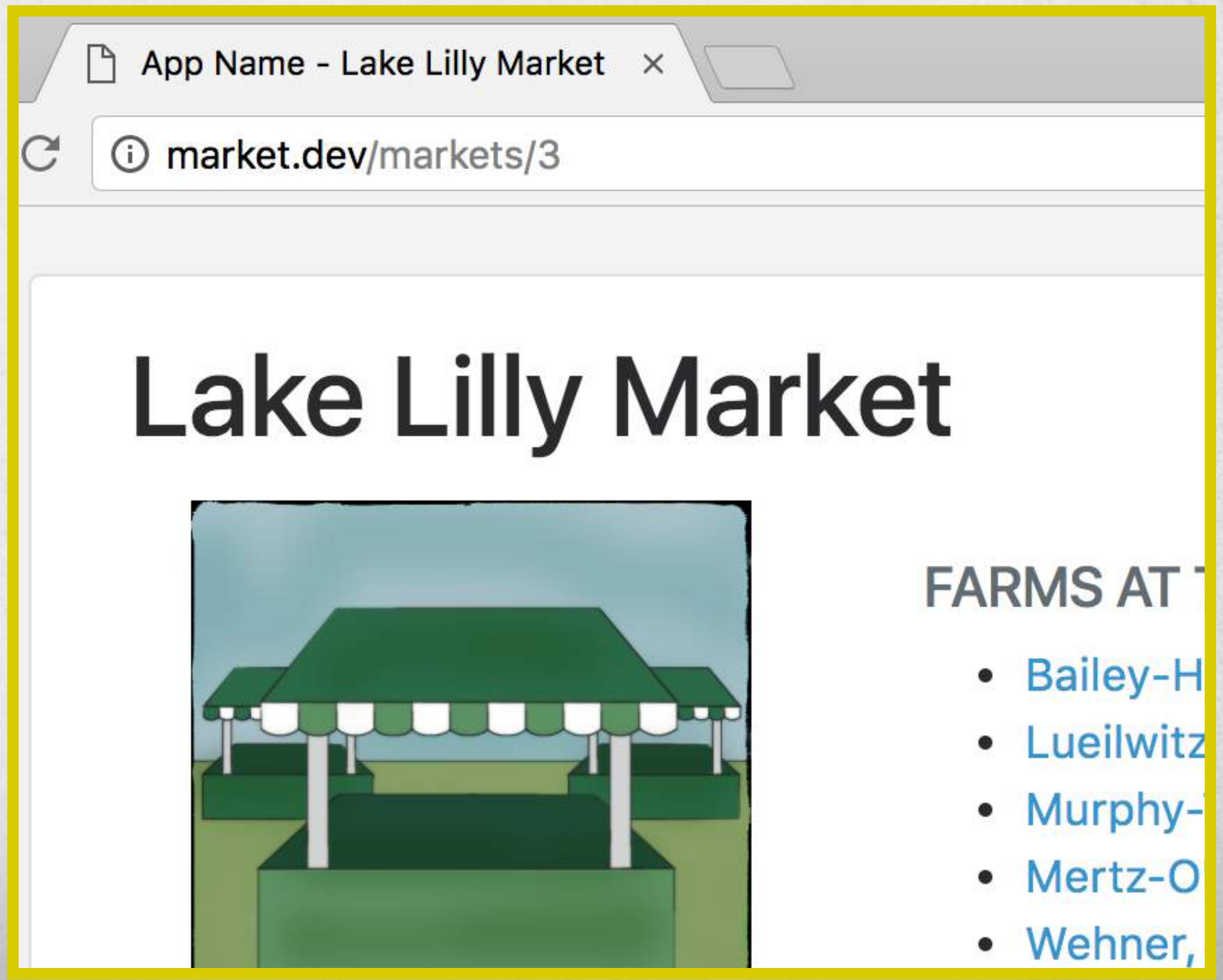
How can we get the market with an id of 3.

markets

id	name	website	city
1	Orlando Market	orlandofarmersmarket.com	Orlando
2	AP Community Market	apcommunitymarket.com	Orlando
3	Lake Lilly Market	lakelillymarket.com	Maitland
4	Cair Paravel Market	cairparavelmarket.com	Narnia

Desired result

```
=> {['id'   => 3,
     'name' => 'Lake Lilly Market',
     'city' => 'Maitland'
  ]}
```



App Name - Lake Lilly Market ×

market.dev/markets/3

Lake Lilly Market

FARMS AT

- Bailey-H
- Lueilwitz
- Murphy-
- Mertz-O
- Wehner,

Using Eloquent to Find in the Database

Accessing collection data through Eloquent with find.

```
$m = Market::find(3);
```

```
=> App\Market{
     'id': 3, 'name': 'Lake Lilly Market','city': 'Maitland'
   }
```

The class name Market links to our table name "markets"

```
$m['name'];
```

```
=> "Lake Lilly Market"
```

```
$m->name;
```

```
=> "Lake Lilly Market"
```

Laravel in Review

What is Laravel?

Laravel’s collection class

Reading from the database with Eloquent

Level 1 – Section 2

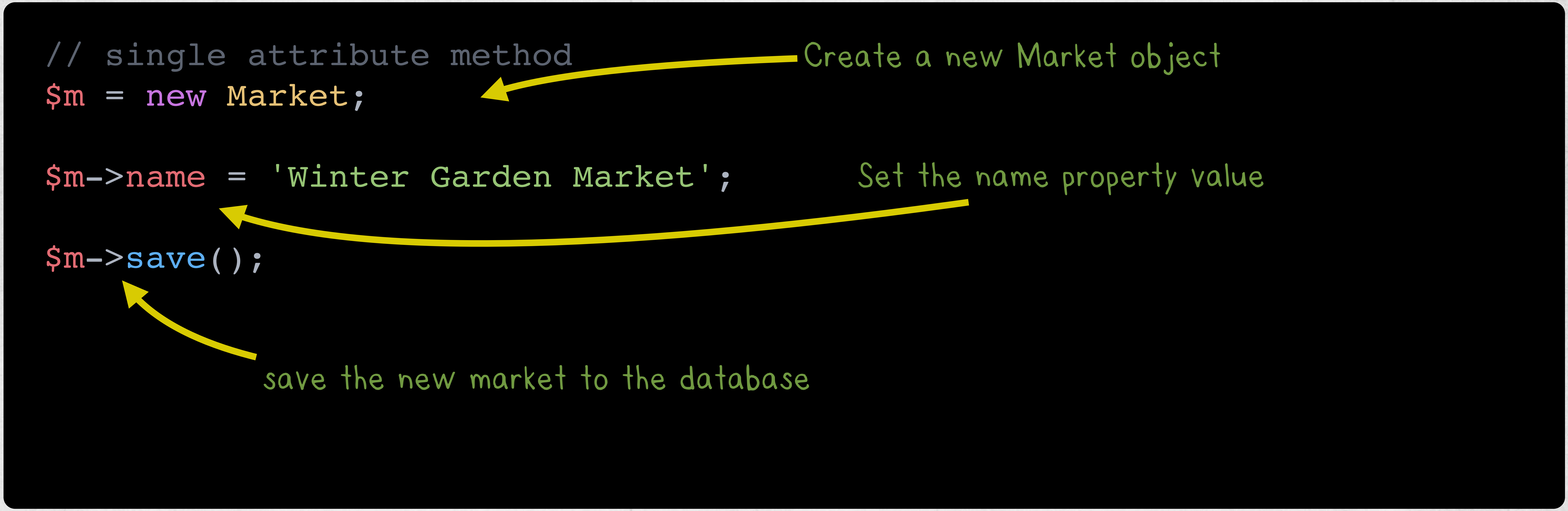
Welcome to Laravel

Cleaning Up CRUD

Creating a Market

Using our model to create a new market

Create a Market



```
// single attribute method
$m = new Market;

$m->name = 'Winter Garden Market';

$m->save();
```

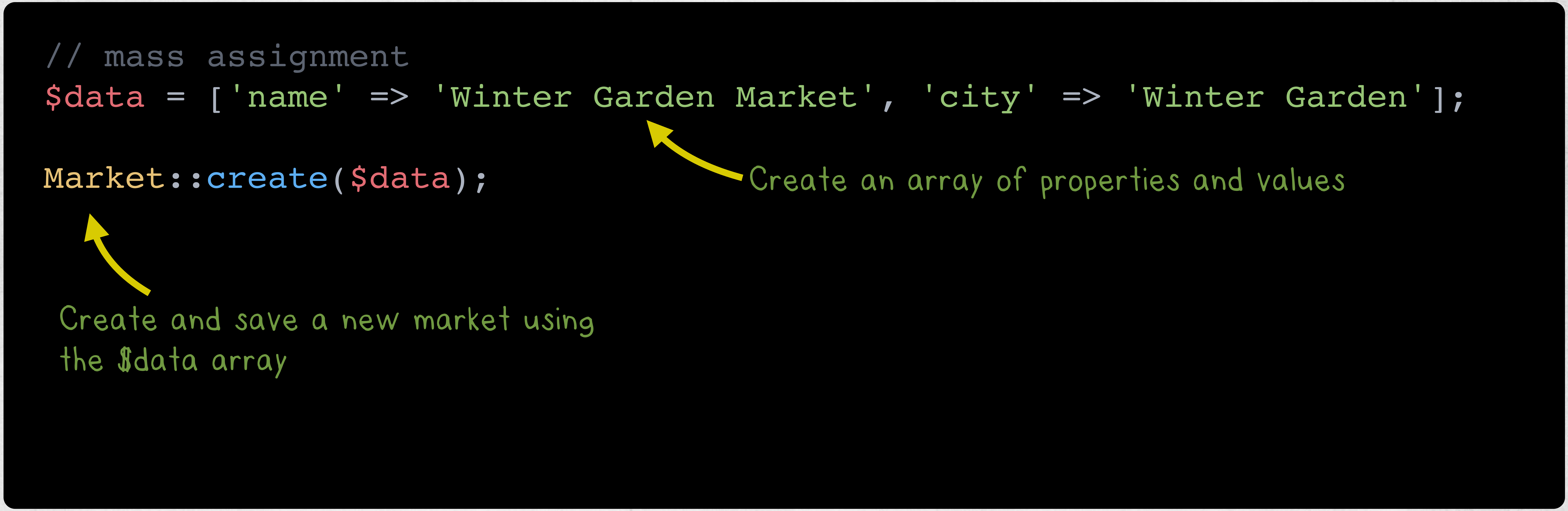
Create a new Market object

Set the name property value

save the new market to the database

Mass Assignment Creation

We can also pass multiple values in an array to create a new item.



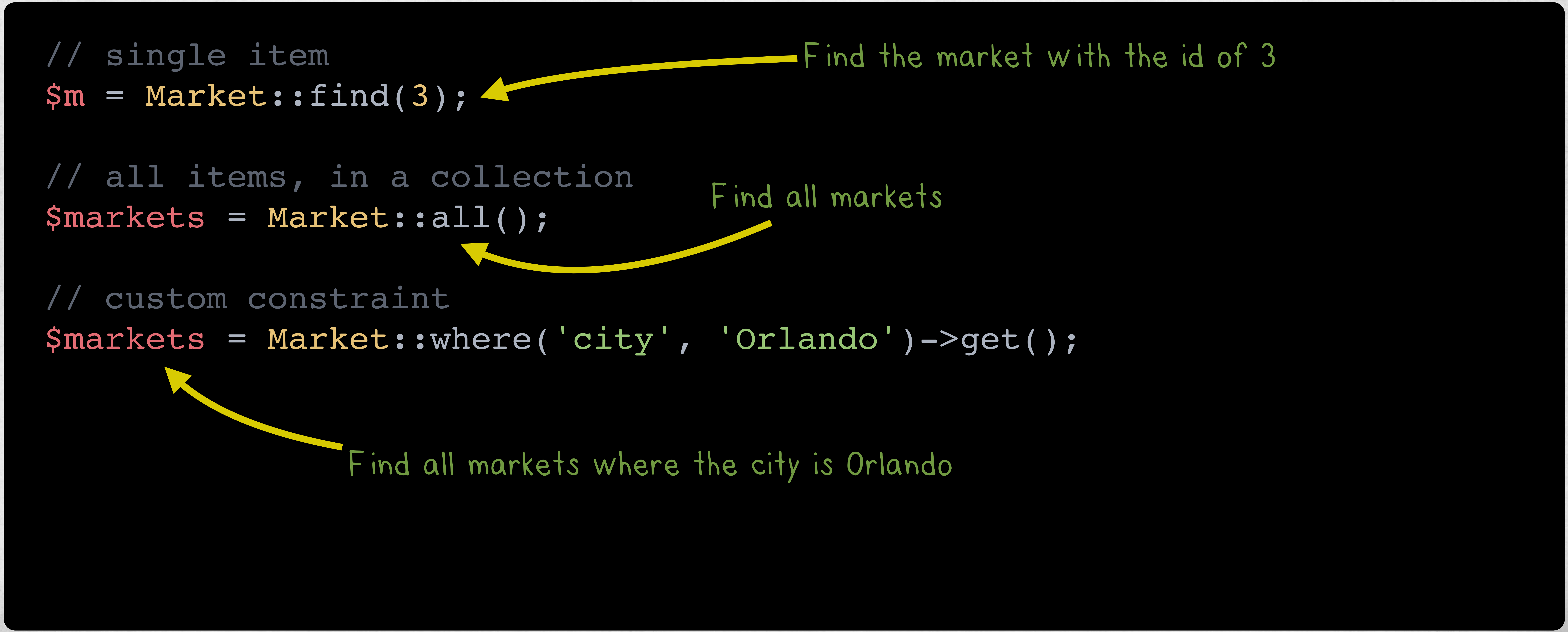
```php
// mass assignment
$data = ['name' => 'Winter Garden Market', 'city' => 'Winter Garden'];

Market::create($data);
```

Create an array of properties and values

Create and save a new market using
the $data array

Reading From a Market

Here’s how we can use our model to read data from a market.



```
// single item
$m = Market::find(3);

// all items, in a collection
$markets = Market::all();

// custom constraint
$markets = Market::where('city', 'Orlando')->get();
```

Find the market with the id of 3

Find all markets

Find all markets where the city is Orlando

Reading Markets With Custom Constraints

Custom methods can be chained together to make a more powerful query.

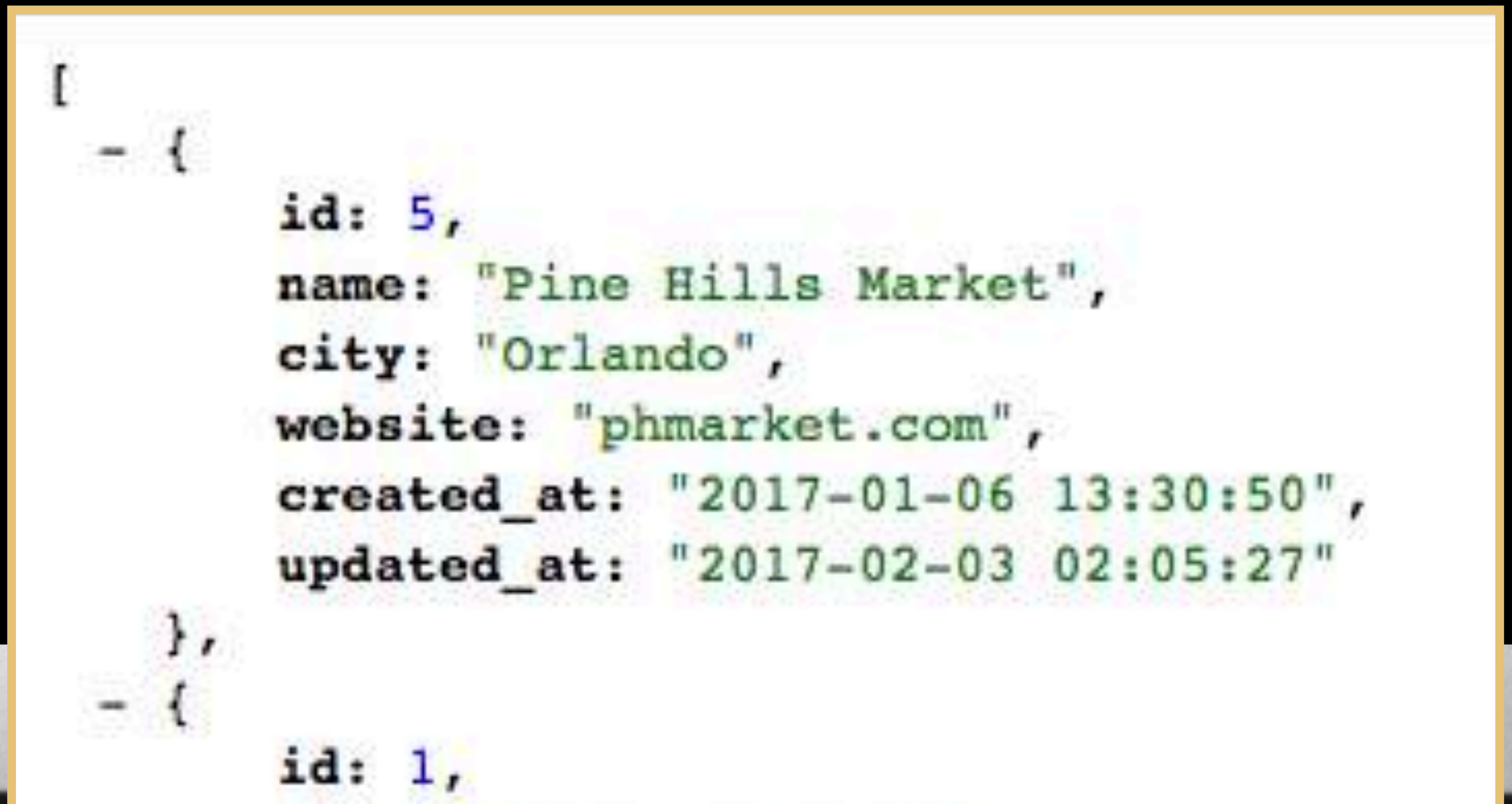
```
// multiple constraints
$markets = Market::where('city', 'Orlando')
                ->orderBy('name', 'desc')
                ->take(5)
                ->get();
```

Find markets in Orlando only

Order by the market name, in descending order

Only the first 5

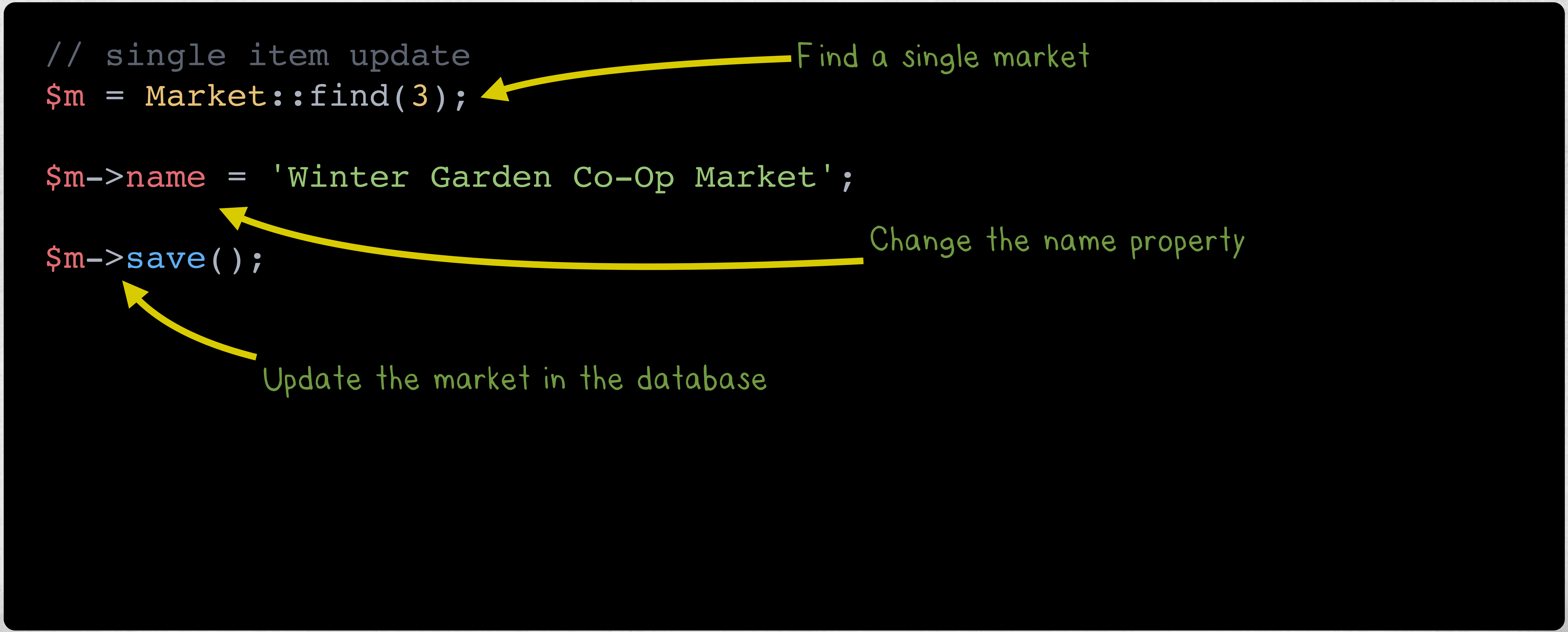
Run the query



```
[
  - {
        id: 5,
        name: "Pine Hills Market",
        city: "Orlando",
        website: "phmarket.com",
        created_at: "2017-01-06 13:30:50",
        updated_at: "2017-02-03 02:05:27"
    },
  - {
        id: 1,
```

Updating a Market

Here’s how we can use our model to update data of a market.



```php
// single item update
$m = Market::find(3);

$m->name = 'Winter Garden Co-Op Market';

$m->save();
```

Find a single market

Change the name property

Update the market in the database

Mass Updates

We can also update multiple attributes at the same time using an array of key-value pairs.

```
// mass update
$m = Market::find(3);

$data = [
          'name' => 'Winter Gar           arket',
          'website' => 'wgcoop.com
        ];

$m->fill($data);
```

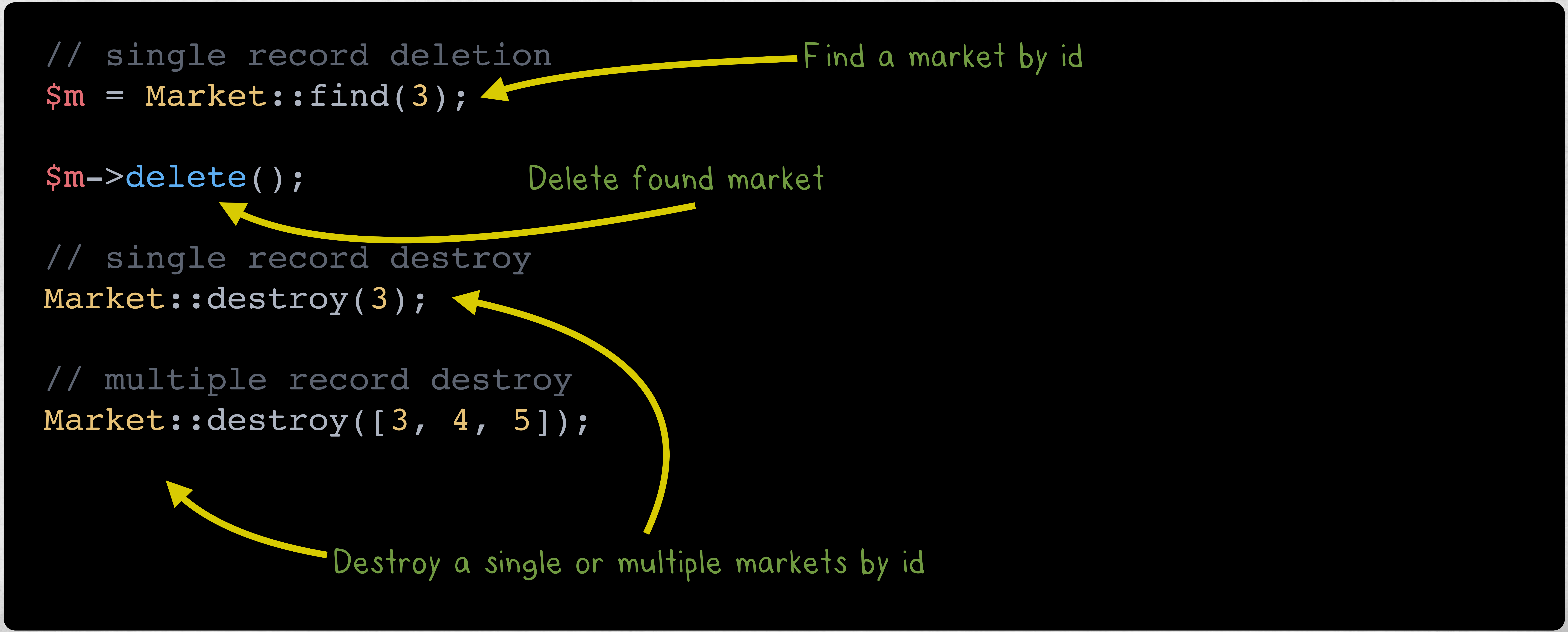
new title

Create an array of values to change

Use fill to update with the new data

Deleting One or More Markets

Here’s how we can use our model to delete a market.



```
// single record deletion
$m = Market::find(3);

$m->delete();

// single record destroy
Market::destroy(3);

// multiple record destroy
Market::destroy([3, 4, 5]);
```

Find a market by id

Delete found market

Destroy a single or multiple markets by id

The Power of Eloquent

Using our model to perform CRUD actions in Eloquent is simple.

Create a Market

```
$m = new Market;

$m->name = 'Winter Garden Market';

$m->save();
```

Read from a Market

```
$m = Market::find(3);

echo $m->name;

=> 'Lake Lilly Market'
```

Update a Market

```
$m = Market::find(3);

$m->name = 'Maitland Market';

$m->save();
```

Delete a Market

```
$m = Market::find(3);

$m->delete();
```

Laravel in Review

Laravel CRUD methods

Mass assignment for create and update