Awesome Tools 2017

geemba

Noel De Martin, CTO

Traditional landscape

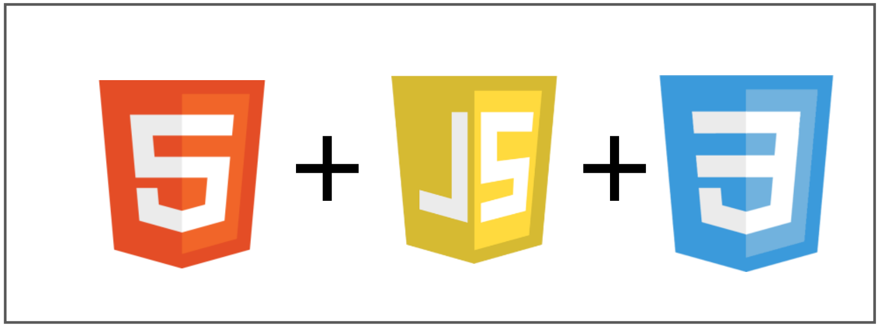
Backend (server)

PHP + MySQL

Mobile Apps

Android → Java / Kotlin
iOS → Swift / ObjectiveC

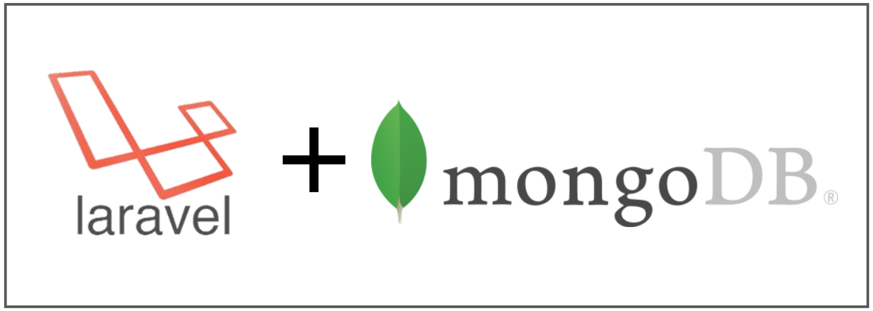
Web Apps



HTML + JavaScript + CSS

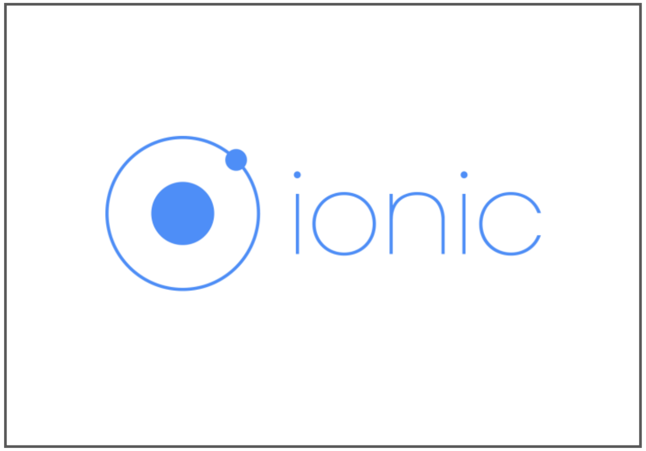
Modern landscape

Backend (server)



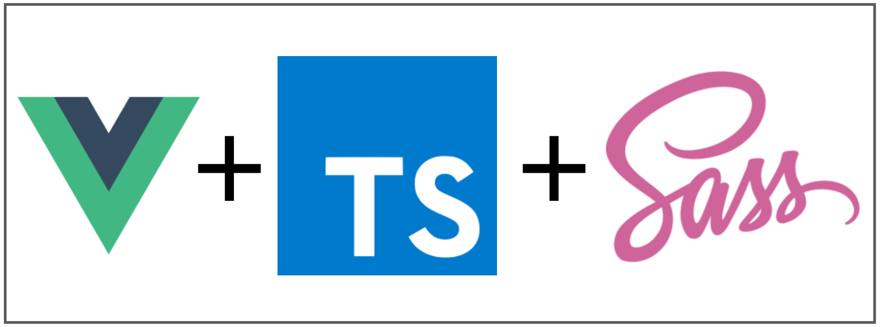
Laravel + MongoDB

Mobile Apps



Ionic

Web Apps



Vue + TypeScript + Sass

mongoDB®

Looks familiar?

```
SELECT associations2.object_id, associations2.term_id, associations2.cat_ID, associations2
FROM (SELECT objects_tags.object_id, objects_tags.term_id, wp_cb_tags2cats.cat_ID, categor
    FROM (SELECT wp_term_relationships.object_id, wp_term_taxonomy.term_id, wp_term_taxono
        FROM wp_term_relationships
        LEFT JOIN wp_term_taxonomy ON wp_term_relationships.term_taxonomy_id = wp_term_tax
        ORDER BY object_id ASC, term_id ASC)
        AS objects_tags
    LEFT JOIN wp_cb_tags2cats ON objects_tags.term_id = wp_cb_tags2cats.tag_ID
    LEFT JOIN (SELECT wp_term_relationships.object_id, wp_term_taxonomy.term_id as cat_ID,
        FROM wp_term_relationships
        LEFT JOIN wp_term_taxonomy ON wp_term_relationships.term_taxonomy_id = wp_term_tax
        WHERE wp_term_taxonomy.taxonomy = 'category'
        GROUP BY object_id, cat_ID, term_taxonomy_id
        ORDER BY object_id, cat_ID, term_taxonomy_id)
        AS categories on wp_cb_tags2cats.cat_ID = categories.term_id
    WHERE objects_tags.term_id = wp_cb_tags2cats.tag_ID
    GROUP BY object_id, term_id, cat_ID, term_taxonomy_id
    ORDER BY object_id ASC, term_id ASC, cat_ID ASC)
    AS associations2
LEFT JOIN categories ON associations2.object_id = categories.object_id
WHERE associations2.cat_ID <> categories.cat_ID
GROUP BY object_id, term_id, cat_ID, term_taxonomy_id
ORDER BY object_id, term_id, cat_ID, term_taxonomy_id
```

Source: How to reduce a long SQL query based on CREATE VIEW? (Stackoverflow)

Looks familiar?

```
SELECT associations2.ob              iations2        ciations2.cat_ID, associations2
FROM (SELECT objects          ct_id, objects_tags.t        cb_tags2cats.cat_ID, categor
    FROM (SELECT w        elationships.object_id, wp_t          my.term_id, wp_term_taxon
        FROM wp_        ationships
        LEFT JO       erm_taxonomy ON wp_term_relat       ps.       xonomy_id = wp_term_tax
        ORDER       ect_id ASC, term_id ASC)
        AS ob      tags
    LEFT JOI      b_tags2cats ON objects_tags        d = wp_cb_ta       s.tag_ID
    LEFT JO      LECT wp_term_relationships       _id, wp_term_t        .term_id as cat_ID,
        FR      erm_relationships
        LE      N wp_term_taxonomy ON wp       relationships.term_        my_id = wp_term_tax
        WH      _term_taxonomy.taxonom       ategory'
        GR      object_id, cat_ID,       axonomy_id
        OR      object_id, cat_ID       _taxonomy_id)
        AS      ries on wp_cb_ta       s.cat_ID = categories.ter
    WHERE o      tags.term_id       b_tags2cats.tag_ID
    GROUP BY      t_id, term       _ID, term_taxonomy_id
    ORDER BY      id ASC,       d ASC, cat_ID ASC)
    AS associa
LEFT JOIN categ       ations2.object_id = categorie       t_id
WHERE association       > categories.cat_ID
GROUP BY object_id,       cat_ID, term_taxonomy_id
ORDER BY object_id, t       ID, term_taxonom
```

Source: How to reduce a long SQL query based on CREATE VIEW? (Stackoverflow)

Looks familiar?

id	name	email	data
1	Ragnar Lodbrok	lodbrok@mail.com	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}
2	Björn Ironside	bironside@mail.com	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}
3	Lagertha	lagertha@mail.com	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}

Looks familiar?

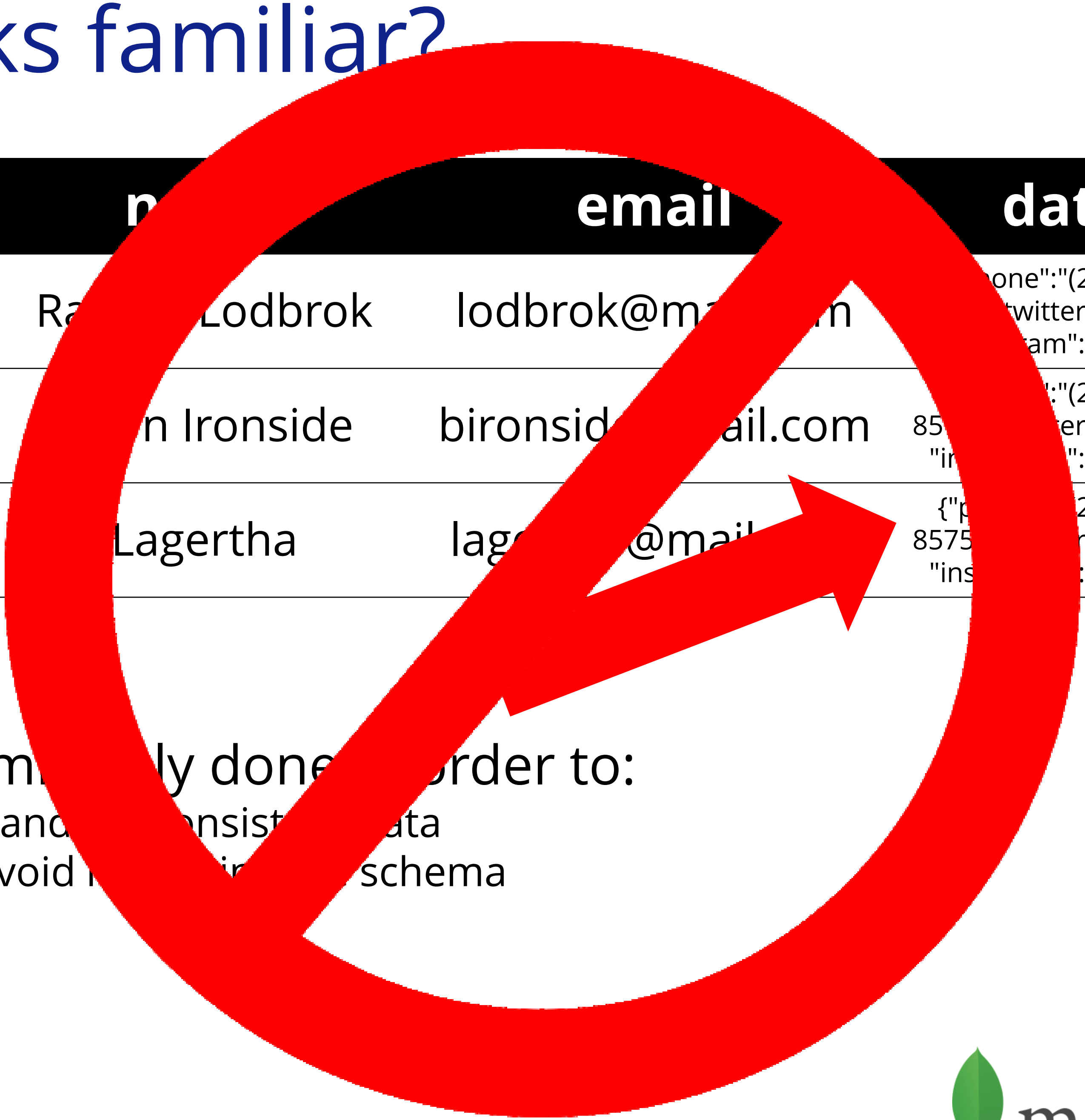
id	name	email	data
1	Ragnar Lodbrok	lodbrok@mail.com	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}
2	Björn Ironside	bironside@mail.com	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}
3	Lagertha	lagertha@mail	{"phone":"(297)-707-8575","twitter":"foobar", "instagram":"foobar"}

Commonly done in order to:

- Handle inconsistent data
- Avoid modifying the schema
- ...

Looks familiar?

id	n	email	data
1	Ra Lodbrok	lodbrok@ma n	one":"(297)-707- twitter":"foobar", am":"foobar"}
2	n Ironside	bironsid ail.com	":"(297)-707- 85 er":"foobar", "i ":"foobar"}
3	Lagertha	lag @mail	{"p 297)-707- 8575 r":"foobar", "ins :"foobar"}



Com ly done rder to:

- Hand nsist ata
- Avoid i schema
- ...

MongoDB to the rescue!!

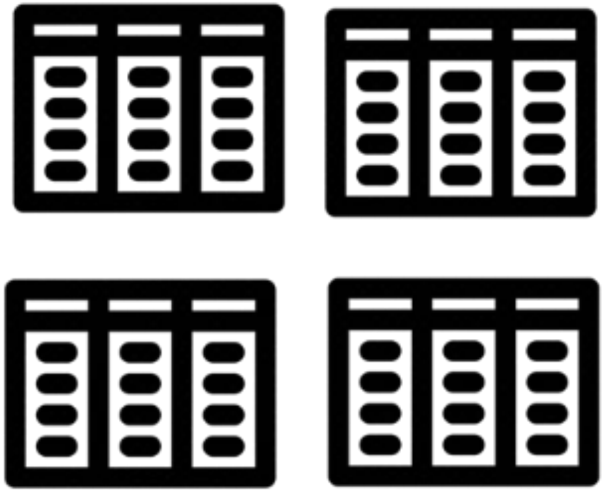
SQL vs NoSQL

SQL	NoSQL
Table	Collection
Row	Document
Column	Field

SQL vs NoSQL

SQL	NoSQL
Table	Collection
Row	Document
Column	Field

TABLES



COLLECTIONS

SQL vs NoSQL

SQL	NoSQL
Table	Collection
Row	Document
Column	Field

ROWS

id	name	age	phone_number
1	Chandler Bing	10	727-647-4415
2	Rachel Green	24	757-328-3857

DOCUMENTS

```
{
    _id: ObjectId(“507f19…”)
    name: “Chandler Bing”,
    age: 10,
    phone_number: “727-647-4415”
}

{
    _id: ObjectId(“2a2ed9…”)
    name: “Rachel Green”,
    age: 24,
    phone_number: “757-328-3857”
}
```

mongoDB®

SQL vs NoSQL

SQL	NoSQL
Table	Collection
Row	Document
Column	Field

COLUMNS

id	name	age	phone_number	twitter
1	Chandler Bing	10	727-647-4415	ChandleringBing
2	Rachel Green	24	757-328-3857	NULL

FIELDS

```
{
    _id: ObjectId(“507f19…”)
    name: “Chandler Bing”,
    age: 10,
    phone_number: “727-647-4415”,
    twitter: “ChandleringBing”
}

{
    _id: ObjectId(“2a2ed9…”)
    name: “Rachel Green”,
    age: 24,
    phone_number: “757-328-3857”
}
```

SQL vs NoSQL

SQL	NoSQL
Table	Collection
Row	Document
Column	Field

COLUMNS

id	name	age	phone_number	twitter
1	Chandler Bing	10	727-647-4415	ChandleringBing
2	Rachel Green	24	757-328-3857	NULL

FIELDS

```
{
    _id: ObjectId(“507f19…”)
    name: “Chandler Bing”,
    age: 10,
    contact: {
        phone_number: “727-647-4415”,
        twitter: “ChandleringBing”
    }
}

{
    _id: ObjectId(“2a2ed9…”)
    name: “Rachel Green”,
    age: 24,
    contact: {
        phone_number: “757-328-3857”
    }
}
```

mongoDB®

Design of a NoSQL database

THREADS

```
{
    _id: ObjectId(“507f1f77…”),
    author_id: ObjectId(“c19729d…”),
    title: “Weather condition”,
    body: “How’s the weather?”
}
```

REPLIES

```
{
    _id: ObjectId(“f77a8w…”),
    thread_id: ObjectId(“507f1f77…”),
    author_id: ObjectId(“29de86…”),
    text: “I think today is sunny”
}

{
    _id: ObjectId(“2f98a0…”),
    thread_id: ObjectId(“507f1f77…”),
    author_id: ObjectId(“c19729d…”),
    text: “Me too”
}
```

USERS

```
{
    _id: ObjectId(“29de86…”),
    name: “John Doe”,
    avatar_url: “http://...”,
    email: “thedoe@mail.com”
}

{
    _id: ObjectId(“c19729d…”),
    name: “Jane Smith”,
    avatar_url: “http://...”,
    email: “iamjane@mail.com”
}
```

mongoDB®

Design of a NoSQL database

Embedding

THREADS

```
{
    _id: ObjectId(“507f1f77…”),
    author_id: ObjectId(“c19729d…”),
    title: “Weather condition”,
    body: “How’s the weather?”,
    replies: [
        {
            author_id: ObjectId(“29de86…”),
            text: “I think today is sunny”
        },
        {
            author_id: ObjectId(“c19729d…”),
            text: “Me too”
        }
    ]
}
```

USERS

```
{
    _id: ObjectId(“29de86…”),
    name: “John Doe”,
    avatar_url: “http://...”,
    email: “thedoe@mail.com”
}

{
    _id: ObjectId(“c19729d…”),
    name: “Jane Smith”,
    avatar_url: “http://...”,
    email: “iamjane@mail.com”
}
```

Design of a NoSQL database

Denormalization

THREADS

```
{
    _id: ObjectId(“507f1f77…”),
    author: {
        id: ObjectId(“c19729d…”),
        name: “Jane Smith”,
        avatar_url: “http://...”
    },
    title: “Weather condition”,
    body: “How’s the weather?”,
    replies: [
        {
            author: {
                id: ObjectId(“29de86…”),
                name: “John Doe”,
                avatar_url: “http://...”
            },
            text: “I think today is sunny”
        },
        …
    ]
}
```

USERS

```
{
    _id: ObjectId(“29de86…”),
    name: “John Doe”,
    avatar_url: “http://...”,
    email: “thedoe@mail.com”
}

{
    _id: ObjectId(“c19729d…”),
    name: “Jane Smith”,
    avatar_url: “http://...”,
    email: “iamjane@mail.com”
}
```

Design of a NoSQL database

“When designing data models, always consider the application usage of the data as well as the inherent structure of the data itself.”

- MongoDB manual

Design of a NoSQL database

“When designing data models, always consider the application usage of the data as well as the inherent structure of the data itself.”

- MongoDB manual

- Duplication can be the right choice.
- Favour few collections over many.
- Data doesn’t have to be structured.

mongoDB®

Advantages

- Flexible schema
- Speed
- Scaling

Advantages

- Flexible schema
- Speed
- Scaling

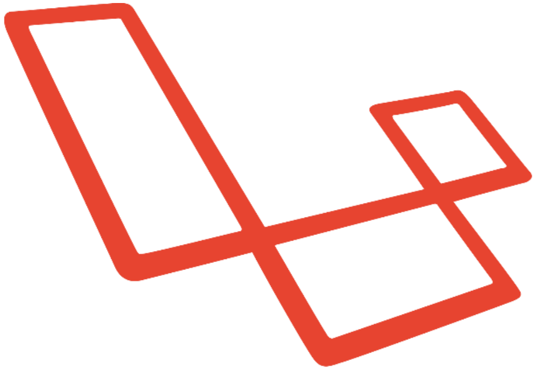
Disadvantages

- Flexible schema
- Memory in disk
- Data integrity (application-side validation)

mongoDB®

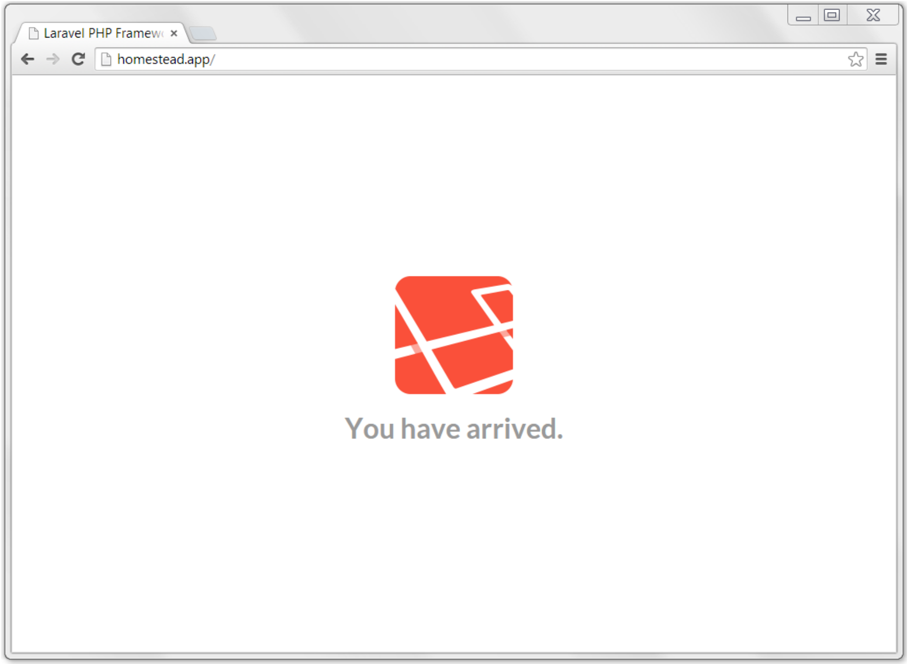
Learn more

- Official Website
- Recommended reads:
  - Thinking in Documents
  - 6 Rules of Thumb for MongoDB Schema Design
- Free online course: Data Wrangling with MongoDB
- MongoDB Source Code



Laravel

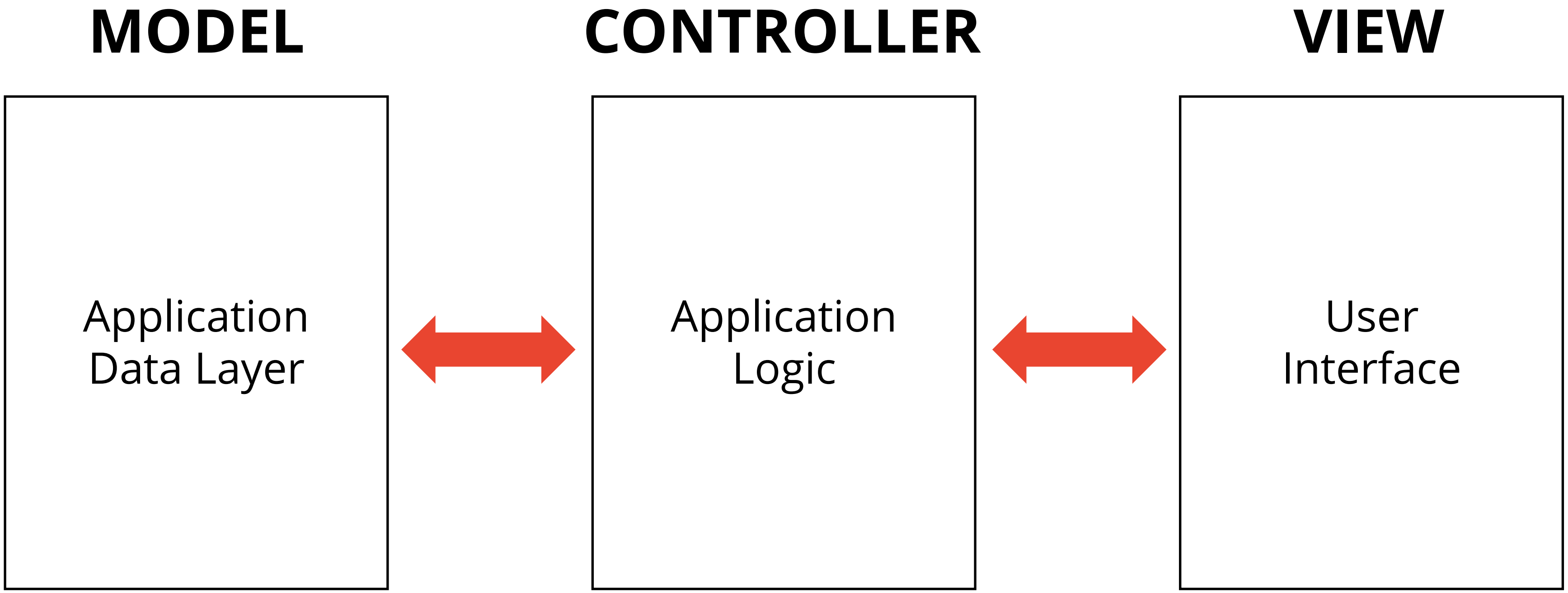
Welcome to Laravel



You have arrived.

PHP framework using Model-View-Controller Design Pattern

Model View Controller



MODEL

Application
Data Layer

CONTROLLER

Application
Logic

VIEW

User
Interface

Laravel

Model – Eloquent ORM

Implementation of Active Record pattern

DATABASE (SQL)

id	name	email
1	Bob Kane	kane@mail.com
2	Jerry Robinson	robinson@mail.com
3	Bill Finger	finger@mail.com

CODE (PHP)

```php
$user = User::find(2);
$user->name;
$user->email;

$user->update([
    'email' => 'new@mail.com'
]);

$user->delete();

User::create([
    'name' => 'Bill Finger',
    'email' => 'finger@mail.com'
]);
```

Laravel

Model – Eloquent ORM

Implementation of [Active Record]() pattern

DATABASE (NoSQL)

```
{
    _id: ObjectId(“d294b8…”),
    name: “Bob Kane”,
    email: “kane@mail.com”
}

{
    _id: ObjectId(“f4c7b2…”),
    name: “Jerry Robinson”,
    email: “robinson@mail.com”
}

{
    _id: ObjectId(“1fb215…”),
    name: “Bill Finger”,
    email: “finger@mail.com”
}
```

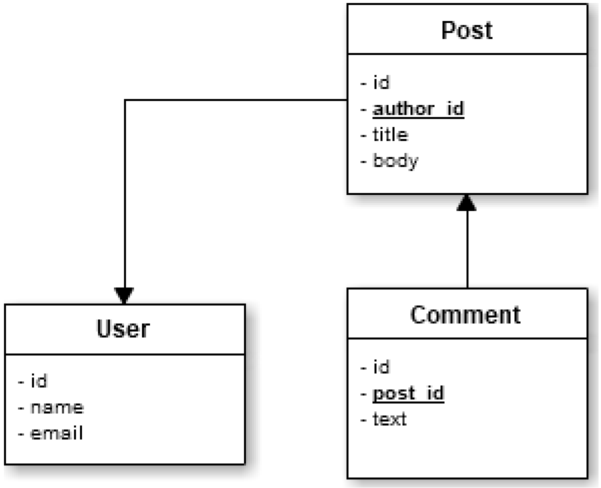
CODE (PHP)

```php
$user = User::find('f4c7b2…');
$user->name;
$user->email;

$user->update([
    'email' => 'new@mail.com'
]);

$user->delete();

User::create([
    'name' => 'Bill Finger',
    'email' => 'finger@mail.com'
]);
```

Model – Eloquent ORM

Convention over configuration

DATABASE



CODE

```php
<?php

use Illuminate\Database\Eloquent\Model;

class Post extends Model {

    public function author() {
        return $this->belongsTo(User::class);
    }

    public function comments() {
        return $this->hasMany(Comment::class);
    }

}
```

Model – Query Builder

COUNT

```
$usersCount = DB::table('users')->count();
```

GROUP BY

```
$usersByStatus = DB::table('users')
                   ->select(DB::raw('count(*) as user_count, status'))
                   ->where('status', '<>', 1)
                   ->groupBy('status')
                   ->get();
```

JOIN

```
$users = DB::table('users')
           ->join('contacts', 'users.id', '=', 'contacts.user_id')
           ->join('orders', 'users.id', '=', 'orders.user_id')
           ->select('users.*', 'contacts.phone', 'orders.price')
           ->get();
```

ETC...

Controller - Routing

GET mywebsite.com/hello

```
Route::get('hello', function () {
    return 'Hello World';
});
```

Controller - Routing

GET mywebsite.com/user/3

```
Route::get('user/{id}', 'UsersController@show');
```

```
class UsersController {

    public function show($id) {
        return User::find($id);
    }

}
```

Controller - Routing

POST mywebsite.com/login

```
Route::post('login', 'HomeController@login');
```

```
class UsersController {

    public function login() {
        $credentials = [
            'email' => request('email'),
            'password' => request('password')
        ];
        if (Auth::attempt($credentials))
            return redirect('/home');
    }

}
```

View - Blade

Laravel Blade can be thought as a **superset** of php when rendering views.

```
@if (auth()->check())

    {!! Form::open(['url' => '/login']) !!}
        {!! Form::text('email'); !!}
        {!! Form::password('password'); !!}
        {!! Form::submit('Login'); !!}
    {!! Form::close() !!}

@else

    <div class="container">
        Hello, {{ auth()->user()->name }}.
    </div>

@endif
```

If the user isn't logged in

```
<form action="/login">
    <input type="text" name="email">
    <input type="password" name="password">
    <input type="submit" value="Login">
</form>
```

If the user is logged in

```
<div class="container">
    Hello, Bill Finger.
</div>
```

Laravel

Out of the box with Laravel

- Authentication
- Mailing
- Testing
- Form Validation
- Command Line Tools
- Sessions
- Database Migrations
- Task Scheduling
- And more...

We just
got started...

Learn more

- Official Website
- Laracasts (video tutorials)
- MongoDB integration with Laravel
- Laravel Source Code

TypeScript

KEEP
CALM
AND
HATE
JAVASCRIPT

TypeScript
to the rescue!!

What can Typescript do for you?

- Javascript **superset**.
- Compiles into Javascript.
  - Compatible in all browsers.
- Provides the following:
  - Classes
  - Interfaces
  - Typings
  - File imports
  - …

Example

JAVASCRIPT

```
function createCar() {
    return {
        run: function() {
            console.log('I am a running car!');
        }
    };
}

var car = createCar();
car.run();
```

TYPESCRIPT

```
class Car{

    public run() {
        console.log('I am a running car!');
    }

}

let car = new Car();
car.run();
```

Example

TYPESCRIPT

```
abstract class Vehicle {

    abstract public run(): void;

}

class Car extends Vehicle {

    public run() {
        console.log('I am a running car!');
    }

}

let car = new Car();
car.run();
```

Example

Vehicle.ts

```
export default abstract class Vehicle {

    abstract public run(): void;

}
```

Car.ts

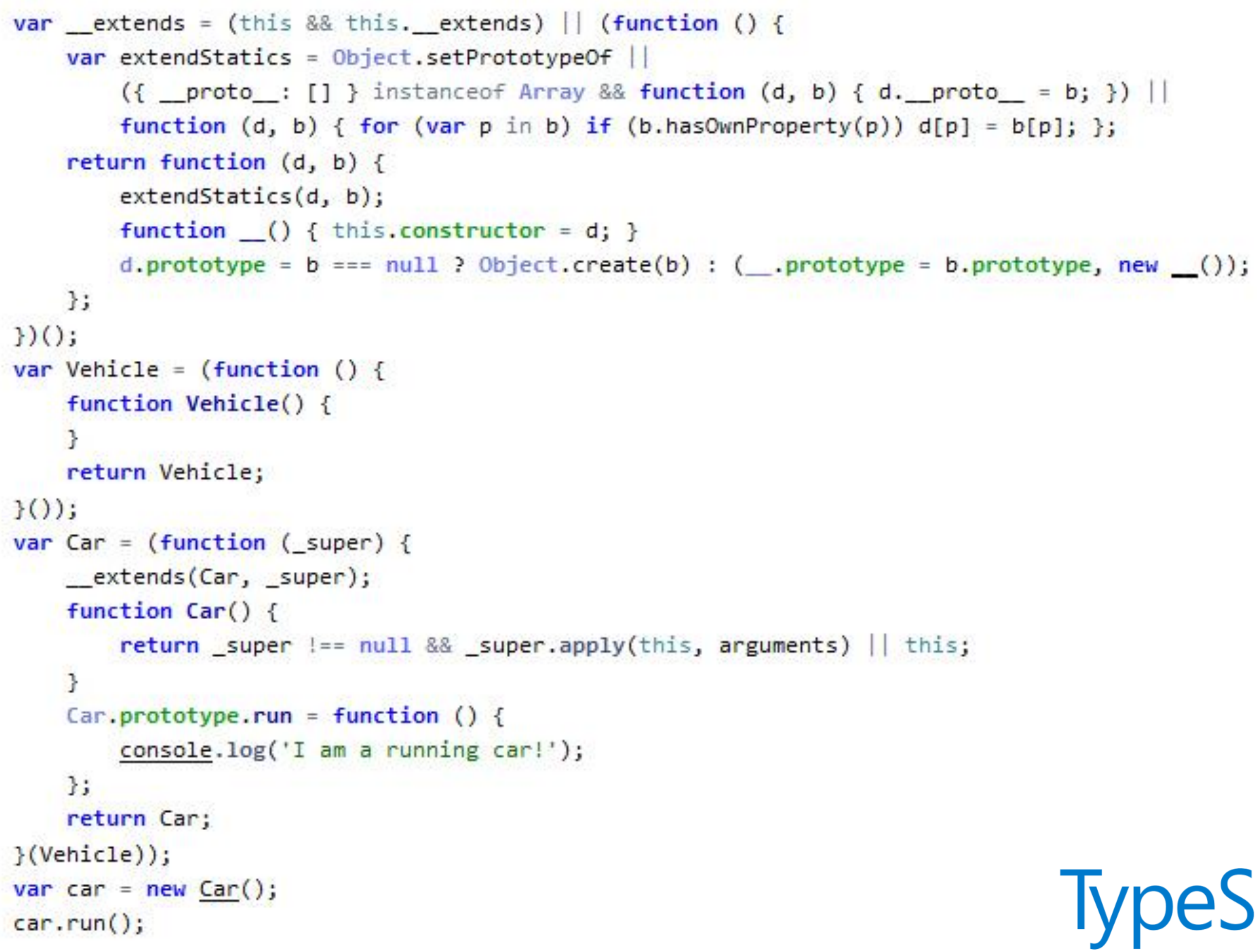
```
import Vehicle from './Vehicle.ts';

export default class Car extends Vehicle {

    public run() {
        console.log('I am a running car!');
    }

}
```

index.ts

```
import Car from './Car';

let car = new Car();
car.run();
```

Example

index.js



```
var __extends = (this && this.__extends) || (function () {
    var extendStatics = Object.setPrototypeOf ||
        ({ __proto__: [] } instanceof Array && function (d, b) { d.__proto__ = b; }) ||
        function (d, b) { for (var p in b) if (b.hasOwnProperty(p)) d[p] = b[p]; };
    return function (d, b) {
        extendStatics(d, b);
        function __() { this.constructor = d; }
        d.prototype = b === null ? Object.create(b) : (__.prototype = b.prototype, new __());
    };
})();
var Vehicle = (function () {
    function Vehicle() {
    }
    return Vehicle;
}());
var Car = (function (_super) {
    __extends(Car, _super);
    function Car() {
        return _super !== null && _super.apply(this, arguments) || this;
    }
    Car.prototype.run = function () {
        console.log('I am a running car!');
    };
    return Car;
}(Vehicle));
var car = new Car();
car.run();
```

TypeScript

Made by

TypeScript

Learn more

- Official Website
- Recommended Text Editor: Visual Studio Code
- TypeScript Source Code

Vue.js

jQuery

~~jQuery~~ Too basic, no structure

ANGULARJS

by Google

~~jQuery~~

Too basic, no structure

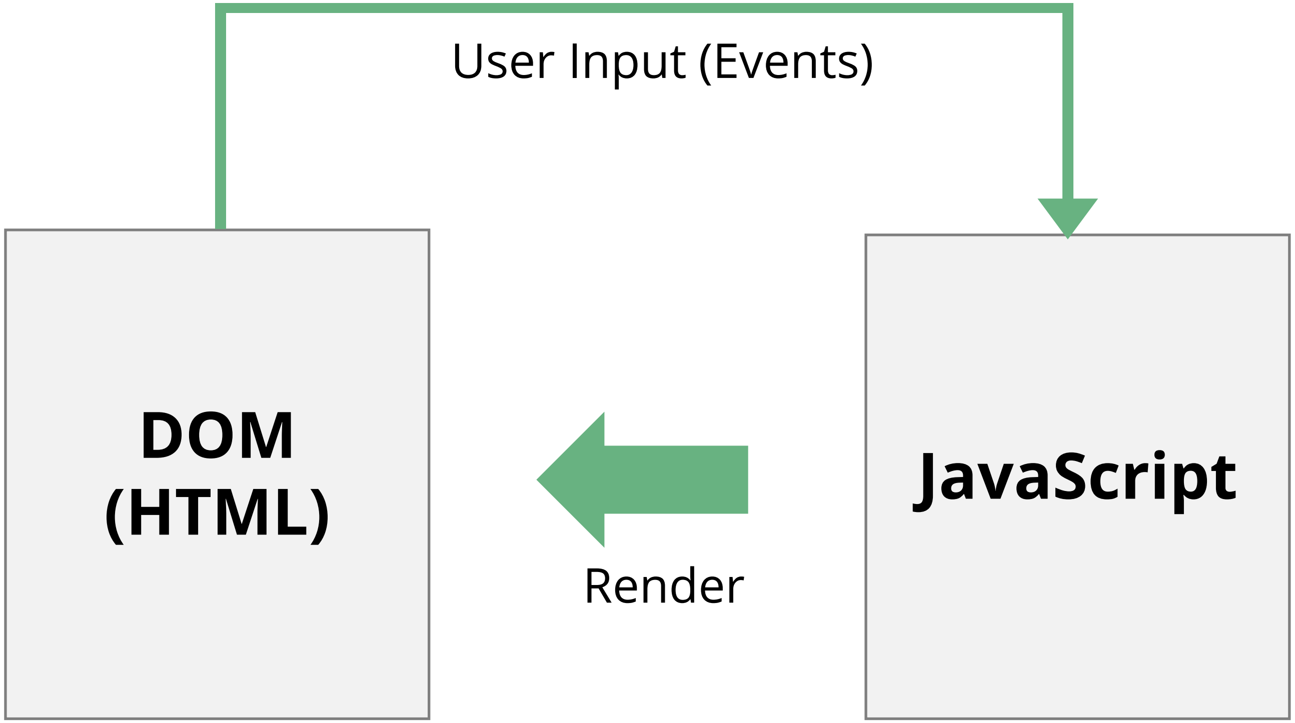
~~ANGULARJS~~
by Google

Too complex,
steep learning curve

Vue.js

✓Structured
✓Incremental
✓Awesome

Vue approach to UI

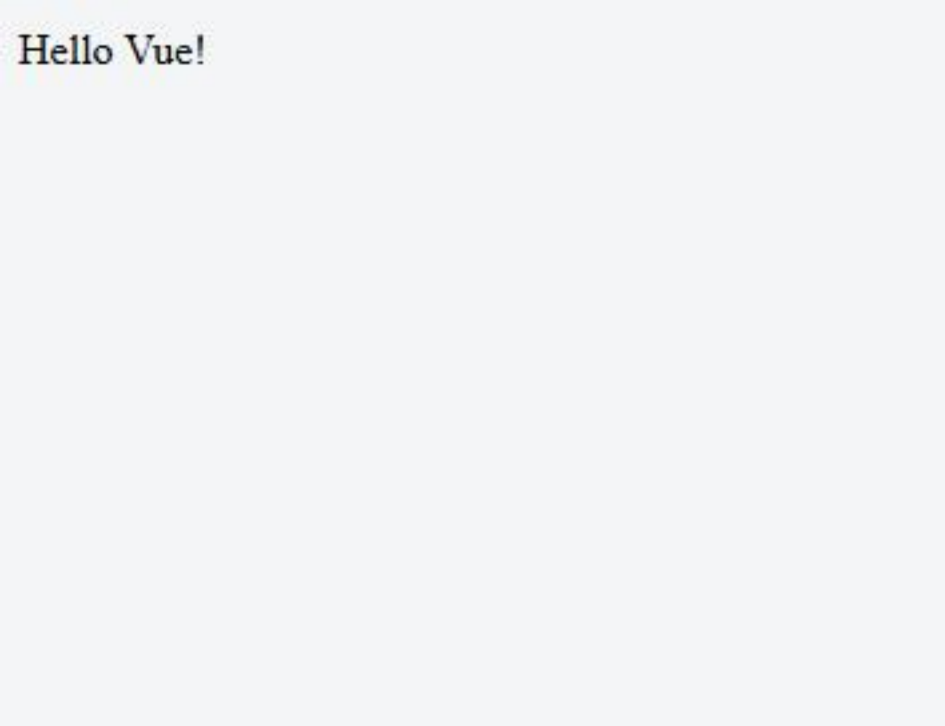


Hello world!

SOURCE

```
<div id="app">
    {{ message }}
</div>

<script>
    new Vue({
        el: '#app',
        data: {
            message: 'Hello Vue!'
        }
    })
</script>
```

BROWSER



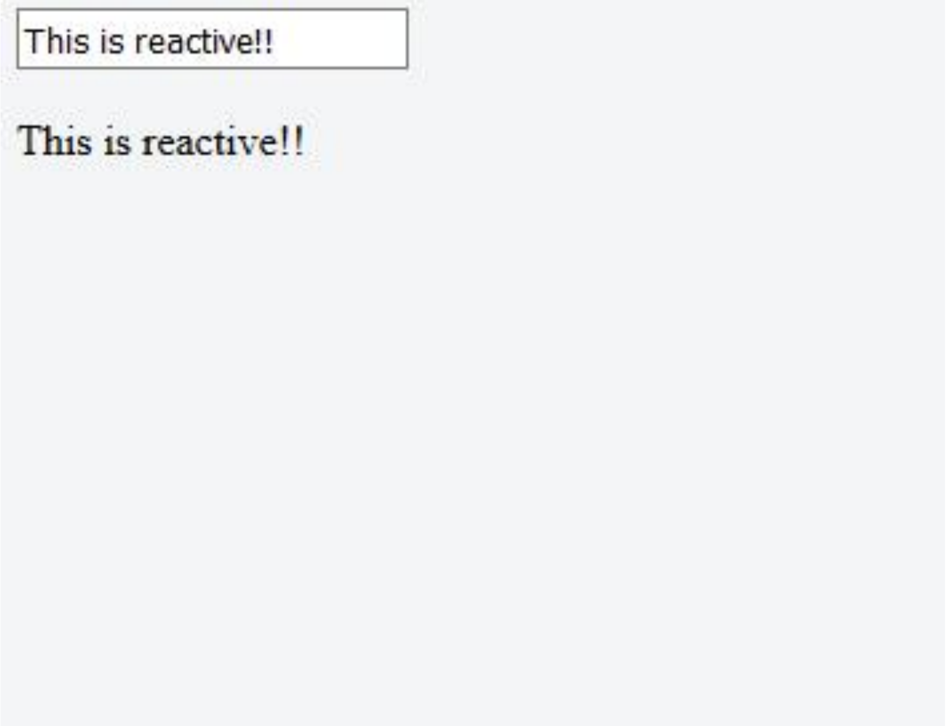
Hello Vue!

Reactivity

SOURCE

```
<div id="app">
    <input v-model="message"/>
    <p>{{ message }}</p>
</div>

<script>
    new Vue({
        el: '#app',
        data: {
            message: 'Hello Vue!'
        }
    })
</script>
```

BROWSER



This is reactive!!

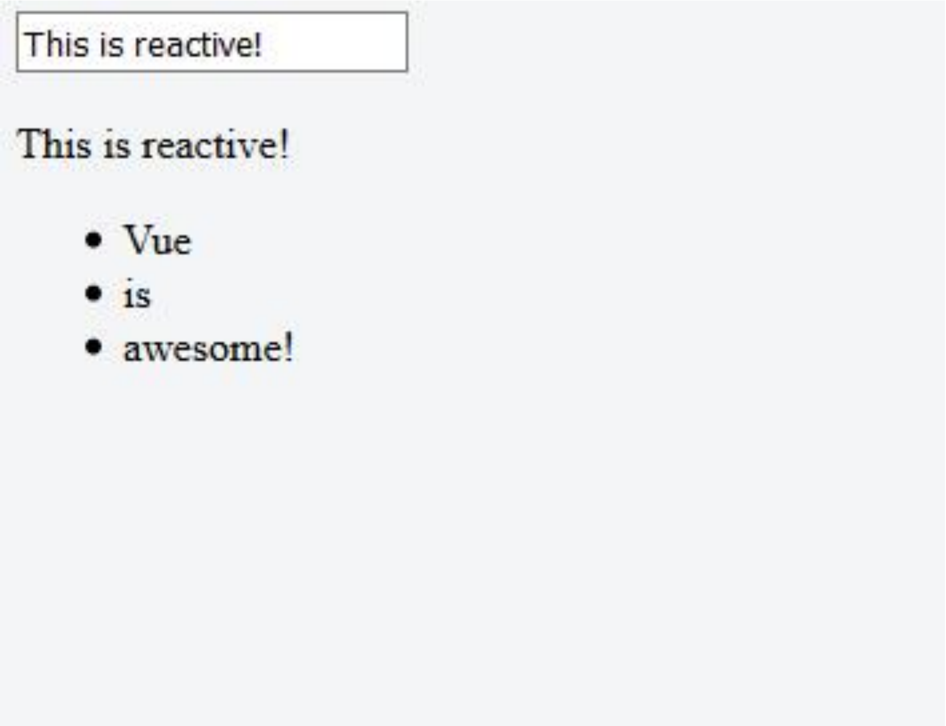
This is reactive!!

Loops

SOURCE

```
<div id="app">
    <input v-model="message"/>
    <p>{{ message }}</p>
    <ul>
        <li v-for="item in items">
            {{ item }}
        </li>
    </ul>
</div>

<script>
    new Vue({
        el: '#app',
        data: {
            message: 'Hello Vue!',
            items: [
                'Vue',
                'is',
                'awesome!'
            ]
        }
    })
</script>
```

BROWSER



This is reactive!

This is reactive!

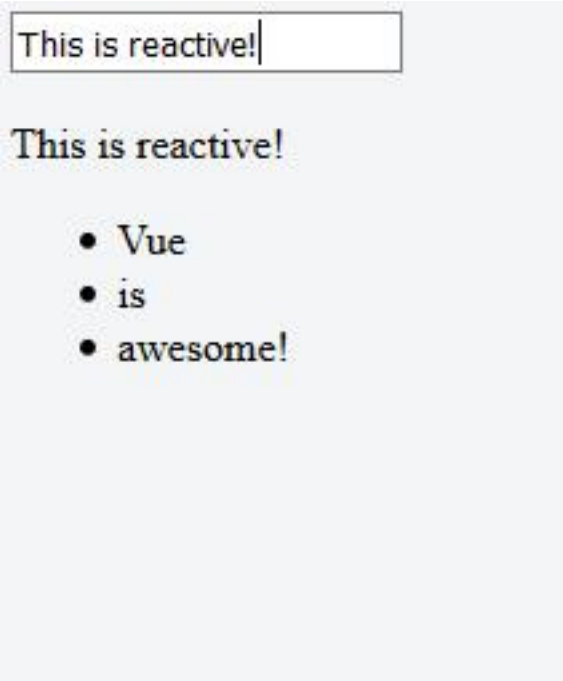
- Vue
- is
- awesome!

Components

SOURCE

```
<div id="app">
    <input v-model="message"/>
    <p>{{ message }}</p>
    <items-list></items-list>
</div>
<script>
    Vue.component('items-list', {
        template: `
            <ul>
                <li v-for="item in items">
                    {{ item }}
                </li>
            </ul>
        `,
        data() {
            return {
                items: [ 'Vue', 'is', 'awesome!' ]
            };
        }
    });

    new Vue({
        el: '#app',
        data: {
            message: 'Hello Vue!‘
        }
    })
</script>
```

BROWSER



This is reactive!

This is reactive!

- Vue
- is
- awesome!

Learn more

- Official Website
- Interactive Examples
- Recommended Introductory Video
- Free Online Course: Learn Vue 2 Step By Step
- VueJS Source Code

Sass

KEEP

CALM

AND

HATE

CSS

Sass
to the rescue!!

What can Sass do for you?

- CSS **superset**.
- Compiles into CSS.
  - Compatible in all browsers.
- Provides the following:
  - Variables
  - Functions
  - Nesting
  - Loops
  - Conditionals
  - File imports
  - …

Example

SASS

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
    font: 100% $font-stack;
    color: $primary-color;
}

nav {

    ul {
        margin: 0;
        padding: 0;
        list-style: none;
    }

    li { display: inline-block; }

    a {
        display: block;
        padding: 6px 12px;
        text-decoration: none;
    }

}
```

CSS

```
body {
    font: 100% Helvetica, sans-serif;
    color: #333;
}

nav ul {
    margin: 0;
    padding: 0;
    list-style: none;
}

nav li {
    display: inline-block;
}

nav a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
}
```

Learn more

- Official Website
- Sass Source Code

ionic

Ionic foundations

Hybrid Mobile Applications

APACHE CORDOVA™ HTML → Mobile

+

ANGULARJS by Google Javascript++

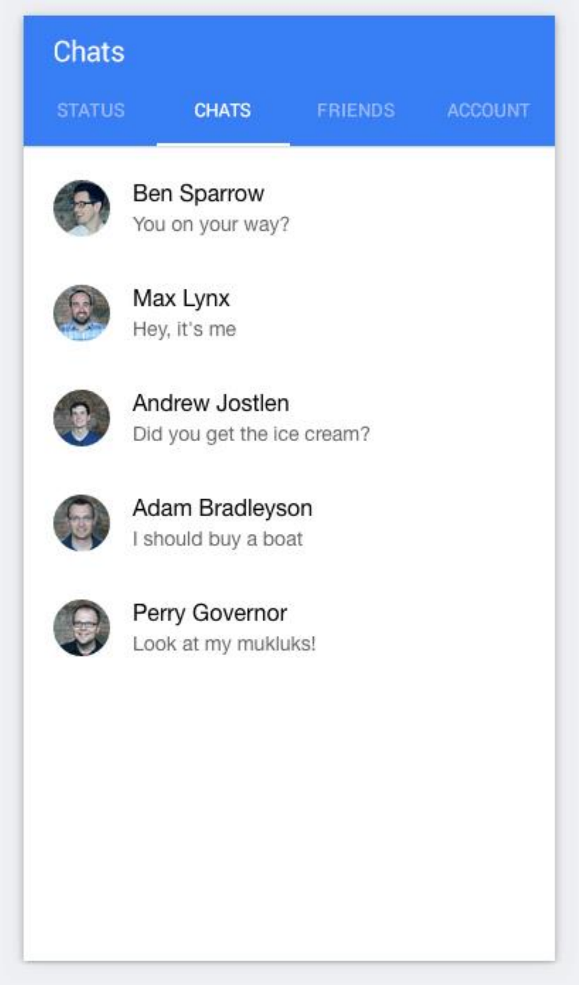
ionic

What can Ionic do for you?

- Develop once, deploy everywhere
- Ready made components
- Implement native functionality when necessary

Develop once, deploy everywhere

ANDROID



Ben Sparrow
You on your way?

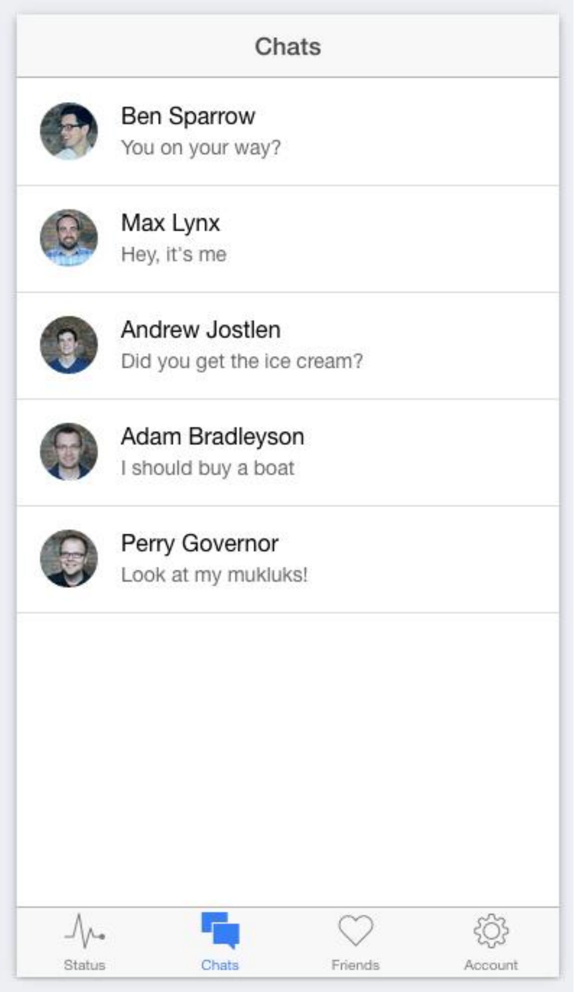
Max Lynx
Hey, it's me

Andrew Jostlen
Did you get the ice cream?

Adam Bradleyson
I should buy a boat

Perry Governor
Look at my mukluks!

IOS



Ben Sparrow
You on your way?

Max Lynx
Hey, it's me

Andrew Jostlen
Did you get the ice cream?

Adam Bradleyson
I should buy a boat

Perry Governor
Look at my mukluks!

Same source code

ionic

Ready made components

- Buttons
- FABs
- Tabs
- Action Sheets
- Menus
- Modals
- Toolbars
- Loaders
- And more...

Native functionality

Via Cordova Plugins

- Code written once per platform, Javascript interface
- Huge collection of available plugins:
  - Camera
  - GPS
  - Push Notifications
  - Bluetooth
  - NFC
  - …
- Program your own!

Learn more

- Official Website
- Components Interactive Documentation
- Ionic Source Code

Wrapping Up

You also need to know…

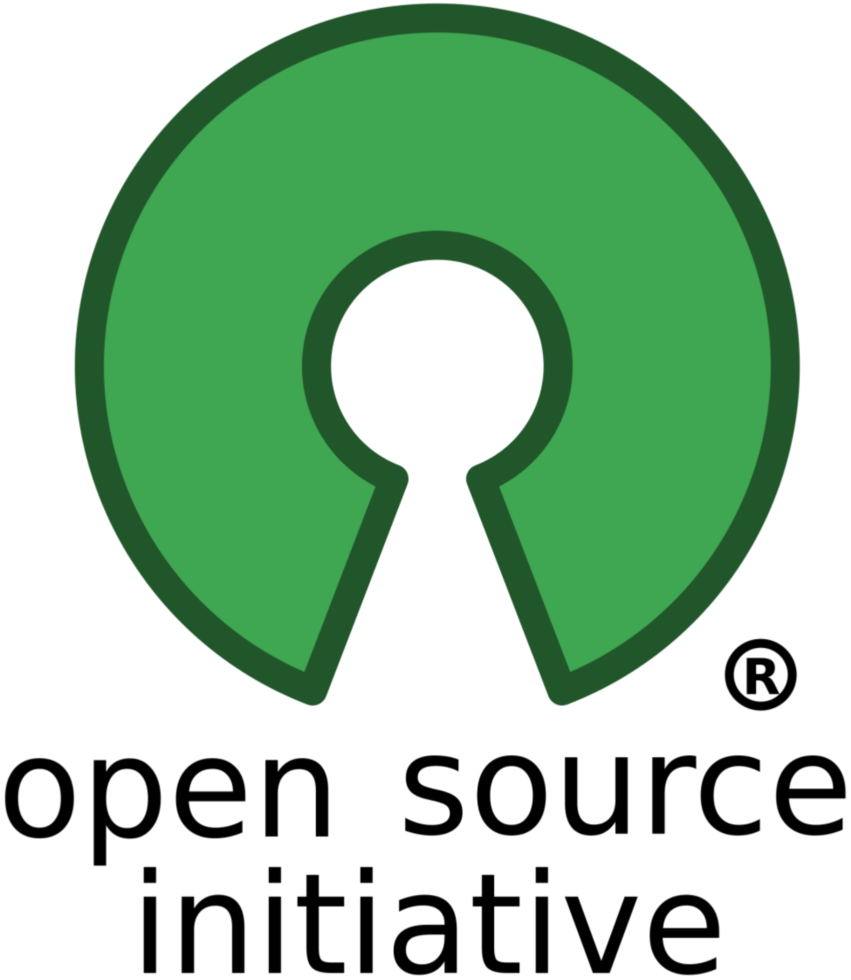
Package managers

- npmjs.com
- Repository for node (Javascript) libraries
- 400.000+ packages

- packagist.org
- Repository for php libraries
- 140.000+ packages

You also need to know…

Open source



Timeline

2006	Sass
2009	mongoDB®
2011	laravel
2012	TypeScript
2013	ionic
2014	Vue.js

Timeline

2006	Sass
2009	mongoDB®
2011	laravel
2012	TypeScript
2013	ionic
2014	Vue.js
1984	[ObjC]
1995	php
1995	MySQL®
1995	Java™
1996	5 JS 3
2011	
2014	

Programming is **Thinking**, not **Typing**.

- Casey Patton

Questions Time

✉ noel@geemba.com

@NoelDeMartin