Как современные библиотеки и фреймворки работают с DOM

Вячеслав Слинько
vslinko@yahoo.com
@vslinko

Agenda

- Общий цикл работы с DOM
- React
- Ember 2.10
- Angular 2

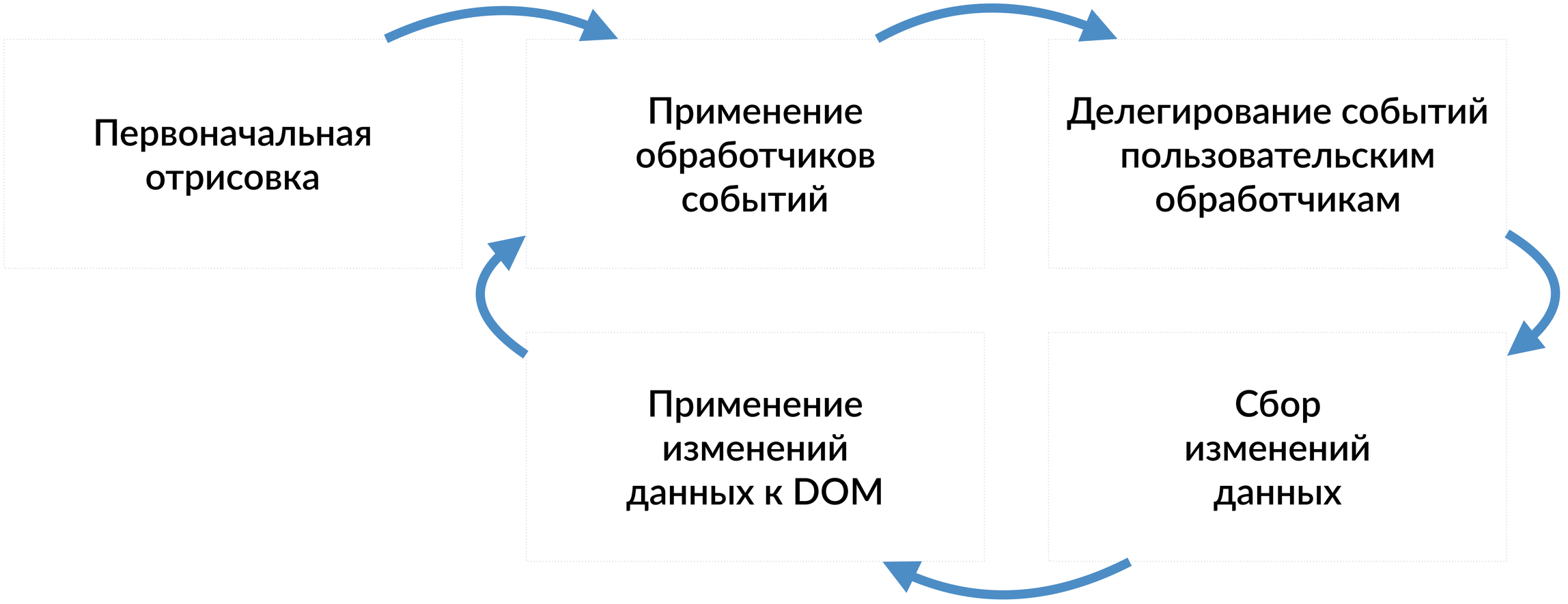
ЦИАН

Disclaimer

- Вся информация про поведение по умолчанию
- Исходники в слайдах — псевдокод
- Информация актуальная для конкретных версий
  - React 15.4.1
  - Ember 2.10.0-beta.1
  - Angular 2.2.3

ЦИАН

Общий цикл работы с DOM



ЦИАН

React

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <App />,
  document.getElementById('root')
);
```

ЦИАН

React: Компиляция JSX

```jsx
function Link() {
  return (
    <a href="/">
      JSX
    </a>
  );
}
```

```jsx
function Link() {
  return React.createElement(
    'a',
    {href: '/'},
    'JSX'
  );
}
```

```jsx
function Link() {
  return {
    $$typeof: Symbol.for('react.element'),
    type: 'a',
    key: null,
    ref: null,
    props: {href: '/'},
    _owner: ReactCurrentOwner.current,
  };
}
```

ЦИАН

React: Слежение за позицией scroll

- Кеширует значения избегая layout thrashing
- What forces layout/reflow. The comprehensive list.

ЦИАН

React: Внутренние классы компонентов

```
class ReactInternalComponent {
  constructor(reactElement) {}

  mountComponent(transaction, context) {}

  unmountComponent() {}

  receiveComponent(nextElement, transaction, nextContext) {}

  performUpdateIfNecessary(transaction) {}
}
```

ЦИАН

React: Внутренние классы компонентов

```
class ReactCompositeComponent extends ReactInternalComponent {}

class ReactDOMComponent extends ReactInternalComponent {}

class ReactDOMEmptyComponent extends ReactInternalComponent {}

class ReactDOMTextComponent extends ReactInternalComponent {}
```

ЦИАН

React: ReactCompositeComponent mountComponent

```
mountComponent(transaction, hostParent, hostContainerInfo, context) {
  const Component = this.reactElement.type;
  const inst = new Component(props, context);
  ReactInstanceMap.set(inst, this);

  if (inst.componentWillMount) {
    inst.componentWillMount();
  }

  ReactCurrentOwner.current = this;
  const renderedElement = inst.render();
  ReactCurrentOwner.current = null;

  const markup = ReactReconciler.mountComponent(
```

ЦИАН

React: ReactCompositeComponent mountComponent

```
  ReactCurrentOwner.current = this;

  const markup = ReactReconciler.mountComponent(
    instantiateReactComponent(renderedElement),
    transaction,
    hostParent,
    hostContainerInfo,
    context
  );

  if (inst.componentDidMount) {
    transaction.getReactMountReady().enqueue(inst.componentDidMount, inst);
  }
}
```

ЦИАН

React: ReactDOMComponent mountComponent

```
mountComponent(transaction, context) {
  const { type, props } = this.reactElement;

  assertValidProps(this, props);

  let mountImage;
  if (transaction.useCreateElement) {
    const el = this._createElement(type);
    this._updateDOMProperties(el);
    mountImage = DOMLazyTree(el);
    this._createInitialChildren(transaction, props, context, mountImage);
  } else {
    const tagOpen = this._createOpenTagMarkupAndPutListeners(
      transaction,
```

ЦИАН

React: ReactDOMComponent mountComponent

```
let mountImage;
if (transaction.useCreateElement) {
  const el = this._createElement(type);
  this._updateDOMProperties(el);
  mountImage = DOMLazyTree(el);
  this._createInitialChildren(transaction, props, context, mountImage);
} else {
  const tagOpen = this._createOpenTagMarkupAndPutListeners(
    transaction,
    props
  );
  const tagContent = this._createContentMarkup(transaction, props, context);
  mountImage = `<${tagOpen}>${tagContent}</${type}>`;
}
```

ЦИАН

React: ReactDOMComponent mountComponent

```
switch (type) {
  case 'audio': case 'form': case 'iframe': case 'img': case 'link':
  case 'object': case 'source': case 'video': case 'input': case 'select':
  case 'textarea':
    transaction.reactMountReady.enqueue(() => trapBubbledEventsLocal(this));
    break;

  case 'input': case 'textarea': case 'select': case 'button':
    if (props.autoFocus) {
      transaction.reactMountReady.enqueue(
        () => AutoFocusUtils.focusDOMComponent(this)
      );
    }
    break;
}
```

ЦИАН

React: Переиспользование DOM

```
<div data-reactroot="" data-reactid="1" data-react-checksum="-1535649998">
  <h1 data-reactid="2">
    Hello World!
  </h1>
</div>
```

ЦИАН

React: Вызов отложенных функций

- Восстанавливается фокус и каретка в поле ввода
- Инициализируются обработчики событий
- В случае вызова setState во время отрисовки, алгоритм запускается еще раз

ЦИАН

React: Работа с событиями

- Основной механизм — Event Delegation
- На некоторые события подписка напрямую
- Обход разных странных проблем в бразуерах

ЦИАН

React: Делегирование событий

ЦИАН

React: Transaction

```
class Transaction {
  perform(callback, ...args) {
    const wrappers = this.getTransactionWrappers();

    try {
      wrappers.forEach(wrapper => wrapper.initialize());

      callback(...args);

    } finally {
      wrappers.forEach(wrapper => wrapper.close());
    }
  }
}
```

ЦИАН

React: ReactDefaultBatchingStrategyTransaction

```
const FLUSH_BATCHED_UPDATES = {
  initialize: () => {},
  close: () => {
    ReactUpdates.flushBatchedUpdates()
  },
};
```

```
class ReactDefaultBatchingStrategyTransaction
extends Transaction {
  getTransactionWrappers() {
    return [
      FLUSH_BATCHED_UPDATES,
    ];
  }
}
```

ЦИАН

React: ReactUpdatesFlushTransaction

```
const NESTED_UPDATES = {
  initialize: () => {},
  close: () => {
    if (dirtyComponents.length > 0) {
      ReactUpdates.flushBatchedUpdates();
    }
  },
};
```

```
class ReactUpdatesFlushTransaction extends Transaction {
  constructor() {
    super();
    this.reconcileTransaction = new ReactReconcileTransaction();
  }

  getTransactionWrappers() {
    return [
      NESTED_UPDATES,
    ];
  }

  perform(callback, ...args) {
    return super.perform(
      () => this.reconcileTransaction.perform(callback, ...args),
    );
  }
}
```

ЦИАН

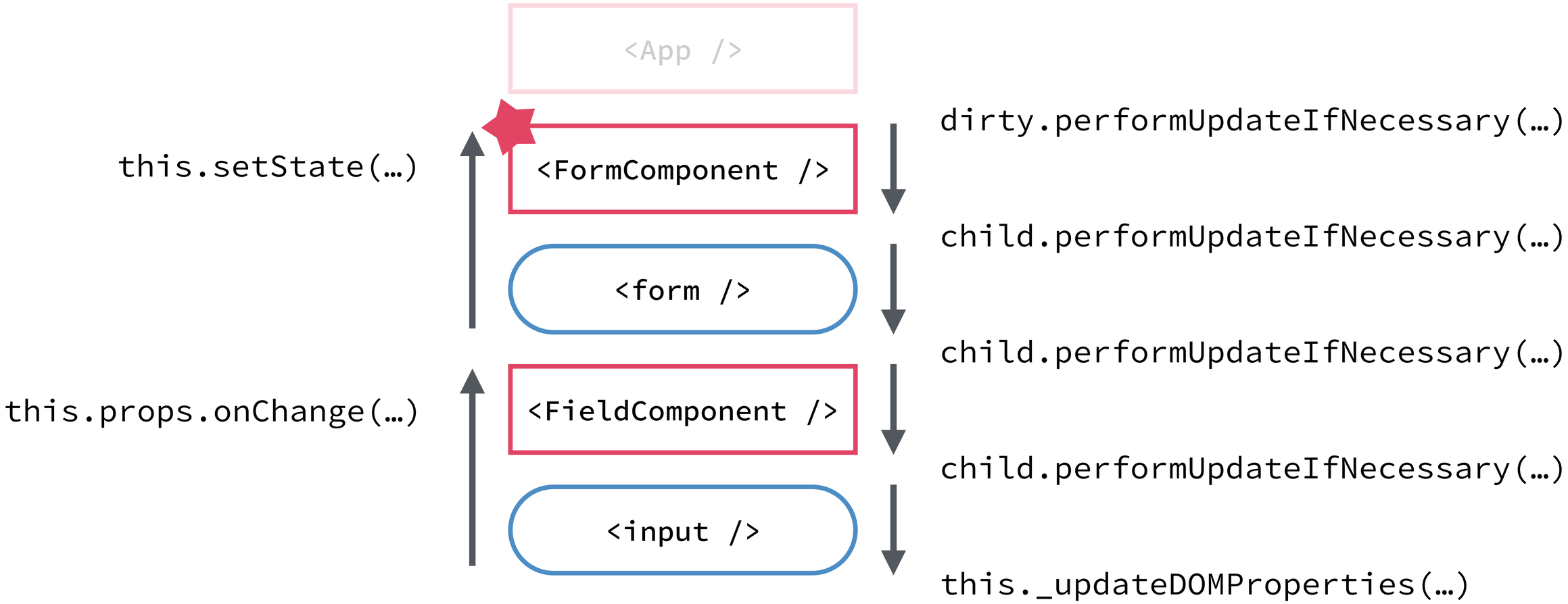
React: ReactReconcileTransaction

```
const SELECTION_RESTORATION = {
  initialize: ReactInputSelection.getSelectionInformation,
  close: ReactInputSelection.restoreSelection,
};

const EVENT_SUPPRESSION = {
  initialize: ReactBrowserEventEmitter.disable,
  close: ReactBrowserEventEmitter.enabled,
};

const ON_DOM_READY_QUEUEING = {
  initialize: (transaction) => {
    transaction.reactMountReady.reset()
  },
  close: (transaction) => {
    transaction.reactMountReady.notifyAll()
  },
};
```

```
class ReactReconcileTransaction extends Transaction {
  constructor(useCreateElement) {
    super();
    this.useCreateElement = useCreateElement;
    this.reactMountReady = new CallbackQueue();
  }

  getTransactionWrappers() {
    return [
      SELECTION_RESTORATION,
      EVENT_SUPPRESSION,
      ON_DOM_READY_QUEUEING,
    ];
  }
}
```

ЦИАН

React: ReactReconciler



ЦИАН

Ember 2.10

```
import Ember from 'ember';
import Resolver from './resolver';
import loadInitializers from 'ember-load-initializers';
import config from './config/environment';

let App;

Ember.MODEL_FACTORY_INJECTIONS = true;

App = Ember.Application.extend({
  modulePrefix: config.modulePrefix,
  podModulePrefix: config.podModulePrefix,
  Resolver
});

loadInitializers(App, config.modulePrefix);

export default App;
```

ЦИАН

Ember 2.10: Компоненты и шаблоны

```
import Ember from 'ember';

export default Ember.Component.extend({
  init() {
    this._super();
    this.set('todos', []);
  },

  actions: {
    onSave(todo) {
      this.set(
        'todos',
        this.get('todos').concat([todo])
      );
    },
  },
});
```

```
{{#x-form onSave=(action 'onSave')}}{{/x-form}}

<ul>
  {{#each todos as |todo|}}
  <li>
    {{#x-todo todo=todo}}{{/x-todo}}
  </li>
  {{/each}}
</ul>
```

ЦИАН

Ember 2.10: Wire Format (AST)


```
{ "statements":[
    ["text","\\n\\n\\n\\n\\n\\n"],
    ["block",["x-form"],null,[["onSave"],[["helper",["action"],[["get",[null]],"onSave"],null]]],2],
    ["text","\\n\\n"],
    ["open-element","ul",[]],
    ["flush-element"],
    ["text","\\n"],
    ["block",["each"],[["get",["todos"]]],null,1],
    ["close-element"],
    ["text","\\n\\n\\n\\n"]
  ],
  "locals":[],
  "named":[],
  "yields":[],
  "blocks":[
    {"statements":[],"locals":[]},
    {"statements":[
      ["text","  "],
      ["open-element","li",[]],
```


ЦИАН

Ember 2.10: Подпрограммы

```
OPEN-PRIMITIVE-ELEMENT("div")
STATIC-ATTR(name="class", value="contacts")
FLUSH-ELEMENT
TEXT("\n")
PUT-ARGS(positional=[self.contacts], named={key: "id"})
ENTER("BEGIN [144]", "END [145]")
LABEL("BEGIN [144]")
PUT-ITERATOR
JUMP-UNLESS("END [145]")
ENTER-LIST("BEGIN [149]", "END [150]")
LABEL("ITER [152]")
NEXT-ITER
ENTER-WITH-KEY("BEGIN [149]", "END [150]")
LABEL("BEGIN [149]")
EVALUATE(default)
  PUSH-CHILD-SCOPE
  BIND-POSITIONAL-ARGS(["contact"])
  TEXT("    ")
  PUT-COMPONENT-DEFINITION("h-card")
  OPEN-COMPONENT($OPERAND)
  CLOSE-COMPONENT
  TEXT("\n    ")
  OPEN-PRIMITIVE-ELEMENT("hr")
```

```
TRY(guid=162, begin=BEGIN [144], end=END [145])
  ASSERT(expected=true)
  LIST-BLOCK(guid=164, begin=BEGIN [149], end=END [150], map={"1
    TRY(guid=165, begin=BEGIN [149], end=END [150])
      JUMP-IF-NOT-MODIFIED("END [413]")
      UPDATE-COMPONENT("h-card")
      TRY(guid=245, begin=BEGIN [191], end=END [192])
        ASSERT(expected=false)
        TRY(guid=262, begin=BEGIN [253], end=END [254])
          ASSERT(expected=true)
          GUARDED-CAUTIOUS-UPDATE(lastValue="John")
          TRY(guid=290, begin=BEGIN [275], end=END [276])
            ASSERT(expected=false)
          GUARDED-CAUTIOUS-UPDATE(lastValue="Appleseed")
      TRY(guid=293, begin=BEGIN [202], end=END [203])
        ASSERT(expected=true)
        GUARDED-CAUTIOUS-UPDATE(lastValue="Apple Inc.")
      TRY(guid=303, begin=BEGIN [210], end=END [211])
        ASSERT(expected=true)
        PATCH-ELEMENT(element="<a />", type=attribute, namespac
        GUARDED-CAUTIOUS-UPDATE(lastValue="appleseec
      TRY(guid=323, begin=BEGIN [218], end=END [219]
        ASSERT(expected=true)
```

ЦИАН

Ember 2.10: Opcode

```
export abstract class AbstractOpcode implements LinkedListNode {
  public type: string;
  public _guid: number;

  prev: AbstractOpcode;
  next: AbstractOpcode;
}

export abstract class Opcode extends AbstractOpcode {
  next: Opcode = null;
  prev: Opcode = null;

  abstract evaluate(vm: VM);
}
```

ЦИАН

Ember 2.10: Append Opcodes

```
OPEN-PRIMITIVE-ELEMENT("div")
STATIC-ATTR(name="class", value="contacts")
FLUSH-ELEMENT
TEXT("\n")
PUT-ARGS(positional=[self.contacts], named={key: "id"})
ENTER("BEGIN [144]", "END [145]")
LABEL("BEGIN [144]")
PUT-ITERATOR
JUMP-UNLESS("END [145]")
ENTER-LIST("BEGIN [149]", "END [150]")
```

ЦИАН

Ember 2.10: “View” события

```
export default Ember.Component.extend({
  init() {
    this._super()
    this.set('value', '')
  },

  submit() {
    this.get('onSave')({
      id: ++id,
      value: this.get('value'),
    });
  },
});
```

ЦИАН

Ember 2.10: “Action” события

```
<form {{action 'save' on="submit"}}>

  {{input value=value}}

</form>
```

ЦИАН

Ember 2.10: “Property” события

```
<form onsubmit={{action 'save'}}>

  <input value={{value}} oninput={{action 'setValue'}} />

</form>
```

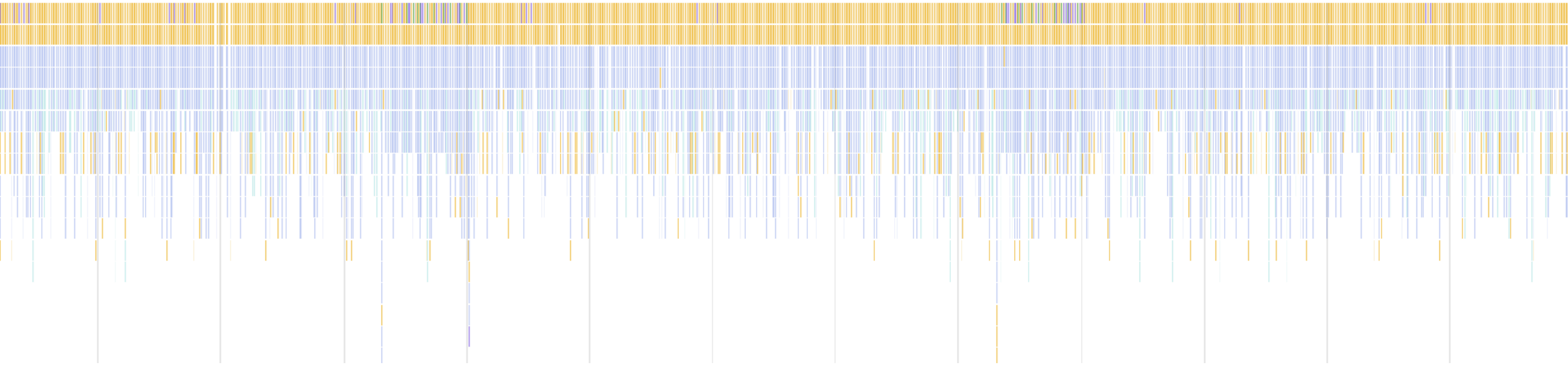
ЦИАН

Ember 2.10: Применение обработчиков событий

```
setup(rootElement) {
  for (event in this.events) {
    if (this.events.hasOwnProperty(event)) {
      rootElement.on(event + '.ember', '.ember-view', function(evt, triggeringManager) {…
      });

      rootElement.on(event + '.ember', '[data-ember-action]', function(evt) {…
      });
    }
  }
},
```

ЦИАН

Ember 2.10: Последствия Event Delegation



ЦИАН

Ember 2.10: Backburner.js

```
const backburner = new Backburner([ 'render' ]);
const state = { name: "Erik" };

function render() {
  $('#name').text(state.name);
}

function updateModel(name) {
  state.name = name;
  backburner.deferOnce('render', render);
}

backburner.run(() => {
  updateModel("Kris");
  updateModel("Tom");
  updateModel("Yehuda");
});
```

ЦИАН

<h-card> Layout

```
<div>{{@name}}</div>
```

Opcodes (<h-car

Opcodes

```
BIND-NAMED-ARGS($1: $ARGS[n
OPEN-COMPONENT-ELEMENT("div
DID-CREATE-ELEMENT($ARGS)
SHADOW-ATTRIBUTES($ARGS)
STATIC-ATTR(name="class", v
PUT-VALUE(`EmberID(...)`)
DYNAMIC-ATTR(name="id", val
FLUSH-ELEMENT
GUARDED-CAUTIOUS-APPEND($1(
CLOSE-ELEMENT
DID-RENDER-LAYOUT
```

ЦИАН

Top-Level Template

```
{{#h-card name=todo.name}}{{/h-card}}
```

<h-card> Layout

```
<div>{{@name}}</div>
```

```
OPEN-COMPONENT($OPERAND)
CLOSE-COMPONENT
```

Opcodes (<h-car

Opcodes

```
BIND-NAMED-ARGS($1: $ARGS[n
OPEN-COMPONENT-ELEMENT("div
DID-CREATE-ELEMENT($ARGS)
SHADOW-ATTRIBUTES($ARGS)
STATIC-ATTR(name="class", v
```

Data

```
{
  "todo": {"name": "c"}
}
```

Top-Level Template

```
{{#h-card name=todo.name}}{{/h-card}}
```

Opcodes (Top-Le

Opcodes

```
PUT-COMPONENT-DEFINITION("h
OPEN-COMPONENT($OPERAND)
CLOSE-COMPONENT
```

Data

```
name": "c"}
```

p-Level Template

```
me=todo.name}}{{/h-card}}
```

Opcodes (Top-Level)

Opcodes

```
PUT-COMPONENT-DEFINITION("h-card")
OPEN-COMPONENT($OPERAND)
CLOSE-COMPONENT
```

Updating Opc

Opcodes

```
JUMP-IF-NOT-MODIFIED("EN
UPDATE-COMPONENT("h-card
GUARDED-CAUTIOUS-UPDATE(
DID-UPDATE-LAYOUT
DID-MODIFY
LABEL("END [1610]")
```

p-Level Template

```
me=todo.name}}{{/h-card}}
```

h-card> Layout

```
}}</div>
```

Opcodes (<h-card>)

Opcodes

```
BIND-NAMED-ARGS($1: $ARGS[name])
OPEN-COMPONENT-ELEMENT("div")
DID-CREATE-ELEMENT($ARGS)
SHADOW-ATTRIBUTES($ARGS)
STATIC-ATTR(name="class", value="ember-view"
PUT-VALUE(`EmberID(...)`)
DYNAMIC-ATTR(name="id", value=$OPERAND)
FLUSH-ELEMENT
GUARDED-CAUTIOUS-APPEND($1(name))
CLOSE-ELEMENT
DID-RENDER-LAYOUT
```

codes (Top-Level)

```
NT-DEFINITION("h-card")
ENT($OPERAND)
NENT
```

Updating Opcodes

Opcodes

```
JUMP-IF-NOT-MODIFIED("END [1610]")
UPDATE-COMPONENT("h-card")
GUARDED-CAUTIOUS-UPDATE(lastValue="c")
DID-UPDATE-LAYOUT
DID-MODIFY
LABEL("END [1610]")
```

ЦИАН

Angular 2

```
import { enableProdMode } from '@angular/core';
import {
  platformBrowserDynamic
} from '@angular/platform-browser-dynamic';

import { AppModule } from '../app.module';

enableProdMode();
platformBrowserDynamic()
  .bootstrapModule(AppModule);
```

ЦИАН

Angular 2: Компиляция DI

```
Object.defineProperty(AppModuleInjector.prototype, '_Title_27', { get: function() {
  var self = this;
  if ((self.__Title_27 == null)) { (self.__Title_27 = new jit_Title22()); }
  return self.__Title_27;
}});
AppModuleInjector.prototype.createInternal = function() {
  var self = this;
  self._CommonModule_0 = new jit_CommonModule23();
  self._ApplicationModule_1 = new jit_ApplicationModule24();
  self._BrowserModule_2 = new jit_BrowserModule25(self.parent.get(jit_BrowserModule25,null));
  self._AppModule_3 = new jit_AppModule26();
  self._ErrorHandler_6 = jit_errorHandler27();
  self._ApplicationInitStatus_7 = new jit_ApplicationInitStatus28(self.parent.get(jit_Token_Application_Initialize
  self._Testability_8 = new jit_Testability30(self.parent.get(jit_NgZone11));
  self._ApplicationRef__9 = new jit_ApplicationRef_31(self.parent.get(jit_NgZone11),self.parent.get(jit_Console32)
  return self._AppModule_3;
};
AppModuleInjector.prototype.getInternal = function(token,notFoundResult) {
  var self = this;
  if ((token === jit_CommonModule23)) { return self._CommonModule_0; }
  if ((token === jit_ApplicationModule24)) { return self._ApplicationModule_1; }
  if ((token === jit_BrowserModule25)) { return self._BrowserModule_2; }
  if ((token === jit_AppModule26)) { return self._AppModule_3; }
```

ЦИАН

Angular 2: Компонент

```
@Component({
  selector: 'todo-form',
  template: `
  <form (submit)="onSubmit($event)">
    <input (input)="value = $event.target.value" [value]="value" />
  </form>
  `,
})
export class FormComponent {
  @Output() created = new EventEmitter<Todo>();
  value = '';

  onSubmit(e: Event) {
    e.preventDefault()
    this.created.emit({ id: ++id, value: this.value })
    this.value = '';
  }
}
```

ЦИАН

Angular 2: Host класс

```
var styles_AppComponent_Host = [];
var renderType_AppComponent_Host = jit_createRenderComponentType0('',0,jit_ViewEncapsulation_None1,styles_AppComp
function View_AppComponent_Host0(viewUtils,parentView,parentIndex,parentElement) {…
}
View_AppComponent_Host0.prototype = Object.create(jit_AppView2.prototype);
View_AppComponent_Host0.prototype.createInternal = function(rootSelector) {
  var self = this;
  self._el_0 = jit_selectOrCreateRenderHostElement4(self.renderer,'my-app',jit_EMPTY_INLINE_ARRAY5,rootSelector,n
  self.compView_0 = new jit_View_AppComponent6(self.viewUtils,self,0,self._el_0);
  self._AppComponent_0_3 = new jit_Wrapper_AppComponent7();
  self.compView_0.create(self._AppComponent_0_3.context);
  self.init(self._el_0,(self.renderer.directRenderer? null: [self._el_0]),null);
  return new jit_ComponentRef_8(0,self,self._el_0,self._AppComponent_0_3.context);
};
View_AppComponent_Host0.prototype.injectorGetInternal = function(token,requestNodeIndex,notFoundResult) {…
};
View_AppComponent_Host0.prototype.detectChangesInternal = function(throwOnChange) {…
};
View_AppComponent_Host0.prototype.destroyInternal = function() {…
};
View_AppComponent_Host0.prototype.visitRootNodesInternal = function(cb,ctx) {…
```

ЦИАН

Angular 2: View класс

```
var styles_FormComponent = [];
var renderType_FormComponent = jit_createRenderComponentType0('',0,jit_ViewEncapsulation_None1,styles_FormCompone
function View_FormComponent0(viewUtils,parentView,parentIndex,parentElement) {…
}
View_FormComponent0.prototype = Object.create(jit_AppView2.prototype);
View_FormComponent0.prototype.createInternal = function(rootSelector) {
  var self = this;
  var parentRenderNode = self.renderer.createViewRoot(self.parentElement);
  self._text_0 = self.renderer.createText(parentRenderNode,'\n  ',null);
  self._el_1 = jit_createRenderElement5(self.renderer,parentRenderNode,'form',jit_EMPTY_INLINE_ARRAY6,null);
  self._text_2 = self.renderer.createText(self._el_1,'\n    ',null);
  self._el_3 = jit_createRenderElement5(self.renderer,self._el_1,'input',jit_EMPTY_INLINE_ARRAY6,null);
  self._text_4 = self.renderer.createText(self._el_1,'\n  ',null);
  self._text_5 = self.renderer.createText(parentRenderNode,'\n  ',null);
  var disposable_0 = jit_subscribeToRenderElement7(self,self._el_1,new jit_InlineArray28(2,'submit',null),self.ev
  var disposable_1 = jit_subscribeToRenderElement7(self,self._el_3,new jit_InlineArray28(2,'input',null),self.eve
  self.init(null,(self.renderer.directRenderer? null: […
  ]
  ),[…
  ]
  );
```

ЦИАН

Angular 2: Wrapper класс

```
function Wrapper_FormComponent() {…
}
Wrapper_FormComponent.prototype.ngOnDetach = function(view,componentView,el) {
};
Wrapper_FormComponent.prototype.ngOnDestroy = function() {…
};
Wrapper_FormComponent.prototype.ngDoCheck = function(view,el,throwOnChange) {…
};
Wrapper_FormComponent.prototype.checkHost = function(view,componentView,el,throwOnChange) {
};
Wrapper_FormComponent.prototype.handleEvent = function(eventName,$event) {…
};
Wrapper_FormComponent.prototype.subscribe = function(view,_eventHandler,emit0) {
  var self = this;
  self._eventHandler = _eventHandler;
  if (emit0) { (self.subscription0 = self.context.created.subscribe(_eventHandler.bind(view,'created'))); }
};
return Wrapper_FormComponent
})
```

ЦИАН

Angular 2: Применение обработчиков событий

```
<form (document:submit)="onSubmit($event)">

  <input (input)="value = $event.target.value" [value]="value" />

</form>
```

ЦИАН

Angular 2: Применение обработчиков событий

- Обработчики применяются через плагины
  - HammerGesturesPlugin
  - KeyEventsPlugin
  - DomEventsPlugin

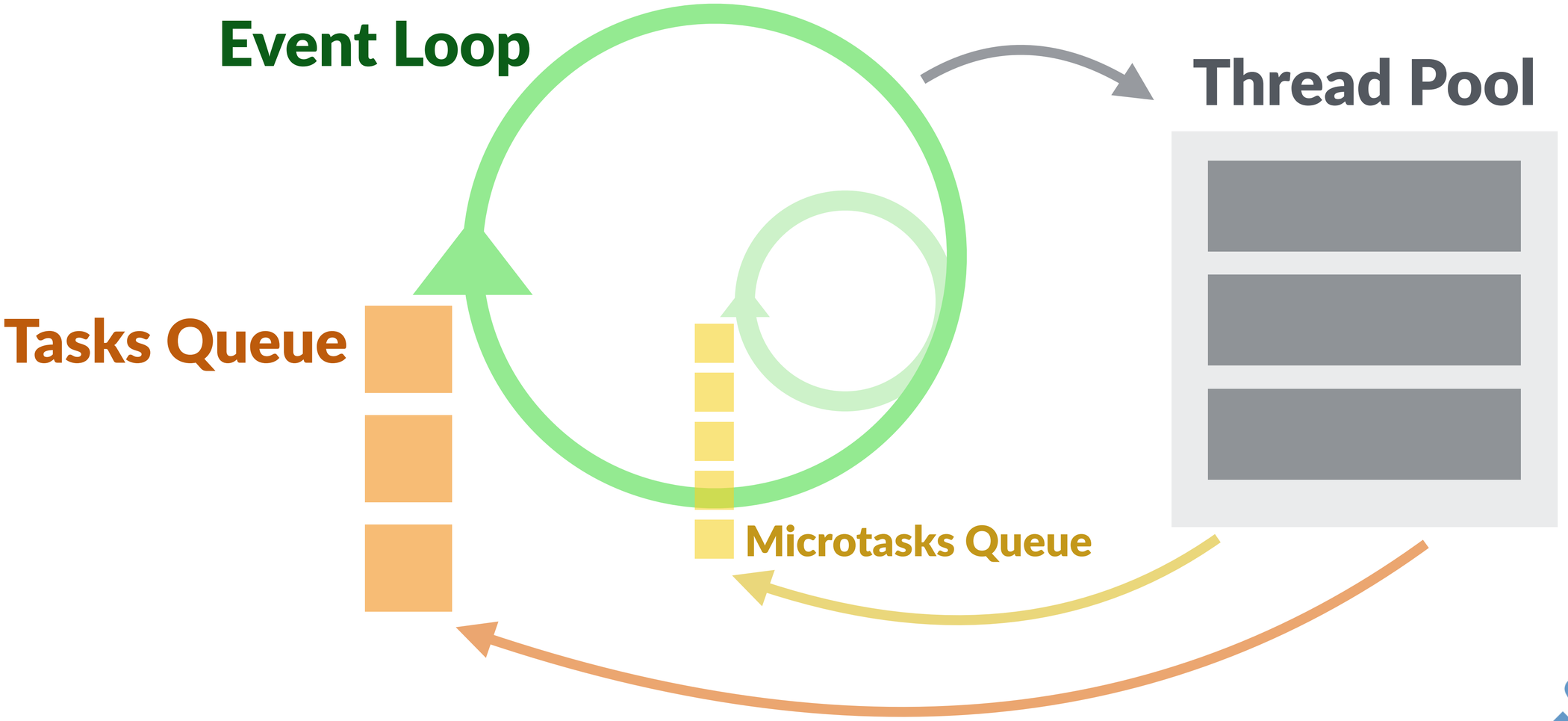
ЦИАН

Angular 2: Делегирование событий пользовательским обработчикам

```
View_FormComponent0.prototype.handleEvent_1 = function(eventName,$event) {
  var self = this;
  self.markPathToRootAsCheckOnce();
  var result = true;
  if ((eventName == 'submit')) {
    var pd_sub_0 = (self.context.onSubmit($event) !== false);
    result = (pd_sub_0 && result);
  }
  return result;
};
```

ЦИАН

Angular 2: Zone.js



ЦИАН

Angular 2: Zone.js

```
export class TimerComponent {
  seconds: number;

  constructor() {
    this.seconds = 0

    setInterval(() => {
      this.seconds++
    }, 1000)
  }
}
```

ЦИАН

Angular 2: Применение изменений данных к DOM

ЦИАН

Angular 2: Применение изменений данных к DOM

```
View_FormComponent0.prototype.detectChangesInternal = function(throwOnChange) {
  var currVal_6 = this.context.value;

  if (jit_checkBinding9(throwOnChange, this._expr_6, currVal_6)) {
    this.renderer.setElementProperty(this._el_3, 'value', currVal_6);
    this._expr_6 = currVal_6;
  }
};
```

ЦИАН

Recap

ЦИАН

Recap: React

- Работает через внутренние компоненты
- Переиспользует разметку
- Система абстракций над событиями
- Все работает внутри транзакций
- Перерасчитывается все поддерево компонентов

ЦИАН

Recap: Ember 2.10

- Glimmer компилирует шаблоны в подпрограммы
- Ember максимально использует Event Delegation
- Существует два вида обработчиков событий
- Все работает внутри Backburner.js
- Подпрограммы выполняются в Glimmer VM

ЦИАН

Recap: Angular 2

- Компилирует компоненты в 3 JavaScript класса
- View классы работают через внешний Renderer
- Использует plugins для работы с событиями
- Весь асинхронный код работает внутри Zone.js
- View классы напрямую обновляют DOM

ЦИАН

Ссылки

- React Codebase Overview
- Glimmer 2 Deep Dive
- Angular 2 Change Detection Explained

ЦИАН

Q&A

Вячеслав Слинько
vslinko@yahoo.com
@vslinko