k

part 1

Hybrid Application เป็นการผสมผสานระหว่าง Web Application และ Native Application

กล่าวคือ เป็นการพัฒนา Application โดยใช้ภาษาที่ใช้พัฒนา Web App เช่น HTML, CSS, JavaScript

แต่ก็สามารถดึงความสามารถของ Native เช่น กล้อง หรือ GPS และสามารถทำเป็น Application

นำไปลง Play Store หรือ App Store ได้เหมือน​ Native App เช่นกัน เรียกได้ว่า พัฒนาครั้งเดียว ได้ทุกแพลตฟอร์ม กันเลยทีเดียว

React Native คือ Javascript Framework หรือ ชุดคำสั่งที่ถูกสร้างขึ้นมาสำหรับพัฒน

Mobile Application แบบ Cross-platform เราสามารถเขียนคำสั่งชุดเดียว หรือมีความแตกต่างกันแต่ไม่มาก

แต่สามารถทำงานได้บนระบบปฏิบัติการที่แตกต่างกัน โดยใช้ Javascript เป็นภาษาหลักในการเขียนโปรแกรม

React Native เป็นเฟรมเวิร์กสำหรับการสร้างแอพมือถือดั้งเดิมใน JavaScript โดยใช้

โต้ตอบไลบรารี JavaScript; React Native Code จะคอมไพล์ไปยังส่วนประกอบดั้งเดิมของจริง ถ้า

คุณไม่แน่ใจว่า React คืออะไร มันเป็นไลบรารี JavaScript ที่โอเพ่นซอร์สและใช้งานภายใน

เฟสบุ๊ค. เดิมใช้เพื่อสร้างส่วนต่อประสานผู้ใช้สำหรับเว็บแอปพลิเคชัน มันมี

ตั้งแต่มีการพัฒนาและตอนนี้ยังสามารถใช้สร้างแอปพลิเคชันฝั่งเซิร์ฟเวอร์และมือถือได้อีกด้วย

(โดยใช้ React Native)

part 1 :

Lifecycle :

เมื่อสร้างคลาส React Native เมธอดจะถูกสร้างอินสแตนซ์ที่คุณเชื่อมต่อได้

วิธีการเหล่านี้เรียกว่าวิธีวงจรชีวิต และเราจะกล่าวถึงในเชิงลึกในบทที่ 2

เมธอดในรายการ 1.1 คือ constructor, componentDidMount และ render แต่

มีอีกสองสามรายการและพวกเขาทั้งหมดมีกรณีการใช้งานของตัวเอง

Lifecycle :

วิธีวงจรชีวิตเกิดขึ้นพร้อมกันและช่วยจัดการสถานะของส่วนประกอบด้วย

เป็นรันโค้ดในแต่ละขั้นตอนของวิธีการ หากคุณต้องการ วิธีวงจรชีวิตที่จำเป็นเท่านั้นคือ

แสดงผล; ส่วนอื่นๆ ทั้งหมดเป็นตัวเลือก เมื่อทำงานกับ React Native คุณคือพื้นฐาน

การทำงานด้วยวิธีวงจรชีวิตและข้อมูลจำเพาะเดียวกันกับที่คุณใช้กับ React

what you will learn :

ในหนังสือเล่มนี้ เราจะครอบคลุมทุกสิ่งที่คุณจำเป็นต้องรู้เพื่อสร้างแอปพลิเคชันมือถือที่มีประสิทธิภาพ

สำหรับ iOS และ Android โดยใช้เฟรมเวิร์ก React Native เพราะ React Native

สร้างขึ้นโดยใช้ไลบรารี React เราจะเริ่มในบทที่ 2 โดยครอบคลุมและทั่วถึง

อธิบายการทำงานของ React

จากนั้นเราจะพูดถึงองค์ประกอบดั้งเดิมมากมายที่มาพร้อมกับเฟรมเวิร์ก

ออกจากกล่องและศึกษาวิธีการทำงานของแต่ละคน ใน React Native ส่วนประกอบ

โดยพื้นฐานแล้วเป็นโค้ดที่มีฟังก์ชันการทำงานเฉพาะหรือองค์ประกอบ UI และ can

ใช้งานง่ายในแอปพลิเคชัน ส่วนประกอบได้รับการคุ้มครองอย่างกว้างขวางตลอดทั้งนี้

จองเพราะเป็นส่วนประกอบสำคัญของแอปพลิเคชัน React Native

มีหลายวิธีในการใช้การนำทาง โดยแต่ละวิธีมีความแตกต่าง ข้อดี และ

ข้อเสีย เราจะหารือเกี่ยวกับการนำทางในเชิงลึกและครอบคลุมถึงวิธีสร้างการนำทางที่มีประสิทธิภาพโดยใช้

API การนำทางที่สำคัญที่สุด เราจะครอบคลุมไม่เฉพาะการนำทางดั้งเดิม

API ที่ออกมาจากกล่องด้วย React Native แต่ยังรวมถึงโครงการชุมชนอีกสองสามโครงการ

ได้ทาง npm

ต่อไป เราจะหารือในเชิงลึกทั้ง API ข้ามแพลตฟอร์มและเฉพาะแพลตฟอร์มที่มีให้บริการ

ใน React Native และวิธีการทำงาน ได้เวลาเริ่มทำงานแล้ว

ด้วยข้อมูลโดยใช้คำขอเครือข่าย, AsyncStorage (รูปแบบของที่จัดเก็บในเครื่อง), Firebase และ

เว็บซ็อกเก็ต จากนั้นเราจะเจาะลึกถึงสถาปัตยกรรมข้อมูลที่แตกต่างกันและดูว่าแต่ละสถาปัตยกรรมเป็นอย่างไร

ทำงานเพื่อจัดการกับสถานะของแอปพลิเคชัน สุดท้ายเราจะดูการทดสอบและความแตกต่างกันเล็กน้อย

วิธีทดสอบใน React Native

what should you know

เพื่อให้ได้ประโยชน์สูงสุดจากหนังสือเล่มนี้ คุณควรจะมีความรู้ระดับเริ่มต้นถึงระดับกลาง

ของจาวาสคริปต์ งานส่วนใหญ่ของคุณจะเสร็จสิ้นด้วยบรรทัดคำสั่ง ดังนั้น ความเข้าใจพื้นฐาน

วิธีการใช้บรรทัดคำสั่งก็จำเป็นเช่นกัน คุณควรเข้าใจด้วย

npm คืออะไรและทำงานอย่างไรในระดับพื้นฐานเป็นอย่างน้อย หากคุณกำลังสร้างใน

iOS ความเข้าใจพื้นฐานของ Xcode นั้นมีประโยชน์และจะทำให้สิ่งต่าง ๆ เร็วขึ้น แต่ก็ไม่เป็นเช่นนั้น

ที่จำเป็น. ในทำนองเดียวกัน หากคุณกำลังสร้างสำหรับ Android ความเข้าใจพื้นฐานเกี่ยวกับ Android

สตูดิโอจะเป็นประโยชน์แต่ไม่จำเป็น

ความรู้พื้นฐานเกี่ยวกับคุณลักษณะ JavaScript ที่ใหม่กว่าที่นำมาใช้ใน ES2015

การเปิดตัวภาษาโปรแกรม JavaScript นั้นมีประโยชน์ แต่ไม่จำเป็น บาง

ความรู้เชิงแนวคิดของเฟรมเวิร์ก MVC และสถาปัตยกรรมหน้าเดียวก็ดีเช่นกัน

แต่ไม่จำเป็น

Understanding how React Native works

าดูกันว่า React Native ทำงานอย่างไรโดยพูดถึง JSX, threading model, React,

ข้อมูลทิศทางเดียว ow และอื่นๆ

1 jsx :

React และ React Native ต่างก็สนับสนุนการใช้ JSX JSX นั้นเป็นส่วนขยายของไวยากรณ์โดยพื้นฐาน

ไปยัง JavaScript ที่มีลักษณะคล้าย XML คุณสามารถสร้างส่วนประกอบ React Native ได้

ไม่มี JSX แต่ JSX ทำให้ React และ React Native อ่านง่ายและง่ายขึ้นมาก

บำรุงรักษา. JSX อาจดูแปลกในตอนแรก แต่ทรงพลังอย่างมาก และคนส่วนใหญ่

เติบโตเพื่อรักมัน

2 Threading

การทำงานของ JavaScript ทั้งหมด เมื่อโต้ตอบกับแพลตฟอร์มดั้งเดิม จะทำในแยกต่างหาก

ซึ่งช่วยให้ส่วนต่อประสานกับผู้ใช้และแอนิเมชั่นทำงานได้อย่างราบรื่น

เธรดนี้เป็นที่ที่แอปพลิเคชัน React ใช้งานอยู่ และการเรียกใช้ API ทั้งหมด เหตุการณ์การสัมผัส

และมีการประมวลผลการโต้ตอบ เมื่อมีการเปลี่ยนแปลงในองค์ประกอบที่ได้รับการสนับสนุนโดยเนทีฟ

การอัปเดตจะถูกแบทช์และส่งไปยังฝั่งเนทีฟ สิ่งนี้จะเกิดขึ้นเมื่อสิ้นสุดการวนซ้ำแต่ละครั้ง

ของวงรอบเหตุการณ์ สำหรับแอปพลิเคชัน React Native ส่วนใหญ่ ตรรกะทางธุรกิจจะทำงานบน

เธรด JavaScript

3 React

คุณสมบัติที่ยอดเยี่ยมของ React Native คือใช้ React React เป็น JavaScript แบบโอเพ่นซอร์ส

ห้องสมุดที่สนับสนุนโดย Facebook เดิมทีได้รับการออกแบบมาเพื่อสร้างแอปพลิเคชัน

และแก้ปัญหาบนเว็บ เฟรมเวิร์กนี้ได้รับความนิยมอย่างมากตั้งแต่

การเปิดตัวโดยมีบริษัทที่จัดตั้งขึ้นหลายแห่งใช้ประโยชน์จากการแสดงผลอย่างรวดเร็ว

การบำรุงรักษา และ UI ที่เปิดเผย เหนือสิ่งอื่นใด

การจัดการ DOM แบบดั้งเดิมนั้นช้าและมีราคาแพงในแง่ของประสิทธิภาพและ

ควรย่อให้เล็กสุด React ข้าม DOM ดั้งเดิมด้วยสิ่งที่เรียกว่า

virtual DOM: โดยพื้นฐานแล้ว สำเนาของ DOM จริงในหน่วยความจำที่เปลี่ยนแปลงเฉพาะเมื่อ

เปรียบเทียบ DOM เสมือนเวอร์ชันใหม่กับ DOM เสมือนเวอร์ชันเก่า นี้

ลดจำนวนการดำเนินการ DOM ที่จำเป็นเพื่อให้ได้สถานะใหม่

4 Unidirectional

การไหลของข้อมูลแบบทิศทางเดียว

React และ React Native เน้นย้ำข้อมูลแบบทิศทางเดียวหรือทางเดียว เพราะว่า

วิธีสร้างแอปพลิเคชัน React Native ข้อมูลทางเดียวนี้ทำได้ง่าย

5 Diffing

React นำแนวคิดของ dif ng และนำไปใช้กับองค์ประกอบดั้งเดิม ต้องใช้ UI และ .ของคุณ

ส่งข้อมูลจำนวนน้อยที่สุดไปยังเธรดหลักเพื่อแสดงผลด้วยส่วนประกอบดั้งเดิม

UI ถูกแสดงอย่างเปิดเผยตามสถานะและ React ใช้dif ngto

ส่งการเปลี่ยนแปลงที่จำเป็นข้ามสะพาน

######## Summary : part 1 :

1 State เป็นวิธีการจัดการข้อมูลในส่วนประกอบ React กำลังอัปเดตสถานะแสดงผล

UI ของส่วนประกอบและส่วนประกอบย่อยใดๆ ที่ใช้ข้อมูลนี้เป็นอุปกรณ์ประกอบฉาก

2 คุณสมบัติ (อุปกรณ์ประกอบฉาก) คือวิธีที่ข้อมูลถูกส่งผ่านผ่านแอปพลิเคชัน React Native

ถึงส่วนประกอบลูก การอัปเดตอุปกรณ์ประกอบฉากจะอัปเดตส่วนประกอบต่างๆ โดยอัตโนมัติ

ได้รับพร็อพเดียวกัน

3 ข้อมูลจำเพาะส่วนประกอบ React คือกลุ่มของวิธีการและคุณสมบัติใน React

องค์ประกอบที่ระบุการประกาศส่วนประกอบ เรนเดอร์เท่านั้น

วิธีที่จำเป็นเมื่อสร้างองค์ประกอบ React วิธีอื่นๆ และ

คุณสมบัติเป็นตัวเลือก

4

มีสามขั้นตอนหลักในวงจรชีวิตของส่วนประกอบ React: การสร้าง (การติดตั้ง)

การอัปเดตและการลบ (ยกเลิกการต่อเชื่อม) แต่ละคนมีวงจรชีวิตของตัวเอง

วิธีการ

5

React lifecycle อยู่ในองค์ประกอบ React และดำเนินการที่

จุดเฉพาะในวงจรชีวิตของส่วนประกอบ พวกเขาควบคุมวิธีการส่วนประกอบ

ฟังก์ชั่นและการอัพเดท

##### Building your 􀁋rst

React Native app

เมื่อเรียนรู้กรอบ เทคโนโลยี ภาษา หรือแนวคิดใหม่ๆ ให้ดำน้ำโดยตรง

ในกระบวนการโดยการสร้างแอปจริงเป็นวิธีที่ดีในการเริ่มต้นกระบวนการเรียนรู้

ตอนนี้คุณเข้าใจพื้นฐานของวิธีการทำงานของ React และ React Native แล้ว มาทำกัน

นำชิ้นส่วนเหล่านี้มารวมกันเพื่อสร้างแอปแรกของคุณ: แอปสิ่งที่ต้องทำ เข้าสู่กระบวนการ

ของการสร้างแอพขนาดเล็กและการใช้ข้อมูลที่เราได้ทำไปแล้วจะเป็น

วิธีที่ดีในการเสริมสร้างความเข้าใจเกี่ยวกับวิธีใช้ React Native

คุณจะใช้ฟังก์ชันบางอย่างในแอปที่เรายังไม่ได้กล่าวถึงในเชิงลึกและ

ความแตกต่างด้านสไตล์บางอย่างที่เรายังไม่ได้พูดถึง แต่ไม่ต้องกังวล แทนที่จะข้ามสิ่งเหล่านี้

ไอเดียใหม่ๆ ทีละอย่าง คุณจะสร้างแอปพื้นฐานแล้วเรียนรู้เกี่ยวกับแนวคิดเหล่านี้

โดยละเอียดในบทต่อไป ใช้โอกาสนี้เพื่อเล่นกับแอพ as

คุณสร้างมันขึ้นมาเพื่อเรียนรู้ให้มากที่สุดเท่าที่เป็นไปได้ในกระบวนการ: อย่าลังเลที่จะทำลายและ x สไตล์

และส่วนประกอบเพื่อดูว่าเกิดอะไรขึ้น

Summary :

1 AppRegistry เป็นจุดเริ่มต้น JavaScript เพื่อเรียกใช้แอป React Native ทั้งหมด

2 extInput คอมโพเนนต์ React Native จะคล้ายกับอินพุต HTML คุณสามารถ

ระบุอุปกรณ์ประกอบฉากหลายรายการรวมถึงตัวยึดเพื่อแสดงข้อความต่อหน้าผู้ใช้

เริ่มพิมพ์ placeholderTextColor ที่จัดรูปแบบข้อความตัวแทนและ

การเลือกสีที่จัดรูปแบบเคอร์เซอร์สำหรับ TextInput

3

TouchableHighlight เป็นวิธีหนึ่งในการสร้างปุ่มใน React Native; มันเทียบได้

ไปยังองค์ประกอบปุ่ม HTML คุณสามารถใช้ TouchableHighlight เพื่อห่อ

มุมมองและทำให้พวกเขาตอบสนองต่อเหตุการณ์สัมผัสได้อย่างถูกต้อง

4 คุณได้เรียนรู้วิธีเปิดใช้งานเครื่องมือสำหรับนักพัฒนาทั้งในอีมูเลเตอร์ iOS และ Android

5 การใช้คอนโซล JavaScript (มีให้ในเมนูนักพัฒนา) เป็นวิธีที่ดี

เพื่อแก้ไขข้อบกพร่องของแอปและบันทึกข้อมูลที่เป็นประโยชน์

#### Summary seting up your Environment

ในบทนี้ เราได้ทำตามขั้นตอนทั้งหมดที่จำเป็นสำหรับการเริ่มต้นใช้งาน

การพัฒนาแอพ React Native รวมถึงการเริ่มต้นโปรเจ็กต์ใหม่ การจำลองการรันของคุณ

โปรเจ็กต์ใหม่บนคอมพิวเตอร์ของคุณ และรันแอพการพัฒนาของคุณบนอุปกรณ์ในโลกแห่งความเป็นจริง

ต้องขอบคุณพลังของ Expo ที่ทำให้กระโดดเข้าและเริ่มทำงานได้ง่ายกว่าที่เคย

เมื่อคุณได้เตรียมทุกอย่างเรียบร้อยแล้ว ก็ถึงเวลาเริ่มสร้าง!

building project in react-native

if expo cli

1 npm install -g expo-cli

2 expo init AwesomeProject

3 npm start (run)

if react-native cli:

1 npm install -g react-native-cli

2 npx react-native init AwesomeProject

3 npx react-native start (run)

part 2 :

ต้องใช้พรสวรรค์ในการสร้างแอปพลิเคชันมือถือ แต่ต้องใช้สไตล์เพื่อทำให้แอปพลิเคชันยอดเยี่ยม ถ้า

คุณเป็นนักออกแบบกราฟิก คุณรู้เรื่องนี้โดยสัญชาตญาณ ลึกลงไปในกระดูกของคุณ ถ้าคุณเป็น

นักพัฒนาคุณอาจจะคร่ำครวญและกลอกตา ไม่ว่าในกรณีใดความเข้าใจ

พื้นฐานของการจัดสไตล์ส่วนประกอบ Reactive Native มีความสำคัญต่อการสร้าง

แอปพลิเคชันที่น่าสนใจที่ผู้อื่นต้องการใช้

คุณมีประสบการณ์กับ CSS มาบ้างแล้ว แม้ว่าจะไม่มีอะไรมากไปกว่า

เห็นไวยากรณ์ คุณสามารถเข้าใจได้ง่ายว่ากฎ CSS ชอบสีพื้นหลังอย่างไร:

'สีแดง' มีขึ้นเพื่อทำ เมื่อคุณเริ่มอ่านบทนี้ อาจดูเหมือนเป็นสไตล์

คอมโพเนนต์ใน React Native นั้นง่ายพอๆ กับการใช้ชื่อ camelCase สำหรับกฎ CSS

ตัวอย่างเช่น การตั้งค่าสีพื้นหลังบนส่วนประกอบ React Native นั้นใช้เกือบ

ไวยากรณ์เดียวกัน backgroundColor: 'red'—แต่ขอเตือนไว้ก่อนว่านี่คือความคล้ายคลึงกัน

จบ.

1 Applying and organizing styles in React Native(การใช้และการจัดระเบียบสไตล์ใน React Native)

React Native มาพร้อมกับส่วนประกอบในตัวมากมาย และชุมชนได้สร้างมากมาย

คุณสามารถรวมเข้ากับโครงการของคุณได้มากขึ้น ส่วนประกอบรองรับชุดสไตล์เฉพาะ

สไตล์เหล่านั้นอาจใช้หรือไม่ใช้กับส่วนประกอบประเภทอื่น ตัวอย่างเช่น,

องค์ประกอบข้อความรองรับคุณสมบัติ fontWeight (fontWeight หมายถึงความหนา

ของฟอนต์) แต่คอมโพเนนต์ View กลับทำไม่ได้ ในทางกลับกัน องค์ประกอบมุมมอง

รองรับคุณสมบัติ flex (flex หมายถึงเลย์เอาต์ของส่วนประกอบภายในมุมมอง) แต่

องค์ประกอบข้อความไม่ได้

2 Applying styles in applications (การใช้สไตล์ในแอปพลิเคชัน)

1 inline style (in)

2 stylesheet (in)

3 stylesheet (out)

4 dynamic style (out)

3 style view components

เมื่อคุณมีภาพรวมที่เหมาะสมของการจัดสไตล์ใน React Native แล้ว มาพูดถึงกันมากขึ้น

สไตล์ส่วนบุคคล บทนี้ครอบคลุมคุณสมบัติพื้นฐานมากมายที่คุณจะใช้

เวลา. ในบทที่ 5 เราจะเจาะลึกและแนะนำสไตล์ที่คุณจะไม่เคยเห็น

วันและรูปแบบที่เป็นแพลตฟอร์มเฉพาะ แต่ตอนนี้ขอเน้นพื้นฐาน: ในนี้

ส่วนที่เป็นส่วนประกอบดู องค์ประกอบมุมมองเป็นส่วนประกอบหลัก

ของ UI และเป็นหนึ่งในองค์ประกอบที่สำคัญที่สุดที่ต้องทำความเข้าใจเพื่อให้ได้สไตล์ของคุณ

ขวา. โปรดจำไว้ว่า องค์ประกอบ View นั้นคล้ายกับแท็ก div ของ HTML ในแง่ที่คุณ

สามารถใช้เพื่อห่อองค์ประกอบอื่น ๆ และสร้างบล็อกของรหัส UI ในนั้น

ในขณะที่คุณดำเนินไปตามบท คุณจะใช้สิ่งที่คุณได้เรียนรู้เพื่อสร้างความจริง

ส่วนประกอบ: การ์ด Pro le การสร้าง Pro le Card จะแสดงวิธีการใส่ทุกอย่าง

ด้วยกัน. รูปที่ 4.3 แสดงว่าส่วนประกอบจะมีลักษณะอย่างไรเมื่อสิ้นสุดส่วนนี้

ในกระบวนการสร้างส่วนประกอบนี้ คุณจะได้เรียนรู้วิธีดำเนินการดังต่อไปนี้:

1 สร้างเส้นขอบรอบ ๆ คอนเทนเนอร์ pro le โดยใช้ borderWidth

2 round รอบมุมของเส้นขอบนั้นด้วยรัศมีรัศมี borderRadius

3 สร้างเส้นขอบที่ดูเหมือนวงกลมโดยใช้ borderRadius ขนาดครึ่งหนึ่งของ

ความกว้างของส่วนประกอบ

4 วางตำแหน่งทุกอย่างโดยใช้คุณสมบัติระยะขอบและช่องว่างภายใน