

Ανάπτυξη Κινητών Εφαρμογών

Εργαστήριο

Δημήτρης Ρίγγας

riggas@ionio.gr

Τμήμα Πληροφορικής
Ιόνιο Πανεπιστήμιο



Εργαστήριο 2 – React Native Development

Βασικές Έννοιες & Εγκατάσταση Εργαλείων



Στόχος:

- ✓ **Node.js:** Εγκατάσταση & Βασικές Αρχές
- ✓ **Expo:** Εναλλακτική Προσέγγιση
- ✓ **Virtual DOM:** Κατανόηση & Σύγκριση με HTML DOM
- ✓ **JSX:** Σύνταξη & Βέλτιστες Πρακτικές
- ✓ **Hands-on:** Πρώτο React Native App



Τι είναι το Node.js?

Node.js = JavaScript Runtime Environment (εκτός browser)

Βασικά Χαρακτηριστικά:

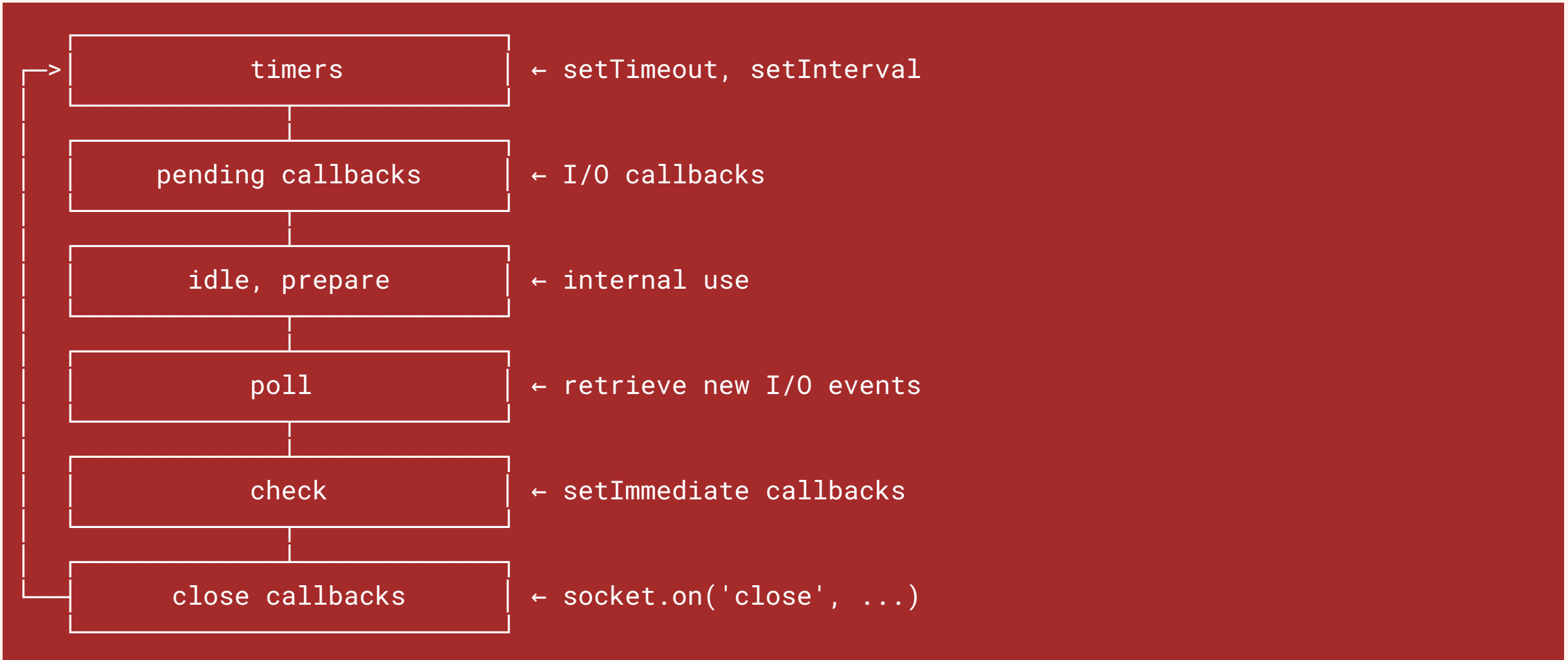
- **Event-driven architecture:** Αρχιτεκτονική βασισμένη σε events
- **Non-blocking I/O:** Ασύγχρονες λειτουργίες
- **V8 JavaScript Engine** (Chrome), **SpiderMonkey** (Firefox), **JavaScriptCore** (Apple)
- **Single-threaded** με event loop για concurrency

Γιατί Node.js για Mobile Development?

- ✓ Κοινή γλώσσα (JavaScript) για frontend & backend
- ✓ Τεράστιο ecosystem (npm packages)
- ✓ Build tools & bundlers (Metro, Webpack)
- ✓ Development server για React Native



Node.js: Event Loop Architecture



“ Πηγή: [Node.js Official Documentation](https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/) ”

Εγκατάσταση Node.js

Official Installer (Προτεινόμενη)

- Download από nodejs.org
- Προτείνεται η **LTS version** (Long Term Support)

Έλεγχος εγκατάστασης

```
node --version # π.χ. v20.10.0  
npm --version  # π.χ. 10.2.3
```

Node.js: NPM Package Manager

NPM = Node Package Manager (μεγαλύτερο software registry)

Βασικές Εντολές:

```
# Αρχικοποίηση νέου project  
npm init  
  
# Εγκατάσταση package  
npm install <package-name>  
npm install react-native --save  
  
# Εγκατάσταση dev dependency  
npm install --save-dev eslint  
  
# Εγκατάσταση global package  
npm install -g react-native-cli  
  
# Ενημέρωση packages  
npm update
```

Node.js: Module System

CommonJS (Traditional)

```
// math.js - Module Export  
module.exports = {  
  add: (a, b) => a + b,  
  subtract: (a, b) => a - b  
};  
  
// app.js - Module Import  
const math = require('./math');  
console.log(math.add(5, 3)); // 8
```

ES6 Modules (Modern)

```
// math.js - Named Export  
export const add = (a, b) => a + b;  
export const subtract = (a, b) => a - b;  
  
// app.js - Named Import  
import { add, subtract } from './math';  
console.log(add(5, 3)); // 8
```


React Native: Τι είναι;

React Native = Framework για native mobile apps με React

Βασικά Πλεονεκτήματα:

- ✓ **Cross-platform:** iOS + Android με ένα codebase
- ✓ **Native performance:** Όχι WebView, πραγματικά native components
- ✓ **Hot Reloading:** Άμεση προεπισκόπηση αλλαγών
- ✓ **React ecosystem:** Χρήση React libraries & patterns
- ✓ **JavaScript/TypeScript:** Γνωστές γλώσσες προγραμματισμού

Architecture:

JavaScript Code → Bridge → Native Modules → Native UI



React Native CLI vs Expo

Feature	React Native CLI	Expo
Native modules	✅ Πλήρης πρόσβαση	⚠️ Περιορισμένη (μέχρι eject)
Setup complexity	🔴 High (Xcode, Android Studio)	🟢 Low (Expo Go app)
Flexibility	🟢 Maximum	🟡 Good
Build time	🔴 Slow	🟢 Fast
Over-the-air updates	❌ Χρειάζεται setup	✅ Built-in
File size	🟢 Optimized	🔴 Larger

Πότε να χρησιμοποιήσετε το Expo: Prototypes, MVPs, standard features

Πότε να χρησιμοποιήσετε το CLI: Native modules, performance optimization, full control

Εγκατάσταση Expo

Expo = Managed workflow για React Native development

Setup (απλό):

```
# 1. Εγκατάσταση Expo CLI
npm install -g expo-cli

# 2. Δημιουργία νέου project
expo init MyExpoApp

# 3. Template selection
# → blank: Κενό template
# → blank (TypeScript): Με TypeScript
# → tabs: Template με navigation

# 4. Είσοδος & εκκίνηση
cd MyExpoApp
expo start
```

Expo: Development Workflow

Expo Go App:

1. Εγκατάσταση Expo Go app στο smartphone (iOS/Android)
2. Scan QR code από το terminal/browser
3. Instant preview στη συσκευή σας

Expo Dev Tools (Browser):

```
expo start
# Opens browser with:
# - QR code για device
# - Εκτέλεση σε emulator/simulator
# - Logs & diagnostics
# - Production build options
```

Advantages:

- Δεν χρειάζεται Xcode ή Android Studio για testing
- Over-the-air updates
- Managed native dependencies

Expo: Ejecting (Advanced)

Όταν χρειάζεστε native code:

```
# Eject από managed workflow
expo eject

# Τώρα έχετε:
# - ios/ folder (Xcode project)
# - android/ folder (Android Studio project)
# - Πλήρης έλεγχος native code

# Trade-off:
# ✓ Native modules access
# ✗ Χάνετε Expo managed features
# ✗ Πρέπει να διαχειρίζεστε native builds
```

Expo Bare Workflow: Alternative που παρέχει native folders από αρχή

JSX: JavaScript XML

JSX = Syntax extension για JavaScript που μοιάζει με XML/HTML

Παράδειγμα:

```
// JSX
const element = <h1>Hello, World!</h1>;

// Transpiled σε JavaScript:
const element = React.createElement(
  'h1',
  null,
  'Hello, World!'
);
```

Γιατί JSX;

- ✓ **Declarative:** Περιγράφουμε "τι" θέλουμε, όχι "πώς"
- ✓ **Familiar syntax:** Μοιάζει με HTML
- ✓ **Type safety:** Errors σε compile time (με TypeScript)
- ✓ **Performance:** Optimizations κατά τη μετατροπή

JSX: Βασικοί Κανόνες

1. Expressions σε Curly Braces:

```
const name = "Νίκος";  
const element = <h1>Γεια σου, {name}!</h1>;  
  
const getGreeting = (user) => `Hello, ${user}`;  
const greeting = <p>{getGreeting("Maria")}</p>;
```

2. Attributes:

```
// HTML-like attributes (σε camelCase)  
<img src={imageUrl} alt="Profile" />  
<div className="container" /> // className, όχι class  
<label htmlFor="name" /> // htmlFor, όχι for  
  
// Event handlers  
<button onClick={handleClick}>Click me</button>
```

JSX: Advanced Patterns

3. Conditional Rendering:

```
// Ternary operator
{isLoggedIn ? <LogoutButton /> : <LoginButton />}

// Logical AND
{showWarning && <Warning message="Be careful!" />}

// If-else με function
{renderContent()}
```

4. Lists & Keys:

```
const users = ['Alice', 'Bob', 'Charlie'];

const userList = (
  <ul>
    {users.map((user, index) => (
      <li key={index}>{user}</li>
    ))}
  </ul>
)
```



JSX σε React Native

Διαφορές από React (Web):

```
// React (Web) - HTML elements
<div>
  <h1>Title</h1>
  <p>Paragraph</p>
  <button>Click</button>
</div>

// React Native - Native components
<View>
  <Text style={styles.title}>Title</Text>
  <Text>Paragraph</Text>
  <TouchableOpacity onPress={handlePress}>
    <Text>Click</Text>
  </TouchableOpacity>
</View>
```

Βασικά Components: `View`, `Text`, `Image`, `ScrollView`, `TextInput`, `TouchableOpacity`

HTML DOM vs Virtual DOM

HTML DOM (Document Object Model):

- **Tree structure** που αναπαριστά το HTML document
- **Browser API** για manipulation (getElementById, querySelector, etc.)
- **Slow operations:** Reflow & repaint costs
- **Direct manipulation:** Αλλαγές απευθείας στο browser

Παράδειγμα DOM Manipulation:

```
// Vanilla JavaScript - DOM manipulation  
const element = document.getElementById('myDiv');  
element.style.color = 'blue';  
element.innerHTML = 'New content';  
  
// Κάθε αλλαγή → Browser reflow/repaint
```

Virtual DOM: Η Έννοια

Virtual DOM = JavaScript representation του πραγματικού DOM

Πώς λειτουργεί:

1. State Change
↓
2. Create new Virtual DOM tree
↓
3. Diff algorithm (compare old vs new)
↓
4. Calculate minimal changes
↓
5. Batch update Real DOM

Επιδίωξη: Ελαχιστοποίηση επεξεργασίας DOM $\Rightarrow$ Βέλτιστοποίηση απόδοσης

Virtual DOM: Diffing Algorithm

React Reconciliation Process:

```
// Initial render  
<div>  
  <p>Hello</p>  
  <span>World</span>  
</div>  
  
// After state update  
<div>  
  <p>Hello</p>  
  <span>Universe</span> // Only this changed!  
</div>
```

Reconciliation:

1. React compares Virtual DOM trees
2. Detects only `<span>` content changed
3. Updates ONLY that specific DOM node
4. Rest of the tree remains untouched

Virtual DOM: Performance Diagram

Traditional DOM Manipulation:

[State Change] → [DOM Update] → [Reflow] → [Repaint]
(Fast) (Slow) (Slow) (Slow)

⚠ Each update is expensive

Virtual DOM Approach:

[State Change] → [Virtual DOM Update] → [Diffing] → [Batch DOM Update]
(Fast) (Fast) (Fast) (Slow × 1)

✅ Multiple changes → One DOM update

Βασική Αρχή:

- Virtual DOM operations: Χιλιάδες/δευτερόλεπτο
- Real DOM operations: Δεκάδες/δευτερόλεπτο

DOM vs Virtual DOM: Σύγκριση

Aspect	HTML DOM	Virtual DOM
Nature	Browser API	JavaScript Object
Speed	● Slow (direct manipulation)	● Fast (in-memory)
Updates	Immediate (expensive)	Batched (efficient)
Reflow/Repaint	Each change	Minimized
Memory	Browser-managed	JavaScript heap
Use case	Direct browser control	React, React Native

Bottom line: Virtual DOM = Abstraction layer για performance optimization

Πρακτικό Παράδειγμα: Counter App

```
import React, { useState } from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';

const CounterApp = () => {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Counter: {count}</Text>

      <TouchableOpacity
        style={styles.button}
        onPress={() => setCount(count + 1)}
      >
        <Text style={styles.buttonText}>Increment</Text>
      </TouchableOpacity>

      <TouchableOpacity
        style={styles.button}
        onPress={() => setCount(count - 1)}
      >
        <Text style={styles.buttonText}>Decrement</Text>
      </TouchableOpacity>
    </View>
  );
};
```

Πρακτικό Παράδειγμα: Styling

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f5f5f5',
  },
  title: {
    fontSize: 32,
    fontWeight: 'bold',
    marginBottom: 20,
    color: '#333',
  },
  button: {
    backgroundColor: '#007AFF',
    paddingHorizontal: 30,
    paddingVertical: 15,
    borderRadius: 8,
    marginVertical: 10,
  },
  buttonText: {
    color: 'white',
    fontSize: 18,
    fontWeight: '600',
  },
});
```


Debugging Tools

React Native Debugger:

```
# Εγκατάσταση  
brew install --cask react-native-debugger  
  
# Χρήση:  
# Shake device → "Debug" → Opens debugger
```

Chrome DevTools:

- Console.log() output
- Network inspector
- React DevTools extension

Flipper:

- Official debugging platform από Facebook
- Network inspector, Layout inspector, Logs
- Plugin ecosystem

Και τώρα.. _hands on_!

Συνεχίζουμε στο:

<https://github.com/riggas-ionio/mobile-apps/>



Πηγές

- Node.js: nodejs.org/docs
- React Native: reactnative.dev
- Expo: docs.expo.dev
- React: react.dev
- React Native Express: reactnative.express

