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 & NPM**: Εγκατάσταση, packages, `package.json`
- ✓ **Modules**: CommonJS vs ES Modules
- ✓ **React Native & Expo**: Πρώτο project, δομή, εκτέλεση στον browser
- ✓ **JSX**: Σύνταξη & βασικοί κανόνες
- ✓ **Virtual DOM**: Κατανόηση & σύγκριση με HTML DOM
- ✓ **Hands-on**: Πρώτο Expo app + hot reload

Τι είναι το Node.js?

Node.js = JavaScript Runtime Environment (εκτός browser)

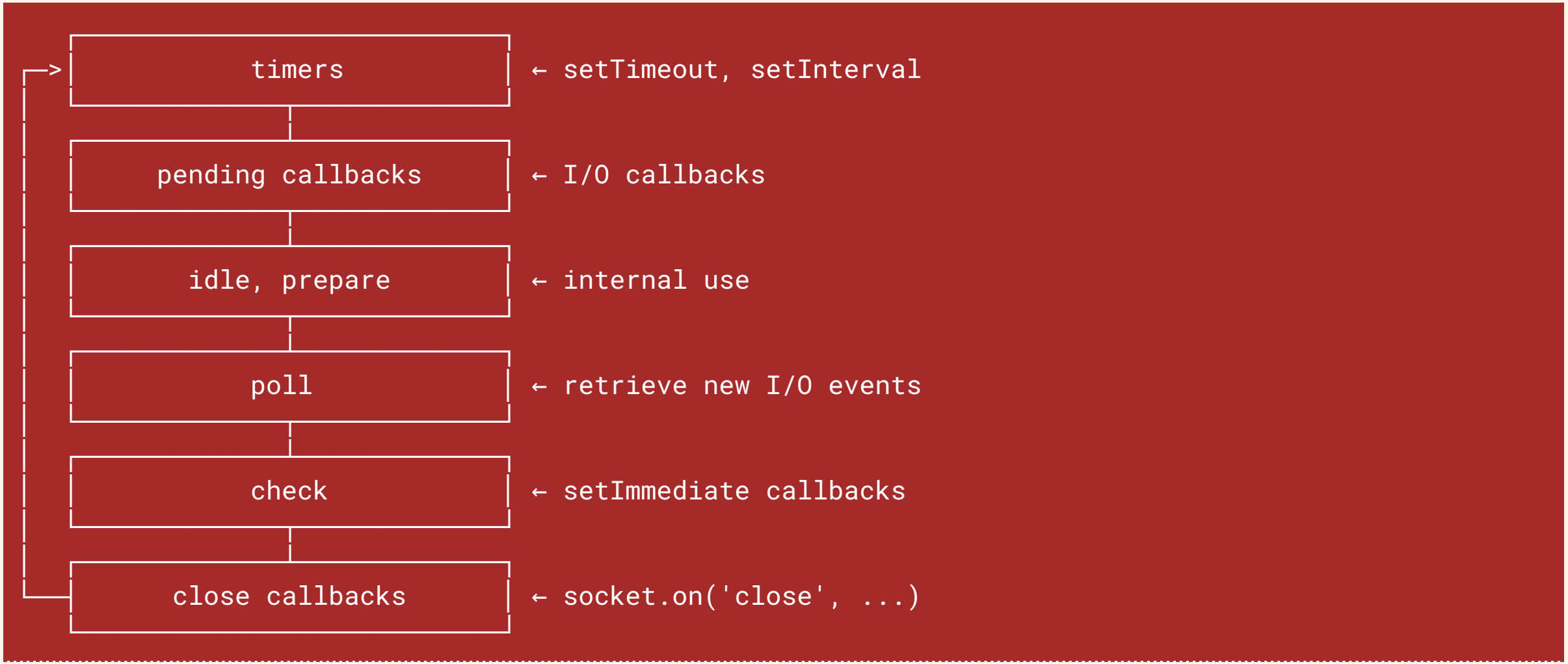
Βασικά Χαρακτηριστικά:

- **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 ”

Εγκατάσταση Node.js

Official Installer (Προτεινόμενη)

- Download από nodejs.org
- Προτείνεται η **LTS version** (Long Term Support)
- Απαιτείται **v22.13 ή νεότερο** (προτείνεται **v24 LTS**) — το Expo SDK δεν υποστηρίζει παλαιότερες

Έλεγχος εγκατάστασης (σε **νέο** terminal)

```
node --version  # π.χ. v24.11.0
npm --version   # π.χ. 11.6.1
node            # REPL → console.log("Hi") → .exit
```

Node.js: NPM Package Manager

NPM = Node Package Manager (μεγαλύτερο software registry)

```
# Αρχικοποίηση (με default τιμές)
npm init -y

# Εγκατάσταση package
npm install <package-name>
npm install chalk@4    # συγκεκριμένη major

# Dev dependency
npm install --save-dev eslint

# Script από το package.json
npm start              # ή npm run <script>

# Εργαλείο ΧΩΡΙΣ global εγκατάσταση
npx create-expo-app@latest firstapp
```

- `npm install` → κατεβάζει στο `node_modules/` και γράφει στο `package.json`
- `^4.1.2` = συμβατό με 4.x.x (όχι 5.0.0)
- `package-lock.json` → ακριβείς εκδόσεις
- Το `node_modules/` **δεν** μπαίνει σε git — αναδημιουργείται με `npm install`
- `npx` → τρέχει εργαλείο από το project ή το κατεβάζει προσωρινά

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
```

ES Modules (Modern)

```
// math.mjs - Named Export
export const add = (a, b) => a + b;
export const subtract = (a, b) => a - b;

// app.mjs - Named Import
import { add, subtract } from './math.mjs';
console.log(add(5, 3)); // 8
```

- ES Modules στο Node: αρχεία `.mjs` ή `"type": "module"` στο `package.json` · η κατάληξη στο `import` είναι **υποχρεωτική**
- ⚠️ Πολλά packages είναι πλέον **μόνο** ES Modules (π.χ. `chalk` ≥ 5) → το `require('chalk')` δεν δουλεύει
- Το Expo / React Native χρησιμοποιεί `import` / `export` (τα χειρίζεται ο Metro bundler)

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 (New Architecture, default από RN 0.76):

JavaScript Code → **JSI** (άμεση κλήση, χωρίς το παλιό asynchronous Bridge) → Native Modules / **Fabric** renderer → Native UI

React Native CLI vs Expo

Feature	React Native CLI	Expo
Native modules	✅ Πλήρης πρόσβαση	✅ Μέσω development builds / `prebuild`
Setup complexity	🔴 High (Xcode, Android Studio)	🟢 Low (browser / Expo Go)
Flexibility	🟢 Maximum	🟢 Πολύ καλή
Build	🔴 Τοπικά, χειροκίνητα	🟢 Τοπικά ή στο cloud (EAS Build)
Over-the-air updates	❌ Χρειάζεται setup	✅ EAS Update

Η επίσημη τεκμηρίωση του React Native **προτείνει** πλέον τη χρήση framework, δηλ. **Expo**
React Native CLI: όταν θέλετε πλήρη έλεγχο χωρίς framework (π.χ. ενσωμάτωση σε υπάρχουσα native εφαρμογή)

Εγκατάσταση Expo

Expo = Managed workflow για React Native development

Setup (απλό) — **χωρίς** global εγκατάσταση:

```
# 1. Δημιουργία νέου project (απλό JavaScript template)
cd ~/ReactNativeLab
npx create-expo-app@latest firstapp --template blank

# Άλλα templates: blank-typescript, tabs, default (Expo Router + TypeScript)

# 2. Είσοδος & εκκίνηση
cd firstapp
npx expo start        # ή: npm start
```

⚠️ Το παλιό global `expo-cli` (`expo init`, `expo start`) έχει **καταργηθεί**.
Το Expo CLI έρχεται μέσα σε κάθε project και καλείται με `npx expo ...`

Expo: Development Workflow

`npx expo start` (Metro bundler, port 8081)

```
› Metro waiting on exp://192.168.1.x:8081
› Scan the QR code above with Expo Go

› Web is waiting on http://localhost:8081

› Press a │ open Android
› Press i │ open iOS simulator
› Press w │ open web
› Press r │ reload app
› Press m │ toggle menu
› Press j │ open debugger
```

Σε αυτό το εργαστήριο: web (`w`)

- Χρειάζεται `react-dom` & `react-native-web` (`npx expo install react-dom react-native-web`)
- Αποθήκευση αρχείου → **Fast Refresh**
- `console.log()` → terminal **και** browser console

Στο Lab 03: Expo Go στο κινητό

- QR code → η εφαρμογή στη συσκευή σας
- Χωρίς Xcode / Android Studio

Expo: Δομή project

```
firstapp/
├── .git/              # ήδη git repository!
├── assets/            # icon, splash, favicon
├── node_modules/      # dependencies
├── App.js             # >>> Main component <<<
├── app.json           # name, version, orientation...
├── index.js           # registerRootComponent(App)
├── AGENTS.md          # οδηγίες για AI assistants
├── package.json       # scripts & dependencies
└── package-lock.json
```

Native code (advanced)

Το παλιό `expo eject` **δεν υπάρχει** πλέον:

```
# Δημιουργία ios/ & android/ φακέλων
npx expo prebuild

# Development build (δικό σας "Expo Go")
npx expo run:android
npx expo run:ios
```

- Συνεχίζετε να έχετε όλα τα Expo εργαλεία
- Builds στο cloud: **EAS Build**

JSX: JavaScript XML

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

Παράδειγμα:

```jsx
// 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 => (
      <li key={user}>{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 ⇒ 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

Προεπισκόπηση (Lab 03): 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>
  );
};
```

Προεπισκόπηση (Lab 03): 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

Στον browser (Expo web):

- **F12 / Cmd+Option+I** → Browser DevTools
- 🧱 **Elements**: πώς τα `View` / `Text` γίνονται HTML elements
- 📊 **Console**: `console.log()` & errors
- 🌐 **Network**: τι κατεβάζει ο browser από τον Metro

React Native DevTools (κινητό / emulator):

- Πατήστε **`j`** στο terminal του `npx expo start`
- Console, breakpoints (Sources), React Components & Profiler
- Αντικαθιστά τα παλιά React Native Debugger & Flipper (καταργήθηκαν)

Στο terminal:

- `console.log()` εμφανίζεται και εκεί · **`r`** reload · **`--clear`** καθαρίζει την cache

Και τώρα.. *hands on*!

Συνεχίζουμε στο:

https://github.com/riggas-ionio/mobile-apps/tree/class-2026/lab02



Πηγές

- **Node.js**: nodejs.org/docs
- **React Native**: reactnative.dev
- **Expo**: docs.expo.dev
- **React**: react.dev
- **React Native Express**: reactnative.express
- **React Native DevTools**: reactnative.dev/docs/react-native-devtools
- **Expo – Create a project**: docs.expo.dev/get-started/create-a-project