QUICK APP PRIMER

Guide to understand Quick Apps v1.0
CONTENTS

Mobile App Trends
Quick App Overview
Use Cases
Development
Architecture
Performance
MOBILE APPs TRENDS

Trends in business; trends in development; new demanded services
“NEW NORMAL” TRENDS

COVID-19 no-touch services

Scan QR-codes everywhere

Too many apps in devices

More transactional online services:
  - Quick iterations to perform concrete tasks
  - Short context of time and/or location
  - Book, pay, arrange services/products
  - Public services for citizens

Photo by Albert Hu on Unsplash
CHANGES IN BUSINESS MODELS

Mobile first approach
App stores to face new regulations
Coexistence of multiple stores
Businesses need to create more apps quickly
Experiences from any device, not only phone
Changes in the advertising models
Innovative ways to acquire users
Towards vendor neutrality & EU sovereignty

Photo by the blowup on Unsplash
QUICK APP OVERVIEW

Quick App paradigm; main features; ...
QUICK APP

A light native-like application that does not require installation and can be opened with just one tap or scanning a QR-code
FRICTIONLESS APPs

Ideal for:

- Simple defined tasks on the move
- Low friction user interactions
- Infrequent interactions
- User’s instant gratification (low time-to-value)

Avoiding complex workflows and intermediate steps
Focus on specific tasks
Instant access (no install)
No registration friction
No payment friction
Quick App is a framework for mobile application development, based on the front-end Web technology stack, that offers maximized performance and enhanced user experience through the use of device’s native resources and services.
PRODUCT’S LIFECYCLE SUPPORT

**DISCOVERABILITY**
Discover the quick app using:
- QR-codes; search results; websites; catalogs; text messages; push notifications;
- marketplaces; in-app links; web links; widgets; global device search; device assistant; ...

**USER ACQUISITION**
Customers access the quick app through:
- scanning a QR-code, tapping on a link, in one step. No installation is required. Native authentication mechanisms to identify the user and personalize the service.

**MONETIZATION**
Developers can monetize the quick app:
- displaying in-app native ads in different formats; in-app reward ad videos; in-app purchases offering different payment methods.

**USER RETENTION**
User engagement in the service (quick app) through:
- push notifications at system level with personalized messages; possibility to bookmark the app; keeping the status of the quick app last interaction for future access; marketing in the stores.
DISCOVERABILITY
USER ACQUISITION
MONETIZATION
USER RETENTION

**Promotion within devices**
- Widgets on dashboard
- Global search results
- No store dependency
DISCOVERABILITY
USER ACQUISITION
MONETIZATION
USER RETENTION

Promotion outside devices
Web sites (URL links)
Links from other apps
Listed in app marketplaces
DISCOVERABILITY
USER ACQUISITION
MONETIZATION
USER RETENTION

QR code & open the app
Quick app marketplaces
Recently used apps
Huge discoverability
Just one tap and access
No installation required
User identification
Open and authorize
DISCOVERABILITY
USER ACQUISITION
MONETIZATION
USER RETENTION

In-app payments
In-app ads
Rewarded ad videos
DISCOVERABILITY
USER ACQUISITION
MONETIZATION
USER RETENTION

Push notifications
Banners in other apps
Tap and access the app
USE CASES

Transactional services; e-commerce; quick social interactions; no-touch services; casual games; ...
PRODUCTIVITY

For eventual needs
Task-oriented tools
Just tap and use
Dispose after use
Light and easy-to-use
SHOPPING

Marketing campaigns
Push notifications
In-App advertising
Tap and buy
QR code and buy
Huawei ID in one tap
COVID-19

- Touchless services
- Emergencies
- Nearby facilities & services
- Alert notifications
- Exposure warnings
- First aid
PUBLIC SERVICES

Touchless, online services
Covering citizens’ needs
Scan QR-code and access
Utility, taxes payments
Smart city productivity
QUICK GAMES

Casual light games
Easy to discover and play
May include Ads and iAPs
One-click authentication
Social interactions
QUICK GAMES

High user acquisition
Simplified version
Trial, preview version
RESTAURANTS & BARS

Touchless service
QR code and access menu
Targeted discounts
Recommendations
Complete dish information
Access to location, orders
Book, order, pay
TRANSPORT

Booking on the go
QR-code, book and pay
In-journey information
Context based on location
Real-time alerts
Status of the service
Push notifications
IN-VEHICLE INFORMATION & ENTERTAINMENT

Third-party apps on the car's head unit

In-car entertainment functions:
• reading news;
• listening to stories;
• watching videos;
• playing games;
• ...

Third-party apps on the car's head unit

In-car entertainment functions:
• reading news;
• listening to stories;
• watching videos;
• playing games;
• ...

...
ALSO AS WIDGETS

Display on the home screen
Direct user interaction
QUICK APP DEVELOPMENT

Features; IDE; MVVM framework; UI components; packaging; distribution...
KEY FEATURES

MVVM framework based on development trends
W3C standard based on HTML, JS, CSS
Declarative UI
Native features (calendar, hardware...)
Supported by 25% of device vendors
Ready-to-use modules
Low learning curve
Easy to convert Web Apps to Quick Apps
IDE WITH TEMPLATES
import router from '@system.router';

export default {
  props: ['websrc'],
  data: {
    title: 'W3C on Quick App',
    loadUrl: 'https://www.w3.org',
    allowThirdPartyCookies: true,
    supportZoom: true,
    wideviewport: true,
    overViewModeLoad: true,
    userAgent: 'default',
    list: ['New RegExp(https://.*)',
    multiwindow: false,
    jumpPolicy: 'default',
  },
  onInit() {
    console.info('onInit: '); 
  },
  onPageStart(e) {
  },
}
CUSTOMIZED WIDGETS: VISUAL UI DEVELOPMENT
MVVM FRAMEWORK

Vue.js-like
Declarative rendering
Reusable UI components
Reactive HTML templates
One-way data flow
Event binding
Virtual DOM manipulation
CSS Stylesheets
QUICK PROTOTYPING

Conditionals and loops
HTML-like markup
UI components
SIMPLE APP SETUP

Configuration in manifest
Description of the app
Main look-and-feel
Per-page customization
Permissions declaration
Per-page routing

```json
{
    "package": "org.example.quickapp",
    "name": "My first quick app",
    "icon": "Common/my-local-logo.png",
    "display": {
        "backgroundColor": "#FFFFFF",
        "titleBarBackgroundColor": "#007DFF",
        "titleBarTextColor": "#ffffff",
        "menu": true,
        "textSizeAdjust": "auto",
        "pages": {
            "main": {
                "fullscreen": true,
                "orientation": "landscape",
                "menu": false,
                "titleBar": false
            }
        }
    },
    "features": [
        { "name": "system.alarm" },
        { "name": "system.clipboard" },
        { "name": "system.router" },
        { "name": "system.contact" },
        { "name": "service.push" }
    ],
    "router": {
        "entry": "main",
        "pages": {
            "main": {
                "component": "main"
            },
            "component/checkout": {
                "component": "index"
            }
        }
    },
    "trustedSslDomains": [ "expired.badssl.com" ]
}
...
Main app-level events
Granular page-level events
Global methods
Global variables

```javascript
<script>
module.exports = {
  onCreate()
  console.info('Application onCreate')
},
onRequest()
  console.info('Application onRequest')
},
onShow()
  console.info('Application onShow')
},
onHide()
  console.info('Application onHide')
},
onDestroy()
  console.info('Application onDestroy')
},
onError(error)
  console.info('Application onError')
},
// Method exposed to all pages
methodApp()
  console.info('This is a app-level method')
},
// Data exposed to all pages
dataApp:
  name: 'this is app-level data'
}
</script>
ADVANCED FEATURES

Internationalization
Multilingual strings
Access to system’s locale
Development
Simulation & Testing
[Generate signature]
Build and sign package
Signed release (RPK)
Ready to publication
**DISTRIBUTION FLOW**

**DEVELOPER**

(1) write

SOURCE CODE (JS, HTML, CSS)

(2) build

APP PACKAGE (ZIP)

manifest.json
app.ux
page
page

(3) upload

APP PACKAGE (app.rpk)

APP STOREs

(3) fetch

**USER**

(1) tap

ENTRY SCENARIOS (link, QR code,..)

(2) launch

QUICK APP PLATFORM (on device)

(4) run

(5) interact

QUICK APP (on device)

**ENTRY SCENARIOS**

(entry, QR code,..)

**QUICK APP PLATFORM**

(on device)
A/B Testing
Change how your app works and looks without requiring user updates.

My apps    My projects    App analytics    AppGallery Download

APP GALLERY CONNECT: PROMOTE THE APP
App information

This information will be displayed on HUAWEI AppGallery. Any changes will be released with the next version of the app.

The app information has been modified. Please continue to fill in the version information. Submit version

* Visual assets

Upload visual assets that will be displayed on the app details page. Uploading in-app screenshots is mandatory. Other visual assets, such as videos, are optional but will greatly improve the display effect of your app in HUAWEI AppGallery.

* Screenshots
HUAWEI Analytics provides you with data such as impressions, downloads, and in-app purchases, as well as the description of some fields. For details about the indicators, please refer to [Guide for Checking Analysis Data in AppGallery Connect](#).

**App Impressions 175**

**Details page views 135**

**New downloads 0**
QUICK APP ARCHITECTURE

Implementation; components; basic APIs; ...
QUICK APP PLATFORM ARCHITECTURE
RUNTIME FRAMEWORK

JS ENGINE

PAGE 1
(JavaScript)

PAGE 2
(JavaScript)

RENTER ENGINE

PAGE 1
(<template>, CSS)

PAGE 2
(<template>, CSS)

API

event

data

API

event

JS BRIDGE

NETWORK

STORAGE

FILE SYSTEM

SENSORs

CAMERA

...
QUICK APP BUILT-IN UI COMPONENTS
PERFORMANCE

Benchmark: Quick Apps, Web applications, Android apps, and games
PERFORMANCE OF QUICK APPS

Comparison of applications performing the same function (source: Huawei 2019)
GAME SPRITES IN USE
PERFORMANCE (in frames per second)

<table>
<thead>
<tr>
<th>HTML5 + Browser</th>
<th>Huawei Quick Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>60 fps</td>
</tr>
<tr>
<td>1000</td>
<td>60 fps</td>
</tr>
<tr>
<td>2000</td>
<td>60 fps</td>
</tr>
<tr>
<td>3000</td>
<td>45 fps</td>
</tr>
<tr>
<td>4000</td>
<td>30 fps</td>
</tr>
<tr>
<td>5000</td>
<td>15 fps</td>
</tr>
<tr>
<td>75 fps</td>
<td></td>
</tr>
</tbody>
</table>

(Source: Huawei 2019)
THANK YOU

Quick App Initiative

https://quick-app-initiative.ow2.io