



The Definitive Guide to Modern Java Clients with JavaFX

Cross-Platform Mobile and Cloud Development

Stephen Chin, Johan Vos and James Weaver

With Contributions by:

Gail Anderson, Paul Anderson, Bruno Borges, Anton Epple,
Weiqi Gao, Jonathan Giles, José Pereda, Sven Reimers,
Eugene Ryzhikov, and William Antônio Siqueira

Apress®

The Definitive Guide to Modern Java Clients with JavaFX

**Cross-Platform Mobile and
Cloud Development**

**Stephen Chin
Johan Vos
James Weaver**

With Contributions by:

**Gail Anderson, Paul Anderson, Bruno Borges, Anton Epple,
Weiqi Gao, Jonathan Giles, José Pereda, Sven Reimers,
Eugene Ryzhikov, and William Antônio Siqueira**

Apress®

The Definitive Guide to Modern Java Clients with JavaFX: Cross-Platform Mobile and Cloud Development

Stephen Chin
Belmont, CA, USA

Johan Vos
Leuven, Belgium

James Weaver
Marion, IN, USA

ISBN-13 (pbk): 978-1-4842-4925-3
<https://doi.org/10.1007/978-1-4842-4926-0>

ISBN-13 (electronic): 978-1-4842-4926-0

Copyright © 2019 by Stephen Chin, Johan Vos and James Weaver

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail editorial@apress.com; for reprint, paperback, or audio rights, please email bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484249253. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Table of Contents

About the Authors	xiii
About the Contributors	xv
About the Technical Reviewer	xix
Foreword	xxi
Chapter 1: Getting Started with Client Java	1
Java Client Technology in Action.....	1
Java Client in Business.....	2
Gaming and 3D.....	3
Mobile Conference Apps	5
A Modern Approach to Client Java.....	6
Target Mobile First.....	6
Build for the Cloud.....	8
Package Your Platform	10
Setting Up Your Environment	11
Mac OS X JDK Installation	12
Windows JDK Installation.....	13
Linux JDK Installation	17
JavaFX Installation	17
Your First Modern Java Client	19
Coding Client Apps with IntelliJ IDEA.....	19
Rapid Application Development with Scene Builder.....	25
The Path to Modern Client Development	32

Chapter 2: JavaFX Fundamentals 33

- JavaFX Stage and Scene Graph 33
 - JavaFX Is Single-Threaded 34
 - Hierarchical Node Structure 34
- A Simple Shape Example 36
- Color 38
- Text Is a Shape 39
- The JavaFX Coordinate System 39
- Layout Controls 40
 - StackPane 40
 - AnchorPane 41
 - GridPane 41
 - FlowPane and TilePane 42
 - BorderPane 42
 - SplitPane 43
 - HBox, VBox, and ButtonBar 43
- Make a Scene 43
- Enhancing the MyShapes Application 44
 - Linear Gradient 44
 - DropShadow 45
 - Reflection 46
- Configuring Actions 47
- Animation 48
- JavaFX Properties 50
 - Property Listeners 51
 - Binding 54
- Using FXML 57
 - Controller Class 60

Putting It All Together	66
Master-Detail UI.....	66
Observable Lists	71
Person UI Application Actions	74
Key Point Summary.....	79
Chapter 3: Properties and Bindings.....	81
Key Concepts	82
Observable and InvalidationListener	83
ObservableValue and ChangeListener	83
WritableValue and ReadOnlyProperty	84
JavaFX Properties.....	84
Creating Bindings.....	89
JavaFX Bindings	89
Create Bindings by Direct Extension	90
Type-Specific Specializations	92
Factory Methods in Bindings	95
Create Bindings with the Fluent API.....	98
Observable Collections.....	104
Factory and Utility Methods in FXCollections	104
Change Listeners for Observable Collections	109
Create Bindings for Observable Collections	124
JavaFX Beans.....	125
Eagerly Instantiated Property	126
Half-Lazily Instantiated Property	128
Fully Lazily Instantiated Property	129
Selection Bindings.....	130
Adapting Java Beans	133
Summary.....	140
Resources	141

- Chapter 4: JavaFX Controls Deep Dive..... 143**
- The UI Controls Module..... 144
- What Is a UI Control?..... 144
- JavaFX Basic Controls..... 145
 - Labeled Controls..... 145
 - Text Input Controls..... 151
 - Other Simple Controls..... 154
- Container Controls 156
 - Accordion and TitledPane 156
 - ButtonBar 157
 - ScrollPane 158
 - SplitPane 160
 - TabPane..... 161
 - ToolBar..... 163
- Other Controls 164
 - HTMLEditor 164
 - Pagination..... 165
 - ScrollBar 166
 - Separator..... 166
 - Spinner 167
 - Tooltip..... 168
- Popup Controls..... 169
 - Menu-Based Controls 169
 - ComboBox-Based Controls 177
- JavaFX Dialogs..... 181
 - Alert 181
 - ChoiceDialog..... 182
 - TextInputDialog..... 183
 - Dialog and DialogPane..... 184

Advanced Controls	185
ListView	185
TreeView	191
TableView	192
TreeTableView.....	198
Selection and Focus Models	200
SelectionMode.....	200
FocusModel	201
Summary.....	202
Acknowledgments.....	202
Chapter 5: Mastering Visual and CSS Design	203
Introduction to Cascading Style Sheets	203
Selectors Based on Class Name	206
Selectors Based on Custom Style Classes	206
Selectors Based on Object ID	206
Applying CSS Styles	207
Loading CSS Stylesheets.....	207
Applying CSS Styles to JavaFX Nodes.....	208
Advanced CSS Techniques	211
Using Descendant Selectors.....	211
Using Pseudo-classes	212
Using Imports	213
Font Loading in the Stylesheet.....	213
Reusing Styles.....	213
Using Advanced Color Definitions.....	214
Using Linear Gradients	214
Using Radial Gradients	215
Using Image Pattern	216
Using RGB Color Definitions	217

TABLE OF CONTENTS

- Using HSB Color Definitions 218
- Using Color Functions..... 219
- Using Effect Definitions 220
- Useful Tips and Tricks..... 221
- Advanced CSS API..... 225
- CSS in JavaFX Applications: Summary 233
- Chapter 6: High-Performance Graphics 235**
- Using Canvas 236
- Giving Life to a Canvas Application 242
- Particle Systems 247
- Fractals 258
- High Performance 271
- Conclusion 284
- Chapter 7: Bridging Swing and JavaFX 285**
- Integrating JavaFX into Swing 285
- JFXPanel: Swing Component with JavaFX Inside..... 286
- Threading 291
- Interaction Between Swing and JavaFX..... 293
- Drag and Drop with JavaFX and Swing 301
- JavaFX 3D Integrated in Swing 304
- Integrating Swing into JavaFX..... 306
- Large-Scale Integrations 308
- Conclusion 309
- Chapter 8: JavaFX 3D 311**
- Prerequisites..... 311
- Getting Started with Shapes 312
- Shape3D 318
- Sphere 319
- Box..... 319

Cylinder	320
Creating User-Defined 3D Shapes	320
Camera.....	334
ParallelCamera	335
PerspectiveCamera	335
Light.....	339
Understanding the LightBase Class.....	340
Understanding the AmbientLight Class	340
Understanding the PointLight Class	340
Material.....	346
Understanding the PhongMaterial Class	347
Adding Texture to 3D Shapes.....	351
Interacting with JavaFX 3D Scenes	356
Understanding the PickResult Class.....	357
Third-Party Software: FXyz 3D	362
FXyz 3D Sample.....	363
Conclusion.....	365
Chapter 9: JavaFX, the Web, and Cloud Infrastructure.....	367
Integrating with the Web.....	369
Displaying a Web Page	369
Adding Navigation and History	370
Showing Loading Progress.....	373
Executing JavaScript.....	374
Registering Java Callbacks	380
FXML and the Model View ViewModel Pattern	382
Writing a Controller for HTML.....	384
Adding the HTML View.....	388
HTML Data Binding Explained	390
Displaying a Map with DukeScript	394
From Web Sites to APIs.....	396

TABLE OF CONTENTS

- Building for the Cloud 396
 - Architecture of a JavaFX Cloud Application 397
 - Use Case: Querying OpenWeather 397
- Conclusion 409
- Chapter 10: Packaging Apps for the Desktop 411**
 - Web vs. Desktop Applications 411
 - Evolutions in Application Deployment 413
 - The jpackage Tool 415
 - What Is jpackage? 415
 - Using jpackage 417
 - Using Graal’s Native 438
 - Requirements 439
 - The Code 440
 - Maven Project 441
 - Gradle Project 443
 - Build the Project 444
 - Compile 445
 - Conclusion 448
- Chapter 11: Native Mobile Apps for iOS and Android 451**
 - Why JavaFX on Mobile 452
 - Different Approaches for Mobile Apps 453
 - OS-Specific Native Controls 454
 - Mobile Web Sites 454
 - Device Native Rendering 454
 - Hello, JavaFX on iOS 456
 - Client Plugins to Reduce Complexity 456
 - The Development Flow 457
 - The Code 459

How Does It Work?	470
Using the Plugin Options	471
bundlesList	472
resourcesList	472
reflectionList	473
jniList	473
delayInitList	473
releaseSymbolsList	473
Creating Real Mobile-Looking Apps	473
Different Stylesheets	474
Mobile-Specific Controls	478
Summary.....	489
Chapter 12: JavaFX 11 on Raspberry Pi	491
Intro to Raspberry Pi	491
Getting Started with a Raspberry Pi.....	492
Initial Kit	492
Installing Java 11	500
Installing JavaFX 11	503
Running JavaFX Applications Remotely	509
Working with Dependencies.....	516
Deploy and Test	548
Conclusions.....	550
Chapter 13: Machine Learning and JavaFX	551
What Is Machine Learning	551
Supervised Learning	552
Unsupervised Learning	552
Artificial Neural Networks	552
Convolutional Neural Networks.....	554

TABLE OF CONTENTS

- Eclipse DeepLearning4J: Java API for Neural Networks 556
- Training Neural Networks from a JavaFX Application 559
- Read an Image from JavaFX to a Neural Network 565
- Detecting Objects in a Video 569
- Chapter 14: Scientific Applications Using JavaFX..... 579**
- JavaFX for Space Exploration 579
- JavaFX for Quantum Computing 581
- Using JShell 584
- Using JShell..... 586
- About ND4J..... 590
- Using ND4J in JShell 594
- USING JavaFX in JShell 598
- Conclusion 609
- Index..... 611**

About the Authors

Stephen Chin is Head of Developer Relations at JFrog and a Java Champion. He has keynoted numerous Java conferences around the world including Devoxx, JNation, JavaOne, Joker, and Open Source India. Stephen is an avid motorcyclist who has done evangelism tours in Europe, Japan, and Brazil, interviewing hackers in their natural habitat. When he is not traveling, he enjoys teaching kids how to do embedded and robot programming together with his teenage daughter. You can follow his hacking adventures at: <http://steveonjava.com/>.

Johan Vos started to work with Java in 1995. He was part of the Blackdown team, porting Java to Linux. His main focus is on end-to-end Java, combining backend systems and mobile/embedded devices. He received a Duke Choice award in 2014 for his work on Javafx on mobile.

In 2015, he co-founded Gluon, which allows enterprises to create (mobile) Java Client applications leveraging their existing backend infrastructure. Gluon received a Duke Choice award in 2015.

Johan is a Java Champion, a member of the BeJUG and Devoxx steering groups, and is a JCP member. He is one of the lead authors of the *Pro JavaFX* books (published by Apress), the author of *Quantum Computing for Java Developers* (Manning), and he has been a speaker at numerous conferences on Java.

Johan contributes to a number of projects, including OpenJFX, OpenJDK, GraalVM., and is the project lead for OpenJDK Mobile and the co-lead for OpenJFX.

James Weaver is a developer, author, and speaker with a passion for quantum computing and JavaFX. He is a Java Champion and a JavaOne Rockstar. James has written books including *Inside Java*, *Beginning J2EE*, the *Pro JavaFX* series, and *Raspberry Pi with Java*. As an IBM quantum developer advocate, James speaks internationally at quantum and classical computing conferences. He tweets as @JavaFXpert and blogs at <http://JavaFXpert.com> and <http://CulturedEar.com>.

About the Contributors



Gail C. Anderson is a Java Champion, Oracle Groundbreaker Ambassador, and a past member of the NetBeans Dream Team. She is the director of research and founding member of the Anderson Software Group, a leading provider of training courses in Java, JavaFX, Python, Go, Modern C++, and other programming languages.

Gail enjoys researching and writing about leading-edge Java technologies. She is the co-author of eight textbooks on software programming, including *JavaFX Rich Client*

Programming on the NetBeans Platform.

Gail has conducted technical sessions and hands-on labs at Oracle Code One and NetBeans Day conferences in San Francisco, Europe, and Latin America. Gail has also presented sessions at Devvxx and Devnexus.

For more information about Gail, visit asgtech.com, the Anderson Software Group on Facebook, and [@gail_asgtech](https://twitter.com/gail_asgtech) on Twitter.



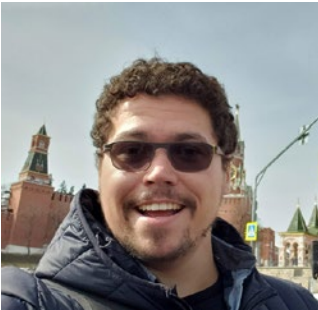
Paul L. Anderson is a Java Champion, Oracle Groundbreaker Ambassador, and a past member of the NetBeans Dream Team. He is the director of training and founding member of the Anderson Software Group, a leading provider of training courses in Java, JavaFX, Python, Go, Modern C++, and other programming languages.

Paul is an experienced speaker and specializes in making the technical aspects of software engineering fun and understandable. He is the co-author of eight textbooks on software programming, including *JavaFX Rich Client Programming on the NetBeans Platform*.

Paul has conducted technical sessions and hands-on labs at Oracle Code One and NetBeans Day conferences in San Francisco, Europe, and Latin America. Paul has also presented sessions at Devvxx and Devnexus and is the author of LiveLesson videos on JavaFX programming and Java Reflection.

ABOUT THE CONTRIBUTORS

For more information about Paul, visit asgteach.com, the Anderson Software Group on Facebook, and [@paul_asgteach](https://twitter.com/paul_asgteach) on Twitter.



Bruno Borges has been developing applications with the Java platform since 2000. Throughout his career, Bruno has developed desktop applications with Swing and JavaFX platforms, as well as dozens of large web systems with Java EE standards and third-party libraries like Spring Framework, Apache projects, and others. His experience building systems for corporations and government agencies led him to join big vendors like Oracle and Microsoft in the fields of product management and developer relations to foster investment in tools and runtimes for Java developers across different industries. Bruno is an active participant of the Java community at large and a recognized conference speaker and influencer and was recently named a Java Champion. Some of his most well-known public projects are the JavaFX port of the game *2048*, the Twitter component for Apache Camel, his work with Oracle to initiate the donation of NetBeans to the Apache Foundation, the Docker images for Oracle products, and the Java EE platform account on Twitter to foster the community. You can learn more about his work at <https://brunoborges.io>.

Anton Epple is a consultant for a wide variety of companies worldwide, ranging from startups to Fortune 500 companies, in many areas, including financial institutions and aerospace. In his spare time, Anton is a member of the NetBeans Dream Team and the organizer of the JayDay developer conference in Munich. In 2013, he joined the Java Champions and received a JavaONE Rockstar Award. In 2014, he received a Duke's Choice Award for his work on DukeScript.



Weiqi Gao is a partner and principal software engineer at Object Computing, Inc. in St. Louis, Missouri, USA. He has decades of software development experience and has been using Java technology since 1998. He is interested in programming languages, object-oriented systems, distributed and cloud computing, and graphical user interfaces. He is part of the original author team that produced the *Pro JavaFX* series for Apress. He is a member of the steering committee of the St. Louis Java Users Group. Weiqi holds a PhD in mathematics.



Jonathan Giles is a principal Java architect at Microsoft. Before this, he was a technical lead in the JavaFX team at Sun Microsystems and Oracle Corp. for 9 years, where he led the development of the JavaFX user interface (UI) controls library and other related libraries. With his considerable contributions to Java releases 7–11, he takes immense pride in having his code deployed on almost every computer on the planet. Jonathan is passionate about creating excellent developer experiences with considered API design, documentation, testing, and tooling. He has toured the world extensively to present and write on these topics, and as a result he is a Java Champion, a JavaOne Rockstar, and a Duke’s Choice Award winner.



José Pereda, PhD who specializes in structural engineering, works as a software engineer at Gluon Software. He is a Java Champion, Oracle Groundbreaker Ambassador, and JavaOne Rockstar. Having worked with Java since 1999, he is a JavaFX advocate, developing Java applications for mobile and embedded platforms connected to the cloud and enterprise systems. He also works on open source projects like OpenJFX and FXyz 3D (<https://github.com/jperedadnr>), co-authors JavaFX-related books (such as *JavaFX 8: Introduction by Example* and *JavaFX 9 by Example*, both published by Apress), blogs (<http://jperedadnr.blogspot.com.es/>), tweets

ABOUT THE CONTRIBUTORS

(@JPeredaDnr), and speaks at JUGs and conferences (JavaOne, Devoxx, JAX, Jfokus, JavaLand, JCrete, JBCNConf, to name a few). José lives with his family in Valladolid, Spain.



Sven Reimers based at Lake Constance in southern Germany, works as a systems engineer at Airbus Defence and Space, creating next-generation ground segment software for space systems. He has more than 20 years' experience building complex software systems and more than 20 years' experience with Java, going back to its early days. In 2009, Reimers was the winner of the Duke's Choice Award in the Network Solutions category for ND SatCom Satellite Communication Management Software. He is part of the Apache NetBeans PMC, a contributor to OpenJFX, and the leader and founder of JUG Bodensee. For his long-term commitment to Java and the community, Reimers was named a Java Champion in 2014.



Eugene Ryzhikov is a software architect with more than 30 years of experience in software development and design. He has created software ranging from real-time maps for very specialized field units and all the way to large energy trading systems. He has been using Java since the very beginning. His main interests are UI/UX, data visualization, advanced systems architecture/design, and serverless technology. Eugene is a cofounder of Gluon, and open source contributor, speaker at multiple conferences, and Java Rockstar.

In addition, his interests also include traditional Okinawan martial arts, jazz piano, and traveling with his lovely wife and daughter.

William Antônio Siqueira is a software engineer working at Red Hat and is a longtime JavaFX blogger and Java developer. He is also a local JUG contributor and has created and has contributed to several Java and JavaFX projects.

About the Technical Reviewer

Carl P. Dea is a principal software engineer. He has been developing software for over 20 years, for many clients from Fortune 500 companies to nonprofit organizations. He has written software ranging from mission-critical applications to ecommerce-based web applications. His passion for software development started when his middle school science teacher showed him a TRS-80 computer. Carl has been using Java since the very beginning, and he has been a huge JavaFX enthusiast since the early days when it was its own language called JavaFX script. His current software development interests are UI/UX, game programming, data visualizations, embedded systems, smartphones, AI, and robotics. When Carl is not working, he and his wife enjoy canoeing and day trips to the beach. Carl and his wife are proud parents of their younger daughter who attends Salisbury University. Carl and his wife are also proud of their older daughter, who is a high school teacher in Anne Arundel County, Maryland. Carl lives in Pasadena, Maryland, USA.

Foreword

The early 1990s saw the introduction of several game-changing technologies, two of which were the Java programming language and the World Wide Web. Prior to that period, client-server architectures were all the rage. One of the big challenges of client-server was developing application user interfaces (UIs) for various client machines and operating systems.

While most of the application development world was content with adapting web-based technologies such as HTTP, HTML, and JavaScript to client application development, a remnant of developers envisioned an alternative future. This future is being realized today in the form of rich, responsive clients that execute on the nearly ubiquitous Java Virtual Machine (JVM). I have had the privilege of co-laboring with many of these talented and revolutionary developers, several of whom have contributed the content that you'll read in the following pages. These champions of JavaFX include Gail Anderson, Paul Anderson, Bruno Borges, Stephen Chin, Carl Dea, Toni Epple, Weiqi Gao, Jonathan Giles, José Pereda, Eugene Ryzhikov, Sven Reimers, William Antônio Siqueira, and Johan Vos.

Here's a sampling of the rich-client treasures buried in this book: First, the groundwork is laid by showing how prevalent Java is in user interfaces that you may not have realized are written in Java. Then, the fundamentals of JavaFX are methodically revealed in a very approachable manner. After this treatment of foundational concepts, deep dives are taken in the important capabilities and libraries contained in JavaFX. Some of these capabilities are behind the scenes, such as JavaFX properties and binding, and some are visual, such as JavaFX controls and 3D graphics. After these JavaFX internal deep dives, several environments in which JavaFX can add much value are explored in depth. These environments include web, desktop, mobile, and embedded devices and the cloud. Finally, some leading-edge uses of JavaFX are discussed, including machine learning and scientific applications.

FOREWORD

It is important to note that JavaFX is developed in the OpenJDK umbrella and has its own release vehicle. As such, there is only one JavaFX, and it works on desktop, mobile, and embedded platforms. JavaFX evolves with and leverages new features of Java, ensuring that it will continue to be a cutting-edge platform.

It is my pleasure and honor to recommend this book and the JavaFX technologies discussed and innovated by my dear friends and colleagues.

—James Weaver
September 2019

CHAPTER 1

Getting Started with Client Java

Written by Stephen Chin

Client technologies are the basis for building any interface that users interact with. Because they are the first part of an application that a user sees, they also leave the greatest impact on your audience. Therefore, it is important that user interfaces look good and also are easy to use and intuitive.

Whether it be desktop, mobile, tablet, or embedded devices, Java client technologies provide a simple and elegant solution to building modern user experiences. Because the Java language is cross-platform, this reduces the effort to build and maintain your application for multiple screens and form factors. Also, as one of the most widely used programming languages, anyone can help maintain your code making it a solid foundation for the future.

In this chapter, we will show some examples of Java client technology in action and guide you through building your own cross-platform client to demonstrate how easy it is to accomplish this.

Java Client Technology in Action

Java client technology has been used for decades for all sorts of applications from business applications to development tools and even games. Also, now that Java runs on mobile and embedded platforms, you also find Java applications on your phone, tablet, and Raspberry Pi devices. It is often hard to tell if you are using a Java application because it is packaged together with the required Java libraries, so it appears just like any other native application.

We will explore several different Java client applications that you may or may not have used to give you an idea of the potential of this technology.

Java Client in Business

Java client technology is a staple in enterprise companies for internal applications. This is because it is great at building highly customized applications with complex controls like graphs, trees, tables, and Gantt charts. By building applications once and taking advantage of Java's cross-platform capabilities, enterprises save on initial implementation cost as well as maintenance.

Common use cases for Java client technology in the industry are high-speed trading, train monitoring and scheduling, supply chain management, medical imaging, and inventory management. MINT systems make a training and resource management system (TRMS) that has been adopted by numerous commercial airlines such as Emirates Airlines, JetBlue Airways, Azul Linhas Aéreas Brasileiras, FedEx Express, Lufthansa Group, and the Avianca-Taca Group.¹

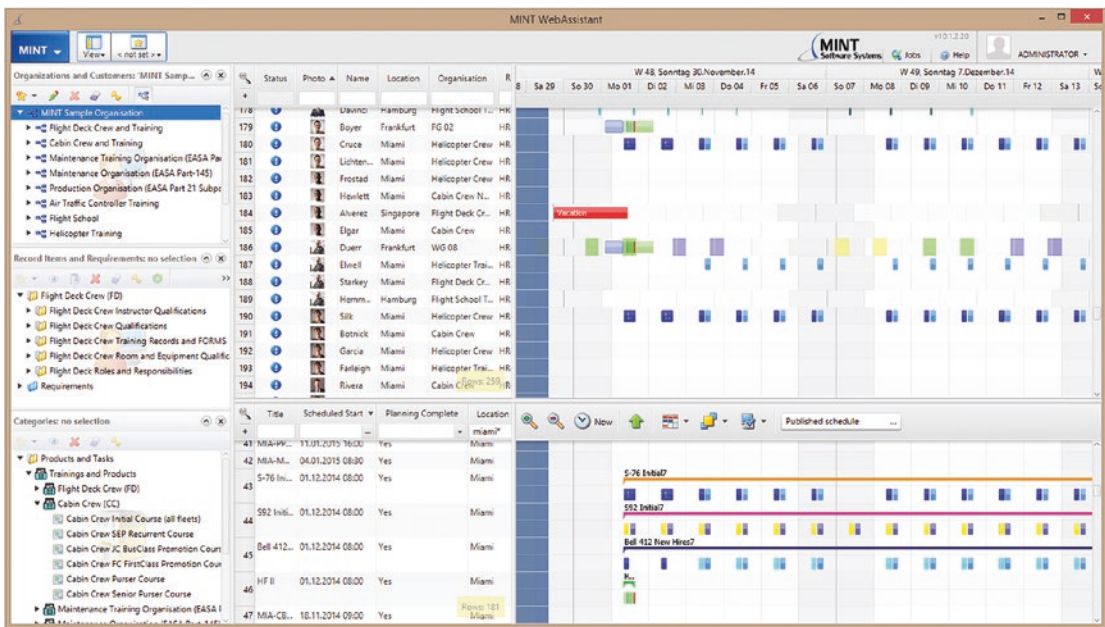


Figure 1-1. MINT software system for airline training and resource management²

¹MINT Software Systems – European Airline Training Symposium (EATS). Retrieved from www.eats-event.com/mint/. July 19, 2019.

²Casall, Alexander. 20 JavaFX real-world applications. Retrieved from <https://jaxenter.com/20-javafx-real-world-applications-123653.html>. February 11, 2016.

Figure 1-1 shows one of the more complex user interface screens from MINT TRMS. It makes use of trees, tables, ribbons, and Gantt charts implemented using JavaFX. This demonstrates a very complex view that would be challenging to implement in any other cross-platform technology.

To find out more about how you can easily build complex applications using pre-built JavaFX controls, check out Chapter 4, “JavaFX Controls Deep Dive.”

Gaming and 3D

Java client technology is also great for building games. One of the most popular games of all time was built by a single person using Java technology. Markus Persson (aka Notch) released a development version of Minecraft in 2009.³ All the initial development was done in his spare time until the alpha release made enough money so he could start his own company, Mojang, and focus on the game full time. It is now the world’s second highest grossing video game with 91 million monthly users.⁴

³Wikipedia. Minecraft. Retrieved from <https://en.wikipedia.org/wiki/Minecraft>. August 2019.

⁴Gilbert, Ben. “Minecraft” is still one of the biggest games in the world, with over 91 million people playing monthly. Retrieved from www.businessinsider.com/minecraft-has-74-million-monthly-players-2018-1. October 2018.



Figure 1-2. *Minecraft server example from Tingsterland created by @tingsterchin⁵*

Much of the success of Minecraft is through the large modding community who builds plugins that change the behavior and enhance the game taking it far beyond the original gameplay limitations. An example of a customer Minecraft server created by a young developer is shown in Figure 1-2. Java offers a great platform for building extensible applications through dynamic class loading and a secure sandbox model. Also with 10 million Java developers worldwide,⁶ there is no shortage of development expertise and talent.

Minecraft is built entirely in Java using client technologies such as Swing and Java 2D and a Java gaming library called LWJGL. The high level of abstraction offered by Java and these libraries made it possible for Notch to develop Minecraft in a short period of time and support a variety of platforms without a large team of developers.

An even easier 3D library to get started with is the built-in 3D support in JavaFX. You can find out more about 3D graphics in Chapter 8, “JavaFX 3D.”

⁵Screenshot from Minecraft server running at <https://tingsterland.com/>

⁶Oracle Java Certification Paths Overview. Retrieved from https://education.oracle.com/oracle-certification-path/pPillar_80. August 2019.

Mobile Conference Apps

Java client technology is not just for the desktop. Using mobile JavaFX technology developed by Gluon,⁷ you can run your Java client on phones, tablets, and embedded devices like Raspberry Pi. Existing JavaFX applications can be ported directly to mobile devices with small changes in styling of controls to make them work on different screen sizes. For handling mobile-specific APIs, Gluon offers Charm Down, which provides cross-platform integration with hardware features.

A great example of JavaFX mobile in action is the Devoxx conference application. This was originally built for the JavaOne conference in San Francisco and contributed to the open source community. The Devoxx conference picked it up and has done a great job extending it to be a general-purpose conference application that serves dozens of Devoxx and Voxxed conferences taking place around the world each year.

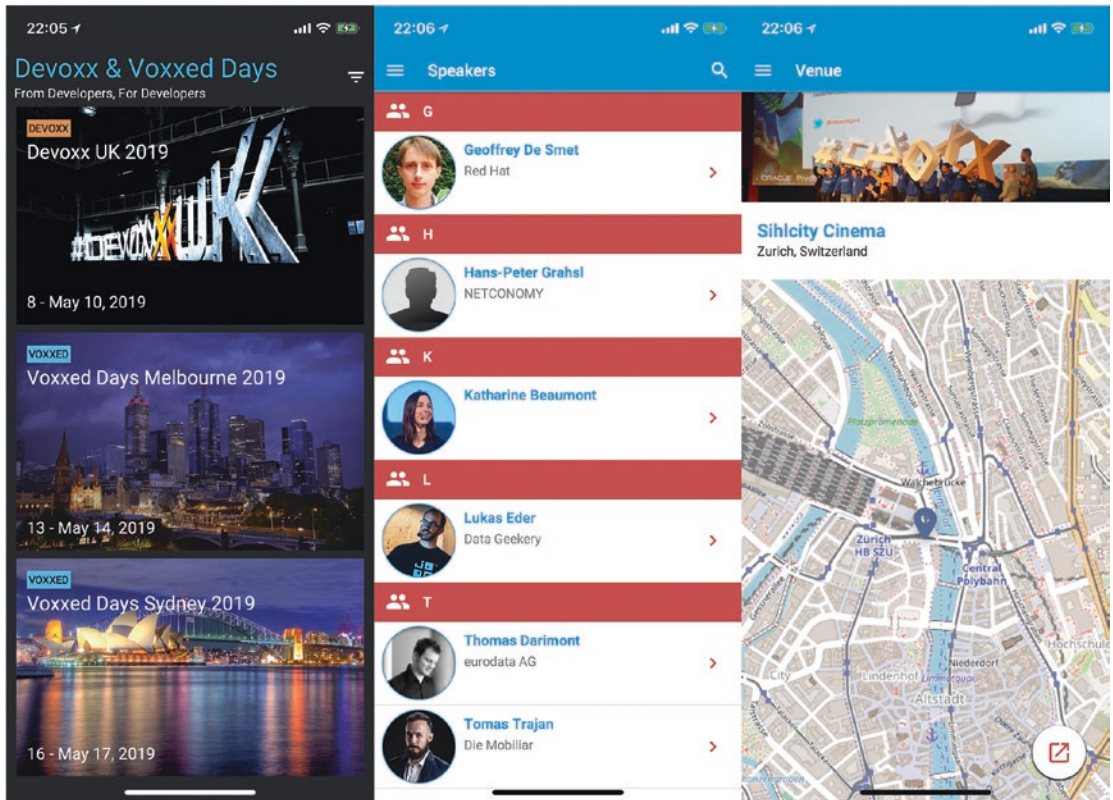


Figure 1-3. Devoxx conference mobile application. From left to right: conference selection, speaker list, and venue navigation⁸

⁷Official web site for Gluon: <https://gluonhq.com/>

⁸Screenshots from the Devoxx iOS conference application. Official conference web site: <https://devoxx.com/>

Figure 1-3 shows several of the different screens within the conference application to select an event, showcase speakers, and navigate to the venue. According to Stephan Janssen, founder of the Devoxx conference family, “JavaFX Mobile technology has helped us to streamline multiple native applications into a single cross-platform application that is well supported on iOS and Android devices. This is a better experience for conference attendees and much easier to keep up-to-date.”

We have a simple mobile example coming up later in this chapter to show how easy it is to use this technology and a more thorough guide in Chapter 11, “Native Mobile Apps for iOS and Android.”

A Modern Approach to Client Java

While client Java technology has been around for a long time, the development ecosystem has been under constant change. There have been significant advances in mobile, cloud computing, and app distribution that affect how you build and distribute your client applications. This book is focused on making you successful as a modern application developer by guiding you toward design and implementation best practices.

The three specific best practices that we are going to describe here and reinforce throughout the rest of the book are as follows:

1. Target mobile first.
2. Build for the cloud.
3. Package your platform.

Target Mobile First

The utilization of smartphones has been increasing steadily since iPhone and Android came out in 2007 and 2008, respectively. As of 2018, mobile smartphones and tablets have overtaken the desktop in web traffic, accounting for 52.2% of all web requests as shown in Figure 1-4. As a result, mobile is not just an option, but a required interface for successful applications.

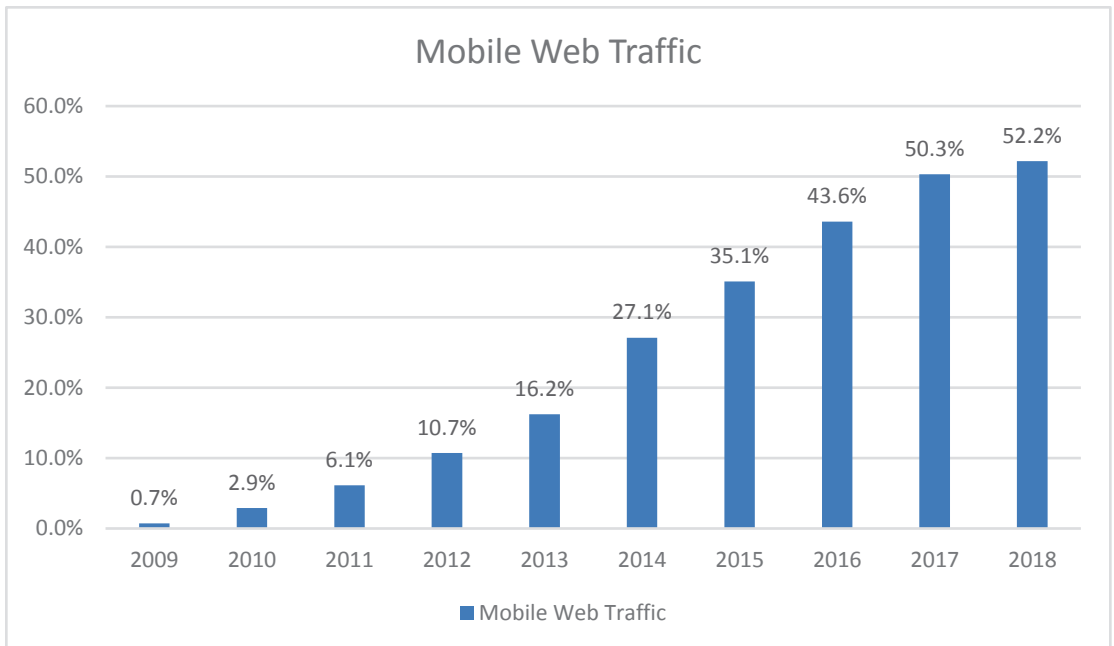


Figure 1-4. Mobile usage since 2009 as a percentage of global web traffic⁹

Smartphones have gotten to the point where they have the processing capability, memory, storage, and resolution to run full applications that were traditionally thought of as desktop-only. A tablet with a Bluetooth keyboard can easily be used as a desktop replacement for many use cases. Also, smartphones and tablets come with built-in Internet, which makes it possible to use them even where broadband is not available.

As a result, there is a rising number of “smartphone-dependent” users who only have Internet access through a phone, but do not have broadband that could be used for a desktop or laptop to connect. As shown in Figure 1-5, 28% of US millennials (18–29 years old) are smartphone dependent. This demographic will only be able to use your application if it has a mobile version available!

⁹Percentage of all global web pages served to mobile phones from 2009 to 2018. Retrieved from www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/. July 2019.

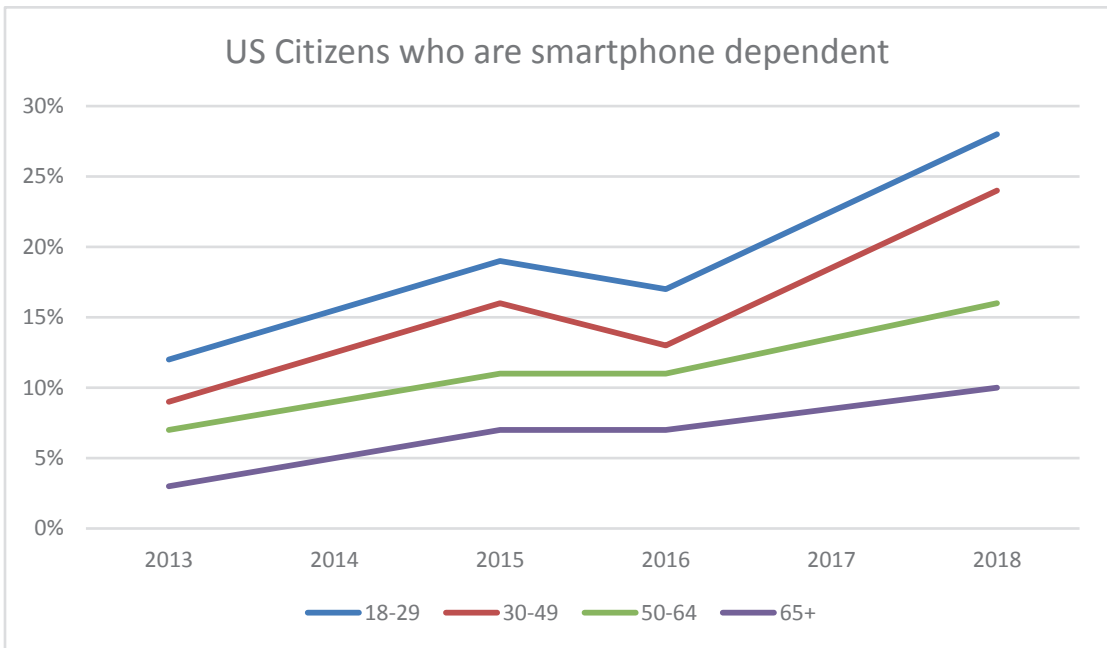


Figure 1-5. Smartphone-dependent US citizens by age bracket according to the Pew Research Center¹⁰

As discussed earlier, JavaFX has great mobile capabilities that are augmented by Gluon, who is an OpenJDK contributor. By using JavaFX mobile, you can write an application code base once and then target multiple screens including smartphones, tablets, and desktops. This gives your application a huge competitive advantage to desktop-only applications that do not allow users to take their work on the road with them. Find out more in Chapter 11, “Native Mobile Apps for iOS and Android”!

Build for the Cloud

The model for application backends has shifted from on-premise to the cloud. The reason for this is that there is an end user shift in expectations on how they interact with data. Historically, users would own and manage their data locally. With the rise of readily available high-speed connections, accessible encryption and security, and multiple screens per user, this expectation has changed. Now users expect data that is

¹⁰Mobile Fact Sheet. Retrieved from www.pewinternet.org/fact-sheet/mobile/. June 2019.

always online and available so it can be used from any device and easily shared and collaborated on.

A good example of this is eteoBoard, a digital collaborative scrum board built by Saxonia Systems AG in Germany. It is designed to solve the problem with distributed teams by creating an extended project team room across multiple locations. This is done by using teleconference equipment on large monitors and an electronic project board displayed on a large touchscreen monitor powered by JavaFX technology as shown in Figure 1-6.



Figure 1-6. Example of the eteoBoard being used to manage a project backlog¹¹

The eteoBoard application uses SynchronizeFX¹² for real-time synchronization of the user interface state between multiple clients. All of the project data gets loaded to and stored from either Atlassian Jira or Microsoft Team Foundation Server, both of which are cloud-based agile lifecycle management packages with REST interfaces. From an end

¹¹ETE0 - One Team - One Office. Retrieved from www.slideshare.net/Saxonia/eteo-one-team-one-office. October 2015.

¹²Open source repo for SynchronizeFX: <https://github.com/saxsys/SynchronizeFX>