

Developer's  
Guide to the

# Amazon Appstore

---

**Publishing, promoting,  
and making money  
with your app**



Free content to help intrepid app developers.

To acquire a free digital download of this book, visit the Amazon Appstore and search the title,  
"Developer's Guide to the Amazon Appstore."

Content by Amazon.com, Inc.

Printed by PrintWest Inc.

6101 238th Street SE

Woodinville, WA 98072

[www.developer.amazon.com](http://www.developer.amazon.com)

Copyright © 2019 Amazon.com, Inc. and/or its affiliates. All rights reserved. Amazon, the Amazon logo, and Amazon Appstore are trademarks of Amazon.com, Inc., or its affiliates. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

This book or parts thereof may not be reproduced in any form, stored in any retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or otherwise—without prior written permission of the publisher, except as provided by United States of America copyright law.

Cover design, formatting, and editing by Indigo Slate.

The content of this book and the website links contained herein are simply for educational purposes, and are not intended to replace professional business advice or guarantee business outcomes. Every effort has been made to ensure that the content provided in this content is accurate and helpful for our readers at the time of publishing. However, this is not an exhaustive treatment of the subjects. No liability is assumed for losses or damages due to the information provided. You are responsible for your own choices, actions, and results.

Second Edition

November 2019

# Contents

---

	<b>Foreword</b>	<b>7</b>
<b>1</b>	<b>Why sell in the Amazon Appstore?</b>	<b>10</b>
<b>2</b>	<b>Migrating your existing app</b>	<b>16</b>
<b>3</b>	<b>Creating a successful app</b>	<b>26</b>
<b>4</b>	<b>Building your app</b>	<b>40</b>
<b>5</b>	<b>Monetizing your app</b>	<b>62</b>
<b>6</b>	<b>Preparing and testing your app</b>	<b>76</b>
<b>7</b>	<b>Publishing your app</b>	<b>88</b>
<b>8</b>	<b>Promoting your app</b>	<b>106</b>
<b>9</b>	<b>Engaging and retaining users</b>	<b>114</b>
<b>10</b>	<b>Measuring and optimizing your app</b>	<b>128</b>
	<b>Conclusion</b>	<b>136</b>



# Foreword

---

Dear Reader,

Congratulations! You've taken the first step to launching your app in the Amazon Appstore. We want to make this process easy and enjoyable—and give you the tools you'll need to help your app find its audience.

As a Developer Evangelist at Amazon—and an app developer myself—I love sharing my experience and learning from the experience of other developers—from the old alums with a continuous stream of successful apps to those who are new to the business and just starting out. Working together makes our work better.

Building an app isn't easy, and the work doesn't end at launch. That's why we wrote this book to help you set up your app in the Amazon Appstore, and then to plan what comes next. From development to determining the look and feel of your app to finding the right audience, growing that audience, and making a profit, this book is designed to guide you through the entire process and share what we at Amazon have learned from our own experience working with developers.

We built the Amazon Appstore to help developers like you create apps and games that are creative, cutting-edge, useful, and engaging to users. And we built the Amazon Appstore to help the millions of Amazon customers discover new apps that will delight them—whether they're calculating the tip at a restaurant or battling space aliens on a bus ride.

Do you have questions or advice for your fellow developers? I'd love to hear from you. Follow me on Twitter (@MikeFHines) and keep up with the Amazon Developer blog to stay up-to-date with the latest news and advice from my team.

Good luck, and welcome aboard.

**Mike Hines**

# How to use this book

The content in these pages has been created to serve both novice and experienced app developers. Each chapter covers a stage in the app development process, so you can reference relatively self-contained segments, reading along as you move through creating, testing, publishing, and promoting your app. You can, of course, choose to read through the entire contents of the book end to end, if that's your preference. The order in which topics are covered in these pages is largely sequential, except for the chapter on the app migration process. This chapter is placed early in the book to provide quick access to those of you who have an existing Android app that you're looking to migrate to the Amazon Appstore. You can find this information in **Chapter Two: Migrating your existing app**.

Most chapters include a "Challenges" section that contains a list of some of the common challenges you might run into as a developer during that specific phase. This outlines the content in the chapter and also provides a point of reference for if/when you encounter a specific challenge.

## Sample Preflight checklist



- Select an Android app to publish
- Determine image assets and metadata for your app
- Create an Amazon Customer account
- Create an Amazon Developer account

## Sample Challenges you might face



- Learning how to get started with migration
- Finding documentation and guidance
- Quickly identifying what changes, if any, are necessary

In each chapter, you'll also find a "Preflight checklist." This provides a quick view into what you need to accomplish before starting on the actions discussed in that chapter—particularly for readers who have bypassed the preceding chapter.

You'll find hyperlinks within these pages that link to external resources like detailed requirements, specifications, or references. It's a good idea to make use of these links, as the information can change from time to time. Keep yourself current to ensure your app development process is a smooth one.

A vibrant photograph of a large crowd at a concert or festival. Many people's hands are raised in the air, some wearing colorful wristbands. Confetti is falling from above, creating a festive atmosphere. The background is blurred with warm, golden light.

# ONE

# Why sell in the Amazon Appstore?

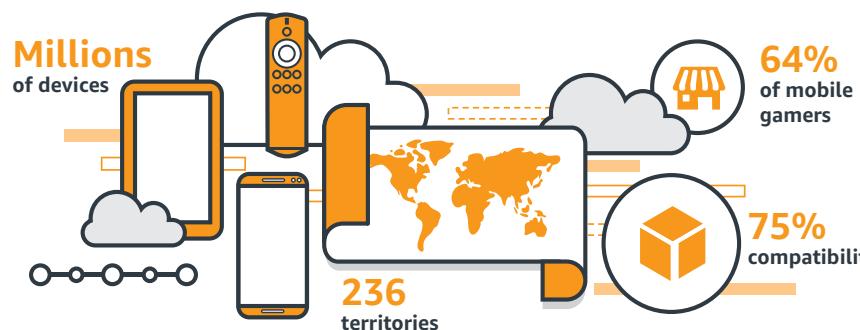
---

The Amazon Appstore connects content creators with Amazon's quality customer base and trusted brand, putting your apps in a prime position to succeed. Because Fire OS is based on Android, the apps and games you publish through the Amazon Appstore will be available on millions of Fire tablets, Amazon Fire TVs, and Android devices in 236 countries and territories. If an app runs on Android, it takes little or no effort to get it to run on Fire OS.

And it's simple to sell in the Amazon Appstore. This is by design. Amazon provides developers with a pain-free publishing process along with unique ways to launch, promote, and monetize their apps.

## Trusted brand

Amazon has hundreds of millions of customer accounts worldwide and has carefully built its brand and customer base. As a result, customers using the Amazon Appstore are the most likely to spend money on your app or game. Here's one example: 64 percent of mobile gamers in the Amazon Appstore spend money on mobile games, compared to 37 percent in the Google Play Store.<sup>1</sup>



## Innovative monetization models

In today's largely commoditized app store landscape, Amazon remains committed to innovating the ways developers monetize their apps. Whatever your platform, Amazon provides Fire OS, Android, and iOS APIs that help you earn more money and create better experiences for your users. In the Amazon Appstore, you'll generate more revenue from your apps and games by using Amazon's unique monetization services, including the In-App Purchasing (IAP) API, which increases conversion with Amazon's

For Fire OS, Android, and Alexa developers, there has never been a better time to publish in the Amazon Appstore. With Amazon's powerful and popular streaming media players, you're able to reach more customers than ever.

1-Click purchasing technology. The IAP API makes it easy to offer subscriptions and digital content—such as in-game currency, expansion packs, upgrades, consumables, and more—for purchase within your app.

Amazon also helps you extend the sales experience beyond your app. With Merch by Amazon, you can provide branded merchandise for your customers—designed by you, and produced, sold, and shipped by Amazon.

## Easy migration

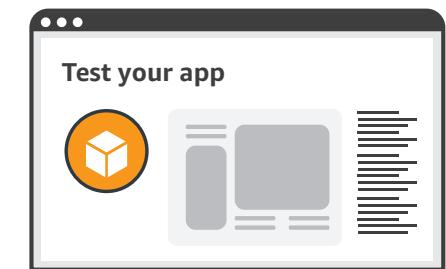
It's easy to bring your Android application package (APK) to Amazon devices. In fact, more than 75 percent of the Android apps that we've tested work as-is in the Amazon Appstore, with no additional development required. Want to find out how your app will work? Access the free compatibility test and get results in just 90 seconds.<sup>2</sup>

<sup>1</sup><http://bit.ly/2nBwG2K>

## How it works

### 1. Upload your app

Drag your Android application package file (the APK) into the upload box, or browse to choose the APK to test.



### 2. Review results

Receive the test results, along with support documentation if you need to make changes. Test results will be saved and available for future use if you're logged in as a registered user of the site.

### 3. Submit your app

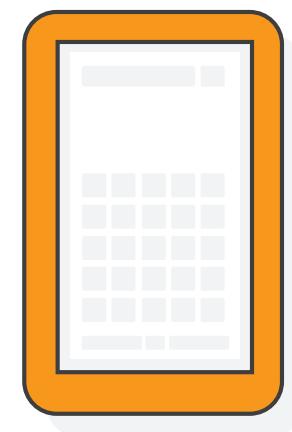
Submit your app to the Amazon Appstore directly from the test results page.

These results will immediately identify potential issues with your app's compatibility. If there are none, you can submit your app right away. You can also integrate Amazon APIs—like In-App Purchasing or Amazon Mobile Ads—to provide a richer customer experience and increase monetization.

# Whatever your platform, Amazon provides Fire OS, Android, and iOS APIs that help you earn more money and create better experiences for your users

## Reach more people on more devices

Amazon's latest Fire tablets ship with Fire OS 7, bringing the best entertainment experience on any tablet, with an updated user interface, integration of Amazon exclusive services, and hundreds of new and upgraded features. Fire OS 7 is based on Android 9 (Pie) and API 22, making it more compatible than ever with existing apps written for other Android devices. Each Fire tablet home screen includes dedicated content pages for books, games and apps, video, music, audiobooks, and periodicals.



Each page is designed to engage users and make apps the center of the tablet experience. Dynamic content lists are personalized for each individual consumer, allowing your app to be discovered by the right customers. Fire tablets are available worldwide—in the U.S., U.K., Germany, France, Italy, and Spain.

## Amazon's commitment

Amazon is dedicated to maintaining consistent quality in the apps we offer. As a developer, you'll receive the support you need to create and maintain your apps. With the continued exponential growth of the Amazon Appstore, ease of app publishing, and the added benefits of monetization, discoverability, and cross-platform support, it's easy to decide to publish your app in the Amazon Appstore.



# TWO

## Migrating your existing app

---

Migrating your existing app to the Amazon Appstore is a fast process, but there are some steps you need to take first to prepare for a successful submission. This chapter covers the important requirements for getting your app ready. These include determining the compatibility of your app and any necessary modification or level adjustments for associated APIs. You'll also want to review the test criteria before submitting your app for approval.



## Preflight checklist

---



- Select an Android app to publish
- Determine image assets and metadata for your app
- Create an Amazon Customer account
- Create an Amazon Developer account

## Challenges you might face

---



- Learning how to get started with migration
- Finding documentation and guidance
- Quickly identifying what changes, if any, are necessary

## Amazon Appstore submission requirements

Before you begin the submission process, it's a good idea to review the requirements and guidelines for app submission.<sup>1</sup> The Amazon Appstore Presubmission

Checklist provides an overview of these requirements and links to more detailed information on app content guidelines, icon and screenshot image requirements, testing requirements, and more.

[1http://amzn.to/2nBOZ01](http://amzn.to/2nBOZ01)

1. Review the Amazon Appstore Presubmission Checklist
2. Pre-test your app
3. Follow the instructions on the Submitting Apps to the Amazon Appstore page to publish your app



## Migrating an existing Android app to the Amazon Appstore

In most cases, if you have an Android app published in the Google Play Store, you won't need to make any modifications to get your app to work on Fire OS devices. In fact, more than 75 percent of migrated apps that we tested work as-is with the Amazon Appstore. However, there is a small subset of Google APIs, services, and features that will require modifications, as covered in the next section.

### Steps for migrating an existing Android app:

1. Migrate any APIs, services, and features to those supported by Fire OS, if necessary.<sup>2</sup>
2. Update your minimum API level as required for the Fire OS devices that you plan to support.<sup>3</sup>
3. Remove any unsupported features from your app.<sup>4</sup>
4. Recompile your app.
5. Submit your app to the Amazon Appstore.

[2http://amzn.to/2oXOtC2](http://amzn.to/2oXOtC2)

[3http://amzn.to/2n93ch9](http://amzn.to/2n93ch9)

[4http://amzn.to/2o0zZUF](http://amzn.to/2o0zZUF)

## Supported APIs, services, and features

If your app uses any APIs, services, or features that aren't supported by Fire OS, you'll need to update your code to use compatible APIs. The table on the next page lists APIs, services, and features that you'll need to modify to make your Android app work with Fire OS.

75% of migrated apps that we tested work as-is with the Amazon Appstore

## Supported API levels

Android API versions have an "API level" value. Fire OS versions are based upon Android API versions, and therefore have an effective "maximum API level." Amazon recommends setting a minimum API level of 10 to ensure compatibility with all Fire tablets. For information about the Android OS versions for the Fire tablets, see Fire Tablet Device and Feature Specifications.<sup>5</sup>

If you're developing your app with the latest API version, your app might not be compatible with older Fire OS devices. If this is the case, you'll need to modify your implementation to avoid calling unsupported methods on older generation devices. For the most up-to-date information, see Check System Version at Runtime in the Android training guide Supporting Different Platform Versions.<sup>6</sup> For details about specifying API levels in your app, see Specify Minimum and Target API Levels in the Android training guide.<sup>7</sup>

<sup>5</sup><http://amzn.to/2pCBzxD>  
<sup>6</sup><http://bit.ly/2oVORo8>  
<sup>7</sup><http://bit.ly/2oW3rMp>

API, Service, or Feature	Action
Google Maps	Use Amazon Maps API <sup>8</sup> <i>Migrating an App from Google Maps v2<sup>9</sup></i>
Google Cloud Messaging	Use Amazon Device Messaging <sup>10</sup>
Google Play In-App Billing	Use Amazon In-App Purchasing API <sup>11</sup> <i>Migrating from Google Play In-App Billing (IAB) v3.0 to Amazon In-App Purchasing (IAP) v2.0<sup>12</sup></i>
GPS geo-location on tablets without a mobile data plan	Use Wi-Fi geo-location or code your app to degrade gracefully
Any phone feature	Code your app to degrade gracefully
Device orientation	See Device Orientation <sup>13</sup>
Unsupported intents	See Android Intents Supported on Fire Tablets <sup>14</sup>
Unsupported screen sizes and layouts	Declare support for screen sizes in the <code>AndroidManifest.xml</code> <sup>15</sup>

- <sup>8</sup><http://amzn.to/2nUXzjQ>  
<sup>9</sup><http://amzn.to/2n8PZFF>  
<sup>10</sup><http://amzn.to/2nBx950>  
<sup>11</sup><http://amzn.to/2f4QhsW>  
<sup>12</sup><http://amzn.to/2n8M8rx>  
<sup>13</sup><http://amzn.to/2nBjg1R>  
<sup>14</sup><http://amzn.to/2oSpJF>  
<sup>15</sup><http://amzn.to/2oPnaen>

## App compliance

Before submitting your app to the Amazon Developer Portal, you need to know which features aren't supported by the Amazon Appstore. See the list of unsupported features to the right.

**Note:** If your app uses an API or hardware technology that isn't supported on Fire OS, we recommend that you remove or gracefully degrade any feature that requires that service. For example, your app might display a message stating that the feature isn't currently available on the device.

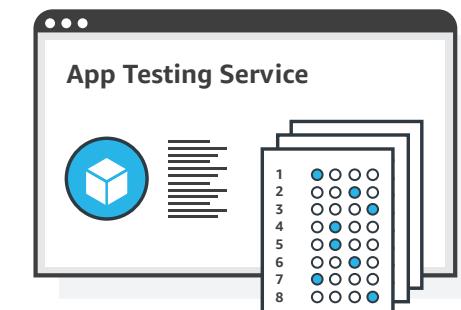
## Testing your app

When you submit your app to the Amazon Appstore, the app must pass a series of tests to qualify for publication. Take a look at the Test Criteria for Amazon Appstore Apps before you submit your app for approval.<sup>17</sup> That way you can avoid time lost in getting your app to market.

Amazon's App Testing Service will check your submission for API compatibility, including unsupported API usage, and perform a device test using actual devices. Once all tests have passed, you can submit your app for publication directly from the test portal. See App Testing Service for additional information and to start the testing process.<sup>18</sup>

## Unsupported features

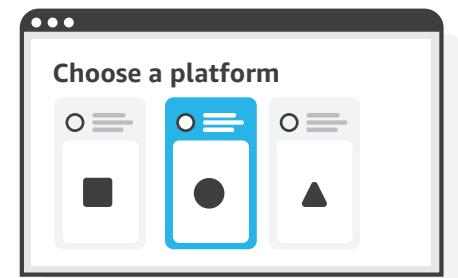
- Custom or third-party on-screen keyboards (use the default on-screen keyboard instead)
- Home screen widgets that manipulate the user interface
- disable\_keyguard permissions
- Lock screen customizations
- Wallpapers
- Screen savers
- Themes



## Publishing your app

Once your app is fully migrated and tested, you're ready to submit it to the Amazon Appstore for publication. Log in to your Amazon Developer account, enter the appropriate information for your app's listing, and upload your app.<sup>19</sup>

You'll also need to provide screenshots and large and small icons for promoting your app on the Amazon Appstore. You can optionally upload a promotional image and videos. For more information, see Image Asset Guidelines for Appstore Submission.<sup>20</sup>



### Steps for publishing your app:

1. Log in and create your app.
2. Enter general information for your app.
3. Enter availability and pricing information.
4. Enter app description(s).
5. Upload image and media assets.
6. Assign a content rating.
7. Upload Android application package (APK) files.
8. Submit your app.

<sup>17</sup><http://amzn.to/2pCBNog>

<sup>18</sup><http://amzn.to/2n8Puel>



THREE

# Creating a successful app

---

To create a successful app, start by asking two fundamental questions:

What problem is your app solving?

Who is the target audience?

The answers to these questions will be the keys to developing your app's core features, as well as identifying which existing apps to include in your competitive analysis. As you look to develop an app that users will want, don't lose sight of the elements that drive customers to adopt your app into their daily lives—like user experience, performance, network connectivity, and security. And never underestimate the importance of testing and feedback to the success of your app.



### Preflight checklist

- Define the problem you want to solve or the story you want to tell
- Identify your target audience



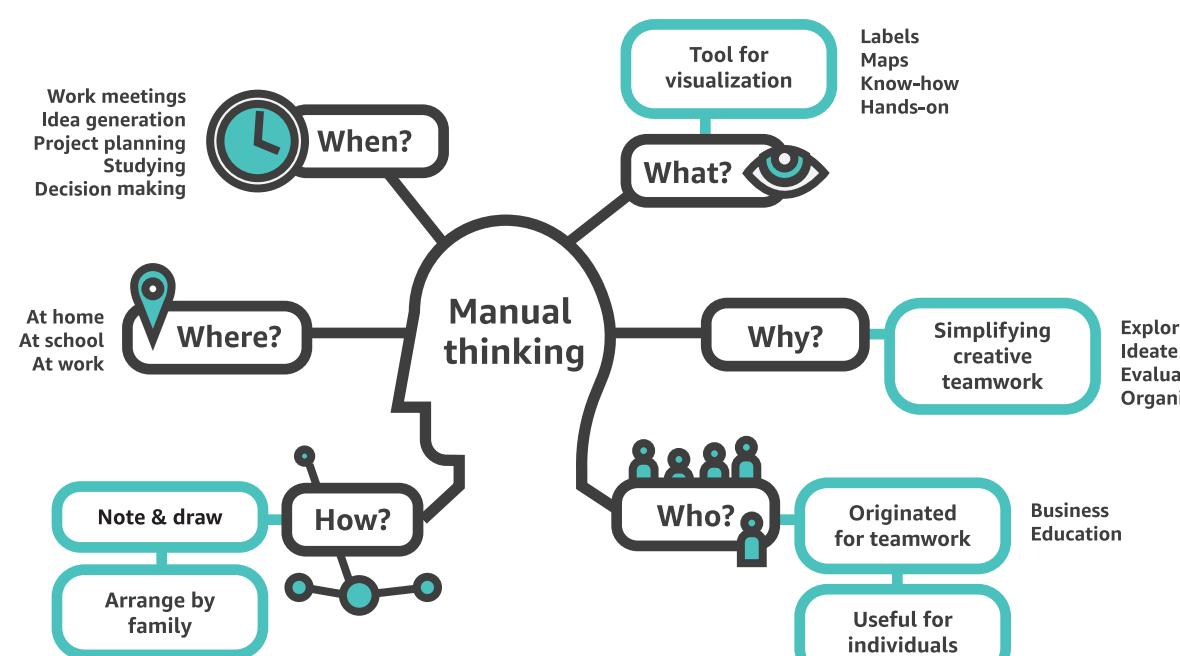
### Challenges you might face



- Developing an app that's appealing to users
- Identifying the correct audience
- Focusing on core features
- Understanding the importance of performance, connectivity, and security
- Getting and responding to user feedback

## Define the problem

What apps have a chance at success on the Amazon Appstore? Any app that solves a perceived problem, saves time, and does it simply. The difficulty isn't always in identifying the need or the solution; sometimes it's staying focused on the intent of that solution without getting distracted before the app is finished. Taking the time to clearly define the problem also helps keep efforts on track. And, eventually, it leads to a solid app.



Defining the problem

## Identify your target audience

You might be tempted to skip the problem definition step if your app is a game or a novelty app. But apps created for pure entertainment serve functions too. Is your game's intent to fill five minutes of time while the user waits in line? To draw the user through an epic saga of high adventure? Even the simplest problem definition still offers vision and focus.

Before you begin developing or even planning your app, consider your audience. Think about the profile or persona of those who will use the app, and how your solution will solve the problem you've defined. This is especially important when you're not a part of your target audience. Knowing the customers who will be using your app provides a framework for determining the features that will be most important to those users. This keeps development efforts focused on what's best for your customers.

User expectations of an app's functionality will differ across platforms. This can be especially true with Amazon devices, which range from small handheld tablets to high-definition TV displays. The device being used should be a key consideration in determining how your app will function for prospective users.

It's worth bringing users into the process early—even before the first line of code is written. Start a conversation with your users to get insight about their needs. Build feedback mechanisms into your app, and ensure your team is paying attention to those responses as they come in.



Customer experience journey

## Plan, starting with core features

To create a reliable app, you need to plan ahead. Start with the core features of your app—the aspects most effective at solving the problem you've defined for your target audience. These features are the critical pieces of your solution; without these, your app fails.

In your planning, ensure that app processing is solid, that the interface to the backend database is locked down, and that the results returned to the app are correct. And be sure to integrate the user story, allowing for some flexibility to deal with unexpected results or conditions.

## Conduct competitive analysis

Take a look at what else is out there. Research similar ideas on the various app stores; test competitors' mobile apps and note the benefits of each. Understanding what's working well—and not so well—with your competitors' apps gives you a significant advantage in your initial development efforts. Pair this with your own users' feedback and analytics, and your time will be well spent.

## Invest in user experience

It's easy to get caught up in the bells and whistles that you believe will wow your users. Keep in mind, however, that snazzy UI and innovative interfaces don't matter as much as the functionality at the core of your app. These core features are your app's essential business function. Users can be temporarily blinded by pretty features, but they won't continue to use an app that doesn't function well.

Your greatest app idea won't catch on with customers if the app is difficult to use. Aim for a mobile app that's intuitive and user-friendly by investing resources in the UX of each platform. Usability drives adoption and acceptance rates, determining whether your app succeeds.

Consistent functionality across all platforms is important to your app's branding, and presents a more polished feel. Keep in mind that users will expect their interactions with your mobile app to be similar to those of other apps on their platform. At the same time, however, your app should retain its look and feel across all platforms, so that users can easily transition to using your app on another device.



Core features drive satisfaction

### Know that performance matters

No one likes a slow app. Performance can be a tricky part of development because there are many things that can have an impact and it can be difficult to anticipate these impacts ahead of time. It's important to constantly monitor the app and its components (network, services, etc.) and ensure performance is within acceptable ranges. Use alerts to find out about issues before they become problems—that then become angry phone calls to support.

Design with performance in mind. Make sure that performance stays within design parameters. But don't lose sight of the fact that performance serves your core features. Your decisions for performance tweaks shouldn't interfere with the core functionality of your solution.

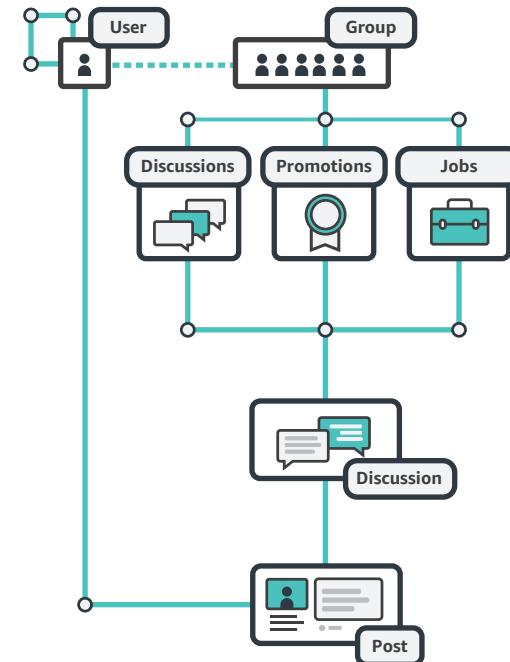
**Design with performance in mind. Make sure that performance stays within design parameters. But don't lose sight of the fact that performance serves your core features.**

### Consider network connectivity

If you're developing a mobile app, network connectivity is almost guaranteed to be a requirement. Fire tablet and Fire TV apps are likely in the same category. Apps that live on their own are rare, so it's important to decide early on what your app will do if it can't find the network.

This will depend heavily on exactly what the app is supposed to do. Will the app store enough data locally to allow it to function for a short time away from the network? Or is it possible that the app can go into store-and-forward mode until connectivity is restored? It might be that you just need to come up with an engaging "waiting for the network" screen. Your app should respond to network connectivity with the proper results.

Network performance and security can get complicated, as users expect smooth transitions from using cellular data to Wi-Fi. So it's important to take the time to consider how your app is affected by high traffic and slow or weak connections.



## Understand mobile application security

Mobile apps are vulnerable—and so, by extension, are their users. Security and privacy are important considerations regardless of whether your mobile app is in a heavily regulated industry like finance or healthcare. You must take measures to protect your users and their data and, if applicable, to comply with industry regulations. Chances are your app involves sensitive information or information that's private to the user. You need to keep that information secure.

Take security into consideration early in your development process. Thinking about this from the start means knowing what kind of input you'll be dealing with, so you can build data testing into the app. Understanding the network components and communications mechanisms allows you to secure data in transit. And you can ensure that only authorized users can access data by knowing which authentication mechanisms are in place and which are available for use.

Some security best practices include encryption of data in transit (e.g., SSL/TLS) when transferring personal information, de-identifying user data (e.g., hashing), user authentication (preferably two-factor authentication), and encrypting all sensitive data stored in the cloud. If you're using a framework for your app, consider using one that provides secure communications and supports several types of authentication. It's important to review how that framework provides these functions. Take some time to understand which protocols are implemented, and research vulnerabilities. Don't assume your chosen framework is invulnerable.

## Test, test, and test

There are lots of arguments about testing. Who should do it? When should it be done? What's the best methodology for doing it? There is, however, no argument about whether testing is necessary. Testing extensively is an absolute must.

Testing is not just another step in the overall process; rather it's a part of each step in the development process. The most effective testing is performed constantly and consistently in every aspect of your app. Test the design, functionality, performance, and interactions with other apps. Test on every platform and multiple browsers and carriers.

---

**There is no argument about whether testing is necessary. Testing extensively is an absolute must.**

---

It's tempting to use your users as beta testers for every new app. Amazon provides an App Testing Service, as well as the ability to release a beta to a specific set of users (likely other developers) to help make sure your app is solid and reliable before releasing it to your customers.

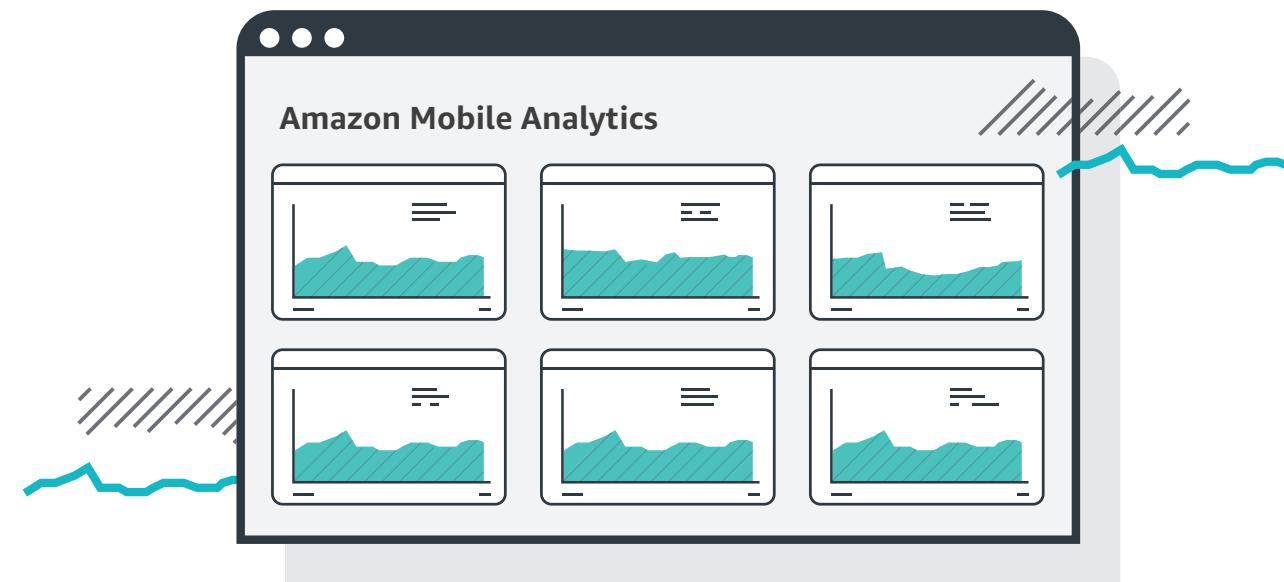
# What apps have a chance at success on the Amazon Appstore?

## Any app that solves a perceived problem, saves time, and does it simply.

### Respond to analytics and feedback

Throughout these pages, there are references to analytics—in fact, there's an entire chapter dedicated to the topic ([Chapter Ten: Measuring and optimizing your app](#)). It's an important tool for collecting information, but in the context of this chapter, analytics is one of several channels for feedback. Other channels include comments, blogs, reviews, and social networks.

Feedback allows you to more easily hear your customers' voices. And you should listen to what they have to say. Avoid becoming defensive when people point out shortcomings in your app. It isn't easy, but getting this right leads to creating better apps. Hearing your users' feedback—positive or negative—and addressing issues objectively play an important role in making your app even better.





# Building your app

Now that you're ready to build your app, it's time to familiarize yourself with the tools that help with development. As you plan, learn how you can use frameworks and integrated development environments to enhance your app's development. Focus on key design considerations like the quality of the code, visuals and audio, mobile connectivity, and data privacy and storage. And use web-based services and analytics that can help support and improve your app.



### Preflight checklist



- Choose a development framework
- Choose your target devices
- Create consistent videos and audio

### Challenges you might face



- Understanding your app's environmental needs
- Planning your app for scale
- Setting the base for collecting feedback and analytics

## Using frameworks

You can accelerate your app's development by using frameworks. Frameworks allow you to work with existing code libraries so you don't have to create these libraries yourself. These libraries perform functions like rendering objects in 3D or connecting to external services. Cross-platform frameworks can help you create platform-agnostic apps. This means you only have to create an app once before deploying it to different types of devices.

Amazon offers plugins for several popular cross-platform frameworks, as described below, making it easy for you to include them in your app.

## Unity

Unity is a cross-platform game development engine. With Unity, you can easily create rich, interactive 3D experiences by assembling your art and assets into scenes and environments, adding physics, and simultaneously play-testing and editing games. When you're finished building and testing, Unity creates a finished binary—such as an Android application package (APK)—that you can publish to your chosen platform, such as Fire tablets and the Amazon Appstore.



### Plugin or extension

Amazon Fling

### Documentation

[Integrating the Amazon Fling Controller in Your Games<sup>1</sup>](#)

[Integrating the Amazon Fling Player in Your Games<sup>2</sup>](#)

Phone

[Using the Unity Plugin in Your Unity Project](#)

In-App Purchasing

[Adding a Plugin to Your Unity Project<sup>3</sup>](#)

[IAP Plugin Methods and Events<sup>4</sup>](#)

[Implementing the IAP Plugin<sup>5</sup>](#)

<sup>1</sup><http://amzn.to/2oVPs9m>

<sup>2</sup><http://amzn.to/2oPrwlR>

<sup>3</sup><http://amzn.to/2oPrNFp>

<sup>4</sup><http://amzn.to/2oPm02y>

<sup>5</sup><http://amzn.to/2oVPfmm>

## Adobe AIR native extensions (ANE)

With Adobe AIR native extensions (ANE), you can package the same code you use for browser-based games—developed using Flash (ActionScript)—into native apps on your chosen platform, such as Fire tablets.



### Plugin or extension

### Documentation

#### In-App Purchasing

- [Adding a Plugin to Your Adobe AIR Project<sup>9</sup>](#)
- [IAP Plugin Methods and Events<sup>10</sup>](#)
- [Implementing the IAP Plugin<sup>11</sup>](#)
- [Using the IAP Plugin for Adobe AIR<sup>12</sup>](#)

<sup>9</sup><http://amzn.to/2oPuAyp>

<sup>10</sup><http://amzn.to/2oVfmm>

<sup>11</sup><http://amzn.to/2oPm02y>

<sup>12</sup><http://amzn.to/2oshWng>

## Apache Cordova

Apache Cordova allows you to develop mobile apps that access native device functions—like the camera and accelerometers—using HTML, CSS, and JavaScript.



### Plugin or extension

### Documentation

#### In-App Purchasing

- [Adding a Plugin to Your Cordova Project<sup>13</sup>](#)
- [IAP Plugin Methods and Events<sup>14</sup>](#)
- [Implementing the IAP Plugin<sup>15</sup>](#)
- [Using the IAP Plugin for Cordova<sup>16</sup>](#)

#### Amazon WebView API

#### Building and Testing Your HTML5 Hybrid App<sup>17</sup>

<sup>13</sup><http://amzn.to/2o4lxML>

<sup>16</sup><http://amzn.to/2onvQOs>

<sup>14</sup><http://amzn.to/2oPm02y>

<sup>17</sup><http://amzn.to/2onQ0BB>

## Xamarin

Xamarin allows you to create native iOS, Android, and Windows apps in C#.



### Plugin or extension

### Documentation

#### In-App Purchasing

[Adding a Plugin to Your Xamarin Project<sup>18</sup>](#)

[IAP Plugin Methods and Events<sup>19</sup>](#)

[Implementing the IAP Plugin<sup>20</sup>](#)

[Using the IAP Plugin for Xamarin<sup>21</sup>](#)

<sup>18</sup><http://amzn.to/2osFKao>

<sup>19</sup><http://amzn.to/2oVPfmm>

<sup>20</sup><http://amzn.to/2oPm02y>

<sup>21</sup><http://amzn.to/2onIOoZ>

## Fire App Builder

With Fire App Builder, you can build streaming media Android apps for Amazon Fire TV while incorporating best practices and techniques. Fire App Builder's code is Java-based and uses Android Studio, Gradle, and other tools common to Android app development. To use it, you'll need to have Android Studio and Java SE Development Kit 8 or later installed on your system.

If you want to extend Fire App Builder with more advanced functionality, you can use Fire App Builder as the underlying framework and build on top of it, since most of the components are modular.

You can accelerate your app's development by using frameworks. They allow you to work with existing code libraries so you don't have to create these libraries yourself.

## Using integrated development environments

An integrated development environment (IDE) helps you create, view, edit, and debug your code. Since Android apps are Java-based, you can choose from several IDEs, including Android Studio and Xamarin Studio.

**Android Studio** is the official IDE for Android. In addition to GitHub and Gradle integration, Android Studio touts a comprehensive device emulator, with emulation support for Android phones, tablets, Android Wear, and Android TV devices.

**Xamarin Studio** for Windows is integrated into Visual Studio, and a Mac OS version is available for download. Xamarin Studio Community is free for individual developers, open source projects, academic research, education, and small professional teams.

## Design considerations for your app

Whether this is your first or your tenth app project, there are a few concepts that you'll want to keep in mind during planning. The preceding chapter (**Chapter Three: Creating a successful app**) covered fundamental elements, but it's worth discussing some basics:

### Keep it simple.

The more complex features your app includes, the more likely something unexpected will happen. Focus on the core object of your app, and the features essential to its function.

### Stay focused on the target audience.

Prioritize the features that provide the most value to your users. Avoid risky investments that seem appealing, but don't add any real value for your target users.

### Pay attention to it and finish.

The most useful, functional app can be overlooked if its UX is cluttered or difficult to navigate. Make your app visually beautiful (as well as useful) so it gets the attention it deserves.

### Ensure a high-quality experience.

Simply put, a few features implemented correctly offer more value to your users than twice as many features implemented poorly.

## Clean, maintainable code

The foundation of your app's success can lie in the quality of the code. This is the case whether you handcraft each function or rely heavily on drag-and-drop design. If your code is overly complex, then so is the process of adding features or addressing feedback. This can lead to new issues being generated with each bug fix.

Focus on keeping code maintainable from the first line written. Implement these concepts from basic programming: unique, descriptive names for classes, members, and methods; single-purpose methods without too much overloading; appropriate use of commenting; and diligent documentation. Refer to best practices and follow them as closely as possible. The Android software development kit (SDK) has a great many of them, as do each of the frameworks described above.<sup>24</sup>

<sup>24</sup><http://bit.ly/2onRNqj>



An IDE helps you create, edit, and debug code

## Clear, consistent visuals and audio

Whether designing UX controls, game characters/avatars, or full-motion cut scenes, it's important to deliver a visual experience that fits. As a part of the user experience, your app should provide a sense of completeness and discoverability. Users should be able to easily identify controls and know what they do. The overall layout should stay consistent throughout screens, tabs, and dialogs. And it's a good idea to provide a visual and audio cue when a control responds to input.

Any art used in your app should look like it belongs. Assets shouldn't appear as though they were arbitrarily added as an afterthought. And you should eliminate any issues with skewing, cropping, or scaling.

Sound effects, voiceovers, and background music can complete the user experience for your app. If implemented poorly, these elements can also ruin the user experience. Make sure that the audio assets for your app have consistent sound levels and sampling quality. And consider including user-adjustable settings to change the volume level for your app.



## Developing for multiple devices

You can use existing tools like Android Studio to develop apps for Fire TV in addition to Android phones and tablets. The experience for your app will likely vary considerably between a 4-inch screen and a 48-inch screen. Choosing the right devices to support can help focus your efforts on the experience best suited for your target users.

### Device targeting provides a number of benefits for your app:

- Better app discoverability. Instead of having multiple apps for a similar title, your reviews, page views, and downloads are consolidated. This helps push your app higher in search results and relevancy rankings.
- Device-specific feature optimization. Your APK is tailored for different platforms and device capabilities including screen size and pixel density, OpenGL compression format, and API version.
- Less customer confusion. Customers won't be confused by multiple device-specific versions of your app.
- Device targeting is also a better and smoother experience for your customers. The customer only has to buy your app once, and the correct, optimized version is automatically delivered for each device they use.

## Mobile connectivity

If your app includes an online presence, you need to consider how much bandwidth your app consumes, as well as how much data your app will use, particularly with cellular data. You might want to consider whether your app needs a connection at all. If your users' needs can be served without an active Internet connection, there's no need to feel compelled to include that capability. It all comes down to the need your app is fulfilling—and how best to meet it.

In addition to Wi-Fi connectivity, certain Fire tablets can connect to mobile networks through a user's mobile network plan. Since some mobile plans have a limit on how much data can be transferred in a month, it's important to give users the ability to defer large downloads or media streaming.

### There are three steps to using device targeting for your app:

1. Build multiple binary APKs for different devices, optimizing each binary for each device or platform.
2. Configure the Android manifest for each APK to use different version codes.
3. Use the Binary File(s) tab on the Amazon Developer Portal when you submit your app to upload your individual APKs as part of your app submission.

Avoid silently transferring data when the device is connected to the mobile network. Instead, batch data and transfer it when the device is connected to a Wi-Fi network. If this isn't possible, consider alerting the user with the option of continuing the transfer or waiting for a Wi-Fi connection.

Because a mobile network connection might shut down when the data limit is reached, your app must deal with the lack of connectivity gracefully. Alert the user when it isn't possible to transfer data.

## Using web-based services

Using web-based services makes sense when your app needs to access data frequently or send messages or requests to other users/resources. Keep in mind that the frequency and size of communications that your app requires will have a significant impact on your users' experience. You'll want to design your app's messages and requests to be small and lightweight and to occur at the right frequency.

The following section on data highlights a few cloud data storage services to consider, depending upon your users' data needs. But data storage is only one aspect of cloud services that can support your app.

Compute services—such as Amazon EC2, Amazon EC2 Container Service, and AWS Elastic Beanstalk—provide a virtual infrastructure that can act as servers for your app if heavy compute cycles are required. One example of this is multiplayer gaming, where a dedicated set of servers can provide the central control needed for a robust gaming experience. An additional benefit of using infrastructure as a service is the ability to build scaling into the experience—increasing capability when demand is high and decreasing it when demand is low.

Depending on your data needs, AWS offers several small-to-large scale data services. These include simple file storage with S3, relational databases with RDS and Aurora, and data warehousing with Redshift. The options for data storage as a service are numerous enough to be described in a bit more detail later in this chapter. (See **Dealing with data**.)



## Using web-based services continued

AWS offers ElastiCache, an in-memory caching service. If your users are consistently requesting the same block of data repeatedly, ElastiCache can greatly improve the performance of data transfer for your app. It offers storage for regularly accessed data in memory, allowing this data to be accessed far faster than data stored on disk. ElastiCache supports both Redis and Memcached caching engines. Mobile services on AWS range from identity management to API management and test and monitoring services. Amazon Cognito manages both user identity and app data synchronization. The API Gateway service enables the publishing of REST APIs, which can provide a common backbone for all your apps—mobile or desktop. AWS Mobile Hub can be used to build, test, and monitor apps, and as a mobile-specific backend for apps. Mobile Hub also ties into Amazon Pinpoint.

AWS includes messaging services for apps as well. SQS message queuing service sends messages between components. Amazon Simple Notification Service (Amazon SNS) is a fast, flexible, fully managed push notification service that lets you send individual messages or fan-out messages to large numbers of recipients. Amazon SNS makes it simple and cost-effective to send push notifications to mobile devices—including iOS, Windows, and Android—worldwide. Amazon Pinpoint is an analysis and notification service built to push targeted notifications to mobile app users; it works with AWS Mobile Hub to analyze user behavior and create targeted campaigns with measurable results.

These are just a few of the offerings available from AWS, mostly focused on app development. For a more comprehensive list of services, or more detail on service offerings, visit [Amazon Web Services](#).<sup>25</sup>

<sup>25</sup><http://amzn.to/2onD3rE>

## Dealing with data



Most users take exception to their information being collected and used without their permission. If you need to uniquely identify users, you can utilize existing platforms to identify and authenticate. However, if your app does collect personal information, be aware that there are countries/regions that have very strict privacy laws around personal data review—so you'll want to review policies around data collection and retention closely.

In most cases, you won't be keeping sensitive information. But beyond configuration information, you might need to store data on a user's behalf to deliver a consistent experience. You'll need to consider whether the volume of data will be a concern. For example, using gigabytes of mobile device storage on a whim isn't advisable. And if data needs to be accessed from another device, avoiding local storage would be ideal.

There are several options for data storage on the web. If you're dealing with small amounts of user data, scalable services such as Amazon Simple Storage Service (S3) provide a cost-effective, expandable solution for your app data storage. S3 allows for individual file storage in containers, which function like files on your local computer—except that they're accessible by web API.

File-based storage with S3 is an unstructured, NoSQL-type solution. If your app deals with a large number of small, unstructured data files, a more mature NoSQL data store might be needed.

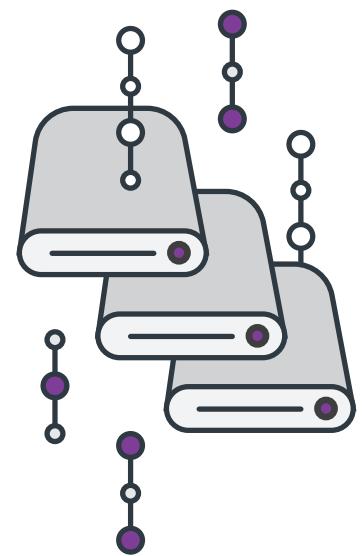
Amazon DynamoDB is an AWS offering that provides scalable NoSQL data storage, which can be well-suited to mobile, web, or gaming applications.

If your application requires medium- to large-scale relational data storage, you might want to consider Amazon Relational Database Services (RDS) to host a relational database like PostgreSQL, MySQL, or MariaDB. As is the case with S3, RDS pricing can be based upon usage, which reduces cost during low access periods.

At the larger end of relational data is Amazon Aurora, an enterprise-class, high-performance, scalable system. In all likelihood, if you're working with 10 to 24 GB of relational data,

you have one or more individuals focused specifically on data management, and their requirements will determine whether an enterprise relational database engine is the right tool for the job.

Ultimately, deciding which data storage solution to use should be based upon the type of data you need, how it's collected or used, the volume of data to be stored, and when and how the data is accessed.



Collecting data analytics will improve your ongoing efforts. Designing the correct telemetry allows you to collect and measure your users' behaviors with your app, and work to confirm your expectations on how your app is used. It can also tell you whether users are abandoning any efforts—either due to malfunction or inaccessibility. Finally, telemetry data can give insight to malicious behavior that might indicate a security flaw in your app.

By tracking key trends like new vs. returning users, app revenue, user retention, and custom in-app behavior events, you can make data-driven decisions to increase engagement and monetization for your app. The Amazon Mobile Analytics Developer's Getting Started Guide describes the process of integrating the SDK, configuring the Auto Export feature to export events data to Amazon S3, and using the REST API to submit events.<sup>26</sup> You can use the Amazon Mobile Analytics Console to view graphical reports for your app, download data in CSV format, or configure the Auto Export feature to export your raw events to your Amazon S3 bucket.

<sup>26</sup><http://amzn.to/2osC6NO>

---

# Designing the correct telemetry allows you to collect and measure your users' behaviors with your app, and work to confirm your expectations on how your app is used

---



# FIVE

## Monetizing your app

---

The Amazon Appstore makes it easy to earn money from your published apps. In addition to the more traditional monetization routes—like In-App Purchasing (IAP) or download fees for premium apps—Amazon Appstore also offers unique opportunities that are exclusive to the Amazon Appstore, including Merch by Amazon and Amazon Coins.



## Preflight checklist

---



- Have a plan for your app
- Identify your revenue goals
- Understand your audience

## Challenges you might face

---



- Selecting an app monetization model that fits with your app, is appealing to your user base, and provides a frictionless experience to users
- Planning and integrating IAP monetization throughout the dev process
- Identifying what does and doesn't work
- Making the needed adjustments throughout the process

## Choosing the best monetization model for your app

As you're planning your app, you'll want to select an app monetization model that fits your use case, appeals to your user base, and provides a frictionless user experience.

## Popular monetization business models

### In-App Purchasing (IAP)

In this model, your app is usually free to download and use. However, users can enhance their experience by purchasing extra features—like advanced services, extra currency, additional lives in a game, and more—or through access to locked content, like bonus levels or mini-games in the app.

### Subscription-based apps

In a subscription-based model, users will pay a recurring amount to access premium or exclusive features or content in the app. The subscription model is most often used by apps for music and video, social networking, or online media. Subscriptions are included in the Amazon Appstore's IAP API and handled the same way as other in-app purchases.

If you charge money for your app, your app includes IAP, or you use Amazon Mobile Ads, you must fill out payment, banking, and tax information in your developer profile.

### Premium (paid) apps

In this model, users pay to download your app. This model is good for apps that show a clear benefit to the user or that already have a strong fan base or independent publicity that drives users to download the app. Because users can't try the app before they decide to make the purchase, premium apps tend to see lower numbers of downloads. Therefore, this model is used less frequently.

## Types of in-app purchases

As you plan your IAP implementation, define what types of items your app will make available to users. You'll also need to determine how you want to deliver those purchased items.

**IAP includes three different categories of purchasable items:**

### Consumables

Purchases that are made and then consumed within the app—such as extra lives, extra moves, or in-game currency. These items can be purchased multiple times.

### Entitlements

One-time purchases to unlock access to features or content within an app or game.

### Subscriptions

Access to a premium set of content or features for a limited period of time.

IAP supports two basic delivery flows: Instantly available content and deliverable content. Instantly available content is available to the customer upon purchase. Under this model, your app already has everything it needs for the customer to use the purchasable item immediately upon purchase. Deliverable content allows you to make new content available to the customer. Under this model, your app will download the new content from your servers and make it available to the customer. (Typically, these are subscription items.)

## Executing the purchase process

Create your purchasable items before submitting your app to the Amazon Appstore. If you decide to add or edit your purchasable items after submitting your app, you'll need to resubmit both the new/changed items and the app itself. Once you've completed the submission of these purchasable items, the Amazon Appstore executes the purchase process for you.

Amazon's IAP API handles the details of purchase flow, payment processing, and managing rights to the purchasable content, so you don't have to code these on your end. When a user initiates a purchase of your app, the Amazon Appstore client app surfaces and presents an Amazon-branded user interface to complete the transaction. You don't need to provide purchase dialogs, transaction timeout logic, or "Thank You" dialogs. The Amazon Appstore provides all of these transaction pieces, including either a receipt for the purchase or a status code in the case of a failed purchase.



*Amazon's IAP API handles the details of purchase flow, payment processing, and managing rights to the purchasable content*

## Setting up a premium (paid) app in the Amazon Appstore

You can set the price of your app during the app submission process (detailed in [Chapter Seven: Publishing your app](#)). The Availability & Pricing tab will ask if you're charging for the app. Select Yes. Then enter a base list price and currency from the drop-down menu. There's a minimum base list price of \$0.99 USD.

## Amazon Mobile Ads API

Mobile ads can be used as the sole monetization method in an app, or you can choose to use them in addition to making in-app purchases available to your users.

The Amazon Mobile Ads API is an in-app display advertising solution to monetize mobile apps and games across platforms, including phones and tablets on Android, iOS, and Fire. The Mobile Ad network is easy to integrate and track. With a single portal, you can measure your success rate with actionable, simple reporting. When you use the network, you'll get paid based on ad impressions served—not on clicks on ads. Users will receive high-quality, highly relevant, mobile-optimized ads from Amazon and brand advertisers.

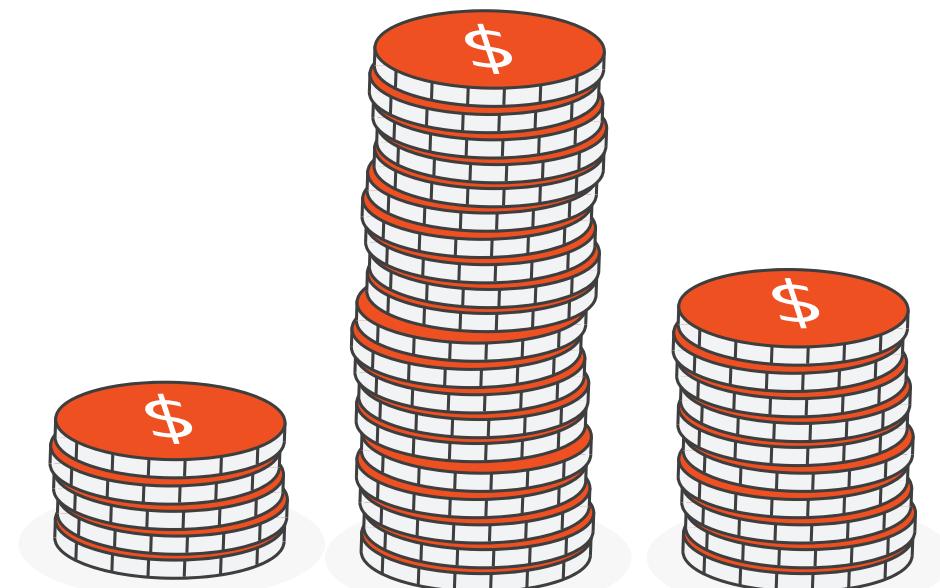
**"[With] Amazon Mobile Ads API... we saw eCPM jump 300% and our revenue doubled"**

—Anatoly Lubarsky,  
Founder, [X2Line.com](#)

**"The \$2.87 eCPM we saw through Amazon far exceeded our expectation"**

—Mahesh Khambadkone,  
Co-founder, [Games2Win](#)

Customers can use Amazon Coins just like cash to buy apps and make in-app purchases. Amazon Coins gives customers more ways to buy, alongside special incentives and discounts. While customers save money when buying Coins, as a developer, you continue to earn your standard revenue share. And because of the incentives customers receive to download and use your apps and games, you get more opportunities to earn money through your app.



**“Working with Amazon is a dream come true. We don’t have to worry about selling the shirts or holding any inventory. It’s a one-stop shop allowing us to grow the brand outside of the game.”**

—Bryan Davis  
Big Blue Bubble,  
creator of My Singing Monsters

## Merch by Amazon

Give your users a fun new way to interact with your brand through the Merch by Amazon service for developers. Merch by Amazon helps you increase revenue through the sale of branded T-shirts designed by you and produced, sold, and shipped by Amazon. You can convert game characters or imagery into wearable merchandise and expand the life of your game beyond the screen.

If you're interested in getting started with Merch by Amazon, you can request an invite to the program. Once you receive your invite, simply set up your Merch by Amazon account, upload the artwork for your T-shirt, and press submit. Amazon takes care of the rest, including worldwide sales, distribution, logistics, shipping, and customer service. You earn a royalty for every T-shirt sold, and there's no out-of-pocket cost.

Every brand is different, but we've found that most have fans who would love to wear branded T-shirts. We suggest that you test out at least 5 or 6 unique designs to determine which designs will appeal to your fan base. Present each design to your fans, see what works, and then iterate.

Through the Merch by Amazon portal promotions section, you can use social and in-app promotions to share your T-shirts with users. Use Pinterest, blogs, YouTube, email campaigns, or web campaigns to reach out to your users and direct them to your T-shirt product page on Amazon. Find your T-shirt's product URL on the Dashboard and Promote sections of the Merch portal.

### Four proven categories that work well



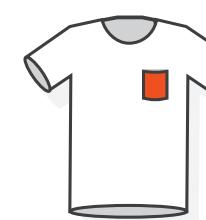
Logos



Funny/inside jokes



Character-driven designs



Subtle designs that look "cool" to non-fans, recognized by true fans

#### Reduce barriers

Your IAP strategy could be perfect, but if people aren't spending any time in your game, you won't be able to generate any revenue. The top developers in the Amazon Appstore make it easy to get into their games immediately. No splash screens, welcome menus, or option screens—users jump straight into the fun.

#### Increase user sessions with cross-platform integration

Successful developers also remove barriers to subsequent sessions through cross-platform integration. The ability to sync gameplay across devices lowers user frustration and increases session use.

#### Simplify the purchase process

Provide users with detailed, accessible instructions on making in-app purchases, and follow-up with clear instructions on how to use the newly purchased items.

#### Target your app catalog

You only have your customer's attention for a short period of time, so it's important to provide the proper pacing and the right IAP catalog to coincide with a user's level of play. Amazon research has shown that players who stay with a game for at least 30 days spent 60 percent more on their in-app purchases. In addition, the top games have a wider variety of items in their catalog to enhance gameplay and keep players progressing through the game. A deeper IAP catalog helps you make the right offers to the right players at the right time.

#### Provide clear price points

While a large number of IAP items in your catalog can help increase sales, too many price points can overwhelm customers. Try to offer no more than 5 price points, and include expensive items in that count (for example: \$0.99, \$4.99, \$9.99, \$19.99, and \$49.99).

#### Test, test, test!

A/B testing can help you to get the most out of your monetization strategy without impacting retention. Set up trials and see what kind of timing, price points, and purchase options are most appealing to your customers.

---

# A deeper IAP catalog helps you make the right offers to the right player at the right time

---



# Preparing and testing your app

---

Before releasing your app to your customers, you want to make sure it's solid. In this chapter, you'll find information on preparing your app for publication, including testing criteria for various devices, and how to navigate Amazon's App Testing Service. Once you've finished taking the actions in this chapter, you'll have an app that's complete and tested—and ready to submit to the Amazon Appstore.



## Preflight checklist



- Understand the app submission workflow
- Have an Amazon Developer account  
(requires an Amazon Customer account)
- Get required images ready for submission
- Target specific devices for your app
- Target locales/regions for your app

## Challenges you might face



- Meeting the requirements for a successful submission
- Resolving issues found in test
- Releasing a beta for your app

## Get your app ready for submission

Develop your app from scratch or migrate an existing Android app for publishing in the Amazon Appstore (see **Chapter Two: Migrating your existing app**). Keep the Amazon Appstore Content Policy Requirements in mind as you design your app.<sup>1</sup>

### Have (or create) an Amazon developer account

You need to have an Amazon Developer account to submit apps on the Amazon Appstore.<sup>2</sup> If you don't already have an account, you'll need to register for one at the Amazon Developer Portal. There are two important things to know about your Amazon Developer account:

- Your Amazon Customer account is tied to your Amazon Developer account.
- Amazon will only send email to the address listed in the company profile when corresponding with you about your app's status.

When you set up a new Amazon Developer account, you are the default account administrator. You can add other administrators and people in other roles to the account, such as analysts, developers, or marketers. Although any developer or administrator can submit an app, Amazon will send correspondence only to the email address in the company profile.

During the registration process, you'll be asked to review and accept the App Distribution Agreement. You'll also be asked whether you plan to monetize your apps, and if so, how. You can change your monetization settings after registration is complete, even if you decline monetization at this point. Once registration is complete, you'll land on the Developer Console Dashboard, where you can begin to add a new app.

<sup>1</sup><http://amzn.to/2onEWoi>

<sup>2</sup><http://amzn.to/1WCYfrt>

## Image assets for production promotion

To promote your app on the Amazon Appstore, you'll need to provide screenshots along with large and small icons. You can also include a promotional image and videos. See Image Asset Guidelines for Appstore Submission.<sup>3</sup>

<sup>3</sup><http://amzn.to/2oW6Mem>



## Test your app

When you submit your app or game to the Amazon Appstore, the app must pass a series of tests to qualify for publication. To avoid time lost in getting your app to market, make sure that your app meets the Test Criteria for Amazon Appstore Apps before you submit it for approval.<sup>4</sup>

<sup>4</sup><http://amzn.to/2pCBNog>

## Test criteria for Amazon Appstore apps

### General qualifications

At a high level, your app should:

- Have a simple and well-planned user experience that doesn't cause user confusion.
- Be thoroughly tested and free of bugs, crashes, and obvious defects.
- Provide visual indication that a user's action has succeeded or failed.
- Fail gracefully if the user tries to access unsupported functionality. (For example, the app could display an error message that reads, "This feature is not currently available.")
- Fully utilize the screen area of the Fire tablet or Fire TV screen.
- Be free of visual defects, such as missing/distorted/pixelated images, misaligned UI elements, or illegible text.
- Have UI text that is free of grammatical or spelling errors.

## More Test criteria for Amazon Appstore apps

**Additional qualifications for Fire TV**  
Apps designed for Fire TV should meet the following additional qualifications:

- Design your app to be displayed in 1080p resolution (1920 x 1080 px) for the best possible experience.
- Limit text entry requirements. Large amounts of text entry hinder the user experience.
- Enable users to read text and UI elements from approximately 10 feet away.

Try to keep your apps to 2 GB or less (recommended). This size includes any first-launch downloads of additional assets or other content. Note that if your app is above or near 4 GB in size, it might not install or run on the device.

## Amazon Fire TV remote and voice remote qualifications

To enable your app to be compatible with Fire TV remotes, follow these guidelines:

- Implement a simple UI that's easily navigable with Up, Down, Left, and Right movements (D-pad).
- Don't require motion events or rapid input for use.
- Enable use with only one hand, and don't require the use of two hands for sustained interactivity.
- If the user needs to hold the remote sideways to use your app, inform the user of that on launch so navigation is handled correctly.
- Avoid simultaneous button inputs, especially for core functionality.

## Additional qualification for Fire tablets

Apps designed for Fire tablets should meet the following additional qualification:

UI should take into consideration and utilize both the notification bar and dynamic soft key menu behavior.

## Get your app ready for submission

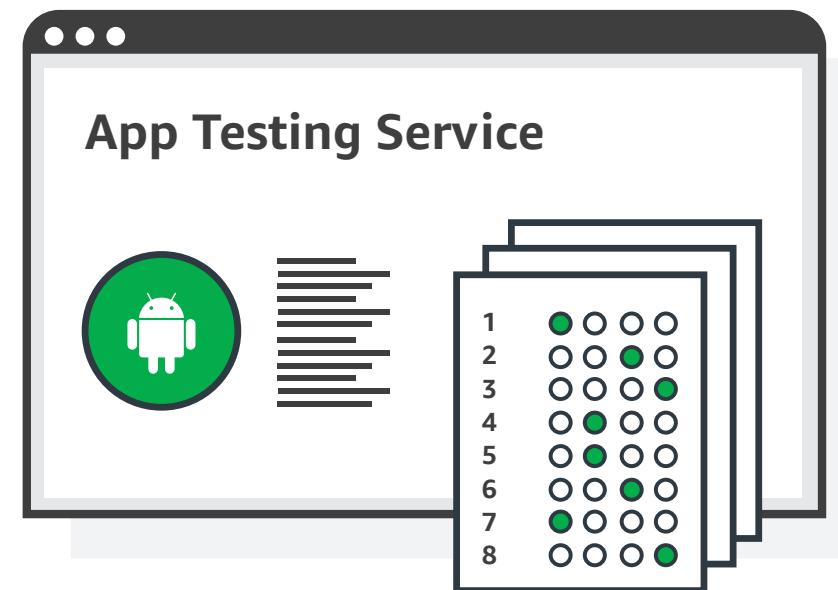
### Game and other controllers

To ensure your app is compatible with game controllers (the Amazon Fire game controller and other Bluetooth game controllers), it should feature a simple UI that is easily navigable with D-pad or analog stick input.

If your app requires the use of any other controller, that controller must support the Bluetooth HID gamepad profile and not require additional software for input.

Amazon maintains a set of test criteria that need to be met for your submission to succeed. For the latest test criteria, see Test Criteria for Amazon Appstore Apps.<sup>5</sup>

<sup>5</sup><http://amzn.to/2pCBNog>

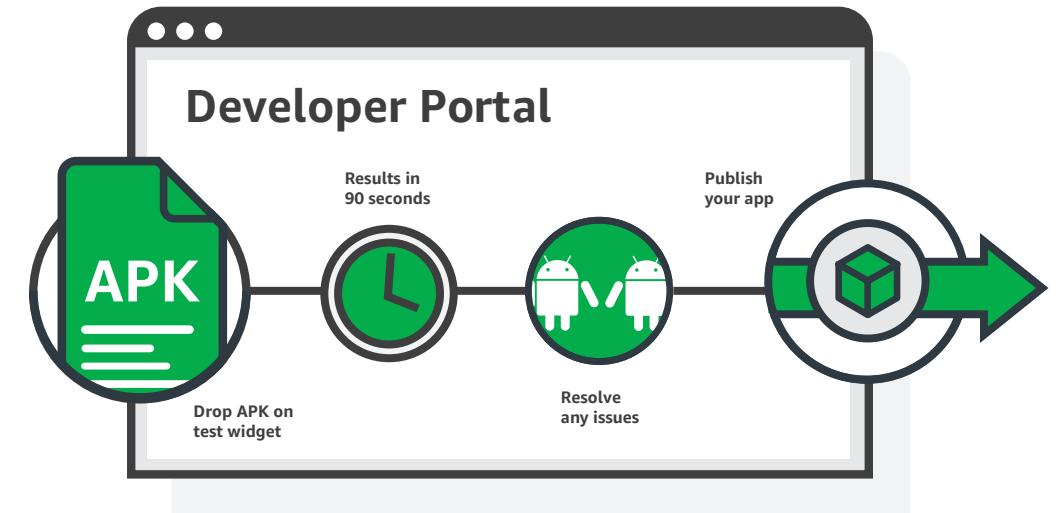


### App Testing Service

Amazon's App Testing Service allows you to upload your app's Android application package (APK) to the Amazon Appstore and receive feedback about whether your app meets submission requirements. You'll also receive suggestions on how to correct potential submission issues. The App Testing Service performs two types of tests on submitted apps: a compatibility test, to verify that unsupported APIs are not in use and check for incorrect API usage or unsupported features, and a device test, which tests your app on actual devices and performs automated tests.

The device test uses AppXplorer, a tool optimized for analyzing native Android UI widgets. Currently, AppXplorer is not optimized for user interfaces based on OpenGL or HTML5. There are also limitations with authentication, and API levels 18 and above cannot be tested with AppXplorer. If AppXplorer is unable to install, launch, or explore the app, an email notification is sent to the developer account email.

If you're looking for a way to beta test your app among hundreds of other developers, it's a good idea to use Amazon's Live App Testing. For more information, see Live App Testing.<sup>6</sup>



## Interpreting test results

Log in to your Developer account to see the test results for your app. Amazon never shares test results publicly.

**Note:** You might get a "Results Not Found" error if the email address used to submit for device testing is different from the email address used for signing in to your Developer account.

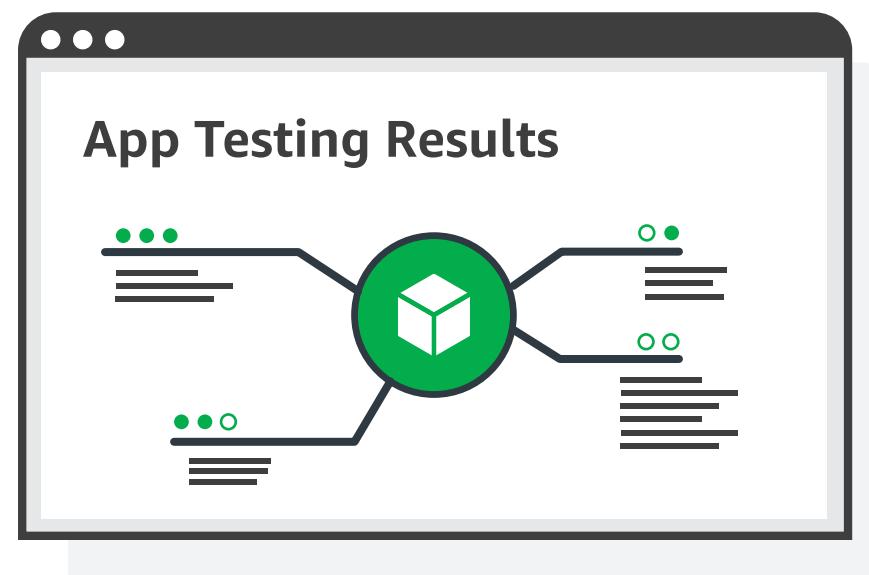
Compatibility test results will be available within 90 seconds. Device test results might take up to 6 hours, depending on traffic on the service. If the AppXplorer testing load is high, test results can be delayed up to 24 hours from the time of app submission.

If you test the same app, unchanged, multiple times by using the App Testing Service, your device test results from AppXplorer might change slightly. This is because AppXplorer can explore the app using different app flows, which might result in navigating to different screens and detecting app crashes only during a specific app flow. This scenario isn't common but can happen with some apps. Your compatibility test results will not vary.

## Submitting your app from the App Testing Service

After you have fixed any issues identified by the App Testing Service and your app passes testing, you can submit your app to the Amazon Appstore for publication directly from the testing portal. If you don't want to submit your app immediately but are signed in to your Developer account when you run the test, Amazon will save your test results and you can submit your app later. To learn more, see the App Testing Service page.<sup>7</sup>

<sup>7</sup><http://amzn.to/2n8Puel>



If you're looking for a way to beta test your app among hundreds of other developers, it's a good idea to use Amazon's Live App Testing

A large school of barracuda fish swims in the upper left corner of the image. In the center, a scuba diver in a black wetsuit and fins is swimming towards the bottom right, creating bubbles. The background is a clear blue ocean.

# SEVEN

## Publishing your app

---

At this point, your app should be fully prepared (or migrated), tested for compliance and functionality, and ready for submission to the Amazon Appstore for publication. This chapter takes you through the publishing workflow. Even if you're not publishing for the first time, it's useful to refer to this chapter to make sure you don't miss a step.



## Preflight checklist

---

- Understand the app submission workflow
- Determine your app Android application package (APK) and image assets
- Choose a monetization model
- Select a security profile (Device Messaging)



## Challenges you might face

---

- Checking the status of your app submission
- Identifying the next steps if your app fails submission



## Submit your app

Log in to your developer account, enter the appropriate information for your app's listing, and upload your app.

### The app submission process includes the following steps:

1. Log in and create your app.<sup>1</sup>
2. Enter general information for your app.
3. Enter availability and pricing information.
4. Enter app descriptions.
5. Upload image and media assets.
6. Assign a content rating.
7. Upload Android application package (APK) files.
8. Submit your app.

## Step One: Log in and create your app

Log in to your developer account on the Amazon Developer Portal.<sup>2</sup> Once you sign in, the Developer Portal displays the Dashboard page for your account.<sup>3</sup> On the bottom of the page, you'll see all your previously created apps listed.

To create a new app, scroll to the bottom of the Dashboard page and click **Add a New App**.

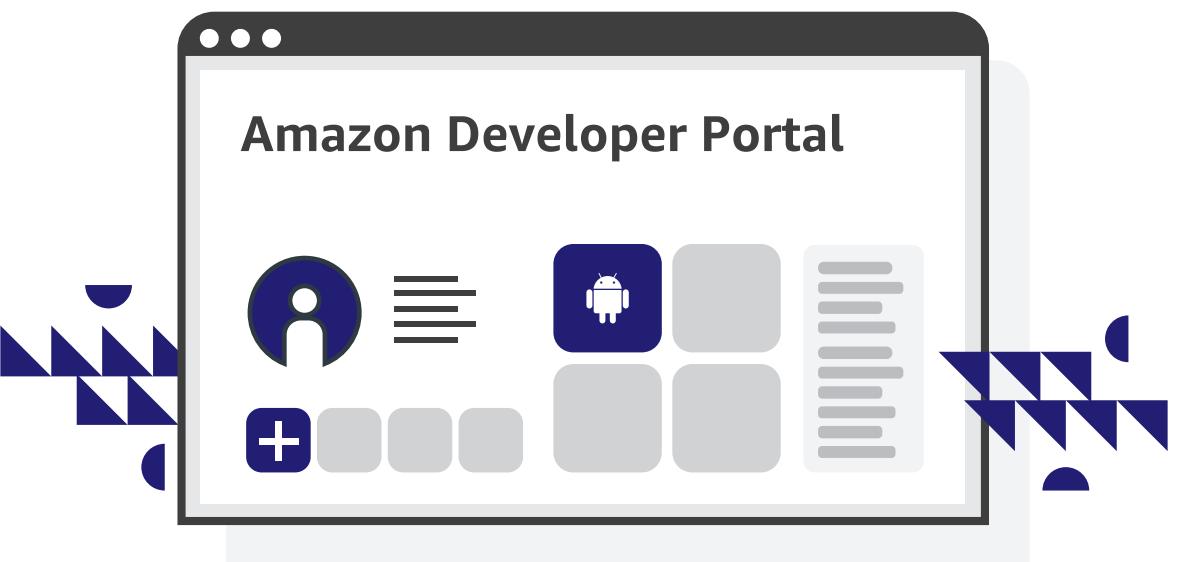
You will be prompted to select a platform: Android or mobile web.

Select **Android**.

The portal displays a page for entering basic information about your app.

<sup>2</sup><http://amzn.to/1gF8KUB>

<sup>3</sup><http://amzn.to/2pSlpwI>



<sup>1</sup><http://amzn.to/2pSlpwI>

**Step Two:**  
**Enter  
general  
information  
for your app**

After adding your app, you'll come to a page containing text fields to capture general information about the app. All text input fields here are plain text only. Do not enter HTML markup. If you enter HTML markup in any field, your app will fail Amazon testing and publishing will be delayed.

**For Android apps, enter the following information:**

- **App title (required)**  
Enter your app's title. Amazon customers see this information in the Amazon Appstore and in certain locations on Fire tablets and other Android devices.
- **App SKU (optional)**  
Amazon doesn't require or use the app SKU. However, adding a unique identifier can help you track multiple versions of your app.
- **Category (required)**  
Select an app type and subtype by using the drop-down menus. This information helps customers find apps like yours.
- **Category refinements (optional—only shown if available)**  
Select any of the refinements that apply to your app. This information helps customers find apps like yours.
- **Customer support contact (optional)**  
If you have configured a customer support email address, phone number, and website in your profile, you can optionally select the **Use My Default Support Information** check box to copy that information. If you don't select the check box to copy the default information, you'll need to enter an email address, phone number, and website in the following text fields.

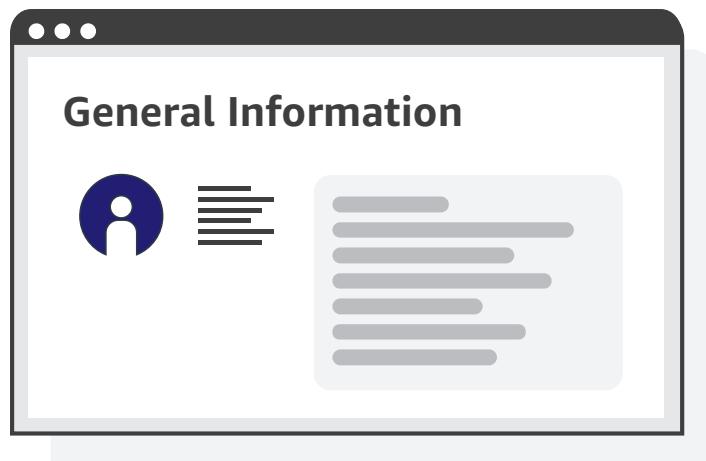
- **Customer support email address (required)**  
Enter an email address where customers can contact you.

- **Customer support phone (required)**  
Enter a phone number where customers can contact you.

- **Customer support website (required)**  
Enter the URL for your app's support website.

- **When finished, click Save.**

A General Information tab containing the information that you supplied on the previous screen appears. If you need to make changes, click **Edit**. You can make changes at any time before you submit your app.



**Step Three:**  
**Enter app  
description**

---

Use the Description tab to provide long and short descriptions for your app's listing and to communicate the special features of your app. **To enter your app's description and other listing information:**

**1. Open the Description tab**

**2. Language (required)**

Displays the default language for your app. All apps except those originating in China require an English (U.S.) translation. To add another translation, click **Save** and **Add localized descriptions**. You'll need to select a language and fill out all the required information on the Description tab in the new language.

**3. Display title (required)**

Enter the app title that will be displayed in your app's listing. The display title should be brief. It doesn't have to be the same as the App title field on the General Information tab.

**4. Short description (required)**

Enter a short description of your app that's appropriate for mobile devices. The maximum length is 1,200 characters, but Amazon recommends a much shorter description.

**!** **Note:** *Keep the following in mind for your short description in regard to Fire TV. The Fire TV user interface uses your short description but only displays the first 200 characters. If you're submitting a Fire TV app, optimize your short description to include the most important information in the first 200 characters.*

**5. Long description (required)**

Enter a description of your app that's appropriate for use on the Amazon Appstore. The maximum length is 4,000 characters.

**6. Product feature bullets (required)**

Enter 3 to 5 key features of your app, one per line. Press the **Return** key after each feature, and the portal will automatically convert each feature to a bullet item.

**7. Keywords (optional)**

Enter comma-separated search terms that will help customers find your app in the Amazon Appstore.

**8. Click Save.**



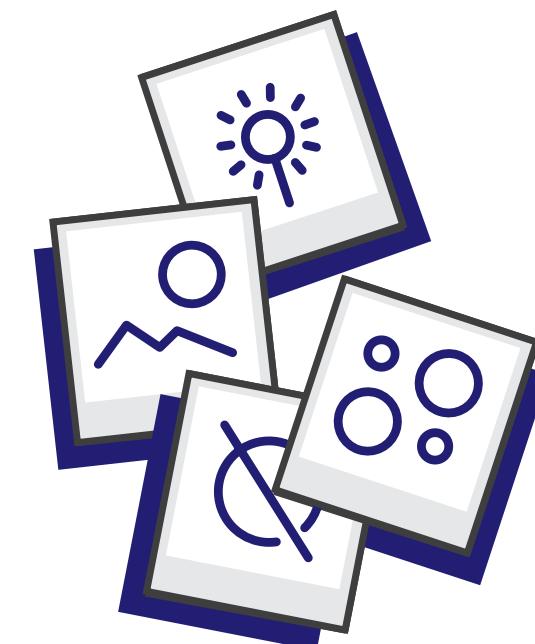
**Step Four:  
Upload  
images and  
media assets**

Use the Images & Multimedia tab to specify the images that represent your app on Amazon product detail pages and in the device user interface. Amazon also uses these images and videos to promote your app.

**To add images and multimedia to your app submission:**

1. Open the Images & Multimedia tab.
2. Read through the Image Guidelines for Appstore Submission and upload the required images for your app.<sup>4</sup>
3. When you're finished uploading image assets, click **Save**.

**!** **Note:** If the file size of a single video is greater than 150 MB, you must upload the video via SFTP.



**Step Five:  
Assign a  
content  
rating**

On the Content Rating tab, you define a content rating for your app. Amazon uses these ratings to target your app to appropriate audiences. An inaccurate representation of your app's content could cause your app to fail Amazon's testing process, or could cause Amazon to remove your live app from the Amazon Appstore.

**To define a content rating for your app:**

1. Open the Content Rating tab.
2. Under Subject Matter, for each of the listed categories, select the radio button in the column that best describes how often your app makes use of or refers to the listed item:
  - **Light to moderate** means that the item occurs once or rarely and is not fundamental to the overall purpose and/or intent of the app.
  - **Strong** means the item occurs regularly and is fundamental to the overall purpose and/or intent of the app.
3. In the Additional Information area, select **Yes** or **No** for each of the following options. Making the appropriate selections here will shorten your app's review period.
  - Account creation or other personal information collection
  - Advertisements
  - Is your app directed at kids under age 13?
  - Gambling
  - Location detection or location-based services
  - User-generated content or user-to-user communication
  - Privacy policy URL
4. Click **Save**.

**Step Six:  
Upload APK  
files**

Use the APK File(s) tab to set permission and device support options and to upload your APK file to the Amazon Appstore.

In addition to specifying which devices are supported by your app on the APK File(s) tab, you need to configure your Android Manifest file to specify which devices are supported. See Getting Started with Device Targeting.<sup>5</sup>

**Uploading  
an APK File**

To upload your app's APK file and set permission and device support options:

**1. Open the APK File(s) tab**

**2. Apply Amazon DRM? (required)**

Choose Yes (Recommended) or No.

When your app starts, it communicates with the Amazon Appstore client to determine whether the current user is entitled to use your app. If you select Yes (Recommended), Amazon digital rights management (DRM) is enabled and your app will only run if the signed-in user has an entitlement to your app. If you select No, DRM is not enabled and the app will run no matter who is signed in. Your app will always communicate with the Amazon Appstore client when it starts, even if you choose not to apply DRM.

Regardless of whether you choose to apply Amazon DRM, Amazon wraps your app with code that enables the app to communicate with the Amazon Appstore client to collect analytics, evaluate and enforce program policies, and share aggregated information with you.

The Amazon DRM is not tied to the key you used to sign your app. Amazon removes your signature and re-signs your app with an Amazon signature that is unique to you, doesn't change, and is the same for all apps in your account.

**3. Binary file (required)**

Drag your app's APK file (also called a binary file) onto the Upload box, or browse to the file to add it.

<sup>5</sup><http://amzn.to/2osJ1GR>

**4. Language support (required)**

Select at least one language supported by your binary. (The default is English.)

**5. Export compliance (required)**

Accept the export compliance statement by selecting the check box to confirm that your app complies with applicable export and import regulations.

**6. Testing instructions (optional).**

Use this field to communicate with the Amazon testing team. For some apps, you can leave this field blank.

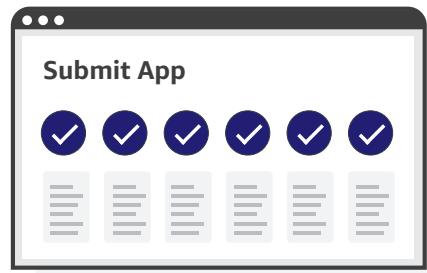
If your app has features that are difficult to discover, include directions here for finding them. Also, if you need to resubmit your app based on feedback from Amazon, or if your app needs user credentials or other information for testing, you can use this field to respond to the test report emails you receive from Amazon.

**9. Click Save.**

## Step Eight: Submit your app to Amazon

Once you've successfully entered all the required information, the portal will display green check marks on all six tabs and enable the **Submit App** button. You can now complete the app submission process.

Click **Submit App**. Respond to any failure notifications (if needed).



If your app fails any submission criteria,

you'll receive a notification that the app has been rejected. Fix your app as directed and resubmit it. Before approving an app and making it live on the Amazon Appstore, Amazon verifies that your app:

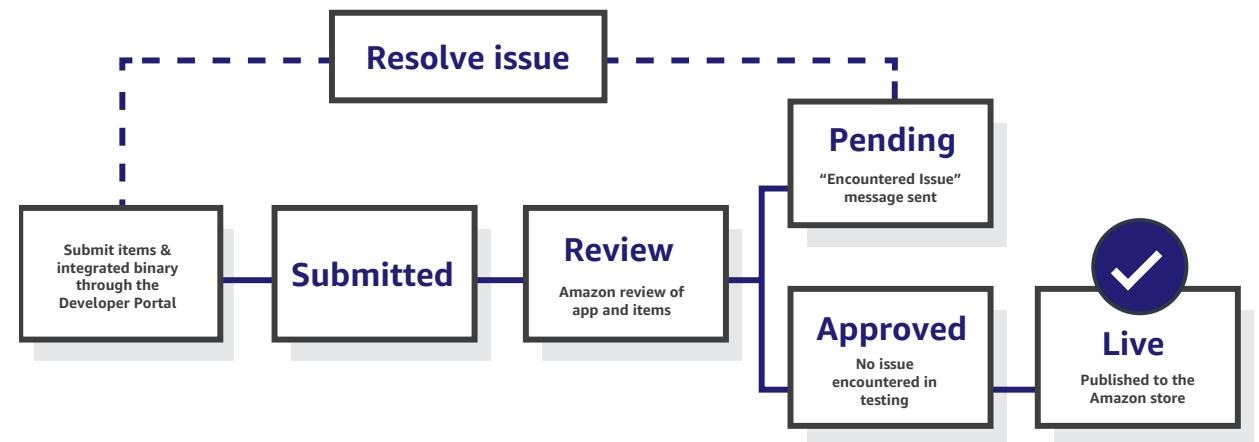
- Works as outlined in your product description.
- Doesn't impair the functionality of the device on which it's running.
- Doesn't put customer data at risk.
- Complies with the terms of the Amazon Program Materials License Agreement<sup>7</sup> and the Amazon Appstore Content Policy Requirements.<sup>8</sup>

<sup>7</sup><http://amzn.to/2osH1yk>  
<sup>8</sup><http://amzn.to/2onEWoi>

## Step Eight: Submit your app to Amazon cont.

After approving your app, Amazon sends an email to the address associated with your company profile. This email address might not be the same as your Amazon Developer Portal account address. Please see the note about this at the top of the page, under Accounts and Correspondence in the section titled Prepare to Publish Your App. If after six days your app is still in review, you can use the Contact Us form to seek clarification.<sup>9</sup>

<sup>9</sup><http://amzn.to/2daxOKs>



Typical flow

Your dashboard in the Amazon Developer Portal displays the status of every app that you submit to Amazon. The status messages are defined here:

- **Incomplete**

You haven't yet completed your app submission. The app is missing some of the fields required for submission to Amazon.

- **Ready to Submit**

Your app is ready to be submitted to Amazon. You can change the metadata and binary until you click **Submit App**.

- **Submitted**

You have submitted the app for review, and it can no longer be edited. At this time, you can cancel the request for review and continue to edit your app by clicking the **Cancel review** link.

- **Under Review**

Amazon is reviewing the app. You can choose to cancel app review and restart the submission process.

- **Approved**

The app has completed at least one stage of the testing and content review process

and might soon be ready for publication. No changes can be made to the app while it's in this status. You must wait until the app is live before making changes.

- **Live**

The app is now live on the store.

- **Pending**

You will receive an email from the Amazon team explaining why the review process for your app was paused and asking you to perform some action. You must address the relevant metadata or binary issues before your app can successfully continue through the process.

- **Rejected**

The app did not pass the review process. You will receive an email from Amazon explaining the reasons for the failed approval.

- **Suppressed**

The app was live at one point but is no longer in the Amazon Appstore.

**Update  
your app  
(as needed)**

If you need to make changes to an app that has already gone live in the Amazon Appstore, you can simply update your published app in the Amazon Appstore rather than creating a new one. Updating an existing app has benefits such as:

- Enabling you to retain the app's customer ratings and download statistics across versions.

- Enabling your customers to receive app updates.

**Transferring  
apps to  
another  
developer  
account**

If a situation arises (such as a company merger or spin-off) where you need to transfer one of your published apps to another Developer account, Amazon support can help. However, note that unless it's absolutely necessary, Amazon strongly discourages developers from transferring apps to different Developer accounts. (See How do I transfer an App from one Developer Account to another?<sup>12</sup>)

<sup>10</sup><http://amzn.to/1qF8KUB>

<sup>11</sup><http://amzn.to/2ITTkU1>

<sup>12</sup><http://amzn.to/2osNzgx>

Once Amazon approves an app and makes it live on the Amazon Appstore, you can keep it up-to-date by submitting updated versions through the Amazon Developer Portal.<sup>10</sup> Through the portal, you can perform any of the following:

- Delete a submitted but unpublished app.
- Modify device support for your app.
- Add or modify APK files to a published app. For more information, see Updating a Published App.<sup>11</sup>

# EIGHT



## Promoting your app

---

Coding your app might be the hard part, but it's not the last part. You still need to sell your app. Developers often ask how they introduce their apps and games to new Amazon customers. This chapter guides you through the basic steps of promoting your app—both in and out of the Amazon Appstore—and reaching potential customers.

There are two steps to marketing your app. You start by creating your detail page content and assets—what customers see once they've already decided to learn more about your app. Then you work on driving traffic to your app through marketing assets or opportunities.



## Preflight checklist

---



- Understand your options and decide where to invest—with both time and money
- Commit to ongoing promotion efforts
- Determine the promotion methods that will best fit your audience
- Choose the right keywords and write effective product copy

## Challenges you might face

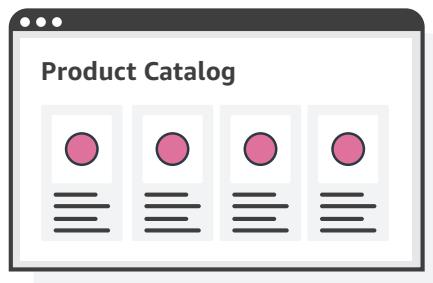
---



- Writing a detailed product description of your app
- Selecting relevant keywords for your app
- Choosing screenshots for your app
- Creating a video of your app in action

### Start with a strong product description

Think of your detail page as a place to create a narrative about your app. A great product description can help to put your app in the best light and convert users. Make sure your description is clear, honest, and straightforward. Introduce your app clearly and succinctly, while describing your app's most notable features. Explain why and how your app benefits users. Make sure not to make any grandiose promises about your app, as customers can see through inflated claims—you don't want to lose credibility. End your description with a strong call to action (CTA), such as "Get started now," or "Download now and play today!"



### Review your copy

Don't forget to edit your product description before going live. It's a good idea to get a fresh set of eyes—especially if you've been looking at your copy for a while! Reach out to friends and ask for feedback to make sure what you wrote is clear and error-free.

### Select keywords

A crucial step in your app's marketing strategy is identifying the right keywords. Choosing these keywords and an appropriate name for your app helps drive discovery. Pick keywords that are uniquely descriptive of your app rather than vague terms like "Fun," "Free," and "Game." Be as accurate as possible.

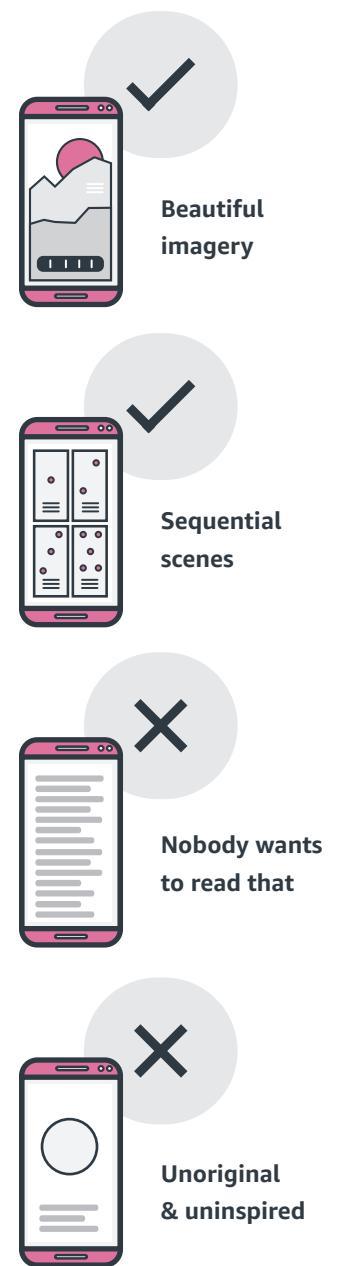
To choose your keywords, jot down things that are meaningful tie-ins to your app from your target audience's perspective. Go to the blogs and newsgroups that your target audience reads and look for frequent terms and descriptors.

Keep tweaking and experimenting! Keyword efficacy is always changing, so make keyword research a regular exercise.

### Choosing screenshots

Customers respond well to great visuals. Screenshots and videos help customers understand your app when they're making a purchase or download decision.

Screenshots show customers how your app looks and feels. These screenshots should show how your app stands out from others in its category. For games, avoid screenshots of splash screens. Instead, include images that demonstrate the action and fun in your game. Take screenshots of different levels or features, displaying them in a logical order to complement the story you tell in your description. A splash screen with stylized art creates a mood and gets users' attention. Many developers over-index on highly designed splash screens and don't focus enough on in-game action. If you must choose, in-game screenshots should always win out.



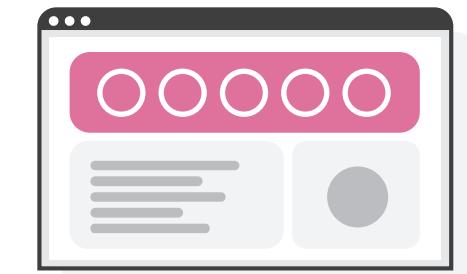
## Investing in video

If your app or game has an exciting in-app experience or gameplay, a video can be an effective promotion tool. Including a gameplay video on your app detail page gives your customers a way to see what it's like to play the game. Capture video directly from your device and create a video that shows basic gameplay while emphasizing what makes your game unique. Show the most stunningly beautiful, captivating, or dramatic part of your game. In post-processing, add a voiceover or text overlays that explain what's happening. Place this video first in the order of screenshots on your app detail page.



## Promotional images

Promotional artwork captures the attention of customers. When you create colorful imagery that reflects the essence of your apps, Amazon can then use this artwork if your app is featured in the Amazon Appstore or in advertisements.



## Advertising

Publishing in the Amazon Appstore gives you access to a self-managed app advertising solution, available exclusively to Amazon mobile app developers. This advertising solution is simple, fast, and effective. By promoting your app on Fire tablet wake screens and through mobile placements on the Amazon Mobile Ad Network, you can reach millions of users.

You can create an advertising campaign in less than 90 seconds. Start by selecting your app, then preview auto-generated ads. Set your campaign details (like budget, bids, and duration). Finally, select your payment option and submit your campaign. It's that easy!

## Promotions

With the Amazon Developer Promotions Console, you can create temporary discounts for your apps and in-app items. Want to run a 24-hour special holiday sale? Or celebrate the launch of a new game by discounting all your mobile apps? It's simple to draw extra attention to your apps through temporary discounts, where you control the pricing options. You can set a percentage-based price drop across all in-app items, or manually adjust the price of your items in the marketplace. Price drops provide incentive for your customers to make in-app purchases, increasing your in-app sales. And through the console, you can view detailed reports that let you see your ROI for a specific price-drop campaign, allowing you to continue to monitor and improve future promotions.

**"Advertise Your App is the most cost-effective, scalable solution to acquire the users I care most about"**

—Kevin Tydlacka,  
Kevin Tydlacka Apps



# NINE

# Engaging and retaining users

---

Now that your app is out there, you want your users to return to it frequently. You can encourage this by continuing to improve their experience. This includes both listening to your users' feedback and learning about how they use your app. Engaging further with your audience gives you important information about your app and provides valuable insights into how to reach new audiences.

This chapter takes you through strategies and solutions for user experience and retention, optimizing your app's visibility, and building your fan base—and better relationships with your users.



### Preflight checklist



- Integrate Amazon Moments to reward customers
- Add leaderboards to your games with Amazon GameOn
- Identify and enact a user strategy to develop your fan base
- Create a content roadmap to plan for ongoing updates to your app

### Challenges you might face



- Building a fan base from the ground up
- Identifying strategies to continually engage users
- Committing to a long-term strategy of updating/refreshing your app and engaging with users
- Identifying your user types and targeting your power users

## User engagement and retention

By planning, designing, and building your app with your users front and center in your mind, you can create an app that will keep your users returning again and again.

In general, people use apps in one of two ways: to gather information or to escape. Information gathering apps deliver news, weather information, social media content, sports updates, or other valuable information. These apps can naturally draw users back. Apps that offer entertainment or distraction allow their users an opportunity to escape—for varying amounts of time. Sometimes a user will want to sit and play a game for an hour, while other times the user might need a 2-minute distraction while waiting in line. Aim

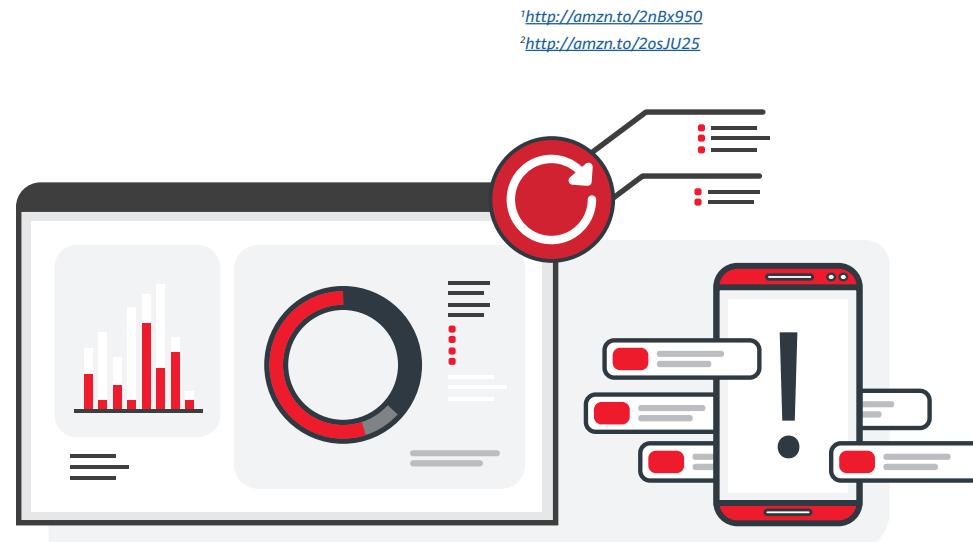
to make an app that can be enjoyable for an extended period, but also for shorter 2 to 5 minute bursts.

Amazon APIs are designed to help you improve player experience and retention. In games, you can easily add leaderboards for cross-platform competition. You can also use Amazon Device Messaging<sup>1</sup> to send push notifications to your customers, or use Login with Amazon<sup>2</sup> to reduce registration friction by making it easy for your customers to sign in. With these tools, it's easier for your users to continue to enjoy a seamless experience across their devices and eagerly engage with your app.

## Updating and refreshing content

Regularly updating your app not only improves your placement in the app store, it also signals to your users that you're engaged as a developer and dedicated to continuing to improve and update the app. A good rule of thumb is to plan to update your app 1 to 4 times per month, depending on what seems most appropriate for your product. Plan your updates in advance and track your ongoing content strategy. Create a roadmap that will outline your content strategy, planning around opportunities to update content, continue the app's story, highlight new features, and address user feedback.

Try to plan ahead by about 2 to 4 updates. This way your updates will be ready to go out at your desired cadence. Be sure to include the release notes with each update. Like the product description, this is an important opportunity to reach users and highlight what makes your app stand out, as well as a chance to plug any new features.



<sup>1</sup><http://amzn.to/2nBx950>

<sup>2</sup><http://amzn.to/2osJU25>

## Using push notifications

Push notifications are alert-style messages that appear on the home screen of a user's device—regardless of whether they have your app open. According to Localytics, push notifications can boost app engagement by 88 percent. Push notifications provide an opportunity to connect and re-engage users with your app. Users can consent to receive push notifications during the app installation process. When setting up your app, provide a solid value proposition to convince new users to opt-in.

While you could send a one-size-fits-all message, the best notifications are tailored to each end user. As a developer, you have access to a wealth of data to help you personalize your messages, such as which new users haven't opened the app since their first day, or which returning users have reached a certain number of sessions or engaged with certain assets. Executed properly, a push notification strategy can increase engagement and revenue in your app.

## Optimizing your placement in the Amazon Appstore

Your app listing plays a crucial role in the success of your app. A detailed, inviting description with well-chosen, relevant keywords is the most important thing you can do to optimize your Amazon Appstore listing. Keywords can also help users find your app through searches on Amazon.com, opening up the audience for your listing even further.

Amazon highlights apps through storefronts like the Amazon Appstore and Amazon.com, but also through e-mail, Facebook, and Twitter. Featured placements could include "Highly Rated Apps and Games" or "Featured New Releases."

## Responding to feedback

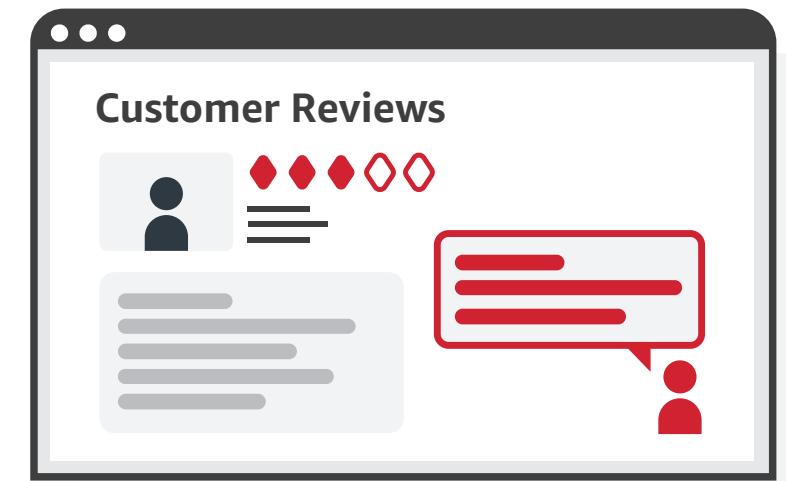
Customers take note when developers are actively engaged with responding to customer questions and comments. Showing that you care about the customer influences whether a customer will choose your app and recommend it to others.

Make a goal to respond to all reviews, both negative and positive—even if it's just to say, "Thanks." This lets your customers know that you're paying attention to their feedback. The best approach for negative reviews is to respond with what you've done to fix legitimate complaints, or with a short explanation.

For example, if a customer writes to complain about having to pay for items in your app, you can say: "Thanks for your note. We looked at several different ways to make this game a financially sustainable project, and we decided to go with In-App Purchasing. We continue to explore new models as they are developed and hope to thrill you with wonderful gameplay in the future."

Within your app, ensure it's easy for users to contact you. There are tools available, like Helpdesk or Help Scout, which can make it easy to set up a way to hear directly from users and respond quickly.

A detailed, inviting description with well-chosen, relevant keywords is the most important thing you can do to optimize your Amazon Appstore listing



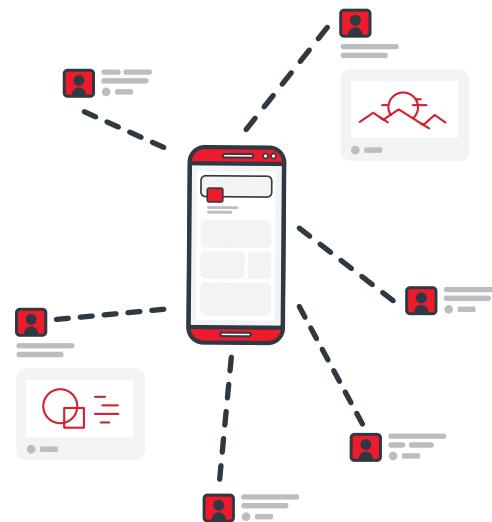
## Promoting your app through social media

Social media sites, including Facebook, Twitter, Instagram, Snapchat, and even LinkedIn can provide effective opportunities to connect with fans and reach new users.

Create a presence for your app on Facebook to share videos, artwork, and screen grabs and also to deliver news about updates, exciting new features, and your upcoming projects. Plus, on Facebook, your users can interact directly with you and share information about your app with their friends.

Set up a page on Twitter as well. Twitter can be a great tool to share your latest activities and app updates. Twitter has also become a common tool for users to reach out with their customer service queries. As with all feedback, be sure to respond to these questions as soon as you can.

Social media sites also have internal features that can be used to advertise and promote your page to social media users outside of your followers, including sponsored posts and paid advertising. Visit the Facebook and Twitter pages to learn more about how these work for you.



## Driving engagement with digital and physical items

Amazon Moments is a cross-platform marketing tool that lets you easily reward customers in more than 100 countries when they perform high-value actions in your apps and websites. With Moments, you can track actions in real time, create targeted campaigns, and reward customers with digital and physical products that are delivered to their doorsteps.

## Building cross-platform competitions

GameOn is a cross-platform, competitive gaming service for developers. GameOn is a set of flexible APIs that allow you to easily build cross-platform competitions into your games, allowing your players to compete for real-world prizes fulfilled by Amazon or other

For example, a video streaming service can encourage customers to watch the first five episodes of a new show by rewarding them with an action figure of their favorite character or a pair of headphones. Notable customers using Moments include: Washington Post, TikTok, Sony Crackle, Sesame Workshop, Bravo, USA TODAY, Sago Mini, and Bell Canada.

in-game rewards. GameOn is built on AWS's cloud infrastructure and works on any operating system, enabling you to scale quickly and invest more time in what you do best—designing great games.

**Tailor rewards delivered by Amazon when customers reach actions that matter in your apps and websites**

As your players develop into a fan base, it helps to understand the types of fans these players become. This allows you to focus on the areas of your app that are most important to them. Two important categories of your fan base are influencers and content creators.

Influencers are the small group of early adopters who can create or contribute to a change in opinion or behavior of a larger audience. Within these influencers are advocates, players that influence peer groups. You can build advocate fans through engagement opportunities—like a chance to chat with the developers, share fan art or photos, or participate in local events or other events and opportunities to show their loyalty. Professionals are different from advocates because they are influencers with a wide reach. These include YouTube stars, celebrities, or other professionals with a lot of fans of their own. Target professionals who you believe will be genuinely interested in your app.

When you contact them, focus on why the app would be a good fit for them and why their fans and readers would care about it. Help them to reach their audiences by providing ways for them to express their excitement about your product. Include a strong description, videos that show the best highlights of your game, and headline-suited screenshots. And don't forget to include a link to download the game.

In addition to influencers in your app's fan base, you'll have content creators. These are fans eager to extend the in-game experience through their own content creation. You can encourage these fans by making it easy for them to share and promote their content within the app through badges, boards, or other in-app opportunities.

---

**A fan base drives revenue for your app through evangelism, player recruitment, and subscription revenue. Outside of the game, this evangelism can also result in product sales. By providing products for sale, you can support your loyal fans and help promote your app at the same time.**

---

## Increasing user engagement in games through Twitch

Another way to grow your audience is through Twitch. This platform helps you engage with users, develop a fan base, and spread the word about your game by giving potential users a chance to preview it. While new users get to see exactly what it would be like to play your game, existing users can connect with each other to discuss their favorite aspects of your game and share tips and tricks.

Through Twitch, you can respond to feedback and talk directly with your customers about what they're looking for. Plus, you can take advantage of opportunities to cultivate your community, like giving out early access to new games before they launch, soliciting feedback from broadcasters, and creating original content through your Twitch channel.

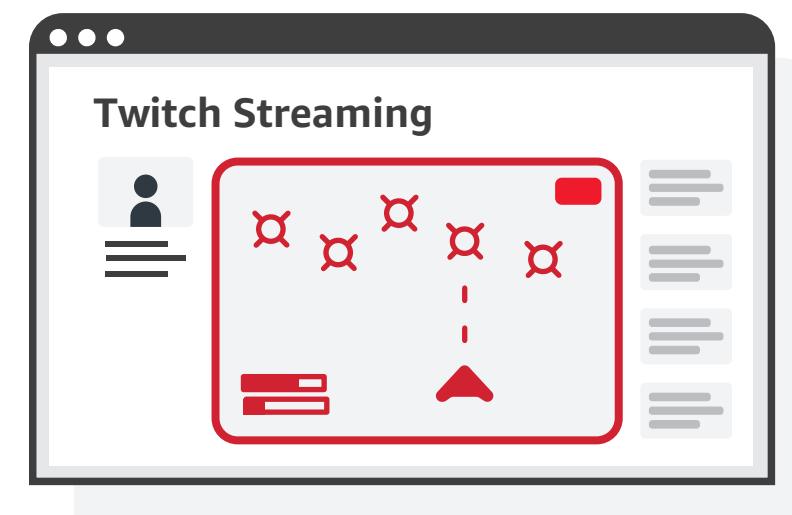
**Note:** While creating and maintaining a Twitch channel is an important step for your game, you'll likely experience a slow growth curve that will require considerable content creation and dedication. Be prepared to spend time engaging with users, creating interesting content, and keeping your Twitch stream updated regularly.

Twitch streamers with an established following, or other gaming professionals in the media, might also be interested in helping you promote your game. There are a few best practices to keep in mind when you're reaching out to Twitch streamers or any other member of the media. Most importantly, ensure that you're contacting the right professional for your game—someone with a demonstrated interest in what you've created. Make it as easy as possible for the professional to understand what your app is and download the app for themselves. If you're hoping they'll publish a post about your app, take care to include all the assets they would need—like screenshots, download information, and links to videos of your game in action. The open-source presskit() (pronounced "do presskit")<sup>3</sup> has become the industry standard in promoting games. This means it has become what people are expecting to see, which makes it an easy choice in sending accessible, relevant information. (See presskit() to learn more.)

<sup>3</sup><http://bit.ly/2osJQ2d>

## Top tips to reach professionals

- Show them why their fans or readers will care.
- Include video that shows the most amazing parts of your game.
- Include headline-suitable screenshots.
- Include a link to download the game. (Research first to ensure that they can play the game on their preferred device.)



twitch



# Measuring and optimizing your app

---

Your app needs analytics. It's tempting to assume you know what your users are trying to do, and how they will (or won't) use your app. As the architect of your solution, it's easy to project your own expectations into your app. But this overlooks the user's actual experience.

Users will surprise you. They'll find alternative solutions or exhibit unanticipated behavior when using your app. Some users will provide feedback via reviews or commentary, but others will remain silent. Without the telemetry to track usage, you

run the risk of missing important insights that can help with addressing your customer's needs. This can lead to lower retention and loss of customers.

In this chapter, you'll learn how to instrument your app to take advantage of Mobile Analytics (now included in the Amazon Pinpoint service), understand the data you collect, and build insights that you can use in future updates for your app.

## Preflight checklist

---



- Make sure you have an Amazon AWS account
- Sign up for Amazon Pinpoint

## Challenges you might face

---



- Understanding the value of collecting user data
- Deciding what data to collect
- Instrumenting your app for data collection
- Learning how to draw insight from your data
- Viewing your analytic data/reports



By using Amazon Mobile Analytics, you can track customer behaviors, aggregate metrics, generate data visualizations, and identify meaningful patterns. This gives you deep insights that can improve your KPIs. Mobile Analytics can automatically copy your app event data to Amazon S3 and Amazon Redshift and analyze app usage patterns with SQL queries or other tools.

Mobile Analytics serves as the collection point for data transmitted from user sessions with the Insights SDK. As telemetry data comes in, the analytics platform builds insights and presents reports on key indicators such as new and returning users, user retention, app revenue, and user behaviors. Data visualizations show performance on your key indicators, and ongoing trends.



Mobile Analytics is now also part of your Amazon Pinpoint service. Pinpoint builds upon the capabilities of Mobile Analytics to help you drive targeted content to your users with greater accuracy.

If you're looking for a way to beta test your app among hundreds of other developers, it's a good idea to use Amazon's Live App Testing

---

## Understanding your data

When you add an app to Mobile Analytics, several metrics are automatically collected from your app. These metrics can be viewed from the main default dashboards in the Mobile Analytics console: overview, active users, sessions, revenue, and retention. If you have custom app events, they will be available on the Custom Events dashboard. You will also have the option of creating custom dashboards.



The default dashboards are built from the measures that come with Mobile Analytics out of the box:

- Active users (daily and monthly)
- Sessions (instances in which your app was opened)
- Revenue (from in-app purchases)
- Retention percentages (daily and monthly)

It's important to note that three of these dashboards actually come from one tracked activity: when your app is opened. When this is tied to anonymized user identity and timestamp information, a great deal of useful information becomes available.

You can gain deeper insights by creating custom events. For example, the default dashboard might show a dip in your app's retention after two weeks, but without additional data, you're left guessing about the cause of abandonment. With a custom event that tracks user progress, you might compare average progress against session count. In this way, you might find that users are experiencing a plateau of progression despite regular participation, leading to early abandonment.

In a more advanced scenario, you can also use custom events to study usability. Logging which controls are used most frequently, or whether users seem to have trouble finding a specific control, can provide insight into how your users use your app. This can be useful during soft launches (early versions of your app) or when adding features, to identify the best layout for your interface.

## Building insights

As pointed out earlier, three of the four default dashboards come from a single activity, one that only holds two data points: a unique user identifier and a timestamp of when that user opened your app. The different dashboards build upon that raw data by framing a question—a need for information—and grouping and aggregating that data to provide an answer for that question. Here are some examples from the default dashboards:

### Active users (daily and monthly)

The question being posed on this dashboard is "How many unique users opened my app during a specific timeframe?" This is basically a count of unique user IDs per day and per month.

### Sessions

The question posed here is "How many times was my app opened?" In the context of these reports, a session is an instance of your app being opened. A second chart shows the average number of times an active user opens your app in a day.

### Retention percentages

The question here is "How many users are continuing to use my app?" Reaching an answer to this question is a bit more complex and requires establishing time elapsed between distinct sessions. But this metric is still built upon two data fields and comparing multiple instances over time.

When it comes to extending these existing capabilities with your own data, there are nearly limitless options. It's important to keep in mind that a few data points—or even just one—can go a long way. Consider this example: **Chapter Four: Building your app** introduced the concept of graceful degradation. The idea is to handle certain errors with a simple notification alerting the user to the condition rather than letting the app crash for no apparent reason. Adding a custom event that logs when these error conditions occur lets you generate reports on error incidents. With a bit of metadata, such as which function or feature generated the error, you can generate reports on errors happening with your app even before your users start to log issues.

With the addition of a few well-planned API calls to Mobile Analytics, you can begin to generate critical data on the health and use of your app. This can easily be used to head off potential issues with your app or provide insights into which features you might want to prioritize in your next update.

For more information, see **Amazon Mobile Analytics**.<sup>1</sup>

<sup>1</sup><http://amzn.to/2osNzNh>

# Conclusion

Congratulations! You're now equipped with everything you need to successfully launch your app in the Amazon Appstore. From publishing your app in the Amazon Appstore to sharing the news of everything your app does; from cultivating a community of dedicated users to designing T-shirts and updating your Twitch page, you're finally beginning to enjoy the rewards of your hard work.

## Next steps

- Plan ongoing content updates to keep users engaged.
- Continue to optimize your app's placement in the Amazon Appstore.
- Continue your marketing and promotion.
- Continue to measure and improve your KPIs.
- Build more apps!

Maintaining your app with bug fixes, upgrades, and improvements is extremely important to the continued success of your app. As the capabilities and features of mobile platforms continue to evolve, you'll need to work to ensure that your app is keeping up—and that your users can get the most out of the new features.

Keeping your app up-to-date will require dedication and planning, but keep in mind that regularly maintaining your app will be easier and more cost-effective than allowing the app to sit and gather a backlog of new feature requests that need to be addressed through major changes, or even a complete rewrite of your app. In addition to keeping up with changing platforms and adding new features, stay engaged with users and address their feedback by adding the features they request—

and expanding on existing features that are seeing the most use. This won't just make your app more beneficial and appealing to your established users—you'll also be delivering a better experience for new users.

As your user base grows, so will your opportunities to market and promote your app and to continue to grow your fan base. Through continued engagement, responding to feedback, and cultivating a community, you can contribute to the ongoing growth and success of your app.

Be sure to follow the Amazon Developer blog to be notified of the latest platform changes, stay up-to-date on new devices available from Amazon, and connect with other developers.<sup>1</sup> You'll see what's working best for them, getting ideas and inspiration from their experiences.

<sup>1</sup><http://amzn.to/2o7n7fH>

**Made by Amazon**

# Getting started in the Amazon Appstore

---



Create and test your app, publish in the Amazon Appstore, build your audience, and retain users. This book will guide you through the entire process and share what we at Amazon have learned—best practices, common challenges, and insider advice—from our experiences in the Amazon Appstore. Whether you're an experienced developer or new to the business and ready to build your first app, this guide will prepare you to publish your app in the Amazon Appstore and make it available across millions of Fire tablets, Amazon Fire TVs, and Android devices in 236 countries and territories.