

How to Pass the Apple App Store Review Process

Introduction

Apple maintains a strict app review process to ensure that every app on the App Store meets high standards of quality, safety, and compliance. Before any iOS app can be published, it must pass Apple's review, which checks for adherence to the **App Store Review Guidelines** organized into five main categories: **Safety, Performance, Business, Design, and Legal** ¹. Apple's review team will examine your app's content, functionality, user experience, and compliance with policies to decide if it can be approved for the App Store. This document provides a comprehensive guide for developers to navigate these requirements and **avoid common pitfalls that cause app rejections**, ensuring your app sails through the review process.

Why this matters: A rejected app can delay your launch and cost valuable time. By understanding what Apple looks for and preparing your app accordingly, you can save yourself from the "rejection loop" and get your app approved on the first try. Use the following best practices and checklists to **pass the standard Apple App Store review** and confidently publish your iOS app.

Pre-Submission Checklist: Before You Submit

Before you even hit "Submit for Review," go through this checklist to catch issues that might trigger an immediate rejection. Apple itself emphasizes these common missteps that can slow down or derail the review ² ³:

- **Test for Crashes and Bugs:** Rigorously test your app on real devices and simulators to ensure **no crashes, freezes, or major bugs** occur ³. An app that crashes during Apple's review will be rejected outright. Use test cases to cover core features, edge cases, and low network conditions (Apple reviewers often test on slow or flaky networks ⁴). Fix all crashes and significant bugs before submission.
- **Complete All Metadata and Info:** Make sure your App Store listing is **fully filled out with accurate information** ⁵. This includes the app name, description, category, keywords, screenshots, and **privacy policy URL** (if required). Inaccurate or misleading metadata (e.g., claiming features your app doesn't have) can lead to rejection under guideline 2.3 for "accurate metadata" ⁶. Double-check that your listed **pricing and in-app purchase details match exactly what's in the app** to avoid being flagged for inconsistency ⁷ ⁸.
- **Update Contact Info:** Ensure the **developer contact information** in App Store Connect is current ⁹. If Apple's review team needs to reach out for clarification (for example, to request a demo account login or ask a question), having up-to-date contact info can speed up the review. Missing emails from Apple due to old contact info could stall your review.

- **Provide Demo Account or Access (if needed):** If your app requires login or special credentials, **provide a demo account** or enable a full-featured demo mode for the reviewers ¹⁰. Include these credentials in the “App Review Notes” field when submitting. For apps with region-locked content or requiring specific hardware (like a Bluetooth accessory or a QR code), supply sample data (e.g., a QR code image) or loan the hardware to Apple if necessary. **Make it effortless for the reviewer to access all app features.** A common fast-rejection reason is when reviewers cannot fully use the app because no test login or access was provided.
- **Backend Services Up and Running:** Keep any servers or backend services **online and testable during the review** ¹¹. If your app fetches data from an API or server, that service must be accessible to Apple. A reviewer who encounters broken content or network errors may reject the app for non-functioning features ¹². Test your app with a “fresh install” on a clean device to simulate a first-time user hitting your backend ¹³.
- **Review Notes for Special Features:** Use the App Review Notes field to **explain any non-obvious functionality**, special configurations, or use of specific APIs ¹⁴. For example, if your app has a hidden gesture to access a beta feature, or if it uses background location in a unique way, describe it to the reviewer. If your app uses In-App Purchases heavily or has complex login flows, you can outline steps for the reviewer in the notes. This **preemptively answers questions** and can prevent misunderstandings.
- **Ensure Compliance with Apple Guidelines and Documentation:** Cross-check your app against Apple’s official documentation and guidelines for design and development. Verify you’re following the **Human Interface Guidelines** for a good UI/UX and any relevant technical guidelines (for example, if you use Apple Pay, HealthKit, etc., ensure you meet those frameworks’ rules) ¹⁵ ¹⁶. Pay special attention to any **entitlement** your app uses (e.g., Push Notifications, Sign in with Apple, etc.) – these must be configured correctly and used only for allowed purposes.

By confirming all the above, you tackle the “low-hanging fruit” issues first. As one expert puts it, *“walk through the app like a reviewer: install fresh, go through main flows, test purchase restore, find the privacy policy, and if applicable, delete an account”* ¹⁷. This reviewer’s-eye view helps catch anything obvious that could prompt an instant rejection.

Adhering to App Store Review Guidelines: Key Focus Areas

Apple’s App Store Review Guidelines are extensive, but below we break down the **key requirements in each category (Safety, Performance, Business, Design, Legal)** that every developer should heed. Ensuring your app meets these will significantly increase your chances of **passing review on the first try**.

1. Safety: User Safety, Privacy, and Appropriate Content

User safety and appropriate content are top priorities for Apple. Guidelines in the Safety category cover what content your app can (or cannot) include, how you handle user data, and ensuring your app doesn't put users or their devices at risk ¹⁸ ¹. Key points to consider:

- **Objectionable Content:** Apps must **not contain offensive, disturbing, or harmful content** ¹⁹. This means no pornography, graphic violence, hate speech, illegal drug references, or defamation. Even user-generated content in your app needs monitoring (more on that below). If your app's content could be considered crude or "over the line," expect a rejection under guideline 1.1. Keep the content appropriate for your target age rating. Apple's review can be subjective here – if in doubt, err on the side of caution with content and provide content filters or warnings as needed.
- **User-Generated Content (UGC) Moderation:** If your app allows users to post content (comments, photos, videos, profiles, etc.), **you must include robust moderation features** ²⁰. Apple explicitly requires apps with UGC or social features to have:
 - A way to **filter out objectionable material** (e.g. profanity filters, image moderation for nudity).
 - A mechanism for users to **report** abusive or inappropriate content, and a plan to respond to such reports promptly.
 - The ability for users to **block other users** who harass or offend them.
 - **Contact information** in-app so users can reach the developer regarding objectionable content or issues ²¹.

Failing to implement these will lead to rejection under guideline 1.2. Apple checks UGC-heavy apps carefully; they often **test posting content** or look for a report button ²². So, add a "Report" option on user posts and ensure it's visible. Even if your community is small, you need basic moderation tools from day one ²³ ²⁴. Also, if your app is primarily user content and it becomes overrun with egregious material (e.g. an app that devolves into harassment or adult content), Apple may remove it later ²⁵. Plan to keep your platform clean and safe.

- **Data Privacy and Permissions:** Apple is extremely strict about protecting user privacy. **Collect only the data you truly need, and declare all uses of it.** If your app accesses sensitive data (location, contacts, health data, camera, microphone, etc.), you must include a usage description in your app's `Info.plist` explaining why the app needs this access. The reviewer will check that requesting permissions makes sense in context. If they find your app **accesses user data without a clear reason**, it's a red flag ²⁶. For example, asking for location or health data at app launch without explanation can lead to rejection ⁸. In the App Review Notes, explicitly mention any sensitive data usage and how it benefits the user ²⁷. Furthermore, as of late 2020, you must fill out the **App Privacy "nutrition label"** in App Store Connect, detailing what data you collect and how it's used. Ensure this is truthful and matches your app's behavior – Apple can reject your app if the information is found to be inaccurate or if your app collects data you didn't disclose.
- **Privacy Policy Requirement:** Every app that collects personal data **must have a privacy policy**, and Apple strongly encourages all apps to provide one ²⁸. You need to **link a privacy policy URL in App Store Connect metadata**, and it should also be **accessible within the app's UI** (for example, in a Settings or About screen) ²⁹ ³⁰. The privacy policy must clearly explain what data you collect, how

it's used, and how users can contact you or request deletion ³¹ ³² . If a privacy policy is missing or hidden, Apple will reject the app for non-compliance with guideline 5.1.1 (data privacy). During review, a tester will try to find your in-app privacy policy – if they can't find it easily, it's treated as if you don't have one ³³ ³⁴ . Put a prominent "Privacy Policy" item in your app's menu or settings.

- **Secure and Safe Experience:** Apps must not harm users or devices. This includes obvious things like no malware, viruses, or spyware. Less obvious: if your app involves potentially risky content (e.g. health-related advice, or user meetups), ensure you include appropriate **disclaimers** and **safety measures**. For example, apps giving medical or health guidance should mention they are not a substitute for professional care. Apps that encourage physical activity should be mindful of user safety (provide warnings for intensive exercise, etc.). If your app uses encryption or security technologies, you must follow export laws and Apple's rules (declare encryption use when uploading the app). **Tip:** Apple will also scan your app for known vulnerabilities or malware automatically in an initial review stage ³⁵ , so keep your libraries up to date and don't include suspicious code.
- **Kids Category and Age Ratings:** If you specifically target children (using the "Kids" category) or if children are a significant part of your audience, extra rules apply. **Apps for kids must be extremely cautious with content and data.** No inappropriate content obviously, but also **no behavioral advertising or unnecessary data collection on kids** ³⁶ . If your app is in the Kids category, you cannot include external links out of the app or purchase opportunities without gating them behind parental consent (e.g. a parental gate challenge) ³⁶ . Ensure your age rating in App Store Connect is accurate; if you have mild profanity or cartoon violence, reflect that in the rating questionnaire. A mismatch (say, rating yourself 4+ but the app has teen content) can cause rejection or later correction by Apple.

In summary, **make user safety and trust a core part of your app**. Filter and moderate content, respect privacy, and be transparent. Apple wants to see that your app will not put users (especially kids) at risk in any way – whether emotionally, physically, or regarding their personal data.

2. Performance: Fully Functional and Efficient Apps

Performance guidelines ensure your app is technically solid and provides a good user experience without glitches. An app that fails to perform as expected won't make it through review. Key aspects of performance Apple will evaluate:

- **Complete, Final Version – No Placeholder Content: Submit a complete app, not a beta or half-built product.** Apple will reject apps that have obvious placeholders like "Coming soon" buttons, dummy content, or sections of the app that are not functional ³⁷ . All primary features that you advertise must be implemented and working. If you have features that aren't ready, **remove or hide them** until a future update. Don't leave dead-end menu items or lorem ipsum text visible. The review team will poke around; any unfinished sections can lead to rejection under guideline 2.1 (App Completeness) ³⁷ .
- **No Crashes or Major Bugs:** This is non-negotiable: **if your app crashes during review, it will be rejected** ³⁸ . Test extensively on all target devices and iOS versions that you support. Pay attention to memory usage and performance to avoid sudden crashes. Also ensure there are no obvious bugs that break user flows (e.g., buttons that do nothing, views that fail to load). Minor UI issues might

pass, but anything that **significantly impairs functionality will cause a rejection** ³⁸. It's wise to do a test on a fresh device with no user data, as Apple will – they perform a *fresh install test* and will catch crashes that occur on first launch or first-run flows ¹³.

- **Optimize Launch Time and Loading:** If your app takes too long to load or fetch initial content, it might be flagged. Apple doesn't set an official timeout, but generally if your app is unresponsive for more than a few seconds on launch, it's a bad sign. Make sure any initial network calls or setup processes are efficient. If content is loading, use a spinner or loading state so the reviewer knows the app isn't dead. **Broken content** (like screens that never populate data due to server issues or slow calls) is treated like a bug ³⁹. Test on slow network conditions (3G speeds or poor Wi-Fi) to ensure the app remains functional and doesn't appear broken to someone with a suboptimal connection ¹³.

- **Compatibility with Devices and iOS Versions:** Your app must run properly on all devices and OS versions you claim to support. This means handling different screen sizes (from small iPhones to iPad screens if universal). **Layout issues** that make the app unusable on certain devices can be a cause for rejection under performance/design guidelines. Also, by Apple's current rules, new apps must be built with the latest Xcode and iOS SDK, and support the latest device displays (e.g., no letterboxing on iPhone 14/15 screens). Make sure your UI adapts to notches, safe areas, and so on. If you choose to exclude iPad support, ensure you've set the `UIRequiredDeviceCapabilities` appropriately to iPhone only; otherwise, Apple might test on iPad and find a subpar experience.

- **Use of Approved APIs and Features:** Apple forbids the use of **private APIs** or any undocumented system calls. Your app should only use public, Apple-approved frameworks. The review process includes automated checks for private API usage, which will result in rejection under guideline 2.5.1 if found. Similarly, do not download or execute new code in your app after release (no hidden code or plugin architectures that aren't allowed) – dynamic code via mechanisms like JavaScript running in WebViews is generally okay, but you cannot, say, load a new binary or use reflection to access private methods. Ensure all your third-party SDKs are compliant as well (no hidden malware or private API calls – you are responsible for third-party code ⁴⁰).

- **Efficient and Resource-Friendly:** While not always a direct cause for rejection unless extreme, your app should not excessively drain battery, misuse background permissions, or hog memory. For instance, if you use background location, Apple will scrutinize if your app truly needs it or if it's just running in the background for no good reason (which could lead to rejection for not respecting user resources). Similarly, apps that heat up the device or cause it to hang could get flagged. Test things like location, Bluetooth, push notifications, etc., to ensure they are used sparingly and responsibly.

- **Comply with Specific Technical Guidelines:** Some types of apps have additional rules. For example, **apps that stream music or video** should use the MediaPlayer/AVFoundation frameworks properly and not violate content rules; **apps that use VOIP** must use Apple's PushKit for call notifications, etc. **Games** must properly manage Game Center if used and not gamble if not permitted. If your app uses background modes (like background audio, location, fetch), justify them in your review notes and make sure they are essential – Apple might reject an app that, say, keeps GPS on constantly without a very clear user benefit. Always ask: *Is this feature implemented in the most Apple-compliant way?*

In short, **polish your app's performance and stability** before submission. One App Store expert advises to *"test your app like you're Apple's QA team trying to break it"* ⁴¹. Apple's reviewers **will try to break your app** – by tapping everywhere, leaving and returning, trying it offline, etc. If your app stands up to this, you're in good shape for approval.

3. Business: Monetization and Payment Rules

Apple's business guidelines cover how you monetize your app and ensure that Apple's in-app purchase (IAP) system is used wherever required. Violating these is a **fast track to rejection**, as Apple is very strict about payment rules and fair use of their ecosystem. Key business-related rules:

- **In-App Purchases for Digital Content/Features:** If your app sells **digital goods, content, subscriptions, or features**, you **must use Apple's in-app purchase system** for those transactions ⁴². This includes things like premium features unlock, virtual currency, game levels, e-books, etc. You cannot direct users to an external payment system or website to purchase these items – doing so will violate guideline 3.1.1 and get your app rejected ⁴². For example, an e-book app cannot have a button that sends users to a web store to buy books; it should use IAP. Apple wants their 30% cut and a consistent user experience for purchases. **Exceptions:** If your app only sells physical goods or services delivered outside the app (e.g. ordering food, ride-hailing, e-commerce for real items), you can use other payment methods. Also, so-called "reader apps" (apps primarily for consuming content you bought elsewhere, like Netflix or Kindle) can allow account login for content access without using IAP, but even those apps cannot *link* to external purchase mechanisms. Be very cautious here – if there's any doubt, use IAP for digital content.
- **Restore Purchases:** If you have IAP (especially non-consumables or subscriptions), you **must implement "Restore Purchases"** functionality ⁴³ ⁴⁴. This lets users who reinstalled the app or got a new device regain their past purchases. Apple's reviewers will test this. A common check is: they buy your IAP, then delete and reinstall the app (or use a second device with the same Apple ID), and see if using "Restore" properly unlocks the content without a repurchase. If restore is missing or broken, expect a rejection under 3.1.1. Place the "Restore Purchases" button in an obvious spot (often on the purchase screen or in Settings) so the reviewer can find it ⁴⁵. Also ensure it actually works (test with a sandbox account if possible before submission).
- **Subscriptions and Auto-Renewables:** If your monetization involves subscriptions, make sure you follow Apple's rules for subscription transparency. **Clearly disclose pricing, duration, and terms** of the subscription **within the app**. Apple has cracked down on confusing subscription screens – the user should easily see the cost and what they get, and understand the subscription will auto-renew (if it does) ⁴⁶. Avoid any dark patterns (like tiny fine print or misleading buttons). Also, provide a way in-app (or via system settings) for users to manage/cancel subscriptions; typically iOS handles the cancellation in the App Store app, but your app should not obscure how to cancel. If offering a free trial, clearly state the trial length and what will be charged afterward. Many apps get rejected for vague subscription offers, so design that flow with clarity and align with Apple's subscription guidelines.
- **Prohibited Monetization Tactics:** Certain ways of making money are not allowed. For instance, **no in-app mechanisms for donating money to others** unless using Apple Pay or IAP (guideline says you can't use IAP for donations either, except if it's a gift to the developer with no benefits). **No**

lotteries or raffles without proper permits and pre-approval. **Crypto-currency mining** in apps is disallowed (unless it's cloud-based mining, but generally don't do it as it violates resource usage). If your app involves cryptocurrency in any way, ensure it complies with guideline 3.1.5 (like it can facilitate storage, but not mining on device, and exchanges must be legal, etc.). **No multi-level marketing (MLM) schemes** or apps that appear fraudulent or pyramid-scheme-like – Apple will reject those under spam or fraud guidelines.

- **Ads and Advertising:** Ads are allowed, but if your app is primarily an ad aggregator (like it shows more ads than content), it may be rejected for poor experience. Ensure that your ad providers are reputable and don't violate user privacy (e.g. use the AdSupport framework properly). If targeting kids, you **cannot serve behavioral ads** (targeted ads) inside kids category apps. Also, using push notifications purely for advertising is not allowed unless users have explicitly opted in (and even then, be careful – Apple's guideline 4.5.4 says push notifications cannot be used for promotions or marketing unless the user consents, and you must provide a way to opt out). So if you plan to monetize via push marketing, think twice or implement an opt-in prompt specifically.
- **Sign in with Apple (If Applicable):** Apple's rules (guideline 4.8) require that if your app allows account creation and supports third-party login (like Google, Facebook, Twitter logins), you must **offer Sign in with Apple** as an equivalent option, with some exceptions. This is a *legal/business* requirement to ensure users have a privacy-friendly choice. If your app uses only your own email/password system, or uses only enterprise SSO, you might be exempt. But if you let users log in with, say, a Google account, you **must include Sign in with Apple** or your app could be rejected until you add it. Check Apple's documentation on this requirement and implement it if needed to avoid rejections for login options.
- **App Store Payments and Metadata:** Be careful about what you say in your app and App Store listing regarding pricing and payment. **Don't include language like "subscribe on our website for a discount" or "cheaper prices available on our site."** Apple forbids steering users to external purchase methods for content that could be in-app. Also, do not use your App Store description or in-app text to complain about Apple's fees or encourage jailbreaking, etc. Such commentary can irritate reviewers or violate guidelines. On the flip side, **ensure the pricing you mention in the app or description is correct.** As noted earlier, even a small mismatch (e.g., description says "\$4.99/month" but actual IAP is \$5.99) can cause rejection ⁷.

In summary, **follow the money rules closely.** Apple's review team quickly rejects apps that try to bypass in-app purchases or confuse users about cost. If you monetize with IAP or subscriptions, integrate them properly and test them thoroughly (including edge cases like purchase restoration). If your app is free with no purchases, you have fewer worries here – just avoid any appearance of charging outside allowed methods. Remember: **any digital content or feature must be bought via Apple's system**, and your app should not even hint at other payment methods (unless you're a legitimate "reader app" under Apple's rules, and even then, no direct purchasing links). Compliance in this area not only avoids rejections but also provides a more consistent experience for users.

4. Design: User Experience, Look-and-Feel, and App Quality

Apple places a high value on apps being well-designed and providing a good user experience. The **Design guidelines** cover everything from the UI/UX to the app's uniqueness and adherence to Apple's ecosystem

standards. While Apple is somewhat subjective here, there are clear principles to follow to avoid design-related rejections:

- **Follow the Human Interface Guidelines (HIG):** While your app doesn't need to win a design award, it should feel at home on iOS. This means using standard controls appropriately, respecting spacing and layout (especially with new screen sizes), and providing an overall polished interface. Apple's reviewers notice when an app is very clunky or non-standard in a bad way. Common issues include text that is cut off on certain screens, buttons that are too small to tap, or using UI elements that don't behave as expected. **Ensure navigation is clear and consistent**, and that the app's visuals aren't jarringly low-quality or unprofessional. If an app looks like an unfinished prototype or a direct web page screenshot, it might be rejected under guideline 4.2 (minimum design quality).
- **No Misleading or Outdated Screenshots/UI:** The visuals you present in your App Store listing **must match the actual app UI**. If you underwent a UI change, update your screenshots accordingly. Apple will compare your app's functionality to the screenshots and description you provided ⁴⁷. Showing features that don't exist or outdated layouts is considered "misleading" and can trigger rejection ⁴⁸ ⁴⁹. For example, don't use concept art or images from a different platform. Always **upload screenshots from the final build** you're submitting. Similarly, within the app, don't label anything inaccurately (like a button says "Free" but it actually triggers a purchase). Consistency builds trust with both Apple and users.
- **Avoid Spam and Duplicative Apps:** Apple will reject apps that are **too similar to existing apps**, or multiple apps that are basically the same. Ensure your app has unique value and isn't just a re-skinned copy of something already on the store (especially if you control those apps too). For instance, if a developer submits 10 flashlight apps with different names, Apple will likely reject most of them under the spam rule (4.3). If you have a portfolio, make sure each app is distinct in content and purpose. Additionally, refrain from stuffing your app with irrelevant keywords or content just to game rankings – Apple checks metadata and will reject apps that abuse keywords or use other app names in their description unfairly ⁵⁰.
- **Sufficient Functionality (Don't Be Too Simple):** Apps that do very little or are basically glorified websites might be rejected for not being "useful" enough. Apple's guideline 4.2.2 notes that apps should have **enough content and functionality to stand on their own**. If your app just wraps a mobile website (WebView) without adding native features or value, that's risky. Try to include some native iOS features (like push notifications, native navigation, etc.) to enhance a web content app. Also, avoid releasing essentially the same app multiple times for different content (e.g., one app per city with minimal changes); consider combining them into one app or using in-app content differentiation. Apple wants to avoid clutter in the App Store, so they will reject low-effort or redundant apps.
- **App Name, Icons, and Branding:** Design extends to your app icon and how you present your brand. Make sure your app's name and icon are not infringing on trademarks (don't use "Instagram" in your app name or a WiFi icon that looks like Apple's own). Avoid using Apple trademarks in your metadata (e.g., don't say "for iPhone X" in the app name). These can cause rejections (guideline 5.2.5 covers trademark misuse). Keep your app name professional and not excessively long. The icon should be a high-quality image – a pixelated or generic icon can signal low quality. While Apple might not reject solely for an ugly icon, a very misrepresentative icon (showing features or content not in the app)

could be an issue. Ensure your launch screens and initial impressions are also up to par, as reviewers do take note of those.

- **User Experience & Engagement:** Apple may consider whether your app provides lasting value. They have been known to reject apps that they feel **don't offer enough meaningful content or utility to users** ⁵¹. For example, an app that just displays a single joke or an app whose primary purpose is self-promotion might not be approved. Make sure your app description and actual experience highlight how the user benefits over time. Apps that appear to be “demo” or trial versions with limited functionality (without indicating that it's a lite version, etc.) can confuse reviewers. If your app is content-driven, ensure there is enough content available at launch, not just a shell awaiting future updates.
- **Push Notifications and Permissions UX:** If your app uses push notifications or requests permissions, handle these in a user-friendly manner. Don't bombard the user (Apple might test by launching the app fresh to see if you immediately ask for a dozen permissions – which is poor UX). Only ask for what you need, when you need it. Moreover, **push notifications should not be abused:** Apple's design guidelines say they are for timely, relevant content for the user, **not for ads or promotions unless explicitly agreed to by the user**. They have enforced rejections for apps that spam users with irrelevant pushes. So design an opt-in or settings for marketing notifications if you plan to send them.

Overall, **present a well-crafted, user-friendly app**. A good way to assess this is to use Apple's own perspective: *Would Apple be proud to showcase this app on the App Store?* While not every app will be featured, thinking this way helps you strive for a baseline of quality. Remember, Apple's reviewers are also judging the “feel” of your app. One report noted that Apple rejected an app because they felt the paid content **“didn't provide enough long-term value to users”** ⁵². This is subjective, but it underscores that you should be confident in your app's usefulness. If your app is niche or simple, justify its purpose clearly in the description and ensure a quality execution of whatever it promises. And finally, **never include hidden or undocumented features** (e.g., secret debug menus or Easter eggs that alter functionality) — if discovered, that's grounds for rejection (or removal later).

5. Legal & Privacy: Compliance with Laws, User Rights, and Data Protection

The Legal section of Apple's guidelines overlaps with some safety and business aspects but focuses on complying with laws and protecting user rights. Key legal considerations when preparing your app for review:

- **Privacy Law Compliance:** Beyond having a privacy policy, your app must comply with privacy laws (GDPR in Europe, COPPA for children in the US, etc.) where applicable. For instance, if you target users in the EU, ensure you provide GDPR-related features (like a way to request data deletion). If children's data is involved, follow COPPA guidelines strictly (verifiable parental consent for data collection, etc.). Apple will hold you to these; guideline 5.1.1 specifically mentions complying with global privacy laws and protecting kids' data ⁵³. Apps in the **Kids category must not transmit any personal data** or even device identifiers to third parties, so be extremely careful with analytics or third-party SDKs in such apps.

- **User Data and Account Deletion:** As of mid-2022, if your app allows account creation, it **must also allow users to initiate account deletion from within the app** ²⁸. This is now an Apple requirement (Guideline 5.1.1(v)). Ensure there's a clear in-app option for "Delete Account" that actually deletes the user's data (or starts that process). Apple's reviewers may hunt for this if your app has sign-up/login functionality. Not offering account deletion is a common rejection reason now. Put the option in a logical place (e.g., Account settings) and document in your review notes how the deletion works (if it's not obvious). For example, you might say "Users can delete their account by going to Profile > Settings > Delete Account, which permanently removes all user data from our servers." ²⁹ ³⁴.
- **Intellectual Property (Copyright/Trademark):** Ensure that all content in your app (images, text, music, code) is either your original content or properly licensed. Apple will reject apps that infringe on copyrights – e.g., using music tracks you don't have rights to, or artwork taken from the internet without permission. Likewise, avoid using trademarked names or logos of other companies in your app or metadata unless you have permission. This includes not just obvious things like "Facebook" or "Apple" names, but also potentially using celebrity images, or sports team logos, etc. If your app includes user-generated content, you need a mechanism to handle IP complaints (DMCA takedown process, etc., though Apple doesn't always check that in review, it's good practice). Bottom line: if it's not yours, don't include it, or get written permission.
- **Legal Restrictions by Category:** Some app categories have legal requirements. For example:
 - **Healthcare/Medical apps:** If your app provides medical advice or could be seen as a medical device (like an app that claims to measure vitals), ensure you include disclaimers (e.g., "not for diagnostic purposes") and have regulatory clearance if needed. Apple often rejects apps making health claims without proper backing or disclaimers (Guideline 5.1.4).
 - **Finance apps:** If your app deals with cryptocurrency trading, stock trading, etc., make sure it complies with relevant laws and Apple guidelines (e.g., the app must come from a legitimate financial institution or use only public info; no unlicensed gambling or trading platforms).
 - **Gambling and Contests:** Apps with gambling (casino games, sports betting) must be geo-restricted to regions where that gambling is legal and you must have the necessary licenses. You should geofence features appropriately and disclose these restrictions. For contests or sweepstakes, you need to have official rules, make clear no Apple involvement, and not violate any lottery laws (guideline 5.3.4).
 - **Alcohol, Tobacco, CBD:** If your app involves sale of substances like alcohol or tobacco, check guidelines (e.g., liquor sale apps must be geo-restricted to legal ages and regions, and not use in-app purchase for it since that's physical goods). Apps related to cannabis (even legal CBD) are currently not allowed on the App Store if they facilitate its sale/delivery.
- **Export Compliance (Encryption):** When you submit your app, Apple will ask if your app uses encryption. If you use standard HTTPS, you can usually say yes (with exempt). But if you implement or contain encryption algorithms (like you have custom encryption), you might need to submit a year-end self-classification report under US law. Generally, fill out the encryption questionnaire accurately. Misstating encryption usage can lead to trouble or rejection (Apple might hold your binary until you provide more info). It's rare to be rejected for this, but providing correct info is part of compliance.

- **Third-Party SDK compliance:** You are responsible for what your included SDKs do ⁴⁰. For instance, if you include an analytics SDK that gathers data in a way that violates user privacy or Apple's guidelines (like an SDK that fingerprints the device), your app can be rejected or removed. Only use SDKs from reputable sources, keep them updated, and ensure they don't violate any policies (some advertising SDKs were flagged by Apple in the past for using private APIs or collecting too much info). If Apple flags an SDK issue, you may need to remove or update it.
- **No Cheating the System:** This is a general legal/ethics point: do not try to **manipulate App Store systems** – for example, do not buy fake reviews or incentivize users to give good ratings with in-app rewards (both are against policy). Don't hardcode a five-star prompt or link to review in a non-standard way; use Apple's provided Review prompt API if you want user reviews. Also, do not attempt to hide functionality or make a "version" of your app for review that changes later (Apple reviewers are keen to catch any differences between what they test and what users get). If you "cheat" and get caught, Apple can remove your app and even terminate your developer account ⁵⁴.

In summary, the Legal category is about **playing by the rules** – both Apple's rules and the law. Ensure users' rights are respected (privacy, ability to leave the service, etc.), and that your app doesn't venture into legally problematic territory. When in doubt about a legal issue (like offering a certain service or content), consult Apple's guidelines and developer forums; you may even seek clarification from Apple via the "Contact App Review" feature before submission if it's a grey area. It's better to be upfront and get guidance than to risk a rejection or a post-release removal.

Preparing Your App Store Listing for Review

In addition to the app binary itself, Apple scrutinizes your **App Store listing (App Store Connect metadata)**. Many apps get rejected for issues in their title, description, screenshots, or other listing details. Follow these tips to ensure your listing passes muster:

- **Accurate and Clear Description:** Write a factual, clear app description. **Avoid overhyped marketing buzzwords or unverifiable claims** ("#1 best app!" or "revolutionary experience") – Apple doesn't care for those, and they don't help users either ⁵⁵ ⁵⁶. Instead, focus on what the app actually does. Ensure every feature you mention in the description is **present and working in the app** (Apple will test against this) ⁵⁷. If you mention a specific functionality, the reviewer will likely try it. Also, if your app requires certain conditions (e.g., a device or a location), be sure to mention that too. **Keep it honest** – think of the description as a *contract* between you, users, and Apple ⁵⁷. Inconsistencies between description and app behavior can lead to rejection (as seen with the pricing example where a mis-stated price led to rejection ⁷).
- **No References to Unrelated Content:** Your metadata (especially description, keywords) should not reference other platforms or irrelevant content. Do **not mention "Android" or other competing platforms**, and don't include other app names or trademarks to boost search (Apple will reject for keyword stuffing or using trademarked terms that aren't yours ⁵⁰). Also, avoid referring to your app as "beta" or "test" – it should sound like a final product, not an experiment.
- **Proper Use of Keywords and Categories:** Choose relevant keywords that reflect your app's functionality. Do not use offensive or forbidden terms. Apple can reject apps for inappropriate keywords (like trademark violations or irrelevant keywords just to spam). Choose the correct

category for your app; if you pick something that doesn't fit, Apple might reassign it or reject until you fix it. For example, don't put your calorie-counter app in "Games." Misleading categorization can confuse users and thus is frowned upon.

- **Screenshots and App Preview:** Provide high-quality screenshots that **accurately represent the current UI and features of your app** ⁴⁸. Make sure to use the right device frames (e.g., iPhone 14 Pro Max screenshots for the largest size, etc.) as required by App Store Connect. Do not embellish screenshots with content that isn't from the app (like photoshopping in extra elements). You can use captions or text in screenshots to explain features, but keep it professional. If your app has both light and dark mode, showing both can be nice (not required, but good for users). If you include an App Preview video, ensure it follows Apple's rules (no misleading content, mostly actual screen recordings, and proper ratings). Reviewers **will watch the video** and make sure it doesn't show any feature or use that the app shouldn't do. Keep all graphic assets free of any inappropriate content (no adult imagery, etc., even if your app allows that in user content, your screenshots must be clean and appropriately rated).
- **App Icon and Version:** Upload the correct app icon – it should match the one in the binary (the one that shows on the device). Mismatched icons can confuse reviewers (and users). Also, ensure your app version in App Store Connect matches the one in the build's Info.plist. Apple can reject if the version numbers are inconsistent (though rare, it's a detail to get right). The bundle ID in the app must match the listing as well (if you upload via Xcode, this is handled, but just be mindful if you ever change bundle ID, you need a new listing).
- **Privacy Policy URL and Support URL:** As noted, include a Privacy Policy URL (mandatory for any app with account or data collection, and generally recommended for all apps now). Also provide a Support URL that is a working webpage where users (and Apple) can reach you or get help. These URLs should be live and not placeholders. Apple may check that the privacy policy link opens a valid policy document and that it's not blank or a 404.
- **App Review Notes:** We mentioned this earlier, but to reiterate: use the "Notes" field in your submission to communicate with the reviewer. If there's anything unusual about your app (login info, special instructions, use of a feature that might not be obvious), **explain it clearly in the notes** ⁵⁸. This field is your friend – it can preempt a lot of questions. For example, "Note: This app requires a Bluetooth connection to a proprietary sensor device. A demo video of the device in use is provided here [link] for reference." Or "Note: Login with any email/password using our test credentials user@test.com / password123. The app uses a demo sandbox environment for review." These notes can make the reviewer's job easier and show that you're proactive, which can smooth the path to approval.
- **Beta Tests and External TestFlight:** Ensure that if you previously used TestFlight, all test indications are removed from the app. Sometimes developers leave in banners or alerts ("This is a beta build") – remove those. Also, if you invite Apple to TestFlight for an expedited review, make sure you also properly submit the app for App Store review in App Store Connect (TestFlight itself is not a guarantee of App Store approval). Generally, by the time you formally submit to the App Store, your app should have no mention of "Test", "Beta", or debugging info visible.

By meticulously preparing your app listing, you not only avoid rejections but also create a better impression on users once you launch. Apple's review process involves comparing your **app's actual behavior to your app listing claims** ⁴⁷, so alignment between the two is crucial. A well-prepared metadata set also speeds up the review since the reviewer doesn't have to puzzle over what your app is or how to use it.

The App Review Process: What to Expect

Understanding how the review works can help you anticipate and address potential issues proactively. Here's a quick overview of the **Apple App Store review process** once you hit "Submit":

- 1. Automated Scans and Checks (Initial Screening):** When you submit your app, Apple's systems will first do automated checks. These include scanning for viruses/malware, checking that you're not using private APIs, and ensuring the app is properly signed. They also verify the basic metadata format (for example, that you provided a privacy policy if needed, that your screenshots are the correct size, etc.). If an automated scan fails (say, a private API use is detected), your app could be returned as **Rejected almost immediately** with a notice. Most well-prepared apps pass this stage without issue.
- 2. Waiting for Review:** After initial screening, your app enters the queue for human review. Apple claims most reviews are completed within 24-48 hours, though it can vary (peak app submission times or during holidays it might be slower). During this time, *be responsive to email* – if Apple needs more information, they might reach out.
- 3. Human Reviewer Testing the App:** A real Apple reviewer will install your app on a test device and go through it. They often follow a standard test plan: launch the app, navigate through core features, test any user flows mentioned in your description or triggered by common use (like creating an account, starting a first level in a game, making a purchase if your app has IAP – using Apple's test accounts – etc.). They might also specifically check any areas highlighted by guidelines or by your notes. For example, if your app uses location, they may check that the permission prompt appears with proper purpose text and that the app behaves if permission is denied. They may switch network off to see offline handling, etc., based on best practices they know.
- 4. Checking Against Guidelines:** As they test, the reviewer is basically checking your app against the guidelines we've outlined. They have an internal interface to mark if a guideline is violated and provide notes. You can imagine them almost with a mental (or actual) checklist: *Content okay? No crashes? All purchases via IAP? UI matches screenshots? Privacy policy present? Account deletion present (if applicable)?* – they will tick off these boxes. They'll also read your App Review Notes and any clarifications you gave.
- 5. Approval, Rejection, or Additional Information:** After testing, the reviewer will either mark the app **Approved** (at which point its status in App Store Connect changes to Ready for Sale or Pending Developer Release), or **Rejected** (with one or more guideline violation codes and notes in the Resolution Center). Sometimes, a reviewer might not outright reject but instead contact you via the Resolution Center for **Additional Information**. For example, they might say, "We were unable to log in. Please provide a valid demo account." or "Please clarify how feature X works." You should respond promptly via the Resolution Center messages. If you provide what they need quickly, they often continue the review without requiring a new submission.

6. **If Rejected:** Don't panic. Apple will cite at least one **guideline number** (e.g., 2.1 or 5.1.1) and a brief explanation. Read it carefully. They may not list every issue in one go; often they stop at the first major issue. Fix that issue (and anything else you suspect might be problematic), then resubmit the build. If you believe they made a mistake or misinterpreted the app, you have a few options: you can reply in the Resolution Center to ask for clarification or explain how you do comply. Occasionally, this can lead the reviewer to re-evaluate without a new submission (rare, but possible if it was a misunderstanding). If you strongly feel the rejection is unjustified and you've tried reasoning with the reviewer to no avail, there is an **appeal process** to the App Review Board via Apple's developer portal ⁵⁹. Use that as a last resort – usually it's faster to adjust your app or metadata to comply, if at all possible.
7. **App Approval:** If approved, congratulations! Your app can now be released. If you set it to "Manual release," it will wait for you to release it; if "Automatic," it will go live shortly. It's good practice to do one last sanity check on the live App Store listing to ensure all looks good (sometimes typos or minor metadata issues slip through and can be updated without a new app version in App Store Connect).
8. **Post-Approval Monitoring:** Even after approval, maintain compliance. Apple can pull apps from the store if they later find guideline violations or if there are user reports of issues. Also, with each app update you submit, the app goes through review again (though updates can sometimes be quicker if changes are minor). Always read the updated App Store Review Guidelines before major updates – Apple refines rules frequently (for example, they added rules about account deletion and push notification marketing in recent years). Keep your app updated accordingly so you don't get a surprise rejection on an update.

Important: The review process can sometimes feel unpredictable, because reviewers are human. One might be stricter or more detail-oriented than another. If you believe you got unlucky (for example, the reviewer cites something that you know is actually not a violation or something inconsistent with a prior review), you can try the "**Review Rejection Resolution Center chat**." Being polite and providing evidence (like "According to guideline X, we believe our app is compliant because...") can sometimes get them to reconsider or escalate to a second opinion. In many cases, though, it's faster to fix the problem if it's fixable.

Apple wants to approve apps – they just need to ensure the rules are followed. By **making the reviewer's job easy** (provide all info, test thoroughly, comply with rules), you maximize your chance of a swift approval. As one developer advice blog noted, *your goal is to ship a build the reviewer can verify without guessing* ⁶⁰ – meaning the reviewer should never be confused about how to use your app or whether something is allowed. If they have to guess or dig for info, you've left room for doubt, which can lead to rejection. So strive for clarity and completeness in everything you submit.

Common Reasons for App Rejection (and How to Avoid Them)

Finally, let's summarize some of the **most common reasons apps get rejected** by Apple, drawn from both Apple's documentation and real developer experiences. Avoiding these will put you well ahead of the pack:

- **Crashes and Bugs:** The number one reason for rejections is an app that crashes or has obvious broken functionality ³⁸. Solution: Thoroughly test and fix crashes, even in edge cases. Use TestFlight or a QA team to catch bugs before submission.

- **Incomplete App / Placeholder Content:** Apps that feel half-finished (with “coming soon” features or dummy text) will be rejected ³⁷. Solution: Only submit when your app is feature-complete for this version. Remove or hide any incomplete features.
- **Misleading or Inaccurate Metadata:** If your App Store description, screenshots, or even in-app text misrepresent the app’s capabilities, you risk rejection ⁴⁸ ⁶¹. This includes pricing discrepancies (saying something is free when it’s not, or wrong subscription info) ⁷. Solution: Keep metadata in sync with the app. If you update the app, update the screenshots/description accordingly.
- **Privacy Violations:** Not providing a privacy policy when required, asking for permissions without explanation, or collecting user data in a way that isn’t disclosed will get you rejected under privacy guidelines ⁸. Solution: Always include a privacy policy URL and in-app link, use permission dialogs responsibly with `NSUserDefaults` keys, and be transparent about data usage in the App Privacy section.
- **Lack of Account Deletion:** If your app has account creation but no in-app deletion option, Apple will reject it ²⁸. Solution: Implement account deletion and make sure it truly deletes or anonymizes user data, and is easy to find in the app UI.
- **In-App Purchase Issues:** Common mistakes include offering paid digital content without using IAP ⁴², not providing a restore purchases feature ⁴³, or having broken IAP flows. Solution: Audit your app for any place a user can purchase or subscribe – ensure all use Apple’s IAP and that you’ve tested purchase and restore with a test user. Don’t mention other payment methods.
- **User-Generated Content Lacking Protections:** Apps with user content but no reporting/blocking system will be rejected ²¹. Solution: Add those moderation features (even basic ones) and clearly state in your app listing that your app includes moderation.
- **Spam and Copycat Apps:** Apps that duplicate others or offer no unique value (e.g., many similar apps from same developer) are often rejected ⁶². Solution: Focus on one quality app, or if you have several, make sure each is distinctive. Don’t submit template-based apps in bulk.
- **Poor UI or User Experience:** If navigation is confusing or the app is very un-intuitive, a reviewer might reject it as not meeting the “minimum quality” bar. Also, if your app’s interface appears significantly different from what Apple expects (for example, using an Android-style back button graphic or something) it could raise flags. Solution: Have some users or colleagues use the app without guidance – if they get confused, refine the UX. Follow iOS design conventions for navigation and gestures.
- **Violating Apple’s Ecosystem Rules:** This includes things like not using Sign in with Apple when required, using forbidden frameworks, or not abiding by Apple’s hardware/software integration rules. Solution: If your app integrates with certain Apple services (Apple Health, Apple Pay, HomeKit, etc.), read the specific guidelines for those. For example, HealthKit apps must have a privacy policy; Apple Pay rules forbid certain uses of the Apple Pay logo; HomeKit requires using it only for approved purposes. Make sure you’re not breaking those sub-rules.

- **Offensive or Prohibited Content:** If your app has content that is pornographic, overly violent, or otherwise objectionable without proper controls, it will be rejected ¹⁹ ⁶³ . Solution: Remove or moderate such content, use age ratings appropriately, and implement content filters for user content.

To put it succinctly, apps often fail review because they **break a rule, crash, or make the reviewer's job difficult**. By adhering to guidelines and using the checklists provided, you mitigate these issues. As one resource summarizes, *reviewers will reject an app if it crashes, has broken/missing features, misuses data, or doesn't provide lasting value to users* ³⁸ ⁵¹ . Also, seemingly small things like a **"Try for free" tagline that isn't truly free** have caused rejections ⁶⁴ – Apple watches for anything that could be seen as deceptive.

If you do get a rejection, **don't be discouraged**. It's often a chance to improve your app. Fix the issue, test again, and resubmit. Many successful apps had a rejection or two early on. The key is to **learn from the feedback**: Apple is effectively telling you "Here's a problem; resolve it and you're good to go." Address it diligently and try again. Also, use Apple's documentation and developer forums; often, other developers have encountered similar rejections and can offer insight on how to resolve them.

Conclusion

Passing Apple's App Store review is a rite of passage for iOS developers. It might seem daunting with all the rules and potential pitfalls, but with careful preparation and attention to detail, **you can absolutely get your app approved smoothly**. Remember to **think like a reviewer**: ensure your app is stable, content is appropriate, user experience is solid, and all App Store rules are followed from functionality to fine-print. Use the checklists and guidelines in this document as a reference each time you prepare for submission.

In summary, **everything that may get an app rejected is avoidable if you plan ahead**. Test thoroughly, follow the guidelines around content, privacy, and payments, and be transparent in your App Store listing. By doing so, you not only increase your chances of approval but also end up with a higher-quality app for your users.

Good luck with your submission! If you've checked all the boxes above, you can hit that "Submit for Review" button with confidence. And once your app is on the App Store, celebrate the achievement – you earned it by navigating Apple's process successfully. Now users around the world can enjoy the app you've worked so hard to create, and you've set a solid foundation for all future updates and releases to also pass review with flying colors. Happy developing!

Sources: The guidance above was compiled based on Apple's official App Store Review Guidelines ⁶⁵ ¹⁹ , Apple Developer documentation, and expert insights from industry blogs and developers ⁶⁶ ³⁸ . Always refer to the latest official Apple guidelines as rules can evolve over time.

¹ ² ³ ⁵ ⁹ ¹⁰ ¹¹ ¹⁴ ¹⁵ ¹⁶ ¹⁸ ¹⁹ ²⁰ ²¹ ²⁵ ³¹ ³² ³⁵ ³⁶ ⁴⁰ ⁵³ ⁵⁴ ⁵⁸ ⁶³ ⁶⁵ App Review

Guidelines - Apple Developer

<https://developer.apple.com/app-store/review/guidelines/>

4 6 13 17 22 23 24 28 29 30 33 34 37 42 43 44 45 47 60 66 App Store Review Guidelines (2025):

Checklist + Top Rejection Reasons

<https://nextnative.dev/blog/app-store-review-guidelines>

7 8 12 26 27 38 39 41 46 48 49 50 51 52 55 56 57 61 62 64 iOS App Store Review Guidelines:

How to Pass & Avoid Rejection

<https://appfollow.io/blog/app-store-review-guidelines>

59 App Review - Distribute - Apple Developer

<https://developer.apple.com/distribute/app-review/>