How to Use a Domain for Public Examples in Documentation

When you write technical documentation, you often need to show examples that include a web address. You might want to demonstrate how an API call works, how to configure a URL, or how a link appears in a user interface. The standard guidance for years has been to use example.com or example.org. These are reserved by the internet standards community specifically for that purpose. They are not real services. They are a domain for public examples.

Understanding the purpose of these reserved domains is important for anyone who writes docs, builds tutorials, or creates training materials. Using a real domain in an example can cause confusion. Someone might try to visit the address and expect to find a working site. Worse, the example might accidentally point to a live service that the reader mistakes for the intended target. That is why the concept of a domain for public examples exists. It keeps examples clean, safe, and clearly fictional.

What Makes a Domain Suitable for Examples

The internet assigned numbers authority, known as iana, reserved several domain names under the top-level domains .example.test.invalid, and .localhost. The most familiar are example.com and example.org. These domains are not registered to any organisation. They are maintained by iana and used by the internet engineering task force for documentation. The key property is that they are guaranteed to never be assigned to a real service. That means you can use them in examples without worrying about misleading readers.

When you pick a domain for an example, you want something that is instantly recognisable as a placeholder. Example.com does that well. It has been used for decades. Its sibling, example.org, serves the same purpose and is often preferred when the example needs an organisation-style address. Both work. The important thing is that the reader understands immediately that the address is not real. If you pick a domain that looks too plausible, you risk the reader trying to visit it and finding something unexpected, or worse, finding a malicious site that someone registered after you published.

Why Not Use a Random Domain

It might be tempting to grab any unused domain for an example. That approach has several problems. First, domains are cheap and can be registered by anyone. If you use a random domain in your documentation, someone could register it later and serve content that you never intended. That could break your examples or, in a worst case, expose readers to harmful

material. Second, a random domain does not signal that it is a placeholder. Readers might think it is a real service and try to use it. Third, using random domains creates inconsistency. Different writers pick different addresses, and the reader never knows which are real and which are not.

Reserved domains avoid all of that. They are stable, well-known, and clearly marked as examples. They also work in testing environments because they are not routable on the public internet in the usual way. Some browsers and tools will even warn you if you try to visit example.com, reinforcing the message that it is not a real site.

Practical Tips for Using Example Domains

When you write documentation, use the reserved domain consistently. If you are showing a url in a code snippet, use example.com or example.org. If you need to show an email address for an example, use @example.com. The same logic applies. The reserved domains are not just for web addresses. They cover email, ftp, and other services. The iana registry includes example.com and example.org for exactly this breadth of use.

There is a nuance with subdomains. You can use sub.example.com or api.example.org in your examples. That is fine. The subdomain part is arbitrary. The reserved domain part stays the same. That gives you flexibility while keeping the example safe. For instance, you might show a rest api call to https://api.example.org/v1/users. The reader recognises the example address and knows it is not real. The structure of the url is what matters, not the actual host.

Another consideration is the protocol. Use https in examples when you want to show a secure connection. Use http when you want to show a plain connection. The protocol does not change the reserved status of the domain. However, be aware that example.com does not have a real ssl certificate, so browsers will show a warning if someone tries to visit it. That is fine for examples. It reinforces the message.

Common Mistakes and How to Avoid Them

One common mistake is using a domain that looks like an example but is not reserved. For example, something.com or myexample.com are not reserved. They are real domains that someone could own. Using them in documentation is risky. Another mistake is using the reserved domain in a context where it could be interpreted as a real service. For instance, if you write "visit example.com for more details", the reader might try to go there. But if you write "for demonstration purposes, we use the address http://example.org", the intent is clear.

Another mistake is forgetting to use the domain for public examples consistently across all documentation. If some documents use example.com and others use a different placeholder, the inconsistency can confuse readers. Agree on a standard within your team. Many

organisations adopt example.com as the primary domain and example.org as a secondary one when they need to show two different entities. That is a simple rule that covers most use cases.

Also, avoid using the reserved domain in production configuration files. That sounds obvious, but it happens. Someone copies an example config into production without changing the url. The result is that the application tries to connect to example.com and fails. To prevent that, make sure your example configs use the reserved domain clearly, and your deployment scripts check for placeholder values.

The Role of Example Domains in Testing

Beyond documentation, reserved domains are useful in testing. When you write unit tests or integration tests that involve urls, using example.com keeps the test self-contained. The test does not depend on an external service. It also makes the test intention clear: the url is a placeholder. If the test accidentally makes a real network call, it will fail quickly because example.com does not resolve to a real server. That is a good thing. It catches mistakes early.

In automated testing, you can also use the .test top-level domain. The .test domain is reserved for testing purposes and is not resolvable on the public internet. Some testing frameworks use it to simulate network calls without risk. The .test domain is less known to general readers, so it is best used in internal documentation and testing contexts. For public documentation, stick with example.com or example.org.

Keeping Examples Clear and Useful

Ultimately, the goal of using a domain for public examples is to make your documentation clear and safe. The reader should never wonder whether an address is real. They should focus on the concept you are explaining, not on the url itself. The reserved domains serve that purpose well. They have been around for a long time, and they are recognised by developers, testers, and technical writers worldwide.

When you are writing a new tutorial or reference guide, take a moment to think about the examples you are including. If any of them contain a url, swap it for a reserved domain. It is a small change that makes a big difference. It protects your readers from confusion and protects your work from becoming outdated if a domain changes hands. The internet is full of examples that used real domains and later broke. Using example.com or example.org avoids that problem entirely.

So next time you write a code sample, a configuration snippet, or a user guide, remember the domain for public examples. It is a simple tool that keeps your documentation professional, reliable, and easy to follow.