Style guide: Level A - Lowest

This section of the style guide defines the minimum (level A) requirements for displaying attestation information on package registries:

1. Attestation information should be placed in a visible, but not overpowering, location on registry pages
2. It should contain enough information to make attestations understandable to a broad variety of users (as detailed in this introduction)
3. It must not impede users from achieving their primary goal, e.g., “finding out if this package is the best choice for the intended purpose.”

These Level A requirements describe a set of minimal, collapsible, and distinct UI "panels" with icons and headings appropriate to the information that can be expanded in them.

Alongside the following UI examples, variants, and recommendation we will reference and explain the user research rationale and insight the led to that decision.

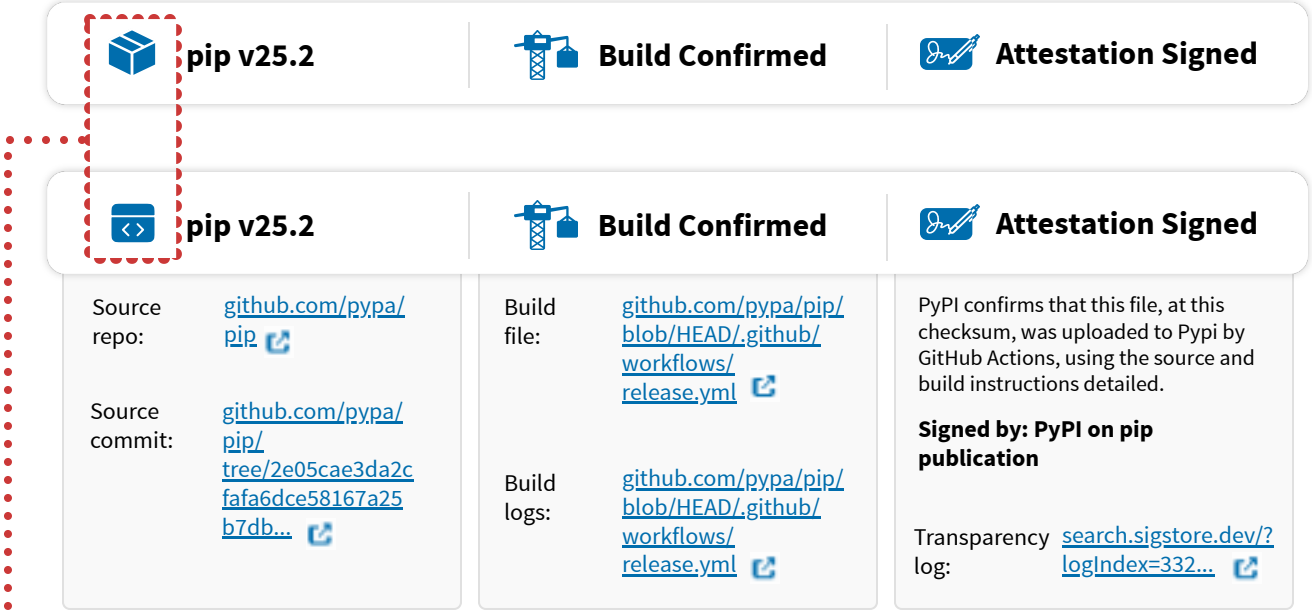
You can find the unedited user research sessions on the open issues related to this project:

- User Testing Sessions Issue #82: https://github.com/ossf/wg-securing-software-repos/issues/82
- Persona & Research Issue #66: https://github.com/ossf/wg-securing-software-repos/issues/66

If you'd like to contribute, comment or iterate on this work then please see the design contribution documentation.

Style guide: Level A - Lowest requirements

The recommended components for Level A, "minimal UI".



The expanded sections have a border stroke and also a background color. You can omit one or the other as long as the expanded section of the panel is in close proximity to appropriate heading.

Depending on the width space you have available you can increase the width of both the panel and the expanded section. Be sure to use minimum text/font sizing available per platform for legibility.

With this Level A component styling, use words and letters conservatively in the attestation statement, balancing clarity of statement with space available. If needed, the attestation can be placed underneath at full width to accomodate complex statements (UI example on next page).

re. the package icon or source code icon. Either can be used interchangeably for this information. Users connected both with this info in research.

This is intended to sit in a main section of a webpage that might use a three-column layout--the main section occupying two-thirds and a sidebar occupying one-third of the width.

In this version, each icon and heading is seperated by a vertical bar/pipe that spans approximately 75% of the total height of the container box.

The panels can have a stroke or a drop shadow effect, yet should be adjusted to match whatever the platform's design system or styles.

While this panel can be used as-is with no interactive expansion, we advise against this.

Since this component is a panel each heading should be clickable to expand additional info underneath. The expand/collapse interaction should be eased and not instantaneous for users.

Style guide: Level A - Lowest requirements

Attestation Signed

PyPI confirms that this file, at this checksum, was uploaded to Pypi by GitHub Actions, using the source and build instructions detailed. With a much more complex and lengthy statement and also possibly two or more 'signed by' statements.

Signed by: PyPI on pip publication

Signed by: An additional signer

Transparency log: search.sigstore.dev/?logIndex=332...

pip v25.2

Build Confirmed

Attestation Signed

pip v25.2

Build Confirmed

Attestation Signed

Source repo: github.com/pypa/pip

Source commit: github.com/pypa/pip/tree/2e05cae3da2cfafa6dce58167a25b7db...

Build file: github.com/pypa/pip/blob/HEAD/.github/workflows/release.yml

Build logs: github.com/pypa/pip/blob/HEAD/.github/workflows/release.yml

PyPI confirms that this file, at this checksum, was uploaded to Pypi by GitHub Actions, using the source and build instructions detailed.

Signed by: PyPI on pip publication

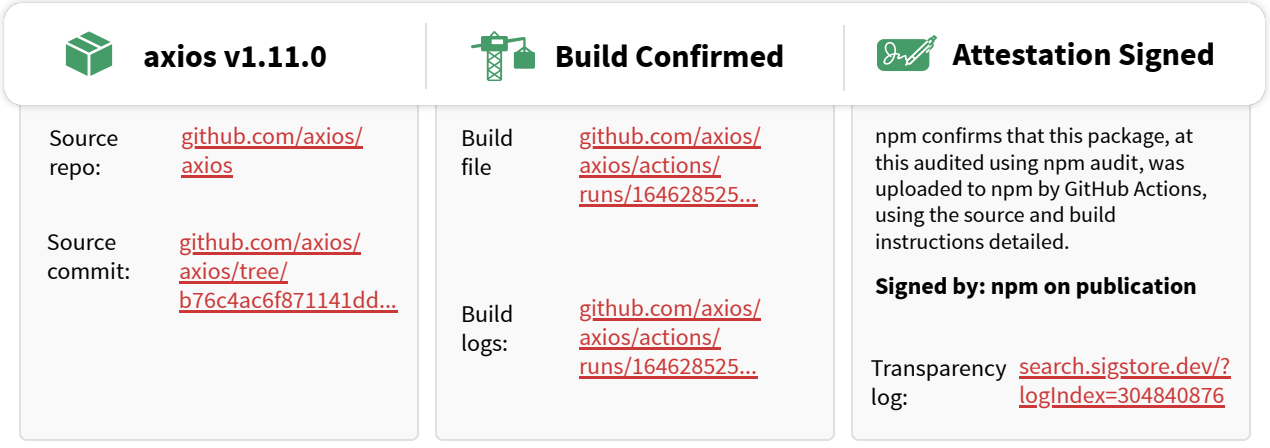
Transparency log: search.sigstore.dev/?logIndex=332...

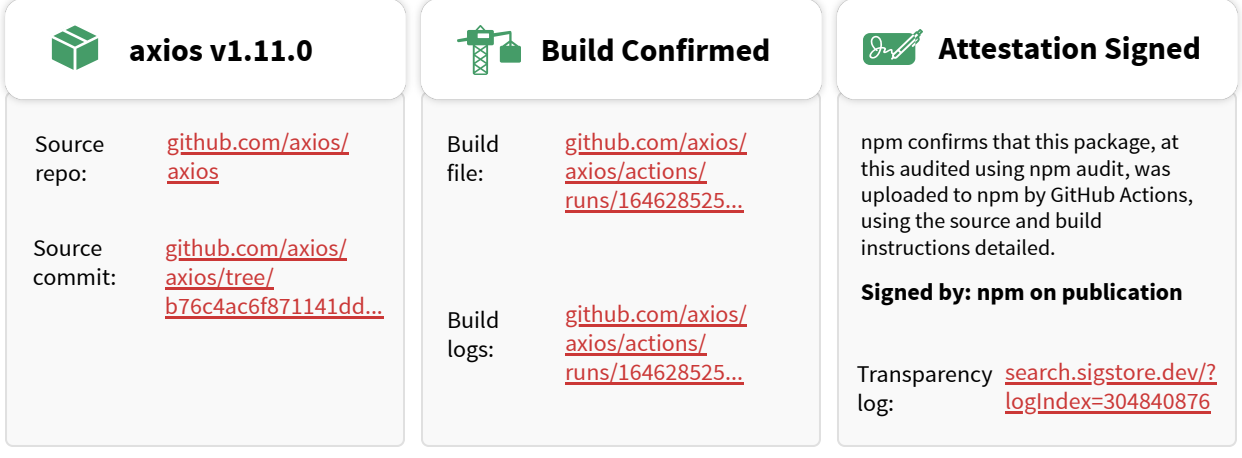
Signed by messages:
"Signed by" messages should be bold to indicate importance. They can flow over two lines but three lines should be avoided.

Links styling and icons:
In this example, as in the last page, we've used PyPI's visual styling with blue hyperlinks and an external link icon next to each link. Both the icon colour and font can be changed to match existing styles. The use of an "open in a new tab" icon is optional.

The component variant to the left here shows when the panels are not grouped, instead they're separate panels aligned in a row. Their expandable sections still can be activated. Some platforms may prefer separated panels like this for their pages' global styles and/or to look more like buttons that can be interacted with.

Style guide: Level A - Lowest requirements





In this example, we've used npm visual styles.

Link labels
The links displayed here have all been carefully tested with users. In our research, we found that users expected the following information to be displayed near the attestation statement:

1. Source repo & source commit
2. Build commit & build logs
3. Transparency log (within the attestation statement section)

We recommend left-aligning link labels, allowing them to span over two lines where required.

Enhancing user confidence
Additional recommendations to help users feel confident about the security of a package are detailed in the medium (level AA) and highest (level AAA) UI recommendations.

Users also rely on "social proof" (number of downloads, maintainers, recognizable project names) typically found on registry pages. Hashes/checksums also encourage confidence, but only if users recognize them and understand their function.

Style guide: Level A - Lowest requirements

rake v13.3.0 | **Build Confirmed** | **Attestation Signed**

rake v13.3.0 | **Build Confirmed** | **Attestation Signed**

Source repo: github.com/ruby/rake

Source commit: github.com/axios/axios/tree/b76c4ac6f871141dd011a21f3b7ca4e66bfc33ae

Build file: github.com/axios/axios/actions/runs/16462852560/workflow

Build logs: github.com/axios/axios/actions/runs/16462852560/attempts/1

RubyGems confirms that this gem file, at this checksum, was uploaded to RubyGems by GitHub Actions, using the expected source and build instructions.

Signed by RubyGems.org on publication

Transparency log: search.sigstore.dev/?logIndex=225255766

rake v13.3.0 | **Build Confirmed** | **Attestation Signed**

rake v13.3.0 | **Build Confirmed** | **Attestation Signed**

Source repo: github.com/ruby/rake

Source commit: github.com/axios/axios/tree/b76c4ac6f871141dd011a21f3b7ca4e66bfc33ae

Build file: github.com/axios/axios/actions/runs/16462852560/workflow

Build logs: github.com/axios/axios/actions/runs/16462852560/attempts/1

RubyGems confirms that this gem file, at this checksum, was uploaded to RubyGems by GitHub Actions, using the expected source and build instructions.

Signed by RubyGems .org on publication

Transparency log: search.sigstore.dev/?logIndex=225255766

This example uses visual styles from RubyGems.

How different personas understand this UI

The 'Security architect' persona already knows what information is needed to feel confident about a package's security. For them, simple visual and text indicators (e.g., this UI in its collapsed state) are typically sufficient; expanding to see more information confirms their existing knowledge.

For 'Pragmatic Developers' and 'Incidental User' personas, simple visual and text indicators (e.g., this UI in its collapsed state) are not helpful and can lead the user to guess what the information means (either correctly or incorrectly).

These users need to see this UI in its expanded state to understand what attestations communicate. Viewing the detailed information typically prompts them to explore further and piece together an understanding of package security. Providing additional documentation links can also help these users learn.

Style guide: Level A - Lowest summary

The positioning of these minimal elements on a page is also critical. The next page in this style guide illustrates their optimal positioning within each example package registry (please note that the next page is large, and you may need to scroll to find the appropriate UI example).

Ideally, this UI should not reside at the bottom of a page, particularly on platforms where extensive README files push other critical information downwards. User feedback indicates a strong preference to understand a package's core functionality before encountering security or attestation details.

Placing these UI elements at the top of a page or within a header component presents a dilemma:

1. **Prioritization:** It effectively "forces" (as users described) immediate consideration of security.
2. **Interruption:** It disrupts the user's primary goal of quickly assessing the package's purpose.

While promoting security awareness is important, it shouldn't overshadow essential package information. Therefore, while placing these elements in the first scroll/fold can prioritize security, it's generally more effective to position them at the top of the second scroll/fold. This aligns better with user expectations and their natural information discovery journey.

Generic webpage structure