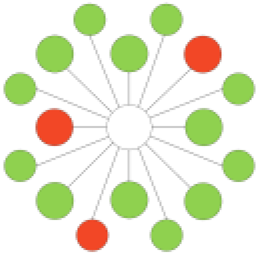
Monitoring Status Pages Built With Atlassian Statuspage

White Paper 2024



IncidentHub

Last Updated 14 Oct 2024

ABSTRACT

Atlassian Statuspage is a status page software used by many SaaS and Cloud vendors. This whitepaper is a summary of the experience of building a status page parser for status pages built with Atlassian Statuspage. The parser is part of IncidentHub which monitors hundreds of public status pages, which are built with Atlassian Statuspage as well as other status page providers.

Introduction

Many Cloud and SaaS service vendors host a publicly viewable status page to inform users about outages, maintenance events, and general information about their services. A status page has information about ongoing, future, and past events that affected the service. It’s primarily designed for human consumption, but many status pages also expose an API that can be consumed by computer programs. Atlassian Statuspage is one such provider.

IncidentHub is a status page monitor that checks hundreds of SaaS/Cloud vendors’ public status pages and provides a single place to monitor them and receive notifications.

The Importance of Monitoring Status Pages

Status pages are often the first indication that something is wrong with a specific service in a Cloud or SaaS vendor. Status pages can offer

- Real-time incident communication.
- Customer trust and transparency.
- Compliance and reporting requirements.
- Historical incident information.

Understanding Atlassian Statuspage

Atlassian Statuspage is a managed status page provider used by many Cloud and SaaS vendors. Updating the status page is completely within the control of the vendor. Like most other status page solutions, Atlassian Statuspage allows the vendor to configure various aspects:

- Configure components - these are services and regions offered by the vendor. Components can be grouped together under a parent component, or be standalone. A status page can have a combination of this.
- Manage incident lifecycle updates.
- Configure which notification channels can be used by users to receive notifications.

Atlassian Statuspage has two kinds of APIs

- The developer API that can be used by the vendor or their developers to manage and update the status page. This API needs authentication. It is used by the vendor's admin or SRE team to update the status page.
- The status page API that is unauthenticated and is open to the public. This can be used by anyone on the internet to fetch incident information for that status page.

Atlassian Statuspage offers the following ways to consume incident data:

- The status page API - a REST API which returns a JSON response
- The status page itself, in HTML, primarily for human consumption
- Notification channels which trigger when there are incident lifecycle updates. These include Email, Text message, Slack, Webhook, and Atom/RSS Feeds.

Monitoring Atlassian Statuspage Pages

Programmatic Monitoring

Programmatically monitoring an Atlassian Statuspage page can be done in various ways

- Scrape the HTML page. This option is error prone. It also has the additional risk that Atlassian can block the scraper from making too many requests.
- Poll the REST API periodically. This is the best option as it fetches well-structured data that reflects the current state of the system.
- Sign up to receive the RSS/Atom feed. This is semi-structured, as there is no standard format for incident notifications in RSS. The incident lifecycle metadata is contained within the content, and has to be parsed further.
- Sign up for the notification channels. This can involve further processing depending on the channel, which can again be error prone, except for the webhook channel.

If you are writing a status page parser for Atlassian Statuspage, the most reliable way is using the REST API.

In all the examples below, "site-address" refers to the status page domain. For white labeled status pages, it will be a subdomain of the vendor's domain, or a dedicated status page domain. E.g. https://status.digitalocean.com/ or https://www.vercel-status.com/

Using the REST API

Atlassian Statuspage has two endpoints in the public status API that are of interest here

- All components https://site-address/api/v2/components.json
- All incidents including resolved and unresolved https://site-address/api/v2/incidents.json

Programmatic monitoring requires a few considerations

- The vendor can configure a list of components, usually specific services that make up the vendor's platform or product, and/or regions for those services. This hierarchical relationship can be observed by making a REST API call to https://site-address/api/v2/components.json
    - Components either form part of a component group, or are standalone. A component group is a logical unit of components. E.g. all datacenter locations under a specific Email service are part of the Email component group.
- The REST API to fetch the list of incidents is https://site-address/api/v2/incidents.json. This returns a list of the 50 most recent incidents, including ongoing and past incidents.
- If you wish to monitor maintenance events, the REST API to fetch upcoming maintenance events is https://site-address/api/v2/scheduled-maintenances/upcoming.json

There are also two other APIs for active (ongoing) and overall maintenance events.

API Response Structure

The general structure of the incidents.json is as follows

- Page metadata
- List of incidents
    - Incident Metadata
    - List of incident updates
        - Each update has information about that update, and the list of affected components. The list of component IDs is a subset of the previously obtained list from the components.json

        - The “status” field of each update event is from a predefined list, and so is the “impact” field.

The API endpoints under api/v2 are unauthenticated, and are read-only, and meant for consuming data.

The API documentation can be found under https://site-address/api. This URL structure works for all Atlassian Statuspage pages.

Monitoring Challenges

The well-formed structure of the Atlassian Statuspage API and the various notification options makes it easy to consume. However, there can be challenges.

Channel Restrictions

Even though a status page is public, it is possible for a vendor to turn off one or more of the notification channels. Both Vercel and Mailgun use Atlassian Statuspage. However, Mailgun’s status page does not have any way to subscribe to notifications, whereas Vercel’s status page has all the options enabled.

However, the REST API is usually always publicly open.

No Filtering Capabilities

The public status API does not offer any filtering capabilities - e.g. there is no way to get only the last 5 incidents, or get only active incidents, and so on. Filtering and advanced querying are part of another API which is authenticated, and hence accessible only to the vendor.

As a third-party, you will have access to only the public status API. Data fetched from this API has to be processed after it is fetched, which can be wasteful if there are many historical incidents in the JSON response as you will have to fetch the entire incidents list each time.

Best Practices for Monitoring

- As of this writing, the public API is not rate-limited according to the Atlassian documentation. However, it is important to be mindful to not hit the API endpoint too frequently.

- Although polling the incident API is one option, another option to consume incident events programmatically is the webhook, which is also real-time. The webhook payload is well-structured and documented.
- The list of valid components can keep changing depending on the vendor's services. If you are monitoring the incident API and doing some processing based on a local copy of components list, it's important to sync your list of components with the components API from time to time.

Example Status Pages

Vercel - https://www.vercel-status.com
Digital Ocean - https://status.digitalocean.com/
GitHub - https://www.githubstatus.com
Reddit - https://www.redditstatus.com/
Asana - https://status.asana.com/
Imperva - https://status.imperva.com/
Render - https://status.render.com

References

1. Atlassian Statuspage APIs
2. Atlassian Webhook Notifications
3. IncidentHub - Status Page Monitor