

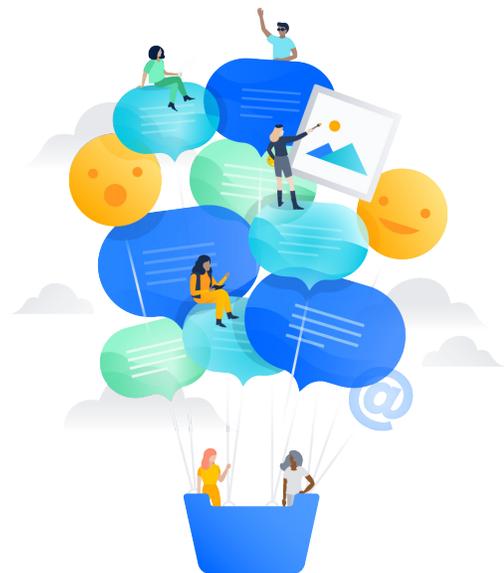
# The complete guide to incident communication

---



# Table of contents

4	Chapter 1: Defining an incident and knowing when to announce downtime
8	Chapter 2: Internal vs. external incident communication
12	Chapter 3: Communication channels
15	Chapter 4: Getting the message out: the right places to share your status page
24	Chapter 5: Stages of incident communication
29	Chapter 6: Incident templates
33	Chapter 7: Real-world examples and case studies
42	Chapter 8: Integrations and features for leveling up your incident comms
50	Chapter 9: Should you automate your status page?
56	Chapter 10: Tools, training, and other resources



# Introduction

We didn't invent incident communication here at Statuspage. But over the last seven years, we've certainly played a role in shaping how it's done today. Teams all over the world, from Dropbox to Reddit to Twilio, turn to Statuspage to alert customers and end users during incidents.

But cloud services and SaaS were still bleeding-edge concepts when our founders first launched "StatusPage.io" in 2013. Back then, any system downtime was announced over email. Or, often, not announced at all.

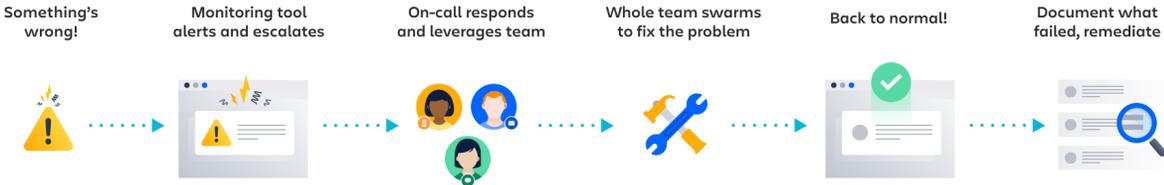
As IT teams have always known, deploying and hosting business-critical applications involves a ton of trust. The relationship between end user and infrastructure team is built on trust. This is true whether that infrastructure team is down the hall or in a data center on another continent.

Nothing erodes that trust like system downtime and poor communication. Downtime can't always be avoided. But poor communication can be. That's why our mission at Statuspage is to build trust with every incident. This ebook is a collection of all the best practices and tips about incident communication we've learned over the years.

---

## WHAT YOU SEE

---

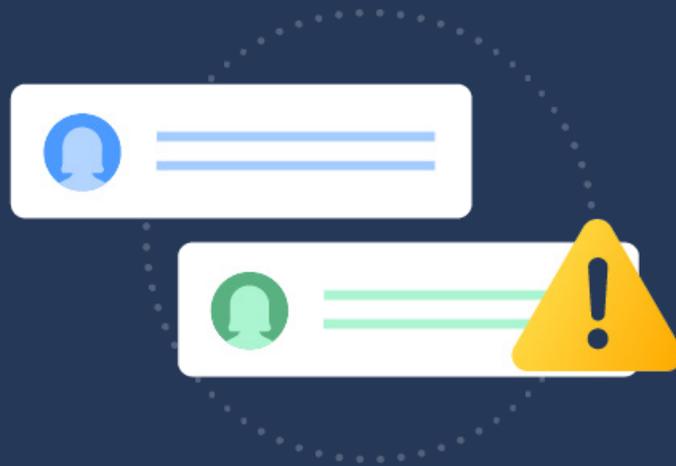


---

## WHAT YOUR CUSTOMERS SEE

---





# 01

---

## Defining an incident and knowing when to announce downtime

# Internal vs. external incident communication

## Defining an incident

Before you can communicate incidents, you need to decide what constitutes an incident.

At Atlassian, we define an incident as an event that causes disruption to or a reduction in the quality of a service, requiring an emergency response.

Teams who follow ITIL or ITSM practices may use the term “major incident” for this instead. When we refer to incident communications, we’re generally referring to the kind of issues that affect multiple (or all) end users of a service and require a one-to-many announcement. Isolated incidents that affect just one customer (the CEO forgot her password, for example) are typically addressed in a service desk ticket. To learn more about service desk and ITSM best practices, check out [Atlassian.com/ITSM](https://atlassian.com/ITSM).

An incident for Atlassian is considered resolved when the affected service resumes functioning in its usual way. This includes only those tasks required to restore full functionality and excludes follow-on tasks such as root cause identification and mitigation, which are part of the postmortem process.

## Incident severity levels

Many web companies define incidents by severity and rely on a standardized 3- or 4-tier severity definition system.

Incident severity levels are a measurement of the impact an incident has on the business. Typically, the lower the severity number, the more impactful the incident.

For example: At Atlassian, we define a SEV (severity) 1 incident as “a critical incident with very high impact.” This could include a customer data loss, a security breach, or a client-facing service going down for all customers.

A SEV 2 incident is a “major incident with significant impact,” including when a client-facing service is down for a sub-set of customers or a critical function within a system is not functioning.

And a SEV 3 incident is “a minor incident with low impact,” such as a system glitch that is causing customers slight inconvenience.

At Atlassian, SEV 3 incidents can be handled during daytime/working hours, while SEV 1 and SEV 2 incidents generate an alert for [on-call](#) professionals for an immediate fix no matter the time of day.

Whatever your thresholds are for incident severity, it’s important to draw a clear line in the sand (ideally around some sort of measurable metric). If you designate an incident at SEV 1, it’s important for everyone on your team to know exactly what that means.

Feel free to adjust definitions based on your services and teams. According to a set of guidelines Google Site Reliability Engineers use, an event is considered an incident if any of the following is true:

- Do you need to involve a second team in fixing the problem?
- Is the outage visible to customers?
- Is the issue unsolved even after an hour’s concentrated analysis?

## Severity levels and incident communications

We say this all because severity levels can help teams set guidelines around what kind of communication is expected for different severity levels.

A note on playing it safe: in our experience, it’s better to overcommunicate if you’re uncertain about an incident. The inconvenience of closing the loop on an expected incident that never took off far outweighs the downside of playing catch-up on incident communications hours into an incident.

“I’ll just fix this quickly before anyone notices,” is a gamble. You might play and luck out once or twice. But try it enough times and eventually you’ll pay for it.

## When to announce downtime

At Atlassian, we train major incident managers to announce incidents on our status page once they feel reasonably confident that the incident is real, and after they've taken initial incident response steps, like notifying the right responders and establishing an incident Slack channel.

## Who's in charge? Incident roles and responsibilities

So whose job is it to communicate with customers and send updates? Without a clear plan going into an incident, it often becomes a case of everyone saying "I thought someone else was doing it."

A proper incident response team designates clear roles and responsibilities. Team members know what the different roles are, what they're responsible for, and who is in which role during an incident.

The roles we use at Atlassian are:



**Incident Manager:** has the overall responsibility and authority during an incident. The incident manager is empowered to take any action necessary to resolve the incident, which includes paging anyone in the organization and keeping those involved in an incident focused on restoring service as quickly as possible. As a rule of thumb, the incident manager is responsible for all roles and facets of the incident, including communications, until they designate the role to someone else. The incident manager is typically the person who updates the status page unless (or until) a specific communications manager is designated. This role is sometimes called the "major incident manager" or "incident commander."



**Communications Manager:** familiar with public communications, possibly from the customer support team or public relations. This person is responsible for writing and sending internal and external communications about the incident. This is usually the person who updates the status page.



**Tech Lead:** a senior technical responder. This person is responsible for developing theories about what's broken and why, deciding on changes, and running the technical team. They work closely with the IM.

The incident manager can also devise and delegate roles as required by the incident. For example, they could set multiple tech leads if more than one stream of work is underway, or separate internal and external communications managers.

Other roles, for public-facing incidents, could be a customer support lead for handling inbound tickets and 1-to-1 comms with customers. Also, a social media lead can help communicate and announce updates over social channels.



# 02

---

## Internal vs. external incident communication

# Internal vs. external incident communication

## When is an incident internal or external?

One important consideration for any incident communication plan is first deciding whether your audience is internal (inside your own company), external (out in the public), or a blend of both.

In deciding whether incidents need to be communicated internally or externally, we find many teams are asking the wrong question.

The question is not:

- Is this an internal- or external-facing service? Instead, ask yourself:
- Is this incident impacting external customers or stakeholders?

For example, an airline might have an incident on a reservation system that only internal agents have access to. Typically only airline workers would need to know about an incident on this service. But if a particular incident caused external customer impact (flights delayed, reservations canceled) it would warrant external communication.

## Internal vs. external messaging

The type of message also varies depending on your audience.

For external audiences, often the best message is to let customers know you're aware of the problem and working on a fix. Boggling them down in the technical details usually isn't necessary.

Internal audiences typically need more information and detail than external audiences. And typically teams are allowed to share more sensitive details with internal colleagues than they could with external customers.

But again, it depends. And since we're communicating with real people, it often becomes more art than science.

For example, if you're telling your sales team the company email is down, they probably don't need to know how many requests per millisecond your load balancer is processing.



# 03

## Communication channels

# Communication channels

Professional support teams and site reliability engineers don't decide on the fly what channels to communicate with. They make a plan ahead of time.

There are six main communication channels for incident communication:

- Dedicated status page
- Embedded status
- Email
- Workplace chat tool
- Social media
- SMS

## Dedicated status page

We recommend teams use a dedicated status page as their primary incident communication source of truth. Whether you build it yourself or go with a SaaS solution like Statuspage, it's important to give your customers and colleagues a clear source of truth during an incident. Statuspage also gives your users the option to subscribe to get updates the moment they're posted. This takes the support burden off teams who should be heads-down fixing the problem.

## Embedded status

At Statuspage, we make it easy to embed status information directly onto any website our customers operate. We know most visitors are likely to check a provider's home page or support page before looking for a status page. The embedded widget is an easy way of letting those visitors know if an incident is underway. Visitors can also click through on the widget to get to the status page.

## Email

Like we just mentioned, a good status page tool will give your audience the option to subscribe to email updates. Even if you're sending directly from your email tool, as opposed to using Statuspage to send email, it's a good channel for incident communication.

## Chat tools

Chat tools like Slack have taken over the workplace in recent years. Many teams set up a dedicated channel for incident communications, or spin up a new room for each incident.

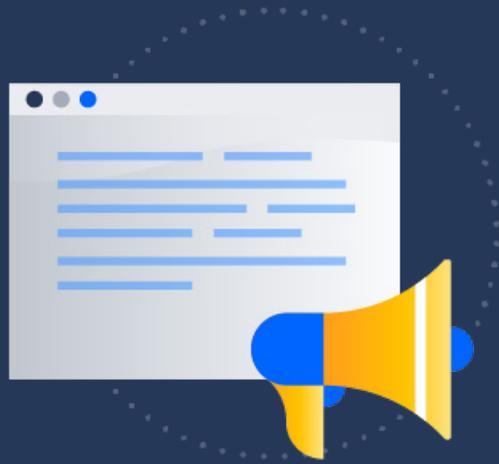
## Social media

Many teams use social channels like Twitter as a means of communication during an incident. It's good to use this as a piece of your strategy, but don't rely on it as your only means of communication.

## SMS

Sending an SMS message, or text message, is often a more immediate way to reach someone, and is the preference of many people when it comes to critical inbound alerts like a downtime announcement. It's also a channel where people can experience message fatigue very fast and will unsubscribe if they see too many messages that aren't relevant to them.

None of these channels are a silver bullet for incident comms. They all have different strengths, and the real power comes when you layer them together. For example, we post incidents to a status page but also sync those updates to Twitter. The updates are also embedded inside our web app. These messages then direct the user back to the status page for more details on the incident. We recommend you identify one channel as your primary communication source of truth, and use other channels to funnel everyone there.



# 04

---

## Getting the message out: the right places to share your status page

# Getting the message out: the right places to share your status page

Once you've settled on a status page and what channels to share incident communications on, it's helpful to get the word out and make it known to customers that you have a page for incident comms.

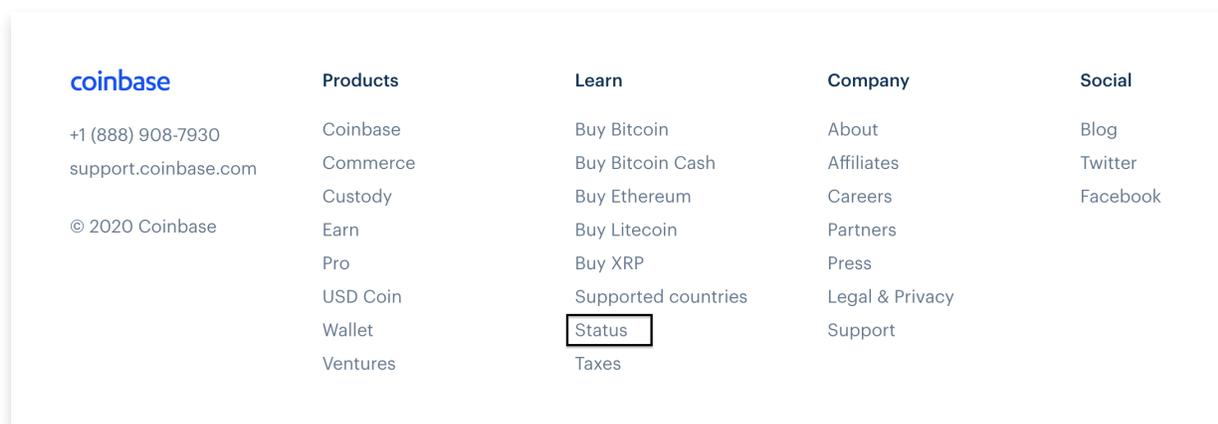
Many teams wait until an incident is happening to let their users and teams know they have a status page. This is a mistake. A little preparation goes a long way. If your customers and colleagues know you have a status page and what it's for, they're more likely to check there first for updates during the next incident, rather than flooding your team with tickets and questions.

Here are a few of the most helpful ways to share your status page and make sure it's visible for the right people.

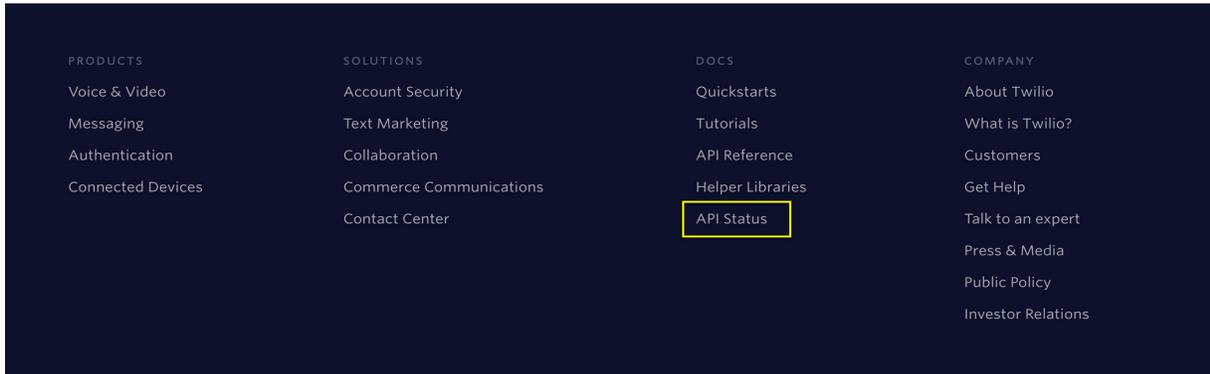
## Add a link to your website footer

Many organizations add a link to their status page in their website footer.

Here's an example from Coinbase:



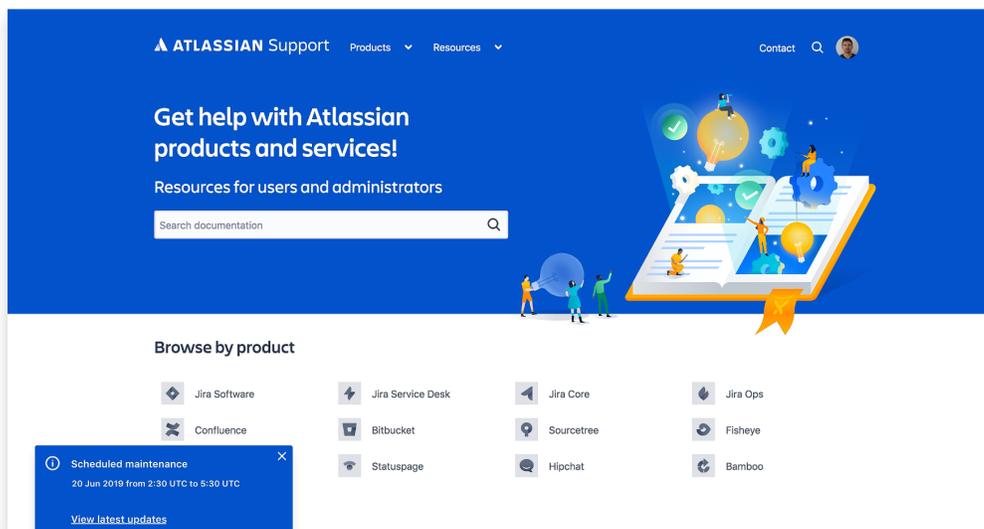
And another from Twilio:



## Status embed

Some page viewers may not know about your status page and will visit your website, knowledge base, or support portal for information on your service availability. You can add a status embed to pages your viewers may visit when they think service is affected by an incident or maintenance. The status embed gives visibility to your status page and makes status information accessible to any page viewer.

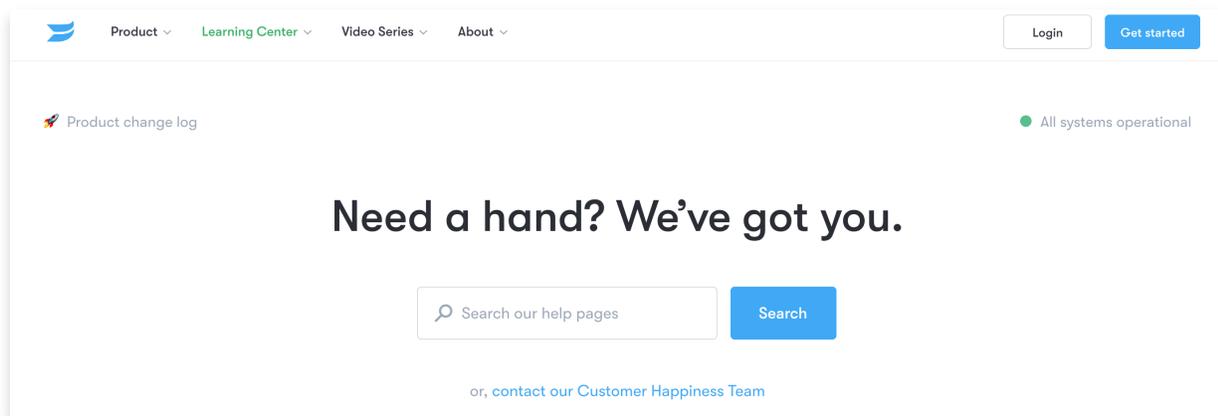
When you enter a new incident or maintenance to your status page, the status embed feature updates in real time, displaying the incident title and description directly on your webpage.



## Embedded status widget

With Statuspage, you can also custom [embed a live widget](#) into your app or website using an API. The widget reflects the current status of your service and links to your status page.

Here’s an example from Wistia (see the “All systems operational” and green button on the top right):



Here’s one from Harvard University’s IT team (see the “All systems operational” and green check in the middle):



## Social media bio

Consider adding a link to your status page in the bio section of your company's social media accounts. These are often the first places people look when they're curious about downtime.

Here's an example from Segment:



## Kickoff email

Many teams send a kickoff email to customers or end users educating them about how the company communicates incidents.

Common times to send this email:

- When new customers begin using the service
- When new teammates join the company
- When the company adds an incident communication channel, like a status page.

## CASE STUDY:

# How Symantec informs customers about their status page

This is an excerpt of a guest post from [Nick Coates](#), Principal ITSM Solutions Engineer at [Symantec](#).

As Symantec's Principal ITSM Solutions Engineer, I was tasked 2 years ago with finding a way to quickly and effectively communicate incidents with our customers after a 4-day outage left our customers in the dark. I found Statuspage and knew it was the right tool for the job, offering a single place to send real-time updates to our customers across various channels (email, SMS, Twitter, etc.)

But I quickly realized that creating a great status page and honing our incident communication plan would be useless if status updates weren't easily discoverable by our customers. Our team found success in marketing our status page like we'd market our product, and we want to help you do the same.

Read on for a few tried-and-true methods you can use to make sure customers know where to go for status information when they need it most.

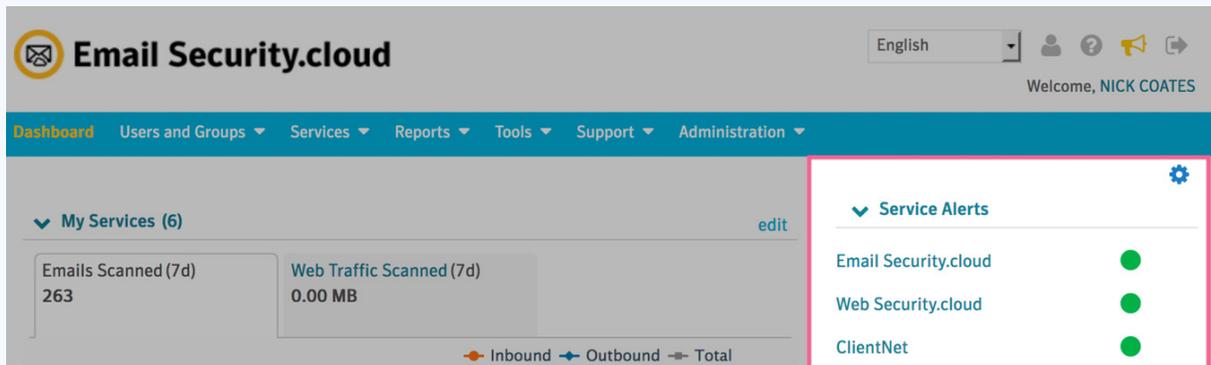
## Introduce your page on day one

After a customer has signed up for your service, do you send an automated email welcoming them with helpful tips and guides?

These emails are a great place to link your status page, directing customers to first visit the page if they suspect your product or service is unavailable, and to subscribe to receive notifications about the parts of your service they care about most.

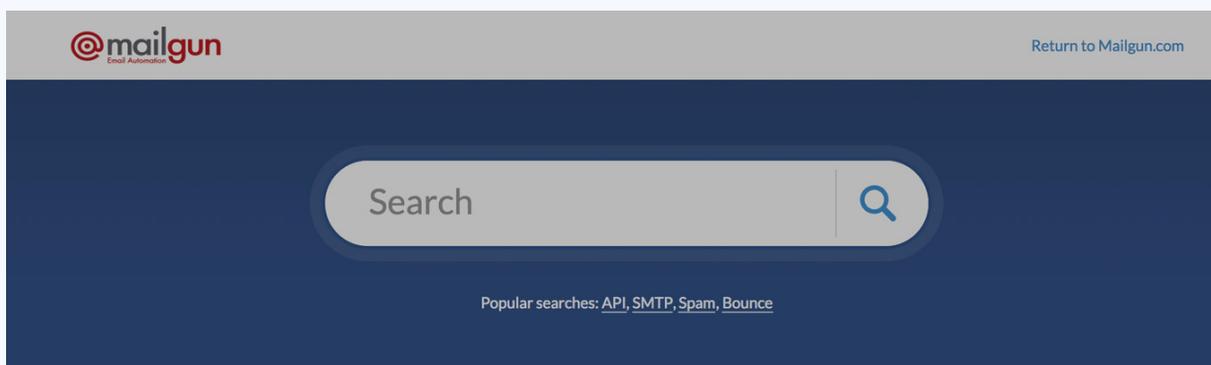
## Embed status where customers already are

We also found it helpful to embed our current status where customers already go when they suspect something isn't right using the status embed widget. Here, you can see that we embedded status on one of our customer portals:



Monzo, a bank in the UK, does this really well. They've used the Statuspage API to embed current status into their banking app (iOS and Android). Anytime there are issues with the service, customers see this in-app and can click through to Monzo's status page for more details.

Mailgun is another example of a company doing this well. They embed their status right on their help site for customers to see as soon as they navigate to the page:



## Promote through social media

Social media has become a mainstream support channel for many companies. Why not use it to broadcast outages, performance degradation, or scheduled maintenance?

Statuspage lets you easily hook up your page to your Twitter account in order to auto-populate tweets when incidents are created or updated.

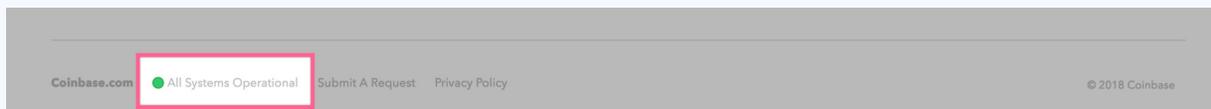


## Add links to your support portal

If you use a support ticketing system, it's helpful to have all support engineers add a status page link to their support email signatures. This helps urge customers to check the status page before filing a ticket the next time something goes wrong.

Most support ticketing tools also allow you to use canned messages or pre-defined templates. Save your support team time by building a template they can use to auto-populate tickets with context and links that direct customers to your status page during an incident.

Finally, you can use the embed widget to display current status on your ticketing portal to deflect duplicate tickets during downtime. Coinbase includes real-time system status [in the footer of their support site](#):

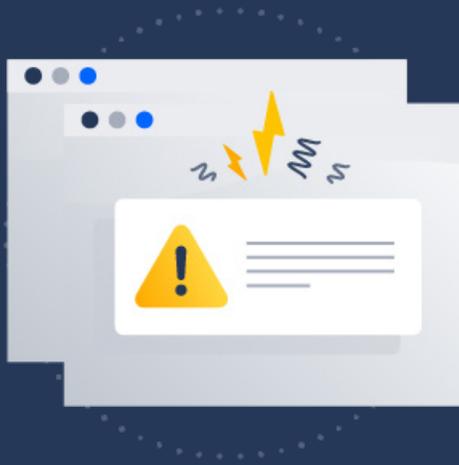


If you use Jira Service Desk, there's an out-of-the-box integration with Statuspage that lets you do this.

## Get creative

You have a lot of options when it comes to surfacing your status page to your customers, and at the end of the day, you know your customers best. Think about where they're already going during an incident, and meet them there. It's easier to surface updates where they already are than to try to completely change behavior.

A little extra time and thought put into marketing your page will go a long way in building customer trust and loyalty.



# 05

---

## Stages of incident communication

# Stages of incident communication

The lifecycle of an incident will likely include several points of contact. Done well, there's a familiar structure to an incident: first contact, updates during the incident, resolution, and postmortem.

Before we get into these parts, it's important to understand the different status names you'll encounter if you're using Statuspage. Incidents in Statuspage have one of four statuses. When creating or updating an incident, you will specify which of these states the incident is currently in. The incident message you send will include this status state.

## SSL Provisioning Unavailable

**Resolved** - This incident has been resolved.

Jan 28, 15:00 PST

**Monitoring** - A fix has been implemented and we are monitoring the results.

Jan 28, 09:47 PST

**Identified** - The issue has been identified and a fix is being implemented

Jan 28, 09:27 PST

**Investigating** - We are currently investigating an issue with our load-balancers which is causing problems to SSL provisioning pipelines.

Jan 28, 09:15 PST

## The four statuses in Statuspage are:

- Investigating - you are seeing the symptoms of an issue but are unaware of what the root cause is.
- Identified - you have identified the incident and are working on a fix.
- Monitoring - you believe you have successfully fixed the issue and are waiting for the symptoms to subside.
- Resolved - you're confident the incident is over and your systems are back to 100% performance.

You may end up skipping a step. For example, if you've identified and confirmed an incident before updating the status page, you would start with "Identified" as your first status. Or you may post several updates in a row using the same status.

Now, here are how these statuses fit into the four parts of an incident.

## Part 1: First contact

The initial update is the most important. Everything, from what you say to how and when you say it, sets the tone for how your response will be perceived. This is where it really helps to have a template set up ahead of time.

Your goal should be to quickly acknowledge the issue, briefly summarize the known impact, promise further updates and, if you're able, alleviate any concerns about security or data loss. It's important to acknowledge there's an issue, even if you don't know the exact details yet.

In Statuspage, you would mark the status at this stage as investigating (if your team is still working to confirm there is an actual incident) or identified (if you already have confirmed there is an incident).

## Part 2: Regular updates during the incident

Mid-incident communication is critical.

Teams often make the mistake of waiting until they have something concrete to report before updating customers again. Sure, it's good to update customers during a key development (incident resolved for partial users, for example). But don't wait until then if nothing's happening and the incident is dragging on. It's okay to update customers to tell them that nothing has changed and you're still working on the incident. This builds trust and confidence and reassures users that you didn't forget about the incident. How often should you send updates like this? There's no one-size-fits-all answer. It depends on the service and the incident. But we've seen many standardize providing updates every 30 or 45 minutes.

Another good practice here: make it clear in your updates when the next update will be. For example:

“We’re still working on restoring company email. We’ll update this page again in 30 minutes or as soon as anything changes.”

The SRE teams at Google list Communication Lead as one of the key roles someone should oversee during an incident.

From [Google’s book Site Reliability Engineering](#) on the role of Communication lead:

“This person is the public face of the incident response task force. Their duties most definitely include issuing periodic updates to the incident response team and stakeholders (usually via email), and may extend to tasks such as keeping the incident document accurate and up to date.”

This person will also be in charge of continuing to update the status page or post updates to other channels as the situation evolves. Even an updating saying “We’re still working on the problem, nothing new to report.” is better than saying nothing and leaving your audiences hanging. People left in the dark start to expect the worst.

In Statuspage, you would mark the status at this stage as investigating (if your team is still working to confirm if there is an actual incident), identified (if you already know there is an incident), or monitoring (if you’ve implemented a fix and are waiting for confirmation the incident is over).

## Part 3: Resolution

When you're confident the incident is over and the service has returned to operating as expected. Impact on customers is over. In Statuspage, you would mark the status at this stage as resolved.

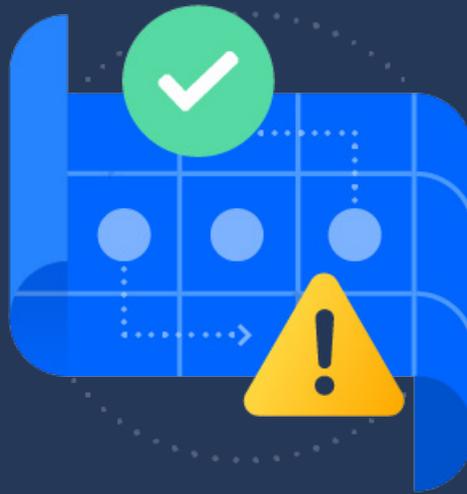
## Part 4: Post-mortem

An incident postmortem brings people together to discuss the details of an incident: why it happened, its impact, what actions were taken to mitigate it and resolve it, and what should be done to prevent it from happening again.

You may also decide to publish takeaways from your incident postmortem with customers or the rest of your organization in the form of a public postmortem. This can go a long way in rebuilding confidence in your service and team. Other customers and teams in your organization, especially leadership, may need to see the details of the problem and what steps were taken to resolve it to head off anyone second-guessing your team in the future.

Partners, customers, and end users may also want to know what happened and what steps you have taken to improve their experience. Making your incident postmortem available on your public-facing website may not be appropriate in all cases, but your marketing or public relations team can help you craft the language so people get the information in a way that is informative and builds trust in your services.

In Statuspage, you have the option to publish a public postmortem to your page.



# 06

## Incident templates

# Incident templates

Communication templates are one of the most helpful tools during an incident. In the heat of a service outage, the response team is under a lot of pressure and every second counts. Sitting down to a blank page to figure out how to update customers is a lot harder than it seems. Templates for common incident updates can be created and saved inside of Statuspage.

We recommend using some boilerplate language to get started.

Be sure to update the placeholder sections before posting. And don't be afraid to add, edit, or delete pieces of these copy to make them yours.



**Detect**  
We know about an issue before our customers do.



**Respond**  
Escalate and communicate, early and often.



**Recover**  
S#\*t happens. Clean it up quickly.



**Learn**  
Blameless accountability with post-incident reviews.



**Improve**  
Never have the same incident twice.

## The incident communication templates we use

Here are the incident communication templates we use at Atlassian, pulled straight from our company [Incident Management Handbook](#). We use separate status pages for internal staff (private pages) and external customers (public pages).

Feel free to copy any of these verbatim or modify to suit your needs:

## Public status page

- **Incident name:** Investigating issues with <product>
- **Message:** We are investigating issues with <product> and will provide updates here soon.

## Internal status page

- **Incident name:** <Incident issue key> - <Severity> - <Incident summary>
- **Message:** We are investigating an incident affecting <product x>, <product y>, and <product z>. We will provide updates via email and Statuspage shortly.

Here are some more templates you could utilize that are specific to the different stages of an incident:

## Investigating a potential outage

Here's an example incident communications template you can use to announce you're investigating a potential outage, but haven't confirmed the impact or scope.

- **Incident name:** Investigating issues with <product>
- **Message:** We're currently investigating reports of a potential service interruption with <impacted services>. We apologize for any inconvenience and will post another update as soon as we learn more.

## Full outage

Here's an example incident communications template you can use during a complete outage.

- **Incident name:** <Product> service outage
- **Message:** We're experiencing a service outage with <impacted services>. Our team is currently working to restore the service. We apologize for any inconvenience. <General impact> users may be affected. Next update in <time until next update>.

## Partial outage

Here's an example incident communications template you can use during a partial outage.

- **Incident name:** <Product> partial outage
- **Message:** We're currently experiencing degraded performance issues with <impacted services>. Our team is currently working to restore normal performance levels. We apologize for any inconvenience. <General impact> users may be affected. Next update in <time until next update>.

## Scheduled maintenance

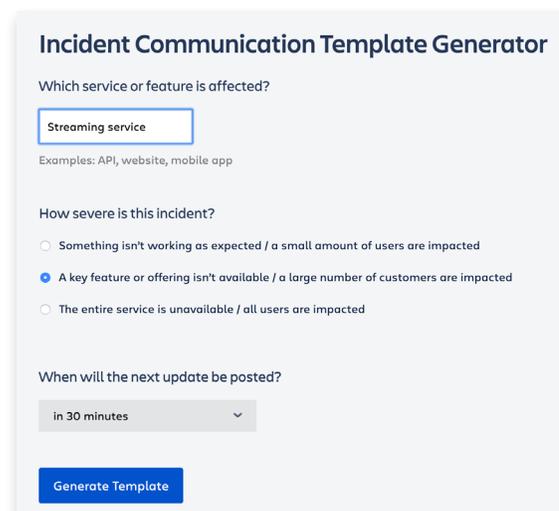
Here's an example incident communications template you can use during a scheduled maintenance window.

- **Scheduled maintenance:** We will be performing scheduled maintenance on <service name> on <Date, start time-end time>. During this time, users can expect <expected impact>.

## Use our free template generator tool

We created a free online tool, called the Incident Communication Template Generator, to help teams write clear and concise incident messages in less than a minute.

Find it at: [www.atlassian.com/incident-management/template-generator](http://www.atlassian.com/incident-management/template-generator)



The screenshot shows a web form titled "Incident Communication Template Generator". It contains three sections: 1. "Which service or feature is affected?" with a text input field containing "Streaming service" and a list of examples: "API, website, mobile app". 2. "How severe is this incident?" with three radio button options: "Something isn't working as expected / a small amount of users are impacted", "A key feature or offering isn't available / a large number of customers are impacted" (which is selected), and "The entire service is unavailable / all users are impacted". 3. "When will the next update be posted?" with a dropdown menu showing "in 30 minutes". At the bottom is a blue "Generate Template" button.



# 07

---

## Real-world examples and case studies

# Real-world examples and case studies

We've seen a lot of status pages over the years. Everything from scrappy DIY pages for side projects to totally bespoke plans for global corporations has crossed our monitors in one way or another.

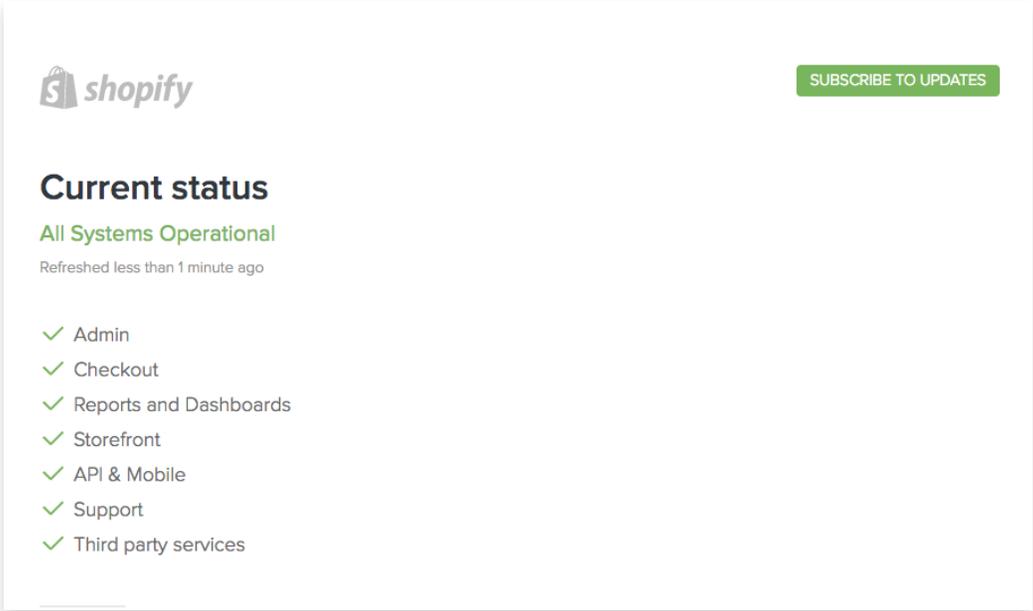
Here are some status pages that showcase some of the excellent design, planning, integrations, and creative thinking that go into [incident communication](#).

## Shopify

Status page: <https://status.shopify.com/>

What they do: Shopify is a leading e-commerce platform helping people around the world buy and sell online.

What's great about their page: Shopify added some custom design elements and a clean list of components. With such a large user base and complex product, they do a good job keeping their page orderly and easy to understand.



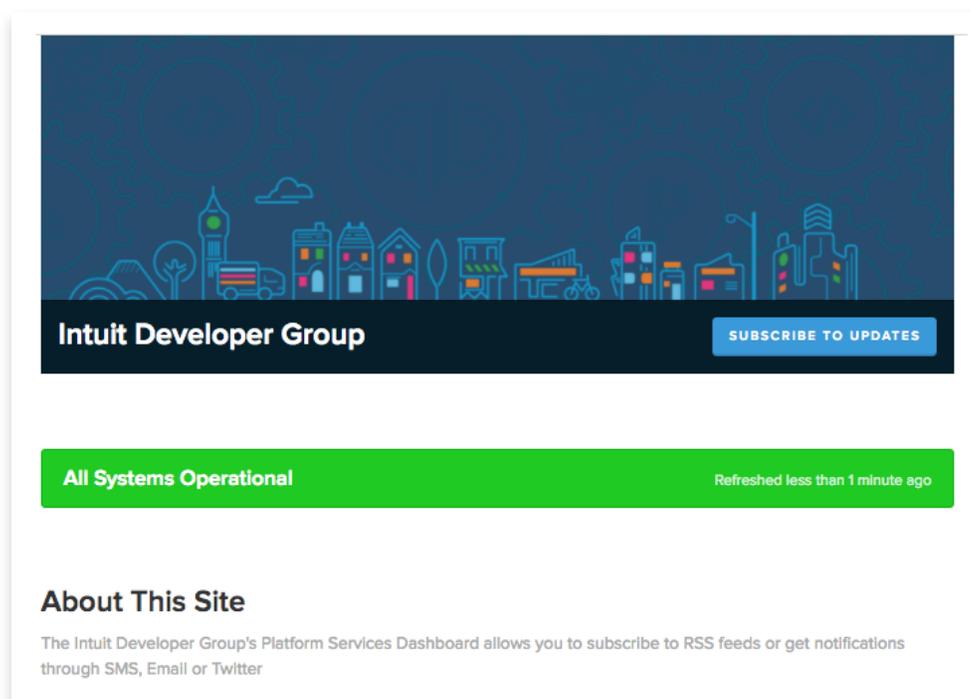
# Intuit

Status page: <http://status.developer.intuit.com/>

**What they do:** Intuit develops some of the world's most recognized financial software, including tools like TurboTax, QuickBooks, and Mint.

**What's great about their page:** Intuit hosts a robust developer program, empowering developers around the world to build products on top of QuickBooks APIs.

This status page is specifically for the developer program, communicating system status for their APIs.

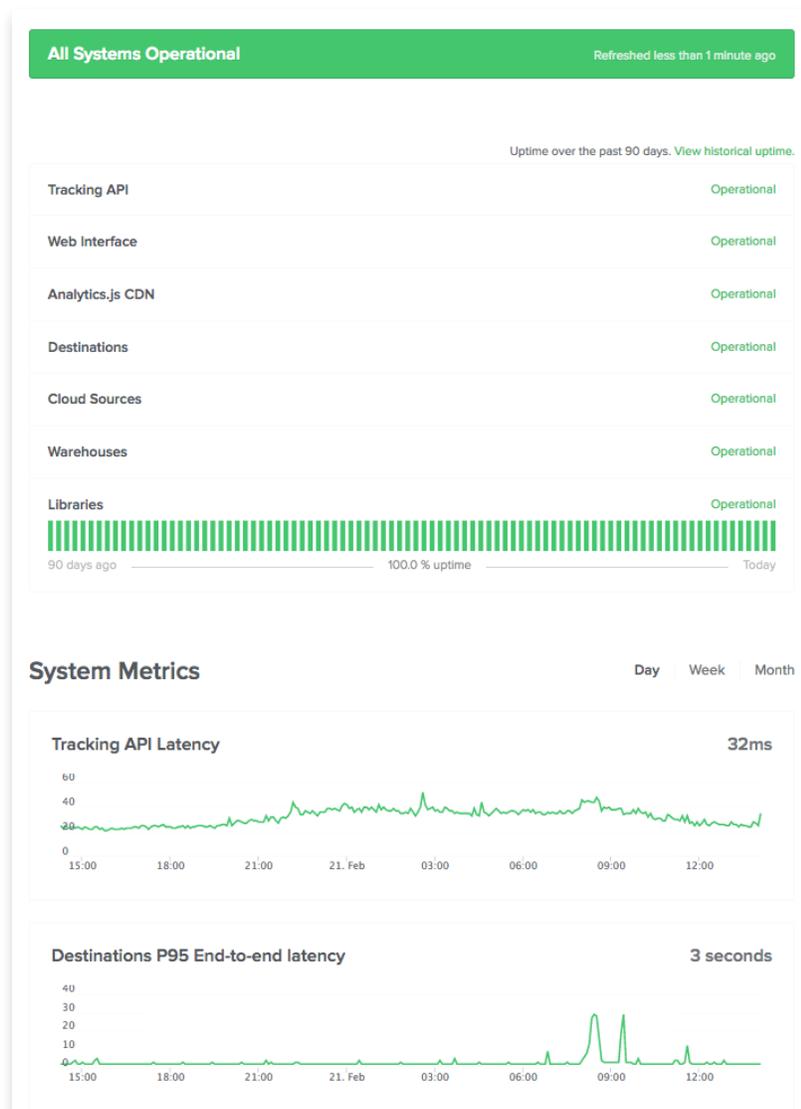


# Segment

Status page: <https://status.segment.com/>

What they do: Segment provides analytics software to help teams manage and measure customer data.

What's great about their page: The Segment status page brings in great data visualizations, including an Uptime Calendar showcasing historic availability and API latency metrics.



## CASE STUDY:

# How Duo Security uses Statuspage to keep thousands of customers in the loop

At Statuspage, we've come across this question a lot.

“I've got my users on all these different deployments. How do I let one group know about an outage without alarming all the users on different servers who aren't affected?”

It's a good question. We talked with the team at [Duo Security](#) and learned about how they're solving this problem.

Duo Security provides two-factor authentication and other security services for thousands of companies and millions of end users. Teams at Facebook, NASA, Yelp, and many more top companies count on Duo to keep their IT secure.

Launched in 2010, Duo puts a lot of effort into what security means for teams that are using cloud tools and working remotely. As a security service hosted in the cloud, Duo's system status is extremely critical to their customers. When incidents occur, their customers need clear, correct, and immediate updates.

“These are the kind of things we want to communicate as quickly and efficiently as possible. If our product goes down it can affect our users' ability to do work,” said Rory McDowell, Technical Support Engineer at Duo.

## Taking advantage of component groups

Duo hosts its applications for customers in 46 (and growing) customer deployments. If there's a problem with one of those 46 deployments, it's critical to get updates to the corresponding customers. At the same time, Duo wants to be careful to not alert customers in other deployments who aren't impacted by the incident.

Rory and his team solve for this by using Statuspage's Component Groups feature. Component Groups allow users to group relevant sub-components together, keeping their page clean and easily digestible for end users.

For Duo, this meant treating creating a Component Group for every deployment. You'll see on their status page that these groups are labeled DUO1 through DUO46.

Under each Component Group are five components that respond to that deployment:

- Core Authentication Service
- Admin Panel
- Push Delivery
- Phone Call Delivery
- SMS Message Delivery

Duo customers can see which group they fall under by logging into their Duo admin panel.

Duo uses the Statuspage API to automatically subscribe new customers to get Statuspage updates for their corresponding Component Group.

The approach has cut down on support tickets during downtime because users are getting the updates from Statuspage. Support agents at Duo also have a pre-written response loaded up in their help desk tool, pointing users to the status page during downtime.

## Twitter integration

Duo also does a great job using the Statuspage Twitter integration to post status updates directly to their [@duohelp](#) Twitter account.

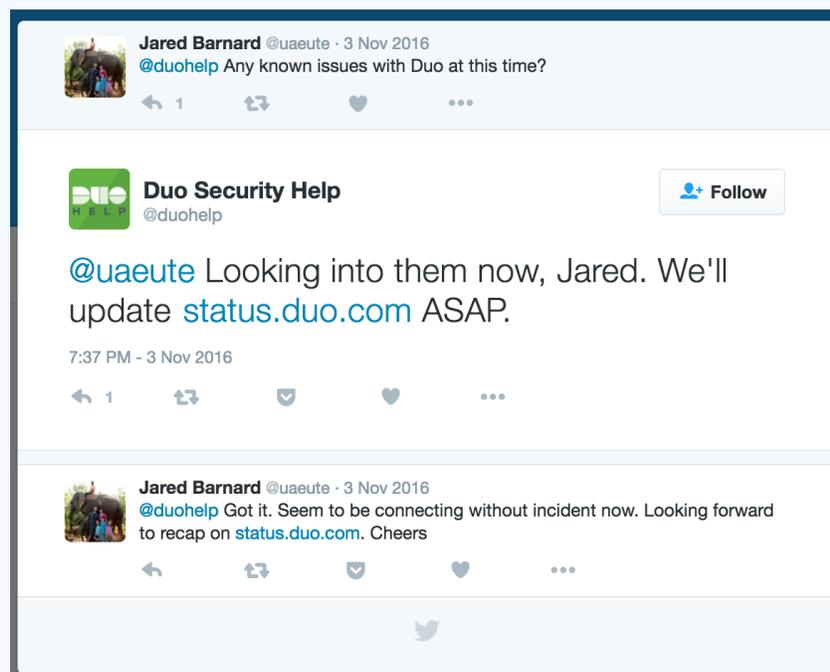
And they aren't just posting updates to the page. You can see they actively monitor the page and use it to answer customer questions and talk to their followers.

Before Statuspage, Duo was simply manually emailing groups of customers during outages.

“This was back when we were a really small startup,” Rory said. “As we grew, we knew we needed to be more efficient.”

The team briefly considered building something itself, but ultimately decided it would be easier and more efficient to use Statuspage.

“Building this kind of thing internally would take up a lot of resources,” Rory said. “It's a great solution for a company that's trying to stay agile and wears multiple hats at once.”



## CASE STUDY:

# How an outage helped Wistia get on board with Statuspage

*The following was written by Jordan Munson, Support Engineer at Wistia, originally published in 2017.*

What do you do when your software is experiencing a critical outage? Post an update to your status page, send out some updates via social, answer emails and calls that come in about it, etc. It all seems pretty obvious what to do in 2017, but for Wistia in 2013, things weren't so clear. A handful of months into my tenure at Wistia, we faced what is still likely the biggest service outage in our company's history. We were not ready, plain and simple.

The Wistia application is effectively three different, connected services: the application portal, the infrastructure that collects/creates stats, and the infrastructure that encodes and hosts our videos. In normal growing-pains fashion, we realized that we were going to outgrow a portion of the stats infrastructure in a few months. Not a big deal, that window was far out; we made the necessary changes to our infrastructure and moved along with the other projects on our plate. Fast forward a few months to one fateful Friday afternoon, one of the engineers gets a page stating that our stats database is no longer being written to. Uh-oh.

As it turns out, we forgot something. A very specific, critical something. The result was our stats infrastructure grinding to a halt. Fortunately it's a modularized system, so we were still collecting data, but we weren't turning that data into things you could see in your account. Over the weekend, the engineering team repaired the issue, but we still had a massive backlog of collected stats that still needed to be processed. For the next two weeks, our stats were behind real time for customers. During this time we were almost all hands on deck covering the work in our support inbox as folks wrote in wondering what the deal was. We were completely swamped.

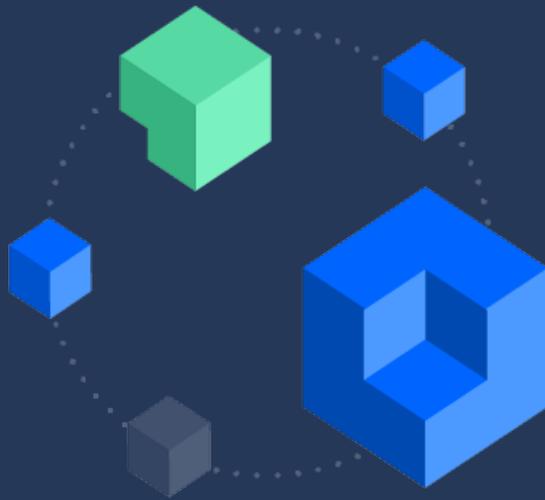
"I think we need a status page."

We realized very quickly that having a status page would have made our lives much, much easier. We thought “We could just build one of those really fast, that should do just fine!” And so we did, and it was fine. We called it “Bugle” because it allowed us to sound the alarm (we’re big fans of clever naming schemes). The problem, however, was that this basically meant we were supporting a new product in addition to our normal work. This might not have been a huge deal for a larger company, but at the time Wistia was roughly 20 people. Supporting another product was simply not a cost we could afford.

After a handful of months of mild frustration around our nearly featureless, but helpful, homegrown solution, we decided we needed something more, something that didn’t require so much tending to. Enter Statuspage. Since the move to Statuspage, we’ve been able to do what we were looking to do along – quickly and easily keeping our customers up to date on the status of our application. It only took one massive outage and building a new product to get there.

Fast forward a couple years to modern times, and our process looks way smoother. Folks get updates from us directly when there’s an outage, they know where to look for updates, and updates made to our status page push directly to a number of places (like Slack, for instance). We’re not impervious to outages here and there, though, especially considering how many services we rely on for critical parts of our business. One of these services is Amazon Web Services (AWS). Recently (the end of February, 2017), AWS was experiencing some serious troubles in their east coast region, which so happens to be the busiest region for us. We were experiencing some serious troubles as a result of this, creating a partial outage for a number of our customers. We faced a similar fate when our DNS provider was slammed with a Distributed Denial of Service attack, and even when our web service was completely inaccessible, things went more or less okay (as well as you could expect, considering our web portal was entirely inaccessible for most customers).

Over the last handful of years, we’ve learned many lessons the hard way. Things can go wrong, and they will go wrong – you should probably be ready for that. Thank goodness we are now!



# 08

---

## Integrations and features for leveling up your incident communication

# Integrations and features for leveling up your incident communication

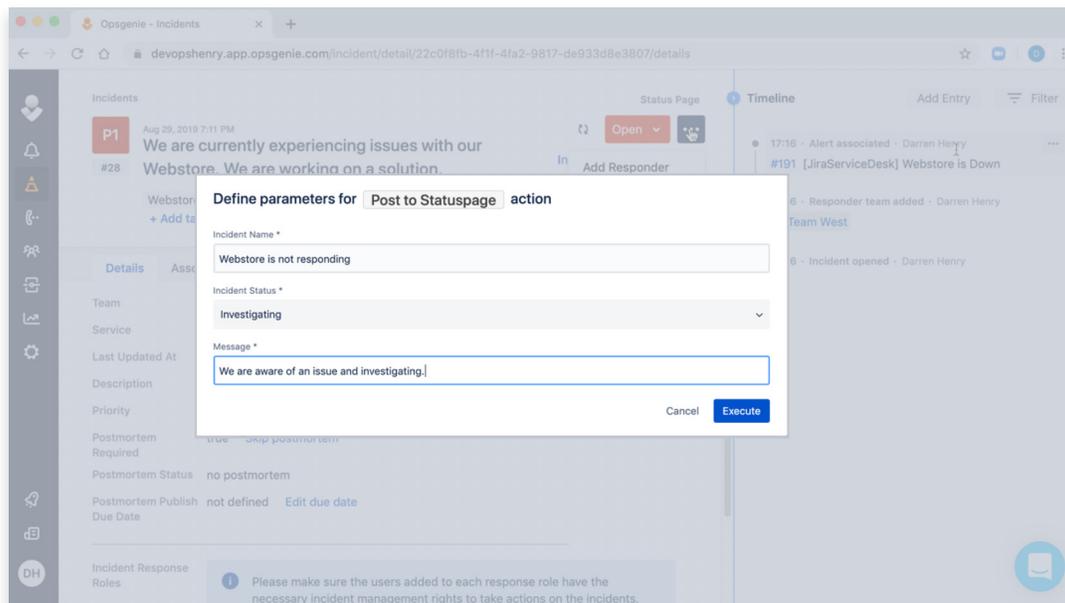
Incident communication is easier to do well (and harder to forget) when your status page is integrated with other incident response tools.

Here are some of our favorite integrations and features for leveling up your incident communication workflow.

## INTEGRATIONS

### Opsgenie

With Opsgenie, teams can post status page updates directly from an incident in Opsgenie.



# Jira Service Desk

Help desk integrations, like the Jira Service Desk integration, put your incident status right at the front door of your ticketing system. Embed a status where your customers are already heading to help squash duplicate support tickets at the source.

## Create an Incident

**Incident Name**

This incident will be posted to page "Teams in Space"

**Incident Status**

Investigating  Identified  Monitoring  Resolved

**Message**

Help center

## Status update

Parts of the website are slower than normal. [View StatusPage](#)

### Teams in Space Support

Welcome! You can raise a service request from the options provided.

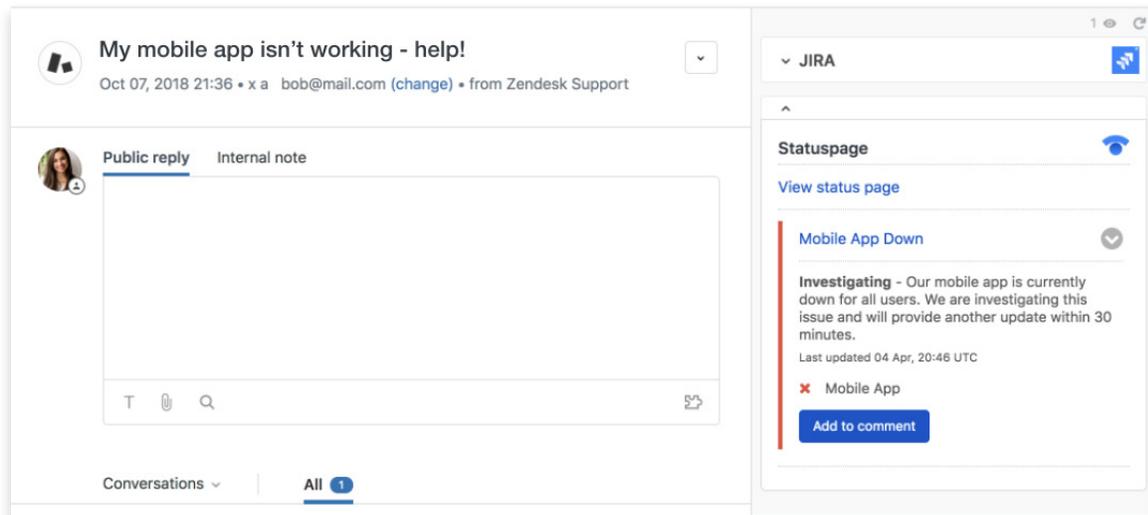
 **Get help**  
Get assistance for general problems and questions

 **Report a system problem**  
Having trouble with a system?

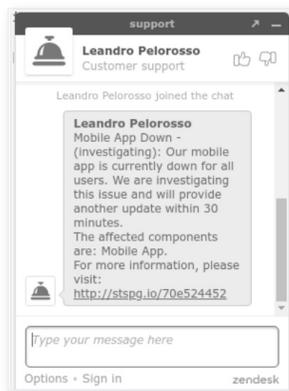
 **Request a new account**  
Request a new account for an internal system

# Zendesk

With the Statuspage and Zendesk Support integration, you'll see a Statuspage icon in the top right nav bar of your Zendesk Support workspace. When an active incident or scheduled maintenance has been posted to your status page, it will be indicated with a badge displaying the number of open incidents/maintenances. Support agents can click into this icon for more details about a given incident, and will have the option to navigate to the overall status page or the specific incident posting for more details.

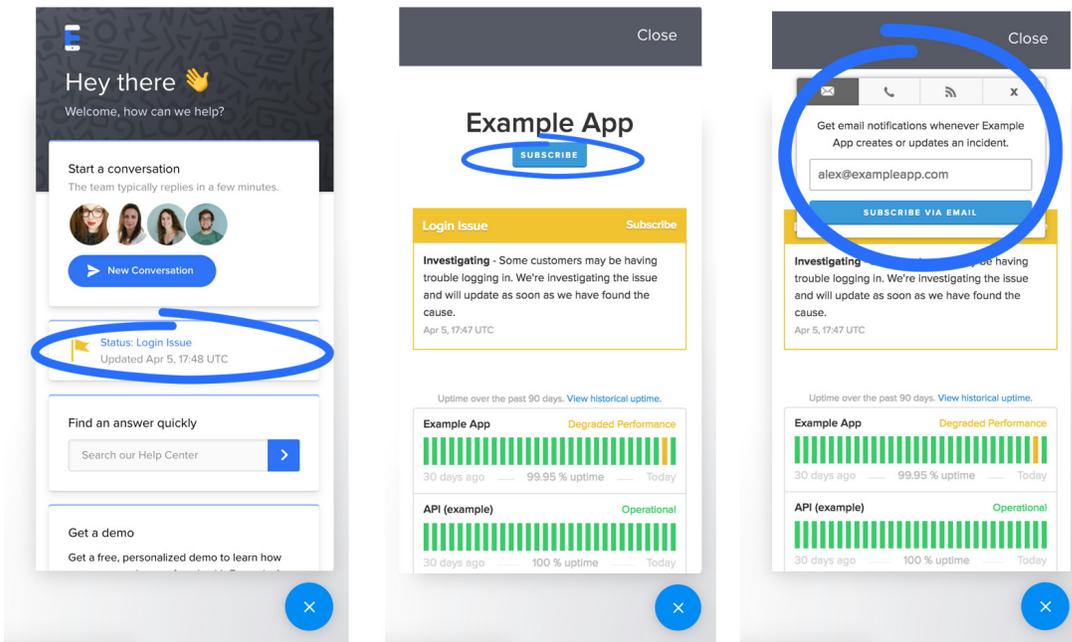


If you also use Zendesk Chat to communicate with customers, you can pipe the current status directly into your chat widget. Similar to the “Add to comment” button above, agents will be able to do this with an “Add to chat” button. Customers can click over to the status page to get more information and subscribe to updates to stay in the loop all the way through resolution.



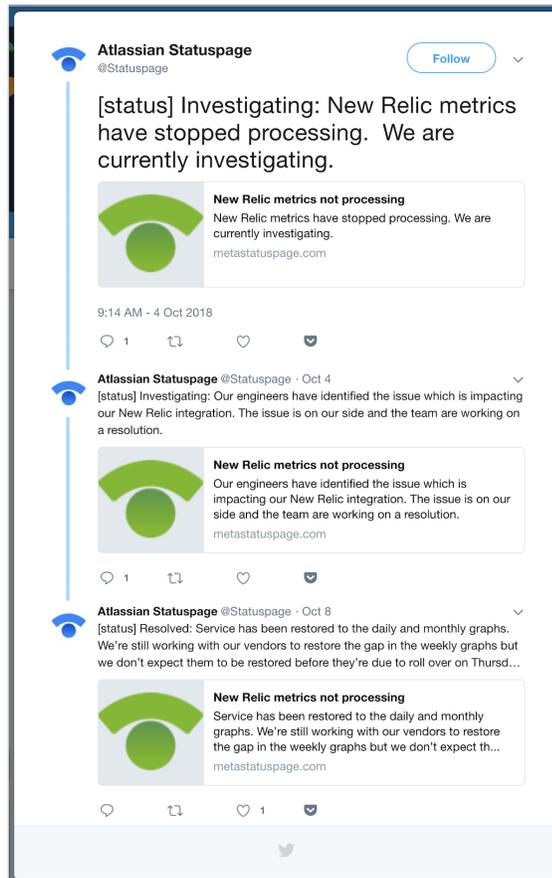
# Intercom

With the Statuspage app for Intercom, you can proactively update your customers on incidents from within the Messenger, before they start a conversation – meaning they get the answers they’re looking for quickly, and your team experiences less of a surge in chats asking “Is the app down?”



# Twitter

Already use Twitter for customer communication? Automate Twitter updates directly from Statuspage with the Statuspage Twitter integration.



# Slack

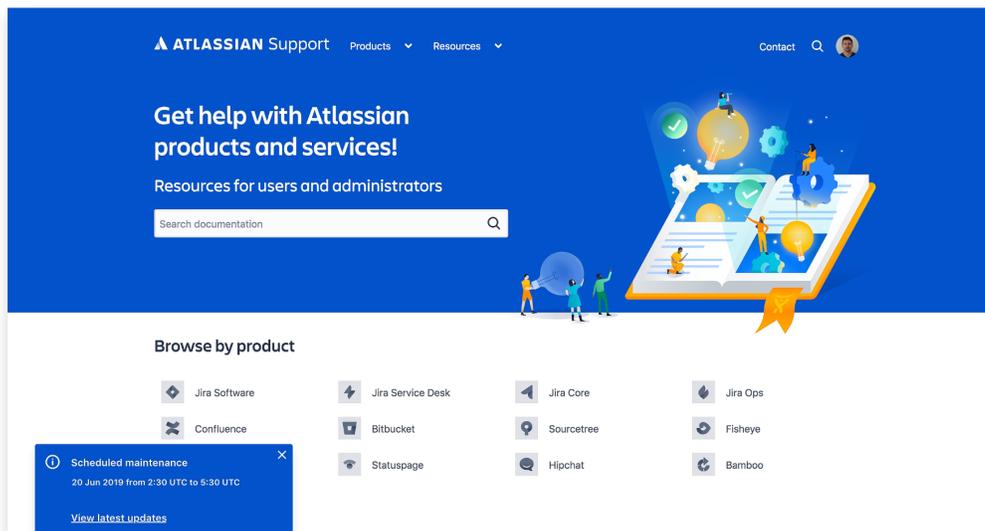
Chat often becomes the central command center during incident response. Use our Slack integration to pipe status updates into designated Slack channels to keep everyone on the same page.



# FEATURES

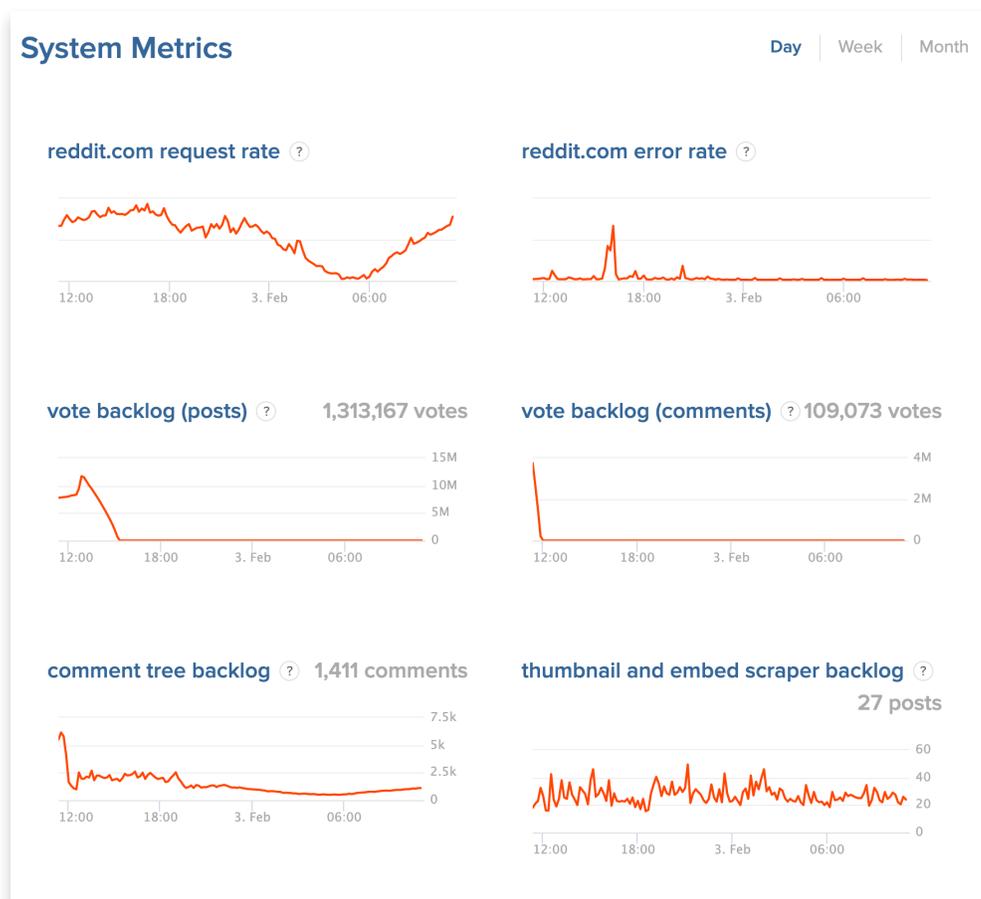
## Status Embed

With Status Embed, you can display active incidents and scheduled maintenance in the places your customers are most likely to see it. Choose custom positioning and color for embeddable widgets, then copy and paste autogenerated code into your website, app, or help center.



## Display system metrics

Seeing is believing. Show current and prospective customers how reliable your services are with system metrics. Connect your page with your monitoring tool to display metrics right on your page. You can also create custom metrics through an API.



## Third-party components

Third-party components are components that you can add to your page that reflect the status of external services. If your service is heavily dependent on a third-party component, you can add the component to your page and the status of that component will automatically be updated.



# 09

---

Should you automate  
your status page?

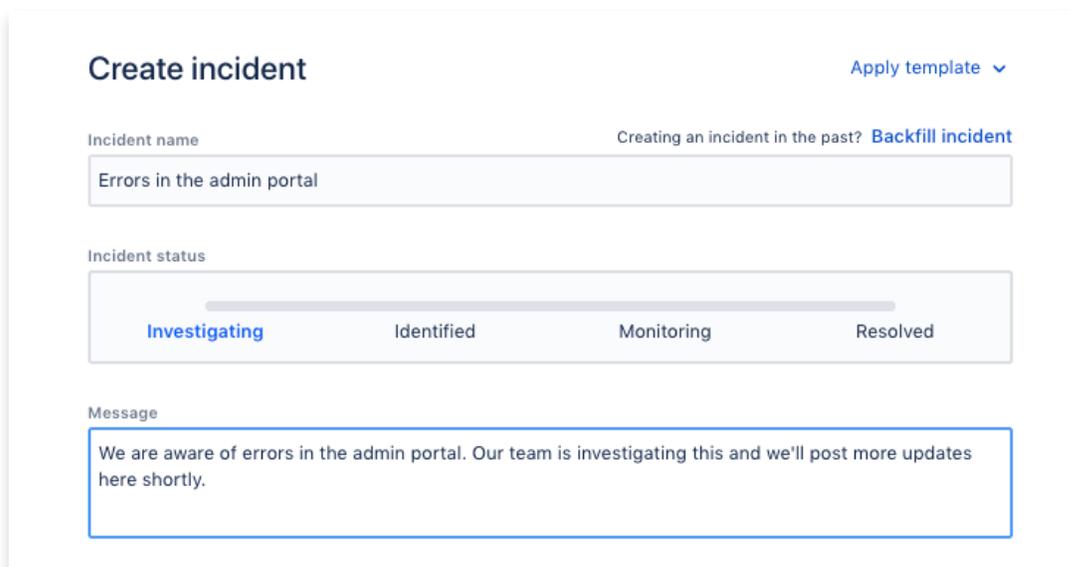
# Should you automate your status page?

Some people confuse Statuspage for a monitoring tool, but that's not really what we do. Statuspage doesn't ping your servers or endpoints to check the status, but we do integrate with several monitoring and alerting tools, which allows you to automate parts of your status page if you want.

## Statuspage is a communication tool

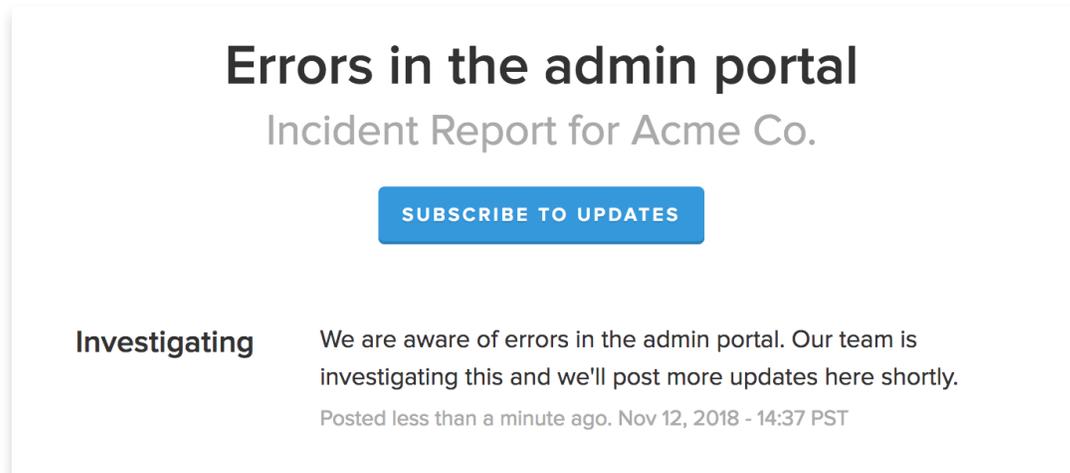
We built Statuspage with one goal in mind: make it easier for teams to communicate with their customers during incidents. When things go wrong, we help you get the message out to the people that matter most – your customers and users. The way most people do that is by dedicating a team to manage their status page.

When an incident occurs, that team updates the status page to let affected users know.



The screenshot shows the 'Create incident' form in Statuspage. At the top left is the title 'Create incident' and at the top right is a link 'Apply template' with a dropdown arrow. Below the title is a text input field for 'Incident name' containing 'Errors in the admin portal'. To the right of this field is a link 'Creating an incident in the past? Backfill incident'. Below the incident name field is a section for 'Incident status' with a progress bar and four buttons: 'Investigating' (highlighted in blue), 'Identified', 'Monitoring', and 'Resolved'. At the bottom is a 'Message' text area containing the text: 'We are aware of errors in the admin portal. Our team is investigating this and we'll post more updates here shortly.'

Here's what that looks like on the status page:



**Errors in the admin portal**  
Incident Report for Acme Co.

[SUBSCRIBE TO UPDATES](#)

**Investigating** We are aware of errors in the admin portal. Our team is investigating this and we'll post more updates here shortly.  
Posted less than a minute ago. Nov 12, 2018 - 14:37 PST

## Human-driven vs. automated

We're big believers in maintaining a human touch with incident communication. A fully automated status page means you have computers talking to people. That's risky, as it can result in a poor user experience by way of false negatives, false positives, lack of context, and a flood of notifications. As a best practice, we rarely recommend fully automating your status page. There are ways you can take a hybrid manual/automated approach, which we'll discuss more below.

You can think of Statuspage as a replacement for using a mass email marketing tool to do your incident communication. Would you set up an automatic email to go to all your customers every time your monitoring tool fired an alert? Likely not. Statuspage provides a single place to communicate with your users during an incident. Your messages are published to your status page and they can be sent out as email or text message notifications to your users (if they opt in), and embedded directly in your own interface or website.

## When to automate your status page

There are some cases where it might make sense to automate your status page.

- Maybe you're using your status page on a smaller scale, to keep your immediate team in the loop on the health of your services. An automatic incident notification might make sense there.
- Or maybe you want to programmatically update components based on monitors you have set up and then use Statuspage as a health/status dashboard.
- Or maybe you want to publish an incident to your status page once you acknowledge an alert in a different tool.

Whatever the case, you can automate your status page a few different ways.

## Ways you can automate your status page

When people ask about automating their status page, they're typically looking to automatically update the status of their components, and/or automatically publish incidents on their status page. There are several ways you can go about doing both.

### Automating your status page using third-party components

Third-party components allow you to display the system status of other services you rely on directly on your own status page. So, for example, if your infrastructure relies heavily on the Sendgrid API, you can enable the Sendgrid third-party component to display their API status on your page, and it's automatically updated when they change the status on their page. You can override the status if need be. We have a large (and growing) network of third-party components you can enable on your page.

## Automating your status page using an integration

There are several monitoring and alerting tools that can be used to automatically update your status page. We've included a comprehensive list below:

Integration	Automation options
Opsgenie	Automatically create incidents, automatically update component statuses
PagerDuty	Automatically create incidents, automatically update component statuses
VictorOps	Automatically create incidents, automatically update component statuses
xMatters	Automatically create incidents, automatically update component statuses
New Relic	Automatically update component statuses
Pingdom	Automatically update component statuses
Pingometer	Automatically update component statuses

## Automating your status page using the Manage API

You can use our API to programmatically update your status page by writing your own integration. In addition to being able to update your page's components and incidents, you can create subscribers, metrics, and much more with the Manage API.

## **Automating your status page using email automation**

If none of the above automation options are a good solution for you, you might want to look into using email automation. This is another way to automatically update the status of your components, and it's done by triggering an email with a subject line containing the word "Up" or "Down" to be sent to a unique email address, which we provide.

We really recommend using one of the integrations or our API for automating your page. We do everything we can do make sure these emails get to the right place at the right time, but the very nature of email is far less reliable than making an API call.



# 10

---

## Tools, training, and other resources

# Tools, training, and other resources

We hope the material here has been helpful. While content, templates, and examples can get your team pointed the right direction, they're only a small part of the bigger picture.

The best incident response teams continuously build and refine their plan. Here are some more resources to help you plan, build, and constantly improve your incident communication journey:

**Atlassian's Incident Management handbook shows the proven and battle-tested incident management process we use at Atlassian.**

[www.atlassian.com/incident-management/get-the-handbook](http://www.atlassian.com/incident-management/get-the-handbook)

**Our Incident Communication Template Generator can help you quickly draft updates during an incident.**

[www.atlassian.com/incident-management/template-generator](http://www.atlassian.com/incident-management/template-generator)

**Get your sheet together: how to create an incident communication plan**

[www.atlassian.com/blog/statuspage/how-to-create-an-incident-communication-plan](http://www.atlassian.com/blog/statuspage/how-to-create-an-incident-communication-plan)

**And check out a lot more incident management best practices and tutorials at [Atlassian.com/incident-management](http://Atlassian.com/incident-management)**

🔍 **Want to dig deeper?**

[www.atlassian.com](http://www.atlassian.com)

📧 **Have questions?**

Contact us at [sales@atlassian.com](mailto:sales@atlassian.com)





 **ATLASSIAN**