

Krish (products.snowpal.com) (00:00.974)

Hey there, hope you're doing well. Welcome to Snowpal software development and architecture podcast. In this podcast, we're gonna take a look at one of Snowpal APIs, the status API. We have a number of APIs and we've done a few podcasts that relate to the other APIs. This is one of those APIs where we don't have a user interface to share with you.

For some of the other ones, we have actually built production apps, web apps and mobile apps. So we could socialize that with you. But this is one of those APIs. Again, as our API suite grows, we're not going to be able to have full -fledged examples, at least, of all of these. So we provide recipes, SDK, and every other way to help you out so you can consume these APIs. This status API, let me actually share my screen so we can take a look at it to get a bit more familiar.

You can go to `status -api .snowpal .com` to check out our API Postman collection. Or you can also go to `developers .snowpal .com` and check out the product API documentation here. If you're a product owner or a product manager, you're in the product team, this might be a better place to start before you go to Postman. But we've seen that developers tend to want to, you know, take a look at the Postman collections directly. So again, to each our own.

and we just have support for both of them so you can pick what works best for you. Let's take a look at some of these endpoints and what is the purpose of this? So our web app and mobile apps have a lot of functionality to manage projects, but we don't have an integration for statuses. So we used to manage statuses, we still do outside of our app. And then we realize, why not build an API so we can actually create an interface at a feature.

to consume my API at some point in our application as well. While we haven't done that yet, the idea is to actually create a robust API much like our other APIs. So anybody else who wants to build any system can actually make status management part of their functionality, which is very helpful because we think about it, you go to these scrum calls, you use different tools to manage projects.

Krish (products.snowpal.com) (02:15.822)

But you're not actually tracking your your status people have seen them do send emails or share it on slack and do one of many ways but not in a very optimal methodical fashion the idea behind this API is to actually make that a possibility For whether you're building a microservice or a web or a mobile app you can use this to implement status functionally now How far you go you could build full -blown status applications using the API at least that's the idea?

But if you didn't want to do that and you want to add just add a support for your API for your existing applications or the new apps you're going to build, you can use some of these endpoints to provide some of the functionality. But let's take a look at what kind of features it supports. You can, you know, there's two facets to it, the highest level. Let's draw this out. So it's actually pictorial. So I want to say status API. The first,

Oops, thing we support is you can report status, but you can report status either as the user, the member. Let's say if I'm a developer, I'm reporting status as user. I can do that as that particular user or the team member. Who status it is are, I can actually report status and an admin can jot it down because we've seen that this happens different ways in different teams. Some teams actually...

how people report statuses directly or update it someplace. Other teams actually have an admin do it in the call. By admin, I mean a project manager or a dev manager do it. So our endpoints support both. So wherever you see as admin, it means you're performing the same

action, but you're actually doing it for a team member. So that's fundamentally the difference. So you'll have actually a member ID in the URL.

alongside the team ID because you know we again have support you can report statuses a member can report status for as part of a team if that member is working part-time in multiple teams they can report status once for each of those teams or the project manager for each of those teams can actually take status update for Krish for each of those teams and use the tool whether it's a app a web app or let's say you build a slack app for this I think it's slack

Krish (products.snowpal.com) (04:38.126)

is probably one of the best apps you could potentially build. There are some Slack apps for status reporting, but they fall well short of what actually is truly needed. So you can build something much better to satisfy the needs of your users, right? So that's simply an example of how you could possibly use this. I have, I guess the sun, there's a shadow. I'm sorry, I should have closed my blinds. My apologies. I didn't do that. Okay. So that's as a user and as an admin. So what does the status comprise?

off. There are three paths to it. So let's say I'm going to pick a different, maybe pick a circle here. Give it a different color. You know, a typical status from what we have seen, at least the one that we've tried different permutations, combinations, this is what works for us. Plan for today. What do you plan to do at the top of the day? You want to share it? Maybe you have a scrum call. Maybe you do this on Slack, however you do it.

And then you say, we have sessions at Snowpal, each session is four hours, if you're working full time, you have two sessions per day, 10 per week. And if you're part-timer, you have one session per day. Sometimes your sessions can be less than four hours as well, depending on the type of engagement you have with us. So let's say there is a session regardless, which is the length of time. Let's call it session one. And then there's gonna be, I'm just gonna make this a little bit smaller. You can have any number of sessions.

So I'm gonna say.

Session two.

Krish (products.snowpal.com) (06:19.598)
Okay, I'm up to session N.

Krish (products.snowpal.com) (06:27.852)
I'm a dot, dot, dot here.

So let's connect this.

Session one, session two, and session end.

Krish (products.snowpal.com) (06:49.87)

Okay, so those are three sessions. And now if you go back here, again, plan for today, what you're working on certain tickets or issues, whatever you want to call them, and then you're...

you define and then there's free text as well. You're saying, yeah, I don't have a ticket or an issue, but this is what I'm gonna be helping somebody on board. So you can report, there's a certain format. We'll not go into the details in this video, but there's certain format you can follow for plan for today. And then you have sessions, you can add any number of sessions. And in the session, you can add tickets to sessions. You can add other items to sessions,

which is again, not a ticket, not an issue, because if it's a ticket, it could be a GR issue. It could be a GitHub issue. It could be some other issue. But if you're saying,

I'm going to be working on something else that doesn't is not tracked necessarily by a ticket Then there's something called other items so you can add sessions You can you know add your tickets and other items to your session while you add the session or you can do them incrementally So that's what sessions is and then the last last but not least is blocked by I want to draw let me copy this I think I should have actually had maybe I'll use a slightly different color because

There are three different categories, if you will. And then for this one, let's pick, okay. So this is gonna be blocked by. Ideally, you don't want anyone in your team to be blocked by anybody else, but that's not always the case. So this tracks what you're blocked by. Am I blocked by this other developer? Am I blocked by approvals? Is blocked by being tracked by a ticket or is it free flowing text?

For admin, I'm not gonna draw these circles because I'm just gonna draw maybe a dotted line. Yeah, I'm just gonna draw a dotted line here. Just remember that all that I'm seeing here is an admin does represents a user. So they do it for another member, right? They're doing it for another member.

Krish (products.snowpal.com) (08:54.67)

But the idea is the same, the type of status, the structure of the status is not going to be any different.

That's what this API at the highest level supports. Now again, the devil is in the detail, meaning there's lots of other nuances. You what are the attributes that it supports when you add a ticket? What are the details that you can add for that, you know, provide for that ticket? You can look through the description, you know, you know, let's take a quick look here. Ticket ID, the type, whether the ticket's on Azure, DevOps, or it's on Jira, whatnot, the URL, the description, how much time did they spend? What is the status of those tickets? Are they pairing with some other?

a member, maybe there's a member ID, maybe there's a non -member, pull requests, URLs, descriptions, those pull requests, and a number of things of this nature.

You can extend this to add custom attributes as well. But the idea is you have a pretty rich set of endpoints associated with this API for you to manage statuses. Imagine creating a Slack app with the API being available. And if it were not to be available, it's going to take your API team all the time. And maybe you don't even have an API team for all I know, for all we know. And if you don't, you don't need to hire staff, maintain, manage, and pay your teams, create a back-end team just to get things.

working because we have a rich suite of APIs. The idea behind that is you can actually not even have a backend team to get things working. Or if you have a team, they probably have better things to do and bigger fish to fry. So they don't have to actually work on rebuilding or reinventing the wheel. That's the idea behind us publishing these fundamental, these APIs as fundamental building blocks. We have SDKs available. So you can go to status SDK, status -sdk .snowpal .com. Currently I

Krish (products.snowpal.com) (10:39.952)

SDKs are available in Golang, but we're expanding that to include other languages. We have recipes published. You can check out those recipes that manufacture examples that showcase

how we can actually use this API. We talked about the developer documentation and then you can go to aws.snowpal.com to go.

find this API and within three clicks and five minutes you'll actually get be able to sign up subscribe to it you'll get the API key and the product code and you can actually start consuming the API. Now there are multiple ways to license and to integrate this API one is a standard SAS based model where you pay by request or subscription annual or monthly whatever works for you but we have customers who asked us hey can you actually provision this in our cloud infrastructure be it AWS Google Cloud or Azure not a problem we can actually make

all of the API gateway, the whole thing available for you. So it's entirely, it's in your environment and we have no access to it. Or we can have, you know, we have multi-tenancy. Suppose you can use our environment for requests, but you can, we can point to your DB. That's a variation of it.

or even other larger customers ask us saying, you know what, can you give us, can we license the API so it becomes part of our DevOps and platform engineering infrastructure, but you have absolutely no control over proceedings because that's how we want to do it, not a problem. And then you can purchase upgrade licenses so you get bug fixes, enhancements and features. That's our licensing model. So we've kind of covered the gamut. We'll go into the details of these endpoints maybe definitely in subsequent videos. But this first one, I just want to socialize.

Thanks for watching guys.

Krish (products.snowpal.com) (12:15.678)

this API and actually you can go to, there's one more place you can actually go to, let me share that as well really quickly. Go to products.snowpal.com and go to API and you have plenty of other API documentation in addition to the developer documentation on developers.snowpal.com and the postman collection. So a combination of all of that hopefully will hold you in good stead in terms of what you actually need to do to integrating

this API. With that I'm going to end this podcast. Thanks and talk to you soon. Bye bye.