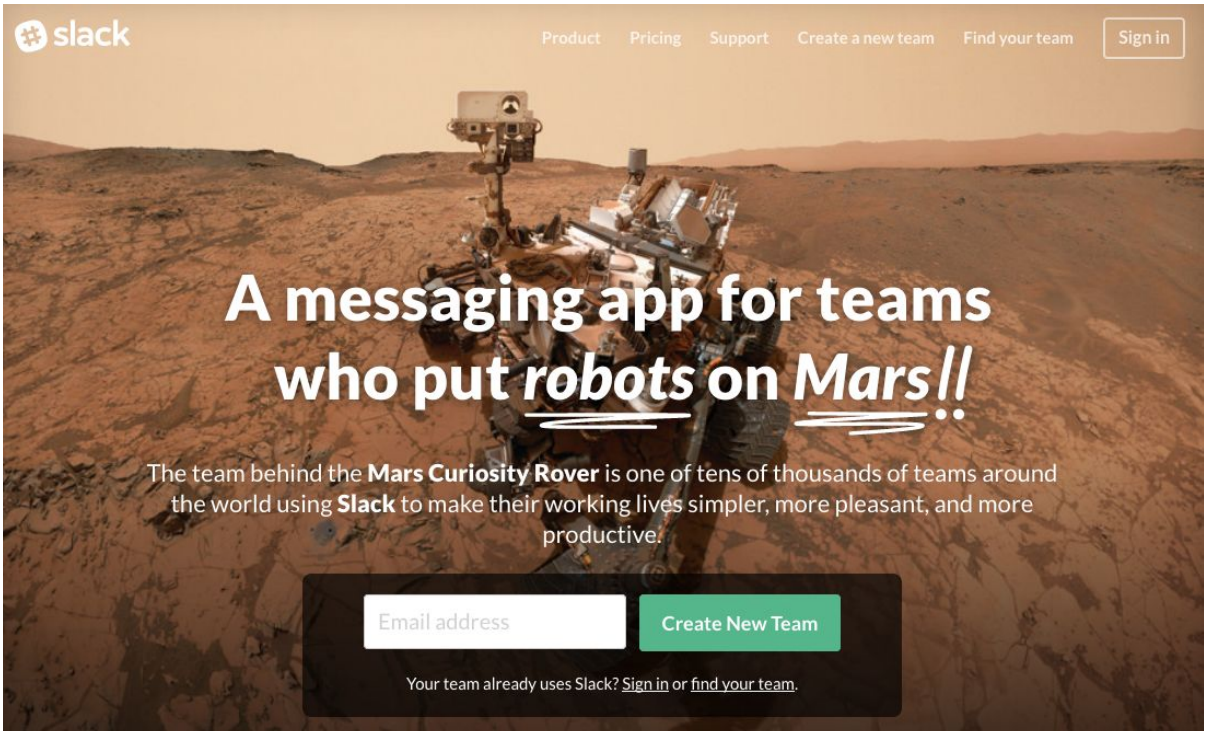
Habby installation manual

Prerequisites

- AWS account
- AWS cli installed and configure with your AWS account (http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-set-up.html or https://github.com/serverless/serverless/blob/master/docs/02-providers/aws/01-setup.md)
- Installed node, npm
- Installed git
- Redis server

1.Creating Slack Team

1. Visit https://slack.com/ and create new Team

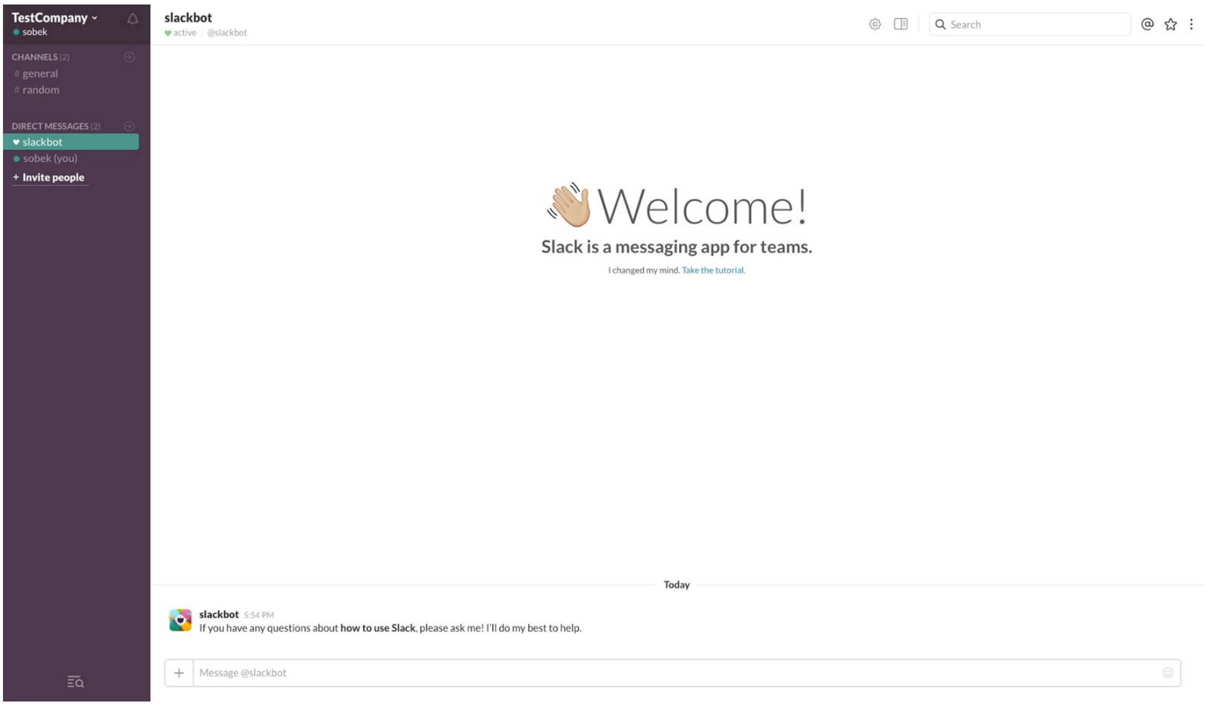


A messaging app for teams who put *robots* on *Mars!!*

The team behind the **Mars Curiosity Rover** is one of tens of thousands of teams around the world using **Slack** to make their working lives simpler, more pleasant, and more productive.

Your team already uses Slack? Sign in or find your team.

2. As final result you should have own Slack team



Welcome!

Slack is a messaging app for teams.

I changed my mind. Take the tutorial.

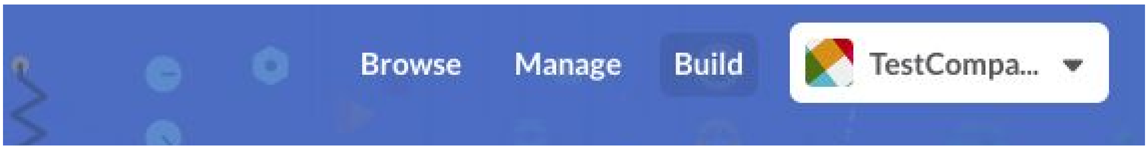
slackbot 5:54 PM
If you have any questions about **how to use Slack**, please ask me! I'll do my best to help.

2. Creating Slack Application

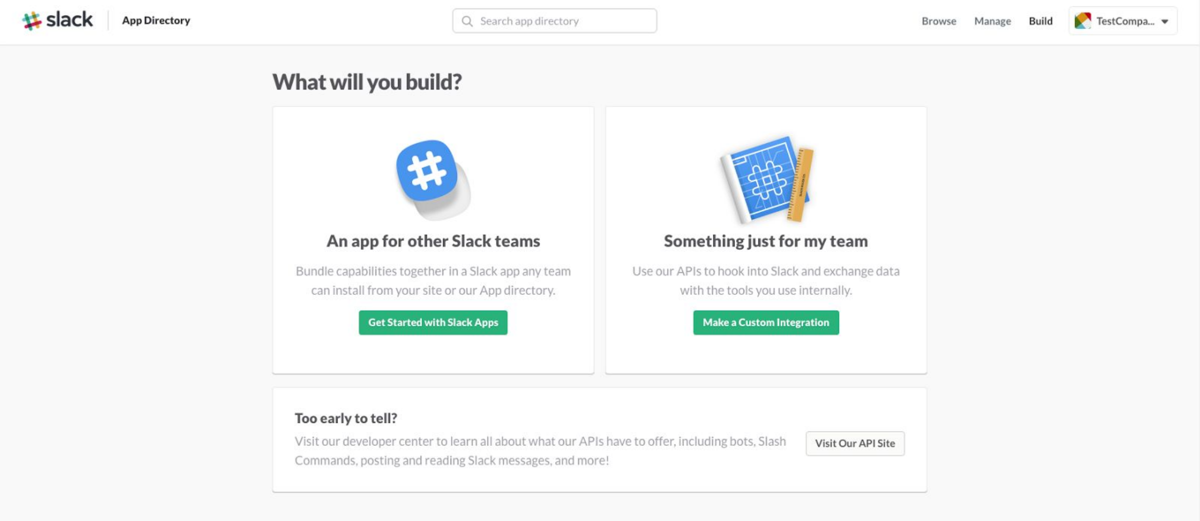
1. Go to your Slack Team application

2. Click on “Team name” (on top left corner). New menu will reveal. Select “App & Integration”.
New page will be opened.

3. On a new opened “App & integration” page. Select “Build” from top right menu.



4. An app for other Slack teams.



What will you build?

An app for other Slack teams

Bundle capabilities together in a Slack app any team can install from your site or our App directory.

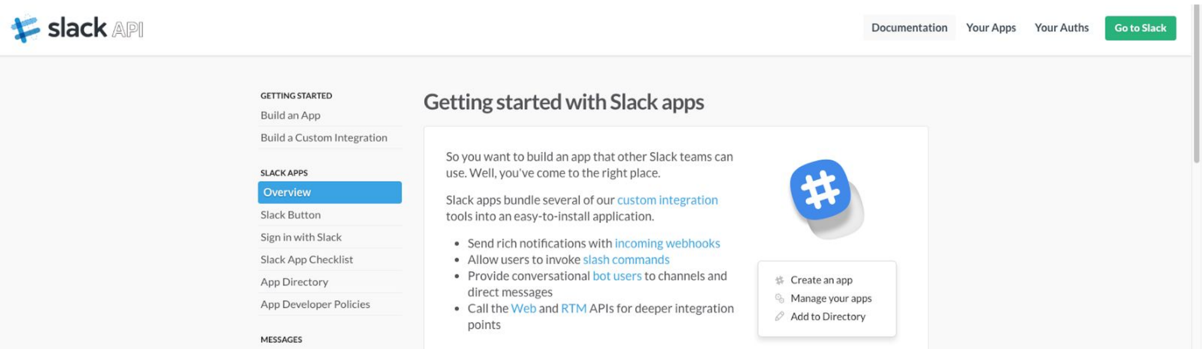
Something just for my team

Use our APIs to hook into Slack and exchange data with the tools you use internally.

Too early to tell?

Visit our developer center to learn all about what our APIs have to offer, including bots, Slash Commands, posting and reading Slack messages, and more!

5. Create an app

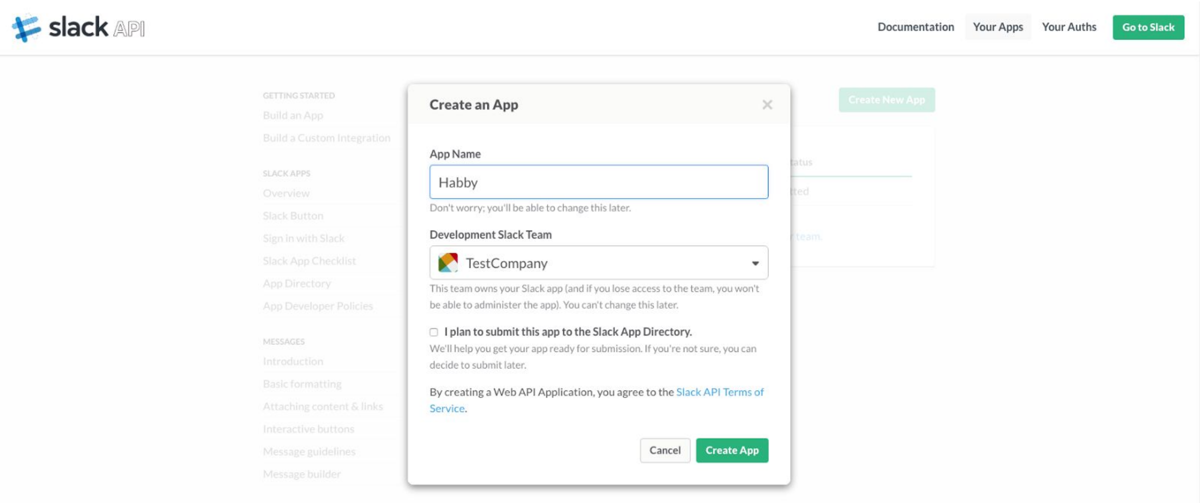


Getting started with Slack apps

So you want to build an app that other Slack teams can use. Well, you've come to the right place.

Slack apps bundle several of our custom integration tools into an easy-to-install application.

- Send rich notifications with incoming webhooks
- Allow users to invoke slash commands
- Provide conversational bot users to channels and direct messages
- Call the Web and RTM APIs for deeper integration points



Create an App

App Name

Habby

Don't worry; you'll be able to change this later.

Development Slack Team

TestCompany

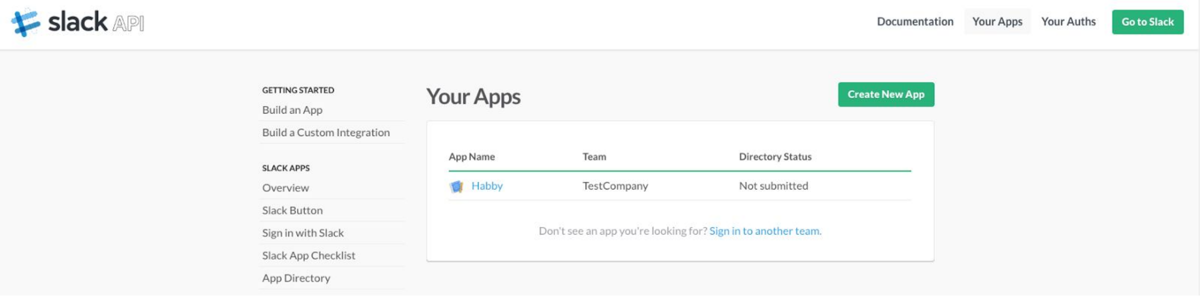
This team owns your Slack app (and if you lose access to the team, you won't be able to administer the app). You can't change this later.

□ I plan to submit this app to the Slack App Directory.

We'll help you get your app ready for submission. If you're not sure, you can decide to submit later.

By creating a Web API Application, you agree to the Slack API Terms of Service.

6. After that your app should be ready to config

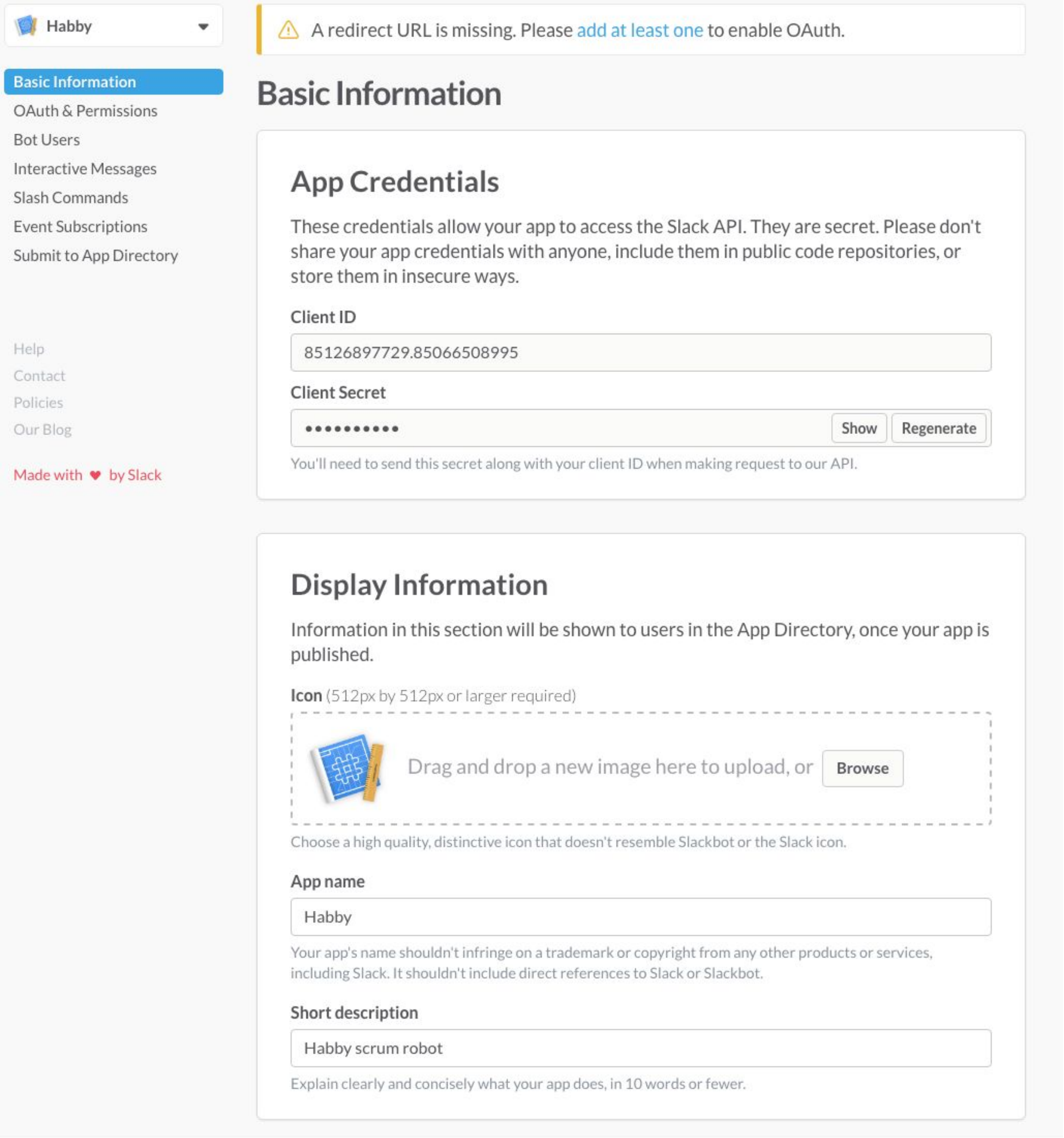


Your Apps

App Name	Team	Directory Status
Habby	TestCompany	Not submitted

Don't see an app you're looking for? Sign in to another team.

7. Copy Slack Client ID and Slack Client Secret



⚠ A redirect URL is missing. Please add at least one to enable OAuth.

Basic Information

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

Client ID

85126897729.85066508995

Client Secret

••••••••••

You'll need to send this secret along with your client ID when making request to our API.

Display Information

Information in this section will be shown to users in the App Directory, once your app is published.

Icon (512px by 512px or larger required)

Drag and drop a new image here to upload, or

Choose a high quality, distinctive icon that doesn't resemble Slackbot or the Slack icon.

App name

Habby

Your app's name shouldn't infringe on a trademark or copyright from any other products or services, including Slack. It shouldn't include direct references to Slack or Slackbot.

Short description

Habby scrum robot

Explain clearly and concisely what your app does, in 10 words or fewer.

3. Deploy application on AWS

1. Clone github repository

```
> git clone https://github.com/mareklabuzeksmt/HABBY.git
```

2. In the app folder, run npm

```
>npm install
```

3. Verify if serverless framework is installed correctly

```
serverless -v
>1.0.0-rc.1
```

4. Edit serverless.yml with own config data

```
...
custom:
  projectName: Habby
  remind:
    last_call: 9:30
    interval: 20
    timeout: 15
  slack:
    client_id: [SLACK_CLIENT_ID]
    client_secret: [SLACK_CLIENT_SECRET_ID]
  redis:
    host: [REDIS_HOST]
    port: [REDIS_PORT]
  writeEnvVars:
      PROJECT_NAME: ${self:custom.projectName}
      REDIS_HOST: ${self:custom.redis.host}
      REDIS_PORT: ${self:custom.redis.port}
      REMIND_TIMOUT: ${self:custom.remind.timeout}
      REMIND_INTERVAL: ${self:custom.remind.interval}
      REMIND_LAST_CALL: ${self:custom.remind.last_call}
      SLACK_CLIENT_ID: ${self:custom.slack.client_id}
      SLACK_CLIENT_SECRET: ${self:custom.slack.client_secret}
```

Please note: Redis server need to be reachable by AWS Lambda

5. Deploy app on AWS. Serverless framework handle deployment, including AWS Api Gateway and AWS Lambda.

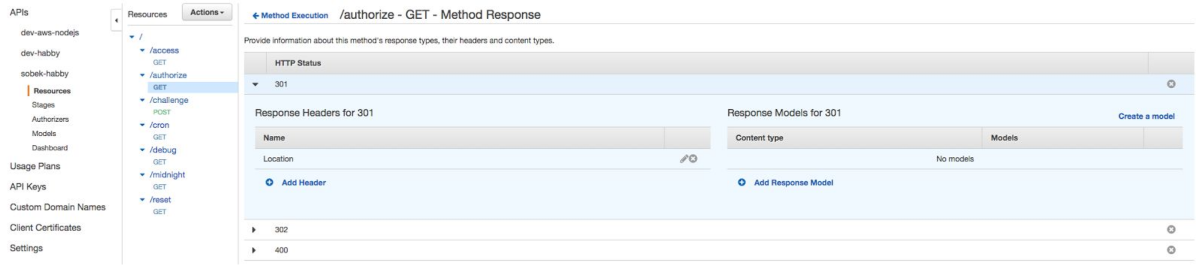
```
> serverless deploy --stage dev --region us-east-1
...
Service Information
service: habby
```

```
stage: dev
region: us-east-1
endpoints:
  GET - https://[id].execute-api.us-east-1.amazonaws.com/[stage]/authorize
  GET - https://[id]execute-api.us-east-1.amazonaws.com/[stage]/access
  GET - https://[id].execute-api.us-east-1.amazonaws.com/[stage]/cron
  GET - https://[id].execute-api.us-east-1.amazonaws.com/[stage]/midnight
  POST - https://[id].execute-api.us-east-1.amazonaws.com/[stage]/challenge
```

6. Go to AWS Api Gateway and configure '/authorize' endpoint. Step is required to setup 301 redirection to Slack Oauth.

7. Open Method Response for '/authorize'.

Add Response: 301
Open newly created response and Add Header "Location"
Remove default 200 rule



/authorize - GET - Method Response

Provide information about this method's response types, their headers and content types.

HTTP Status
301
302
400

Response Headers for 301

Name
Location

Response Models for 301

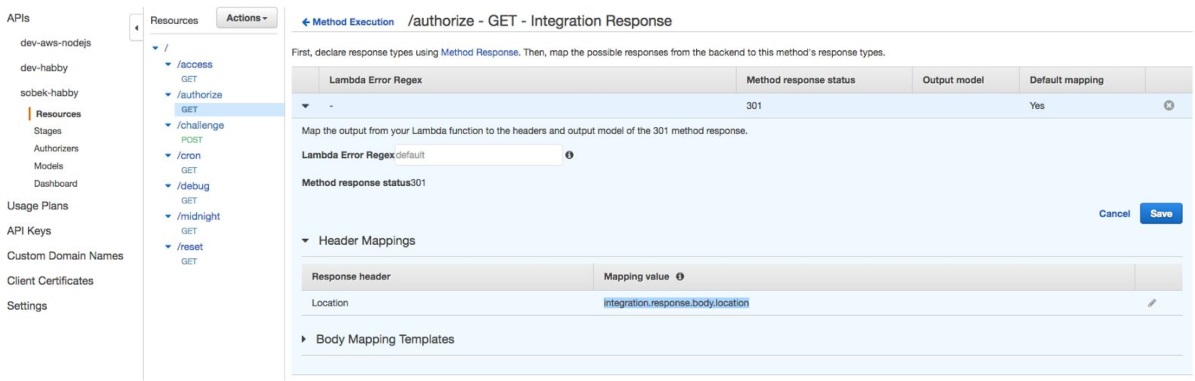
Content type	Models
No models	

8. Open Integration Response for '/authorize'.

Remove default 200 rule
Add integration response.
Lambda Error Regex: blank
Method response status: 301

Save
Open newly created 301 rule
Reveal Header Mappings
Set Location mapping value to: integration.response.body.location
Save

After that all should look as following picture



/authorize - GET - Integration Response

First, declare response types using Method Response. Then, map the possible responses from the backend to this method's response types.

Lambda Error Regex	Method response status	Output model	Default mapping
-	301		Yes

Map the output from your Lambda function to the headers and output model of the 301 method response.

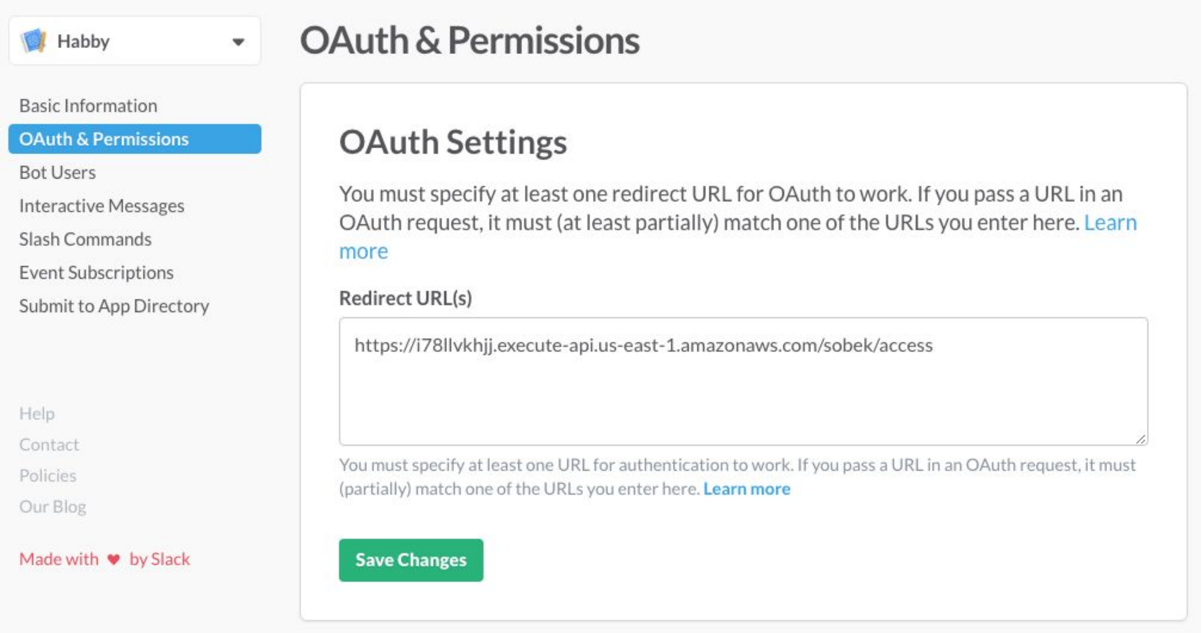
Lambda Error Regex default

Method response status301

Response header	Mapping value
Location	integration.response.body.location

4. Update Slack application with Bot endpoints

1. Go to Slack Team
2. Open “Apps & integrations” page
3. Select “Build” -> “An app for other Slack teams” -> Your Apps
4. Open Habby app -> OAuth & Permissions
5. Put your “../access” endpoint into “Redirect URL(s)” field (full URL is listed in ‘endpoint’ section of ‘serverless deploy’ command output).



OAuth & Permissions

OAuth Settings

You must specify at least one redirect URL for OAuth to work. If you pass a URL in an OAuth request, it must (at least partially) match one of the URLs you enter here. Learn more

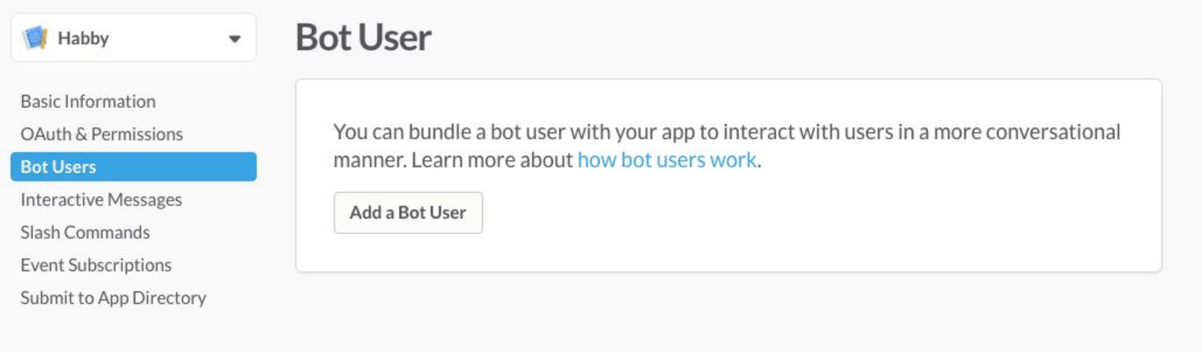
Redirect URL(s)

https://i78llvkhjj.execute-api.us-east-1.amazonaws.com/sobek/access

You must specify at least one URL for authentication to work. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. Learn more

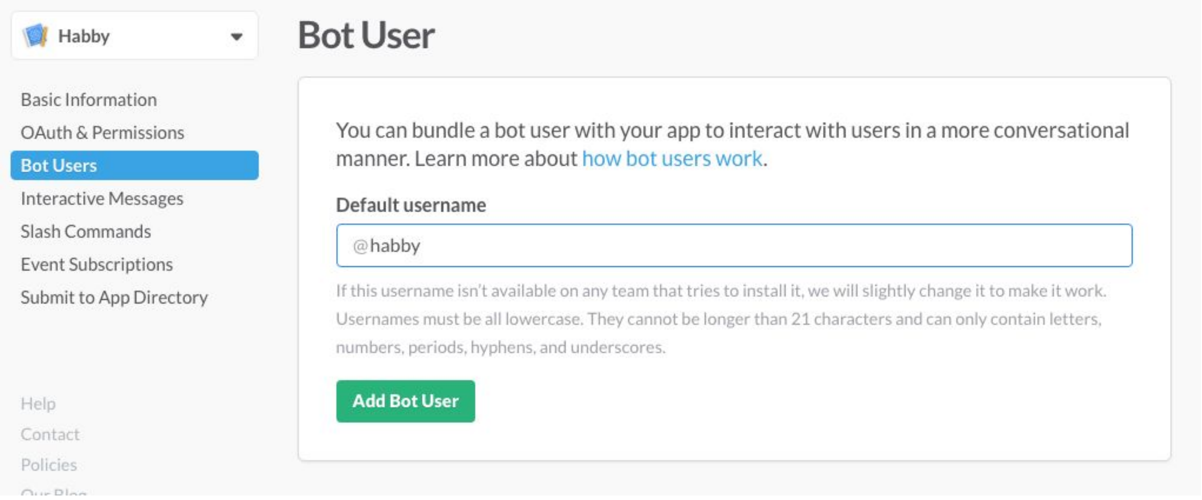
6. Save Changes

7. Go to “Bot Users”
8. Add “@habby” bot user



Bot User

You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.



Bot User

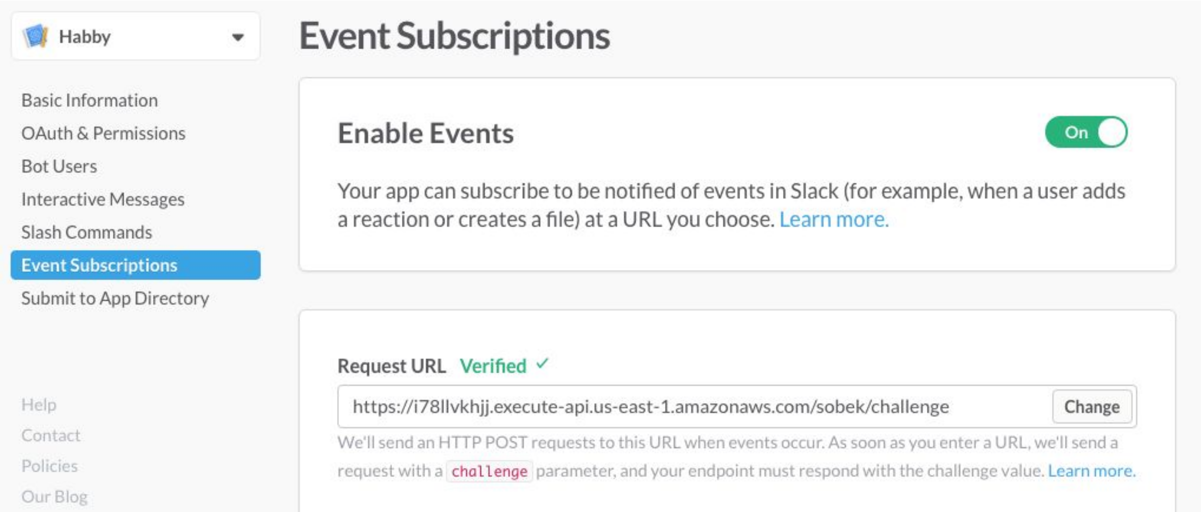
You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.

Default username

@habby

If this username isn't available on any team that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

9. Go to "Event and Subscriptions" to configure Request URL and subscribed events.
10. Enable Events
11. Put ".../challenge" endpoint (full URL is listed in 'endpoint' section of 'serverless deploy' command output) in the Request URL field. You should see that endpoint has been verified.



Event Subscriptions

Enable Events

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. Learn more.

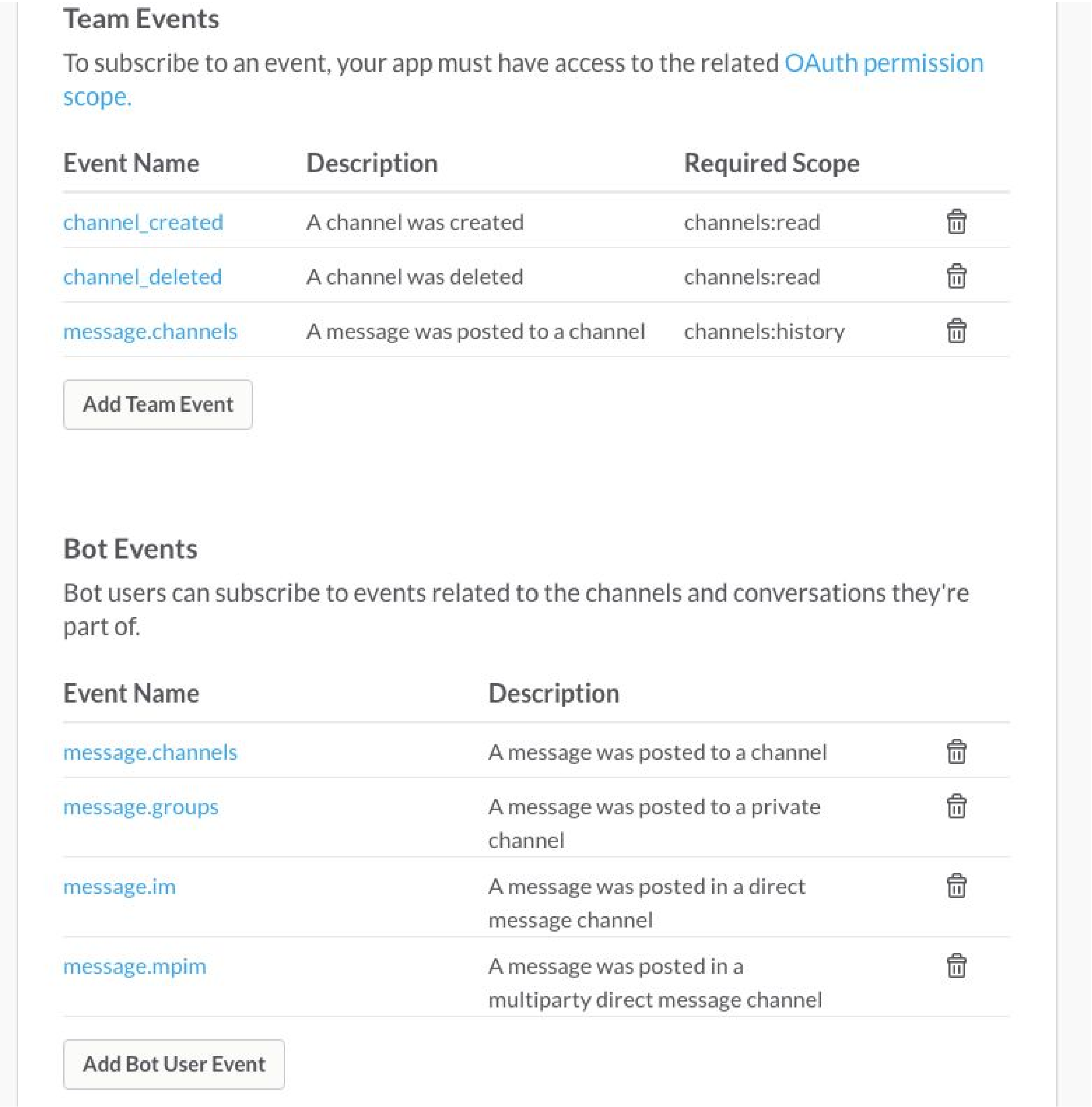
Request URL Verified ✓

https://i78llvkhjj.execute-api.us-east-1.amazonaws.com/sobek/challenge

We'll send an HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. Learn more.

12. Scroll to Team and Bot Events

13. Pickup following events



Team Events

To subscribe to an event, your app must have access to the related OAuth permission scope.

Event Name	Description	Required Scope	
channel_created	A channel was created	channels:read	🗑
channel_deleted	A channel was deleted	channels:read	🗑
message.channels	A message was posted to a channel	channels:history	🗑

Bot Events

Bot users can subscribe to events related to the channels and conversations they're part of.

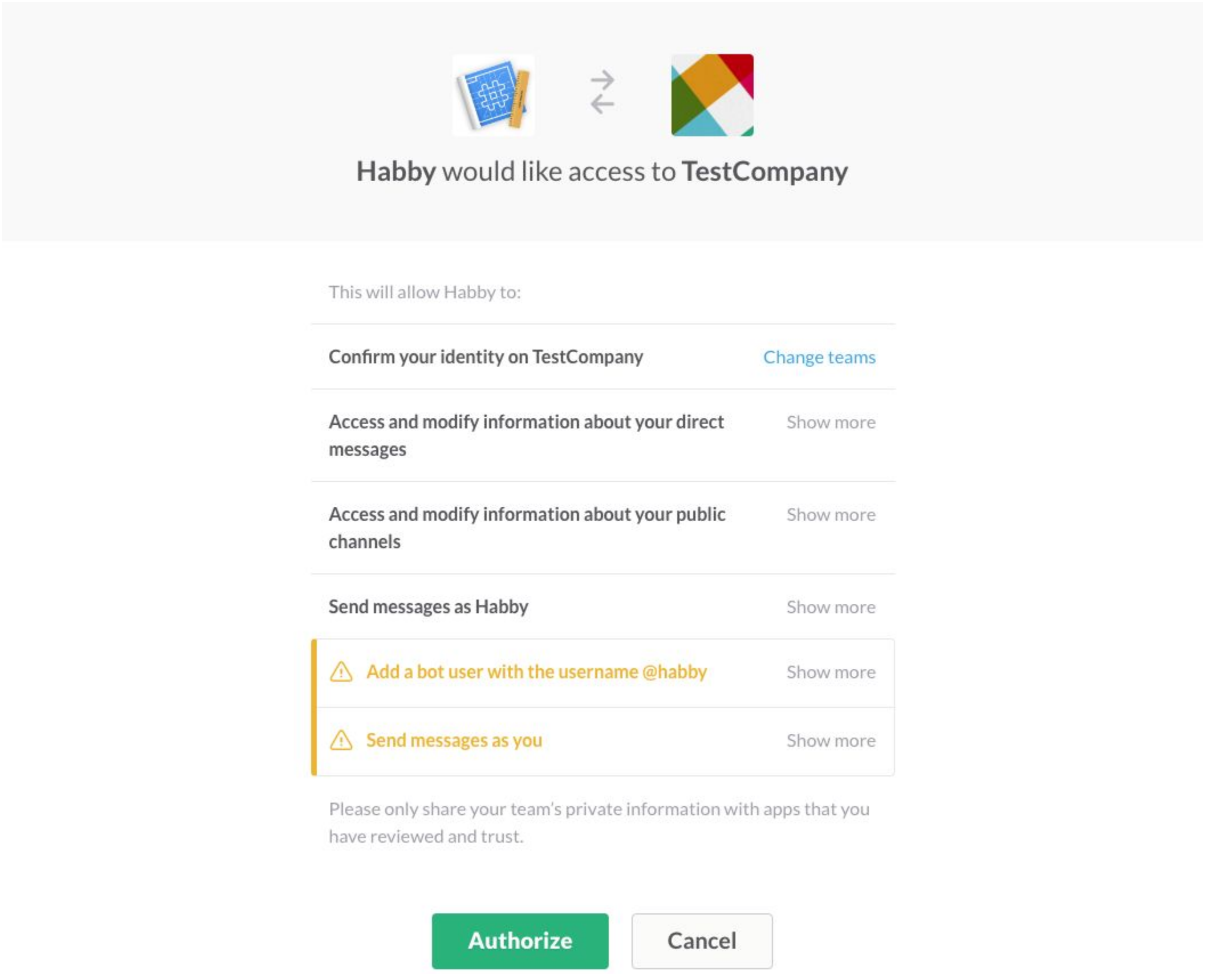
Event Name	Description	
message.channels	A message was posted to a channel	🗑
message.groups	A message was posted to a private channel	🗑
message.im	A message was posted in a direct message channel	🗑
message.mpim	A message was posted in a multiparty direct message channel	🗑

14. Save changes

15. Congratulations! your app is almost ready to use. The last step is to authorize the app. During that step Habby bot will be connected with your Slack Team. Habby will receive authentication token.

16. Final step. Authorize Habby to your Slack Team.

Open in the browser “/authorize” endpoint. For example:
https://[id].execute-api.us-east-1.amazonaws.com/[stage]/authorize
You will be redirected to Slack and you should see following screen.



Habby would like access to TestCompany

This will allow Habby to:

Confirm your identity on TestCompany

Access and modify information about your direct messages

Access and modify information about your public channels

Send messages as Habby

⚠ Add a bot user with the username @habby

⚠ Send messages as you

Please only share your team's private information with apps that you have reviewed and trust.

Important note: In case of lack of redirection: If "/authorize" returns json object:

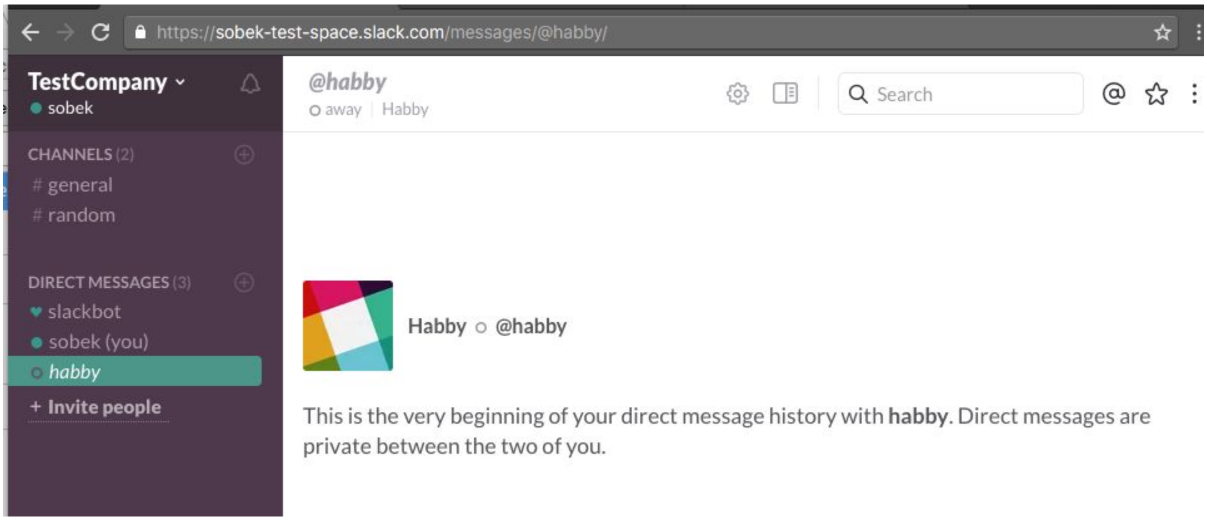
```
{
  "location":
"https://slack.com/oauth/authorize?client_id=[SLACK_CLIENT_ID]&scope=channels:history,channels:read,bot,chat:write:bot,im:read,im:write,im:history&redirect_uri=https://[id]./[stage]/access"
}
```

Please copy url and open in the browser.

17. Click Authorize

18. After few seconds you will see "Habby is authorized to your Slack Team."

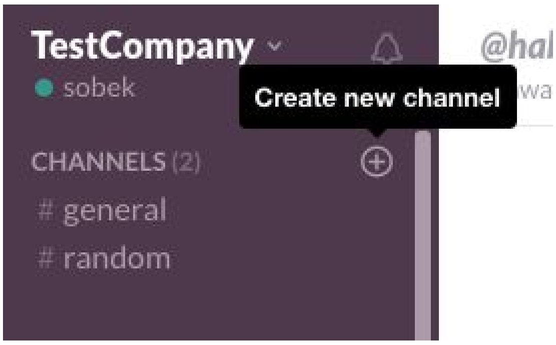
19. As a confirmation Habby bot will be visible on your Slack Team app

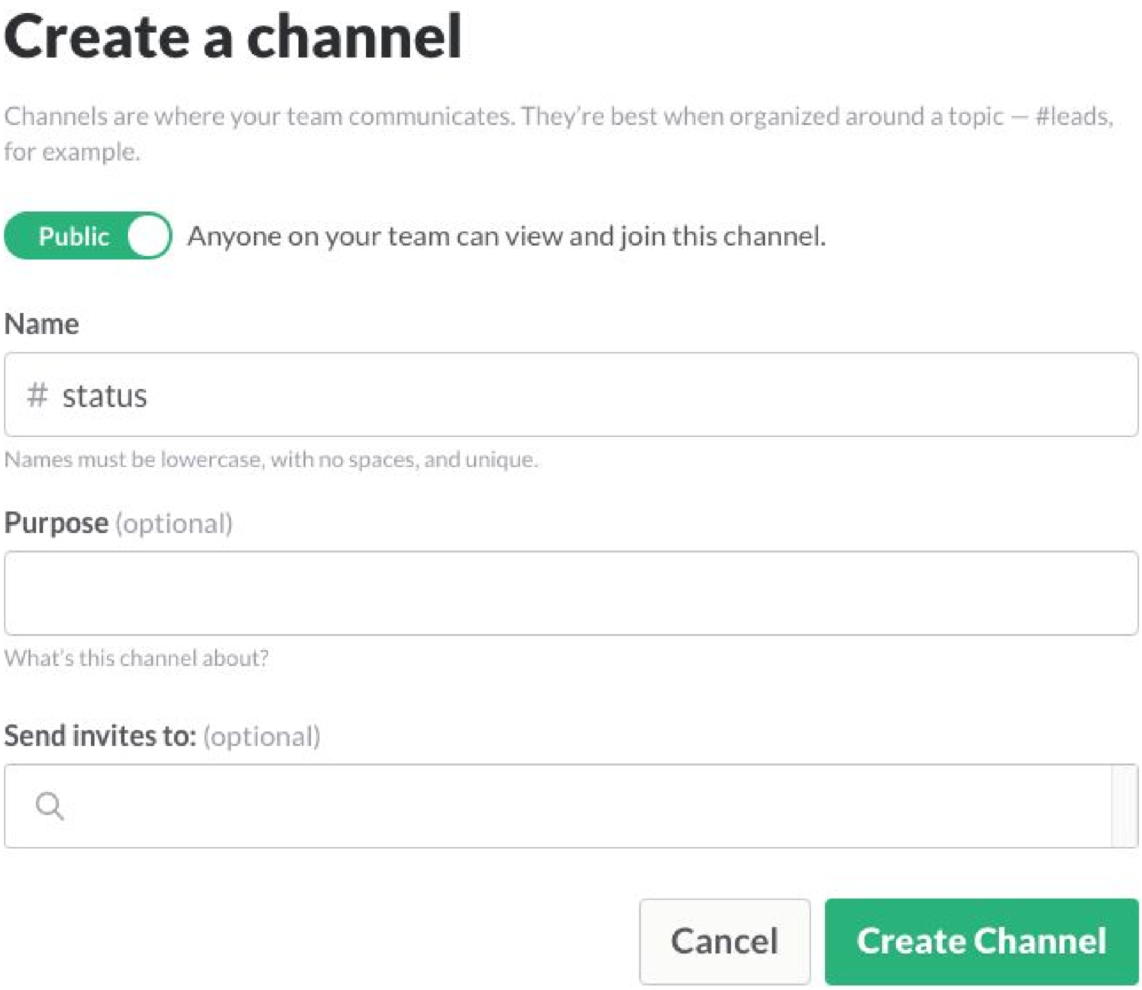


This is the very beginning of your direct message history with **habby**. Direct messages are private between the two of you.

5. Create status channel

1. Create new channel “status”. Habby use this channel to listen “status” updates from other users.





Create a channel

Channels are where your team communicates. They’re best when organized around a topic — #leads, for example.

Public — Anyone on your team can view and join this channel.

Name

status

Names must be lowercase, with no spaces, and unique.

Purpose (optional)

What’s this channel about?

Send invites to: (optional)