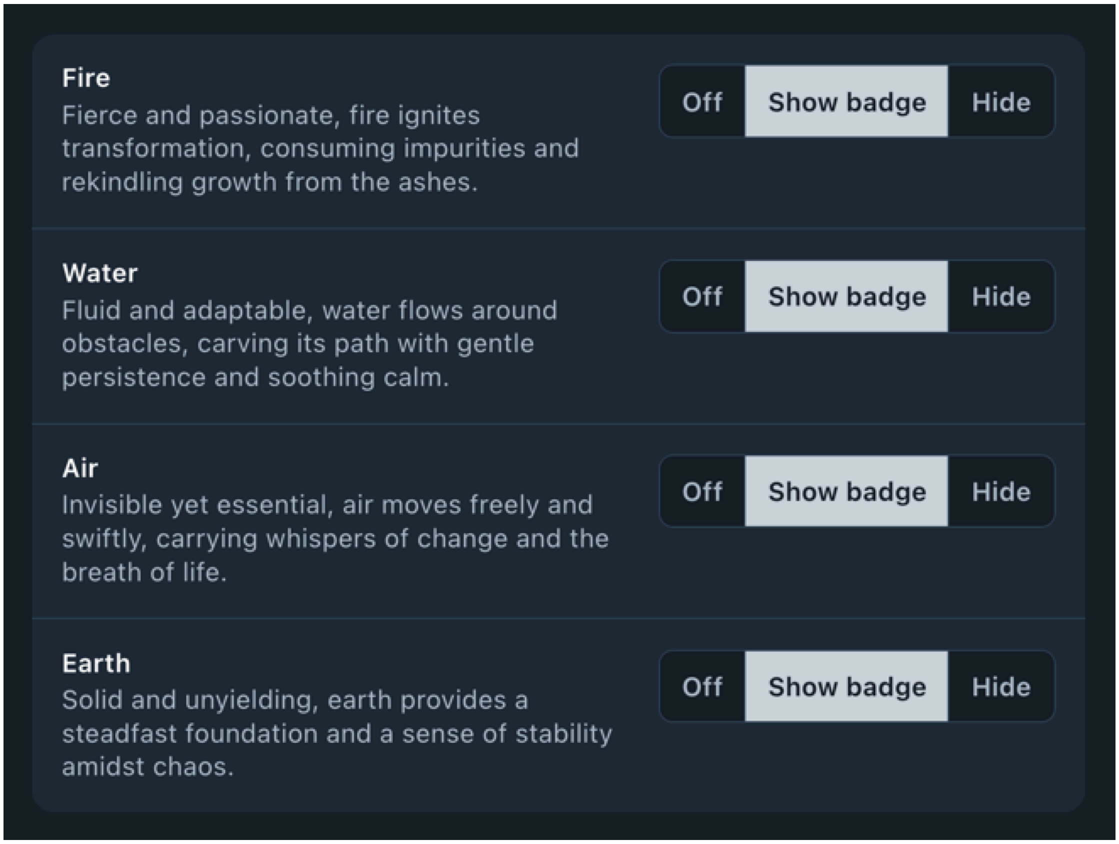
Automated Labeling

This guide assumes you've followed the steps in [Getting Started](https://skyware.js.org/guides/labeler/introduction/getting-started/) and have a labeler set up. If you haven't, do that now!

Now that you're running a labeler server, you might want to start labeling content based on user interaction; for instance, liking a post or sending a DM. This guide will walk you through building a bot that will allow users to select labels by liking posts.

For this guide, we've set up an example labeler with four labels for a user to choose from: `fire`, `water`, `air`, and `earth`. These labels have been set to "inform" because they're intended to be used for informational purposes, "warn" severity so that they appear on user profiles and posts, and no blur because these are not moderation labels.

Here's what the profile we'll be working with looks like, all set up, though it's not ready to start applying labels quite yet.



Fire
Fierce and passionate, fire ignites transformation, consuming impurities and rekindling growth from the ashes.
Off [✓ Show badge] Hide

Water
Fluid and adaptable, water flows around obstacles, carving its path with gentle persistence and soothing calm.
Off [✓ Show badge] Hide

Air
Invisible yet essential, air moves freely and swiftly, carrying whispers of change and the breath of life.
Off [✓ Show badge] Hide

Earth
Solid and unyielding, earth provides a steadfast foundation and a sense of stability amidst chaos.
Off [✓ Show badge] Hide

Setting up

Start by installing `@skyware/labeler` and `@skyware/bot`.

```
npm install @skyware/labeler @skyware/bot
```

You'll also need to create a `.env` file in the root of your project. You can use the `dotenv` package to load these credentials into your project.

```
LABELER_DID=did:...
LABELER_PASSWORD=...
SIGNING_KEY=...
```

`LABELER_DID` is the DID of the labeler account you created in the previous guide. You can look this up at internect.info.

`LABELER_PASSWORD` is the password you set when you created the labeler account.

`SIGNING_KEY` is the signing key you received when setting up the labeler, if you used the CLI for setup.

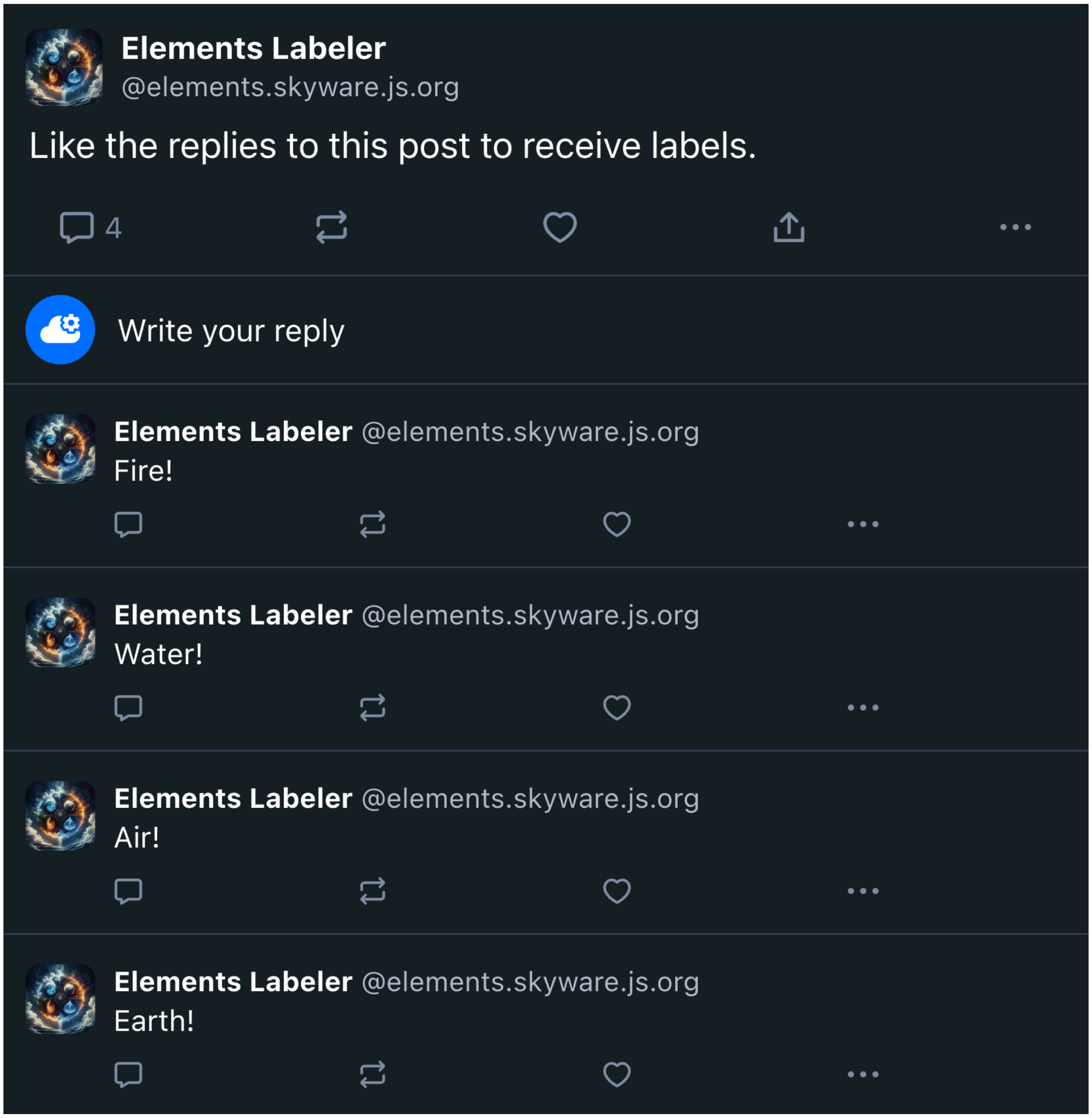
Creating Posts

The first step is to create posts that users can like to receive labels. Using `@skyware/bot`, you can create posts with the Bot#post method. Create a file called `post.ts` with the following code:

```
import { Bot } from "@skyware/bot";

const bot = new Bot();
await bot.login({
    identifier: process.env.LABELER_DID,
    password: process.env.LABELER_PASSWORD,
});

const post = await bot.post({ text: "Like the replies to this post to receive
labels.", threadgate: { allowLists: [] } });

const firePost = await post.reply({ text: "Fire!" });
const waterPost = await post.reply({ text: "Water!"});
const airPost = await post.reply({ text: "Air!" });
const earthPost = await post.reply({ text: "Earth!" });

console.log(
    `Fire: ${firePost.uri}\n`,
    `Water: ${waterPost.uri}\n`,
    `Air: ${airPost.uri}\n`,
    `Earth: ${earthPost.uri}\n`,
);
// Fire: at://did:.../app.bsky.feed.post/...
// Water: at://did:.../app.bsky.feed.post/...
// Air: at://did:.../app.bsky.feed.post/...
// Earth: at://did:.../app.bsky.feed.post/...
```

The `post` method returns a `PostReference` object, which you can use to reply to the post. The `PostReference#reply` method will return a reference to the reply. We're using the `threadgate` property to prevent anyone else from replying to the post, by setting the allowed lists of users to an empty array.

Transpile this code if necessary and run it. You should see the AT URI of each post printed to the console. Take note of these, as you'll need them later to process likes.



Elements Labeler @elements.skyware.js.org

Like the replies to this post to receive labels.

Elements Labeler @elements.skyware.js.org
Fire!

Elements Labeler @elements.skyware.js.org
Water!

Elements Labeler @elements.skyware.js.org
Air!

Elements Labeler @elements.skyware.js.org
Earth!

Running the Labeler

Now that you've got the posts you need, create a new file called `labeler.ts`. This is where you'll run your labeler server and listen for likes.

```
import { LabelerServer } from "@skyware/labeler";
import { Bot } from "@skyware/bot";

const server = new LabelerServer({
    did: process.env.LABELER_DID,
    signingKey: process.env.SIGNING_KEY,
});
```

```
server.start(14831, (error) => {
    if (error) {
        console.error("Failed to start: ", error);
    } else {
        console.log("Listening on port 14831");
    }
});

const bot = new Bot();
await bot.login({
    identifier: process.env.LABELER_DID,
    password: process.env.LABELER_PASSWORD,
});
```

A reminder that you’ll need to set up reverse proxying so that the domain you’re using for the labeler points to the server running this code, on the port you choose for `server.start`. You can use a tool like Caddy↗ to do this.

If no errors are printed to the console, you’re ready to start labeling!

Labeling

Next, we’ll build on this to listen for likes and apply labels to users. To start, we can use the Bot#on method to listen for likes. This method will be called whenever a user likes a post, feed generator, or the labeler profile. Add the following code to the end of `labeler.ts`:

```
// Modify your import to include the Post class
import { Bot, Post } from "@skyware/bot";
// ...
bot.on("like", async ({ subject, user }) => {
    if (subject instanceof Post) {
        console.log(subject.uri);
    }
});
```

Now, whenever the bot receives a like, it will first ensure that the like is for a post (feed generators and the labeler profile can also receive likes!) and then log the AT URI of the post to the console.

Of course, that's not very useful. Let's start applying labels based on which post the user liked. We're going to create an object mapping the AT URIs of the posts we created earlier to the labels we want to apply, then use that object to label the user's account.

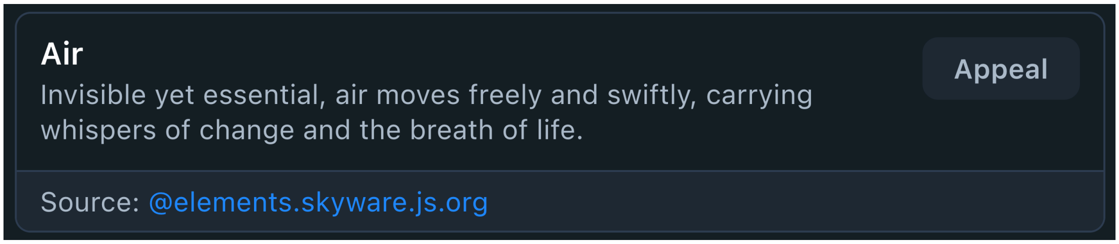
```
const postsToLabels: Record<string, string> = {
    "at://did:.../app.bsky.feed.post/f1234": "fire",
    "at://did:.../app.bsky.feed.post/w3456": "water",
    "at://did:.../app.bsky.feed.post/a5678": "air",
    "at://did:.../app.bsky.feed.post/e7890": "earth",
}

bot.on("like", async ({ subject, user }) => {
    if (subject instanceof Post) {
        const label = postsToLabels[subject.uri];
        if (label) {
            await user.labelAccount([label]);
        }
    }
});
```

Make sure to substitute in the right AT URIs from earlier. We're using the `Profile#labelAccount` method to label the user's account, so that the label will appear on both their profile and posts.

Your final code should look something like this:

```
import { LabelerServer } from "@skyware/labeler";
import { Bot, Post } from "@skyware/bot";

const server = new LabelerServer({
    did: process.env.LABELER_DID,
    signingKey: process.env.SIGNING_KEY,
});

server.start(14831, (error) => {
    if (error) {
        console.error("Failed to start: ", error);
    } else {
        console.log("Listening on port 14831");
    }
});

const bot = new Bot();
await bot.login({
    identifier: process.env.LABELER_DID,
    password: process.env.LABELER_PASSWORD,
});

const postsToLabels: Record<string, string> = {
    "at://did:.../app.bsky.feed.post/f1234": "fire",
    "at://did:.../app.bsky.feed.post/w3456": "water",
    "at://did:.../app.bsky.feed.post/a5678": "air",
    "at://did:.../app.bsky.feed.post/e7890": "earth",
}

bot.on("like", async ({ subject, user }) => {
    if (subject instanceof Post) {
        const label = postsToLabels[subject.uri];
        if (label) {
            await user.labelAccount([label]);
        }
    }
});
```

All that's left is to log into another account, like a post or multiple, and see the labels appear on your profile! (Make sure you've subscribed to the labeler and set the labels you want to see to "Show badge"!)



Air

Invisible yet essential, air moves freely and swiftly, carrying whispers of change and the breath of life.

Source: @elements.skyware.js.org