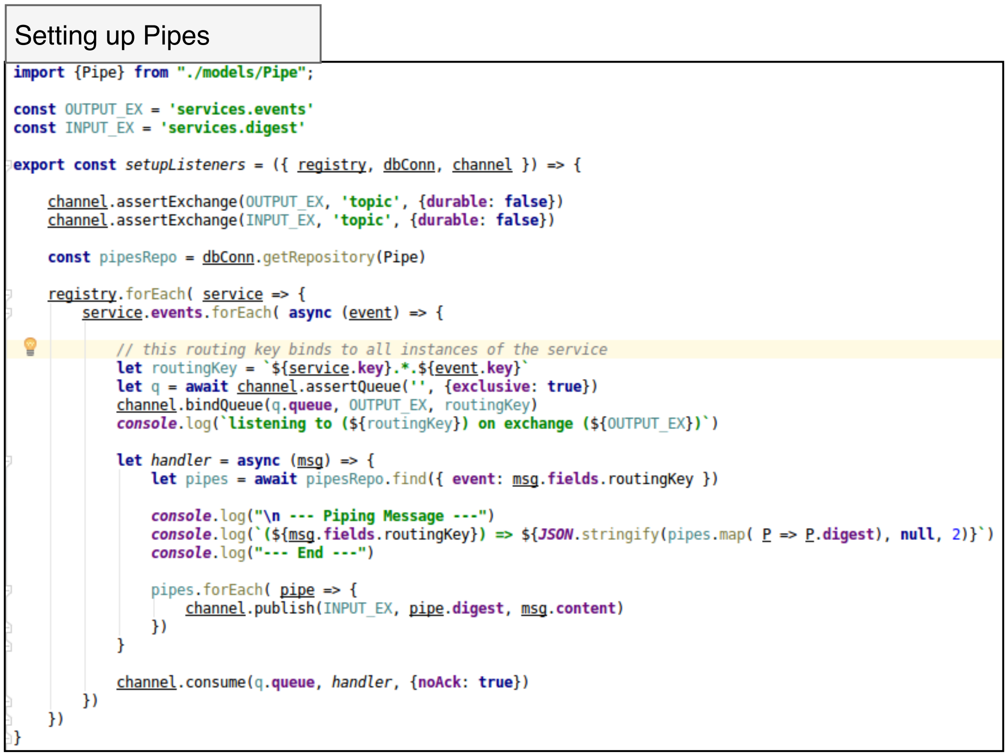
# Show me the code!

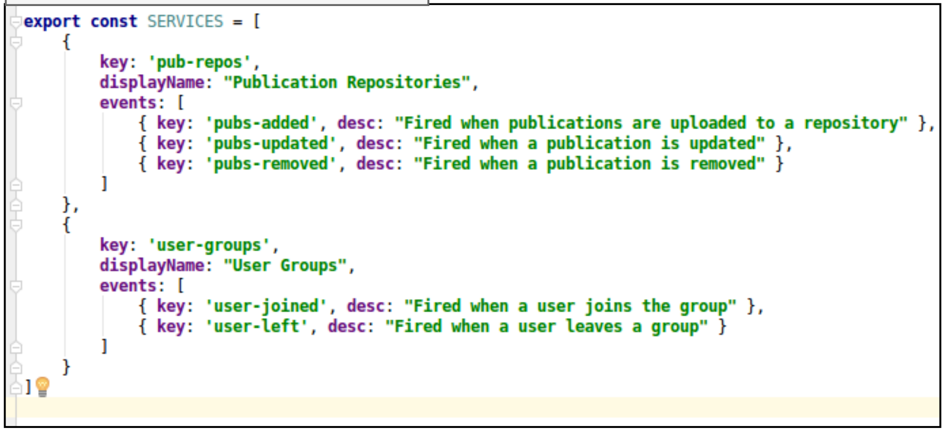
Sound difficult? We can do it in fewer than 30 lines of code!

## Setting up Pipes



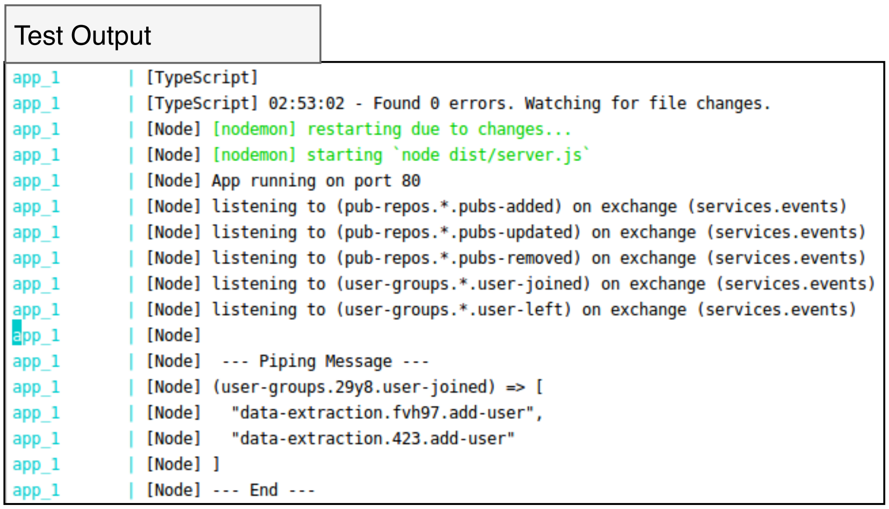
```
import {Pipe} from "./models/Pipe";

const OUTPUT_EX = 'services.events'
const INPUT_EX = 'services.digest'

export const setupListeners = ({ registry, dbConn, channel }) => {

    channel.assertExchange(OUTPUT_EX, 'topic', {durable: false})
    channel.assertExchange(INPUT_EX, 'topic', {durable: false})

    const pipesRepo = dbConn.getRepository(Pipe)

    registry.forEach( service => {
        service.events.forEach( async (event) => {

            // this routing key binds to all instances of the service
            let routingKey = `${service.key}.*.${event.key}`
            let q = await channel.assertQueue('', {exclusive: true})
            channel.bindQueue(q.queue, OUTPUT_EX, routingKey)
            console.log(`listening to (${routingKey}) on exchange (${OUTPUT_EX})`)

            let handler = async (msg) => {
                let pipes = await pipesRepo.find({ event: msg.fields.routingKey })

                console.log("\n --- Piping Message ---")
                console.log(`(${msg.fields.routingKey}) => ${JSON.stringify(pipes.map( P => P.digest), null, 2)}`)
                console.log("--- End ---")

                pipes.forEach( pipe => {
                    channel.publish(INPUT_EX, pipe.digest, msg.content)
                })
            }

            channel.consume(q.queue, handler, {noAck: true})
        })
    })
}
```

## Service Registry Configuration



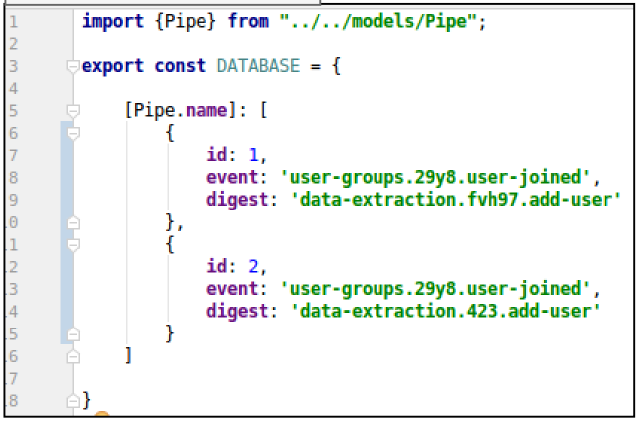
```
export const SERVICES = [
    {
        key: 'pub-repos',
        displayName: "Publication Repositories",
        events: [
            { key: 'pubs-added', desc: "Fired when publications are uploaded to a repository" },
            { key: 'pubs-updated', desc: "Fired when a publication is updated" },
            { key: 'pubs-removed', desc: "Fired when a publication is removed" }
        ]
    },
    {
        key: 'user-groups',
        displayName: "User Groups",
        events: [
            { key: 'user-joined', desc: "Fired when a user joins the group" },
            { key: 'user-left', desc: "Fired when a user leaves a group" }
        ]
    }
]
```

## Test Output



```
app_1       | [TypeScript]
app_1       | [TypeScript] 02:53:02 - Found 0 errors. Watching for file changes.
app_1       | [Node] [nodemon] restarting due to changes...
app_1       | [Node] [nodemon] starting `node dist/server.js`
app_1       | [Node] App running on port 80
app_1       | [Node] listening to (pub-repos.*.pubs-added) on exchange (services.events)
app_1       | [Node] listening to (pub-repos.*.pubs-updated) on exchange (services.events)
app_1       | [Node] listening to (pub-repos.*.pubs-removed) on exchange (services.events)
app_1       | [Node] listening to (user-groups.*.user-joined) on exchange (services.events)
app_1       | [Node] listening to (user-groups.*.user-left) on exchange (services.events)
app_1       | [Node]
app_1       | [Node]  --- Piping Message ---
app_1       | [Node] (user-groups.29y8.user-joined) => [
app_1       | [Node]   "data-extraction.fvh97.add-user",
app_1       | [Node]   "data-extraction.423.add-user"
app_1       | [Node] ]
app_1       | [Node] --- End ---
```

## Database Contents



```
import {Pipe} from "../../models/Pipe";

export const DATABASE = {

    [Pipe.name]: [
        {
            id: 1,
            event: 'user-groups.29y8.user-joined',
            digest: 'data-extraction.fvh97.add-user'
        },
        {
            id: 2,
            event: 'user-groups.29y8.user-joined',
            digest: 'data-extraction.423.add-user'
        }
    ]

}
```