Advanced Testing in Playwright

Steve Harrison / 6 June 2023

https://www.steveharrison.dev

Testing email verification codes

Testing email verification codes

Less ideal options

- Navigating to a web email client in the middle of an automation test, logging in, polling the UI, and extracting the email code when it arrives.
- Building entirely custom infrastructure with your cloud provider to receive an email and then notify the testing client.

testmail.app

testmail.app

PRODUCT PRICING DOCS SIGNUP SIGNIN

Simple email testing

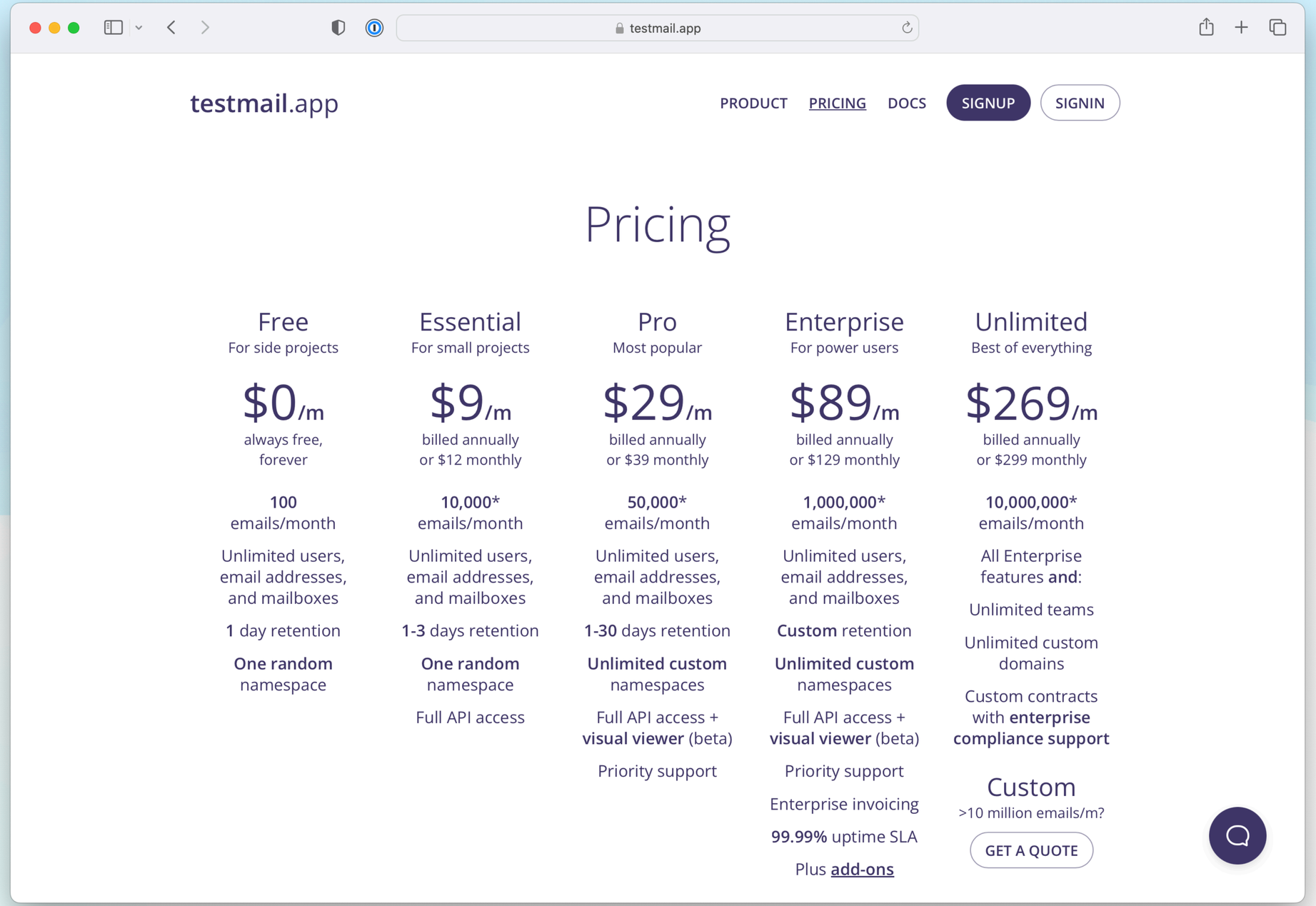
Loved by developers and QA teams

Get **unlimited** email addresses and mailboxes for automating **end-to-end tests** with our simple APIs

- ✓ Test new user signups
  Email verifications, passwordless signins, etc.
- ✓ Test transactional and drip emails
  Are they triggering when expected?
- ✓ Test delivery and spam scores
  Are emails reaching the inbox?

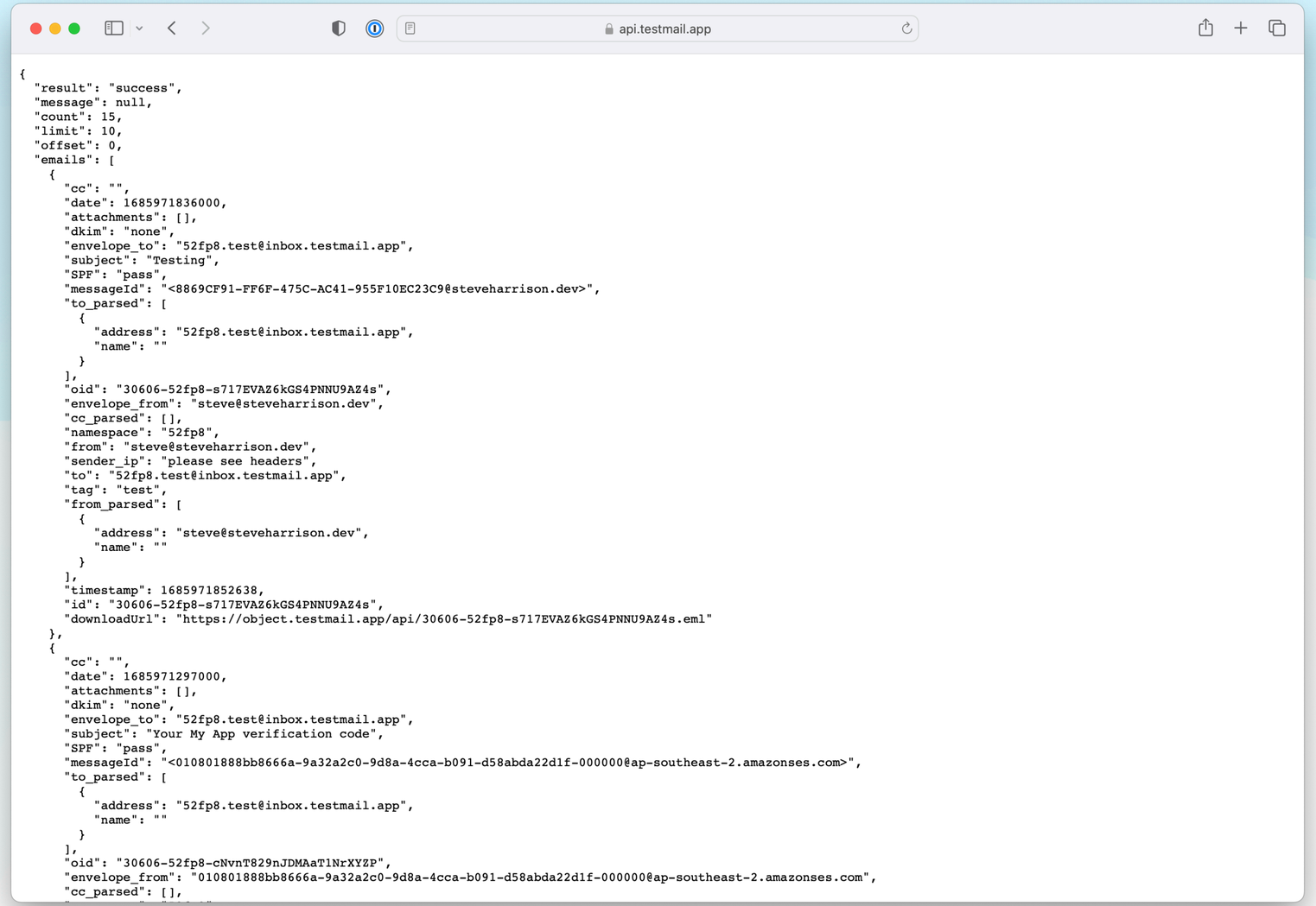
Get started with our **free** forever plan

No credit card required

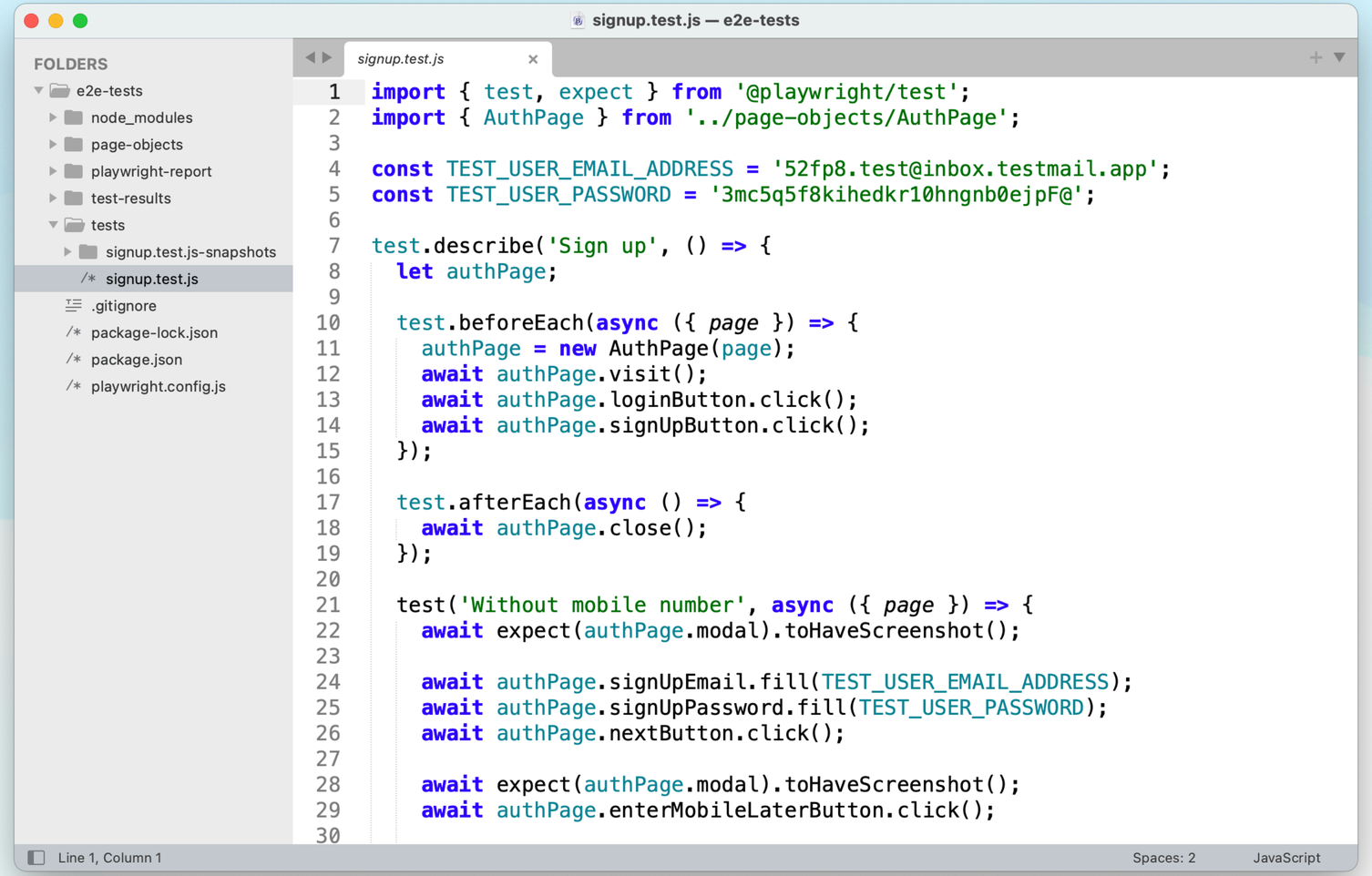

testmail.app
testmail.app
PRODUCT
PRICING
DOCS
SIGNUP
SIGNIN
Pricing
Free
For side projects
$0/m
always free, forever
100 emails/month
Unlimited users, email addresses, and mailboxes
1 day retention
One random namespace
Essential
For small projects
$9/m
billed annually or $12 monthly
10,000* emails/month
Unlimited users, email addresses, and mailboxes
1-3 days retention
One random namespace
Full API access
Pro
Most popular
$29/m
billed annually or $39 monthly
50,000* emails/month
Unlimited users, email addresses, and mailboxes
1-30 days retention
Unlimited custom namespaces
Full API access + visual viewer (beta)
Priority support
Enterprise
For power users
$89/m
billed annually or $129 monthly
1,000,000* emails/month
Unlimited users, email addresses, and mailboxes
Custom retention
Unlimited custom namespaces
Full API access + visual viewer (beta)
Priority support
Enterprise invoicing
99.99% uptime SLA
Plus add-ons
Unlimited
Best of everything
$269/m
billed annually or $299 monthly
10,000,000* emails/month
All Enterprise features and:
Unlimited teams
Unlimited custom domains
Custom contracts with enterprise compliance support
Custom
>10 million emails/m?
GET A QUOTE

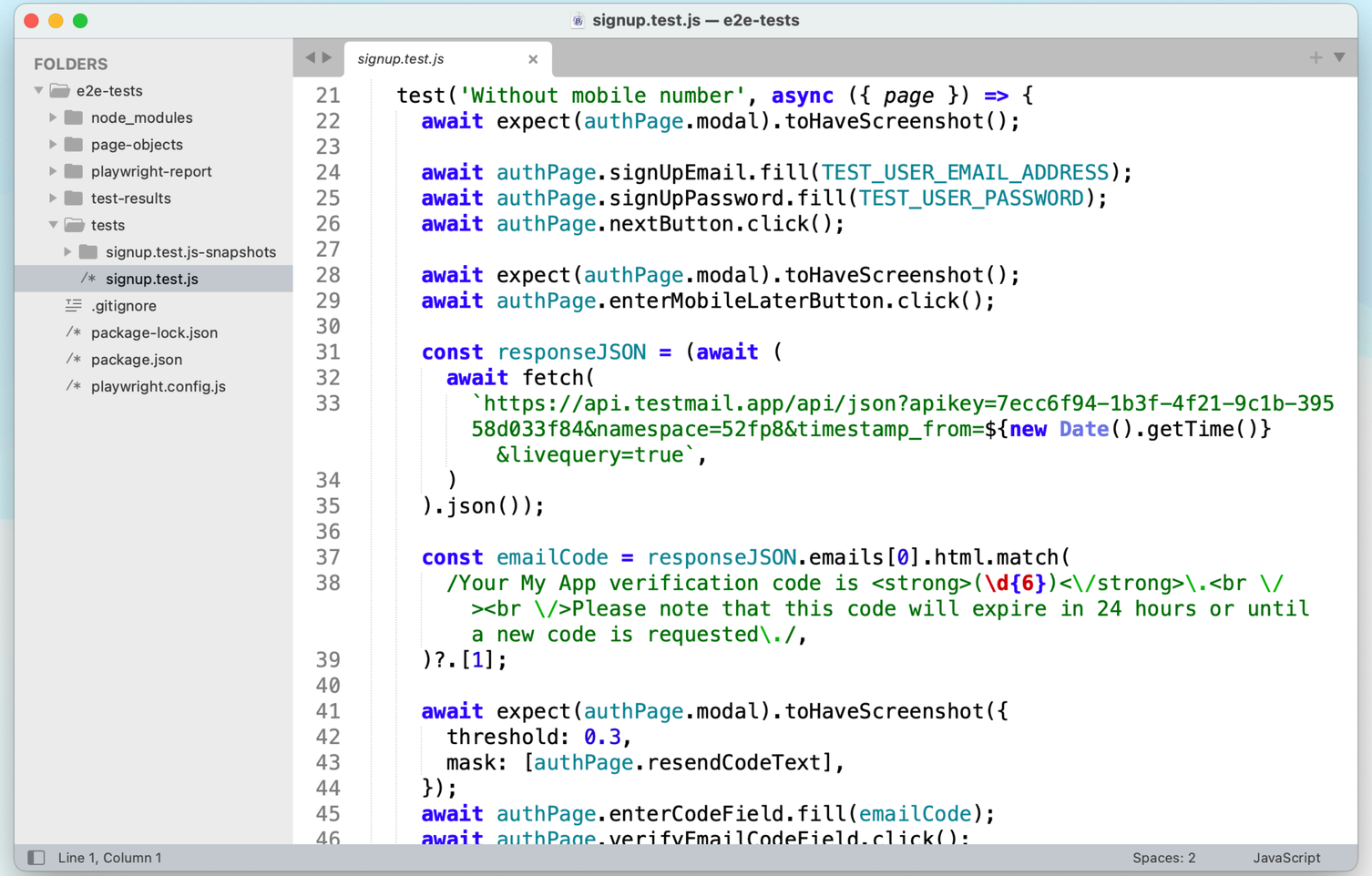
52fp8.test@inbox.testmail.app

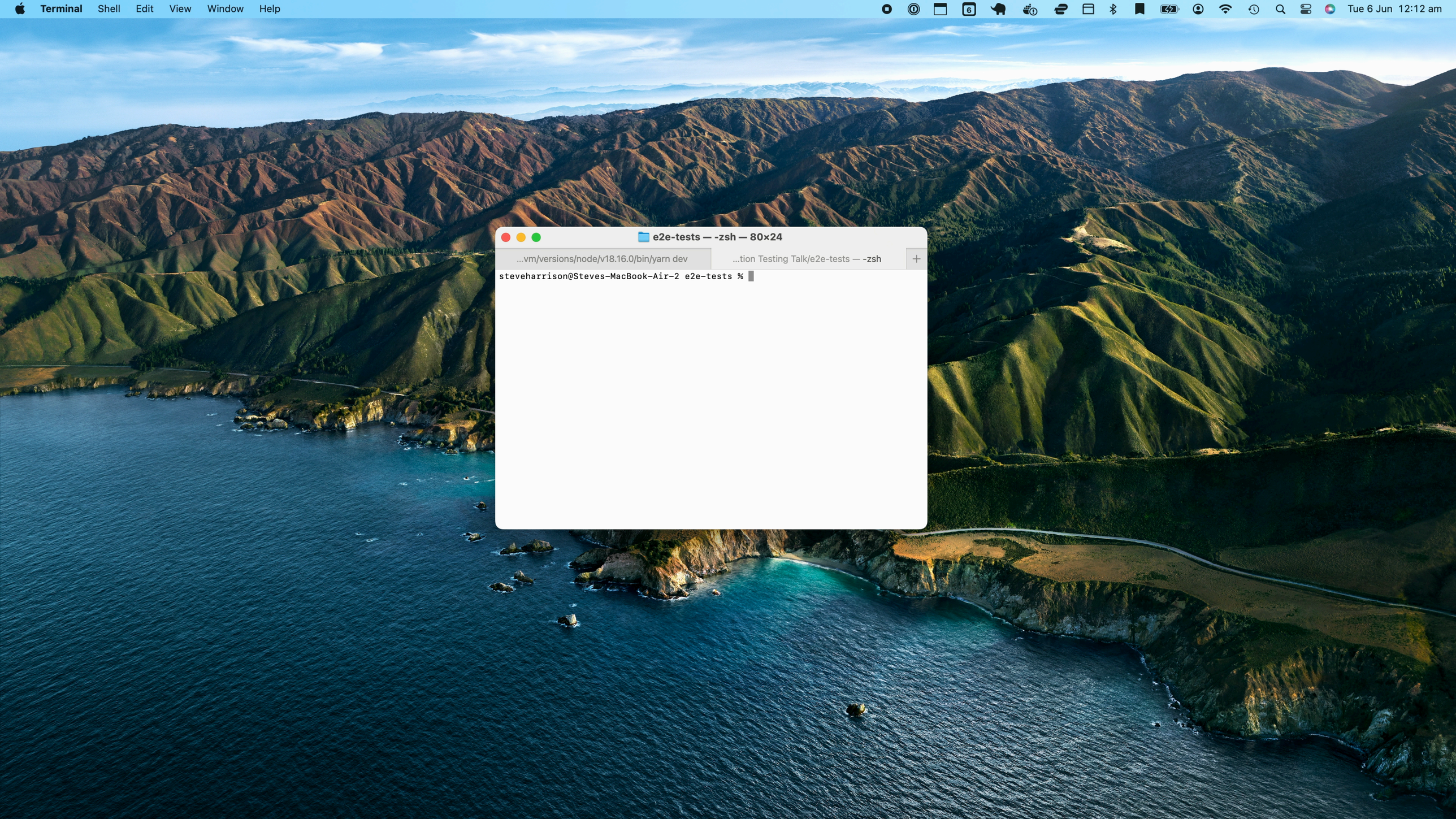
https://api.testmail.app/api/json?apikey=7ecc6f94-1b3f-4f21-9c1b-39558d033f84&namespace=52fp8&pretty=true




```
{
  "result": "success",
  "message": null,
  "count": 15,
  "limit": 10,
  "offset": 0,
  "emails": [
    {
      "cc": "",
      "date": 1685971836000,
      "attachments": [],
      "dkim": "none",
      "envelope_to": "52fp8.test@inbox.testmail.app",
      "subject": "Testing",
      "SPF": "pass",
      "messageId": "<8869CF91-FF6F-475C-AC41-955F10EC23C9@steveharrison.dev>",
      "to_parsed": [
        {
          "address": "52fp8.test@inbox.testmail.app",
          "name": ""
        }
      ],
      "oid": "30606-52fp8-s717EVAZ6kGS4PNNU9AZ4s",
      "envelope_from": "steve@steveharrison.dev",
      "cc_parsed": [],
      "namespace": "52fp8",
      "from": "steve@steveharrison.dev",
      "sender_ip": "please see headers",
      "to": "52fp8.test@inbox.testmail.app",
      "tag": "test",
      "from_parsed": [
        {
          "address": "steve@steveharrison.dev",
          "name": ""
        }
      ],
      "timestamp": 1685971852638,
      "id": "30606-52fp8-s717EVAZ6kGS4PNNU9AZ4s",
      "downloadUrl": "https://object.testmail.app/api/30606-52fp8-s717EVAZ6kGS4PNNU9AZ4s.eml"
    },
    {
      "cc": "",
      "date": 1685971297000,
      "attachments": [],
      "dkim": "none",
      "envelope_to": "52fp8.test@inbox.testmail.app",
      "subject": "Your My App verification code",
      "SPF": "pass",
      "messageId": "<010801888bb8666a-9a32a2c0-9d8a-4cca-b091-d58abda22d1f-000000@ap-southeast-2.amazonses.com>",
      "to_parsed": [
        {
          "address": "52fp8.test@inbox.testmail.app",
          "name": ""
        }
      ],
      "oid": "30606-52fp8-cNvnT829nJDMAaTlNrXYZP",
      "envelope_from": "010801888bb8666a-9a32a2c0-9d8a-4cca-b091-d58abda22d1f-000000@ap-southeast-2.amazonses.com",
      "cc_parsed": [],
```


signup.test.js — e2e-tests
FOLDERS
e2e-tests
node_modules
page-objects
playwright-report
test-results
tests
signup.test.js-snapshots
signup.test.js
.gitignore
package-lock.json
package.json
playwright.config.js
signup.test.js
import { test, expect } from '@playwright/test';
import { AuthPage } from '../page-objects/AuthPage';
const TEST_USER_EMAIL_ADDRESS = '52fp8.test@inbox.testmail.app';
const TEST_USER_PASSWORD = '3mc5q5f8kihedkr10hngnb0ejpF@';
test.describe('Sign up', () => {
let authPage;
test.beforeEach(async ({ page }) => {
authPage = new AuthPage(page);
await authPage.visit();
await authPage.loginButton.click();
await authPage.signUpButton.click();
});
test.afterEach(async () => {
await authPage.close();
});
test('Without mobile number', async ({ page }) => {
await expect(authPage.modal).toHaveScreenshot();
await authPage.signUpEmail.fill(TEST_USER_EMAIL_ADDRESS);
await authPage.signUpPassword.fill(TEST_USER_PASSWORD);
await authPage.nextButton.click();
await expect(authPage.modal).toHaveScreenshot();
await authPage.enterMobileLaterButton.click();
Line 1, Column 1
Spaces: 2
JavaScript


signup.test.js — e2e-tests
FOLDERS
e2e-tests
node_modules
page-objects
playwright-report
test-results
tests
signup.test.js-snapshots
signup.test.js
.gitignore
package-lock.json
package.json
playwright.config.js
signup.test.js
test('Without mobile number', async ({ page }) => {
await expect(authPage.modal).toHaveScreenshot();
await authPage.signUpEmail.fill(TEST_USER_EMAIL_ADDRESS);
await authPage.signUpPassword.fill(TEST_USER_PASSWORD);
await authPage.nextButton.click();
await expect(authPage.modal).toHaveScreenshot();
await authPage.enterMobileLaterButton.click();
const responseJSON = (await (
await fetch(
`https://api.testmail.app/api/json?apikey=7ecc6f94-1b3f-4f21-9c1b-39558d033f84&namespace=52fp8&timestamp_from=${new Date().getTime()}&livequery=true`,
)
).json());
const emailCode = responseJSON.emails[0].html.match(
/Your My App verification code is (\d{6})<\/strong>\.<br \/><br \/>Please note that this code will expire in 24 hours or until a new code is requested\./,
)?.[1];
await expect(authPage.modal).toHaveScreenshot({
threshold: 0.3,
mask: [authPage.resendCodeText],
});
await authPage.enterCodeField.fill(emailCode);
Line 1, Column 1
Spaces: 2
JavaScript


Terminal Shell Edit View Window Help
Tue 6 Jun 12:12 am
e2e-tests — -zsh — 80×24
...vm/versions/node/v18.16.0/bin/yarn dev
...tion Testing Talk/e2e-tests — -zsh
steveharrison@Steves-MacBook-Air-2 e2e-tests %

Visual diffing

Sublime Text File Edit Selection Find View Goto Tools Project Window Help

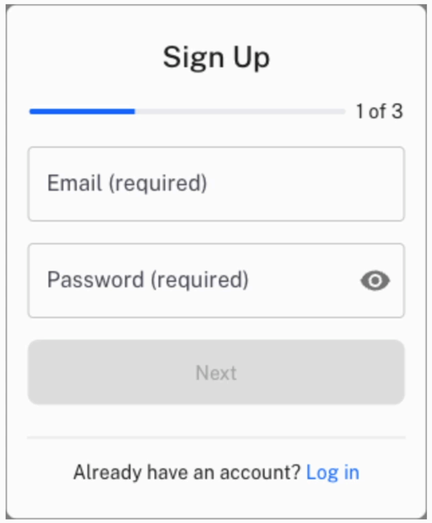
Tue 6 Jun 12:29 am

Sign-up-Without-mobile-number-1-chromium-darwin.png — e2e-tests

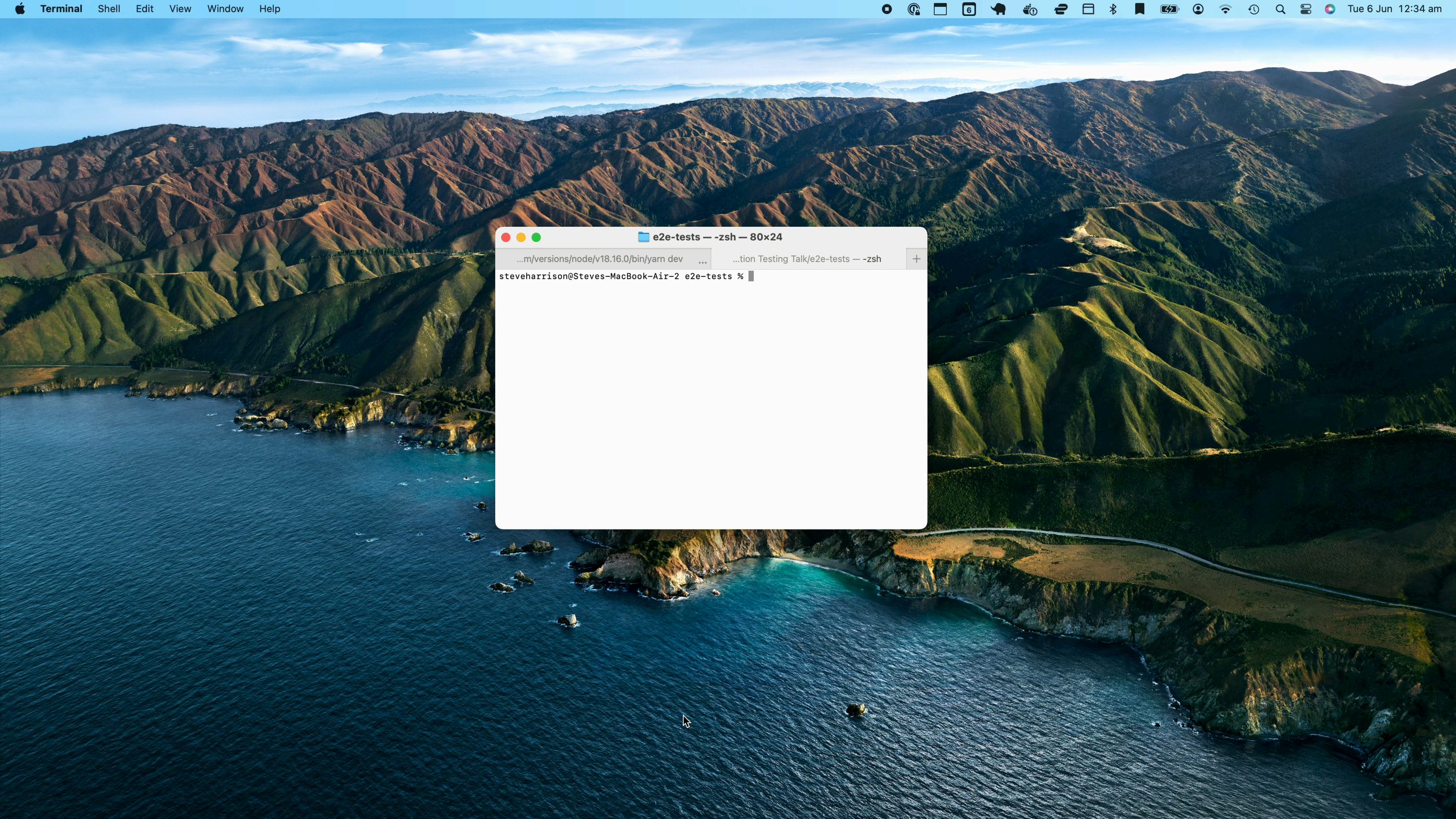
FOLDERS

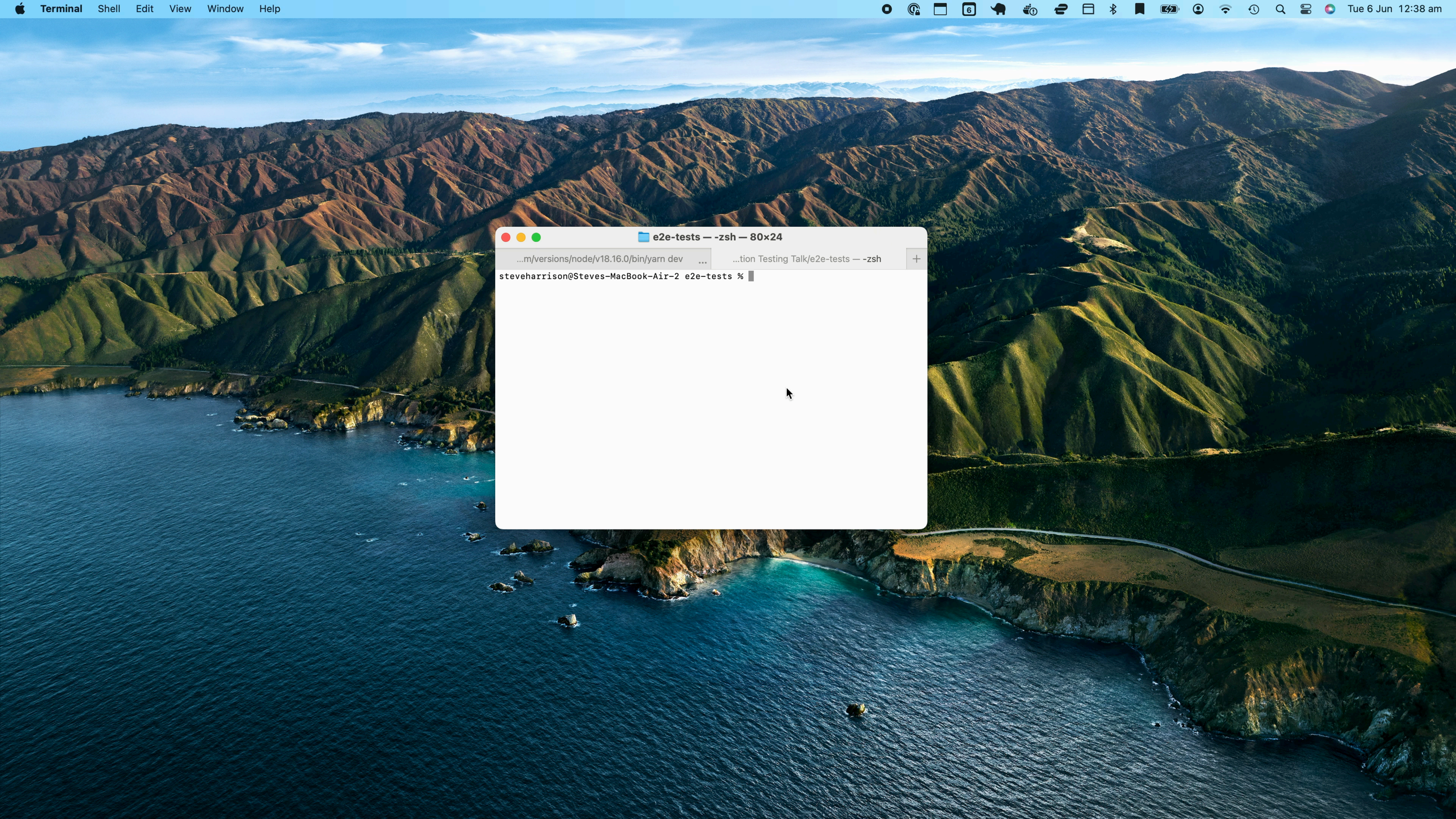
- e2e-tests
  - node_modules
  - page-objects
  - playwright-report
  - test-results
  - tests
    - signup.test.js-snapshots
      - Sign-up-Without-mobile-number-1-chromium-darwin.png
      - Sign-up-Without-mobile-number-2-chromium-darwin.png
      - Sign-up-Without-mobile-number-3-chromium-darwin.png
      - Sign-up-Without-mobile-number-4-chromium-darwin.png
    - signup.test.js
  - .gitignore
  - package-lock.json
  - package.json
  - playwright.config.js

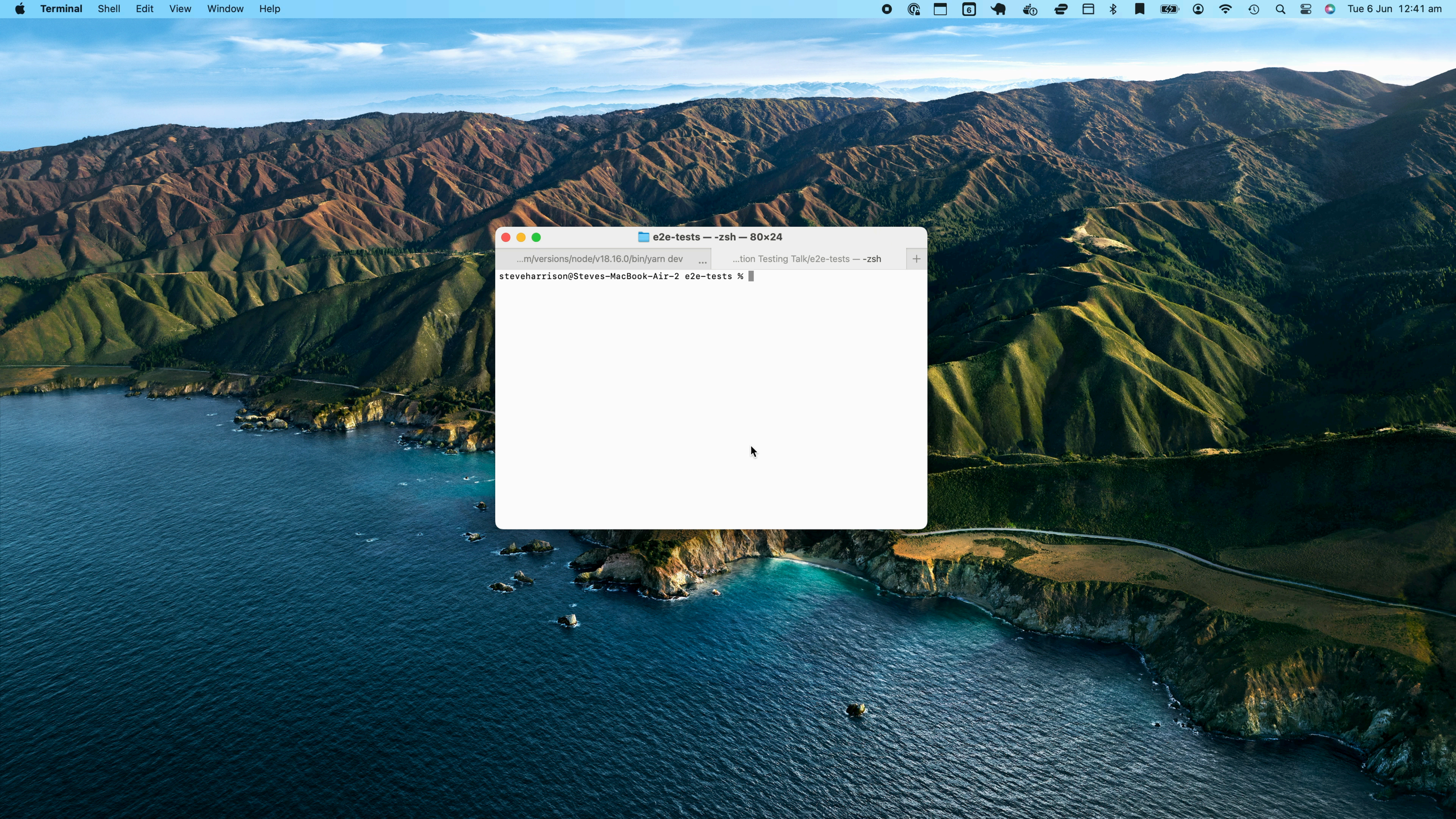
Sign-up-Without-mobile-number-1-chromium-darwin.png

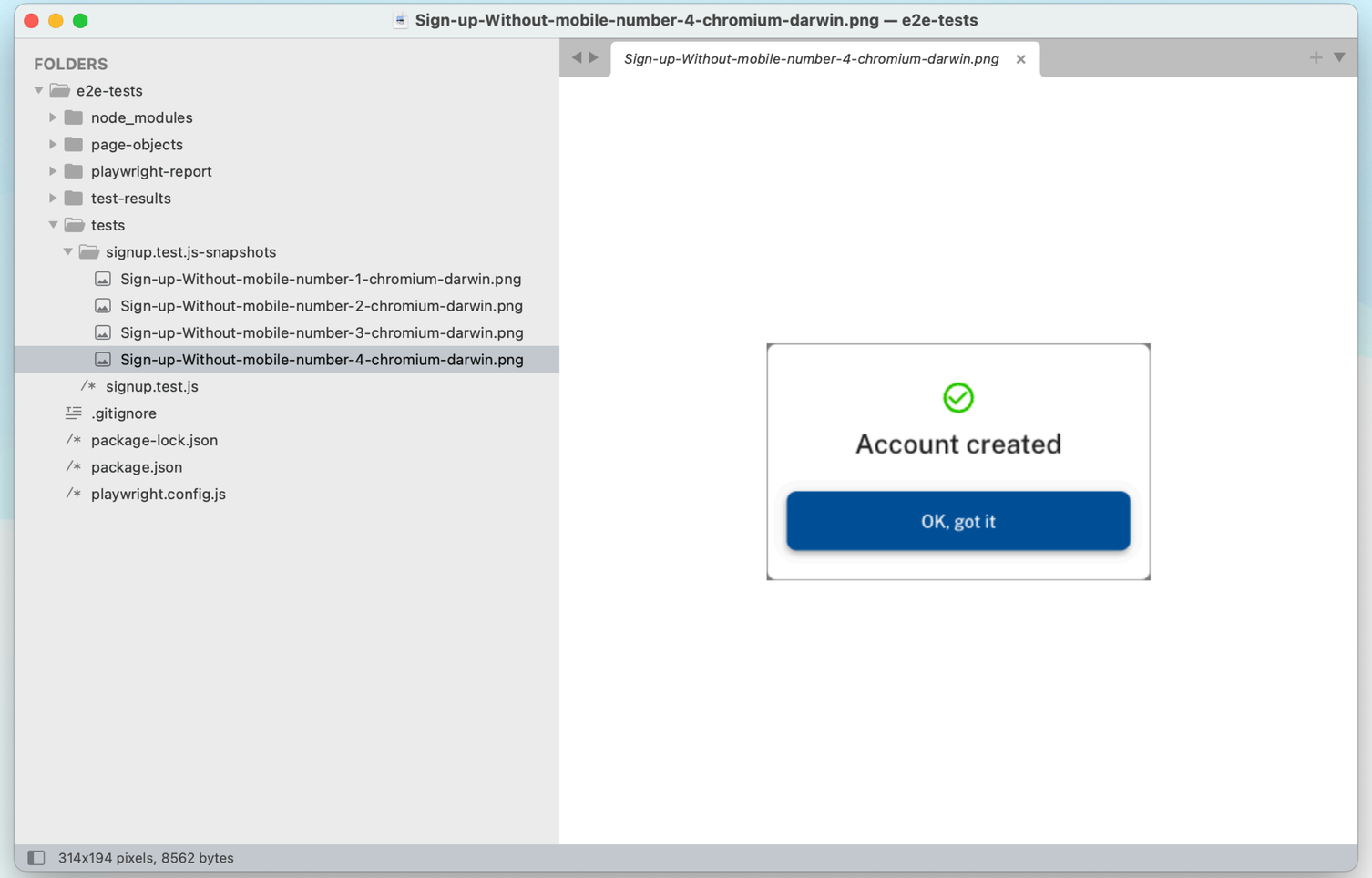


314x384 pixels, 15721 bytes


Terminal Shell Edit View Window Help
Tue 6 Jun 12:34 am
e2e-tests — -zsh — 80×24
...m/versions/node/v18.16.0/bin/yarn dev
...tion Testing Talk/e2e-tests — -zsh
steveharrison@Steves-MacBook-Air-2 e2e-tests %


Terminal Shell Edit View Window Help
Tue 6 Jun 12:38 am
e2e-tests — -zsh — 80×24
...m/versions/node/v18.16.0/bin/yarn dev
...tion Testing Talk/e2e-tests — -zsh
steveharrison@Steves-MacBook-Air-2 e2e-tests %


Terminal Shell Edit View Window Help
Tue 6 Jun 12:41 am
e2e-tests — -zsh — 80×24
...m/versions/node/v18.16.0/bin/yarn dev
...tion Testing Talk/e2e-tests — -zsh
steveharrison@Steves-MacBook-Air-2 e2e-tests %


Sign-up-Without-mobile-number-4-chromium-darwin.png — e2e-tests
FOLDERS
e2e-tests
node_modules
page-objects
playwright-report
test-results
tests
signup.test.js-snapshots
Sign-up-Without-mobile-number-1-chromium-darwin.png
Sign-up-Without-mobile-number-2-chromium-darwin.png
Sign-up-Without-mobile-number-3-chromium-darwin.png
Sign-up-Without-mobile-number-4-chromium-darwin.png
signup.test.js
.gitignore
package-lock.json
package.json
playwright.config.js
Sign-up-Without-mobile-number-4-chromium-darwin.png
Account created
OK, got it
314x194 pixels, 8562 bytes

Testing SMS verification codes

mailslurp.com

MailSlurp

Product Guides Documentation Pricing Support

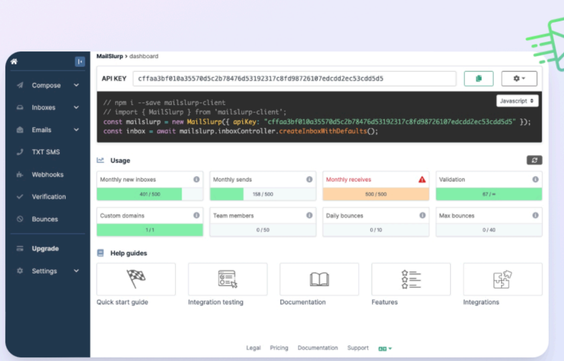
Search ⌘ K

Sign in → Sign up →

Test, build, automate

Powerful **Email** and **SMS** API platforms. Routing, webhooks, mail catchers and forwarding. For developers, QA, and marketing.

Start free account → Documentation →

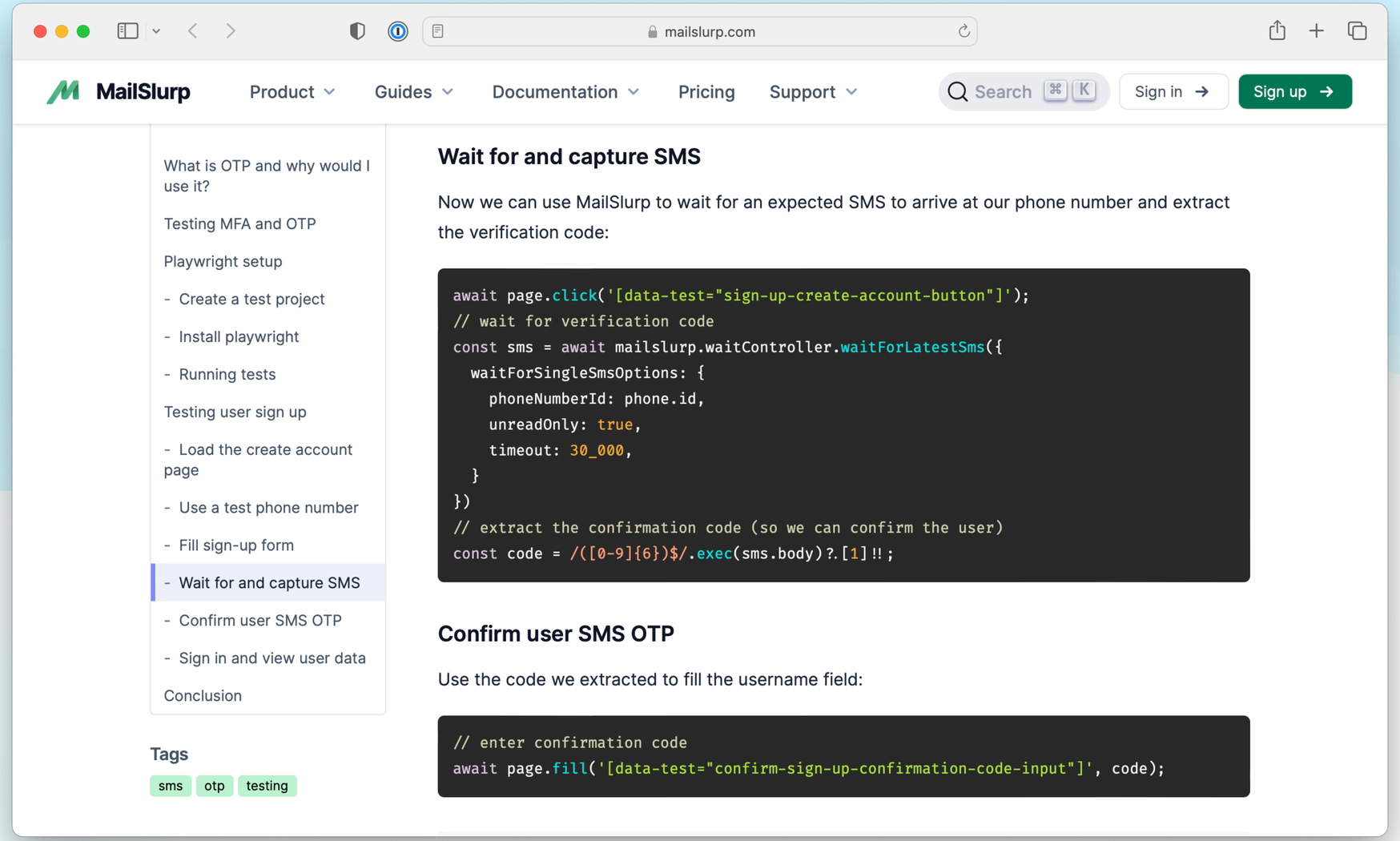


C# Go Java Javascript PHP Python Ruby

```
const MailSlurp = require('mailslurp-client').default;
const mailslurp = new MailSlurp({ apiKey: process.env.API_KEY!! });
// create an email address
```

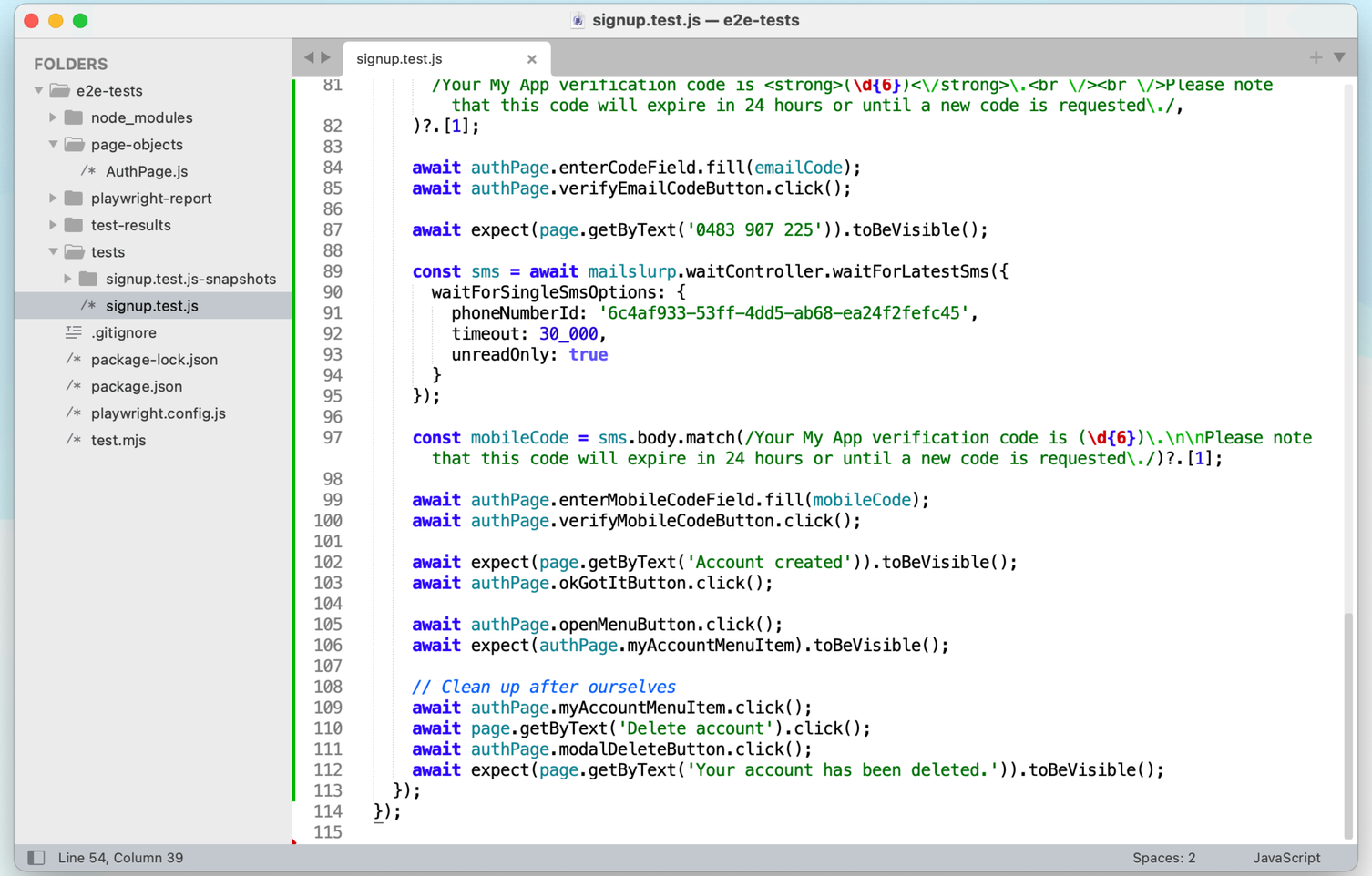
Developer integrations

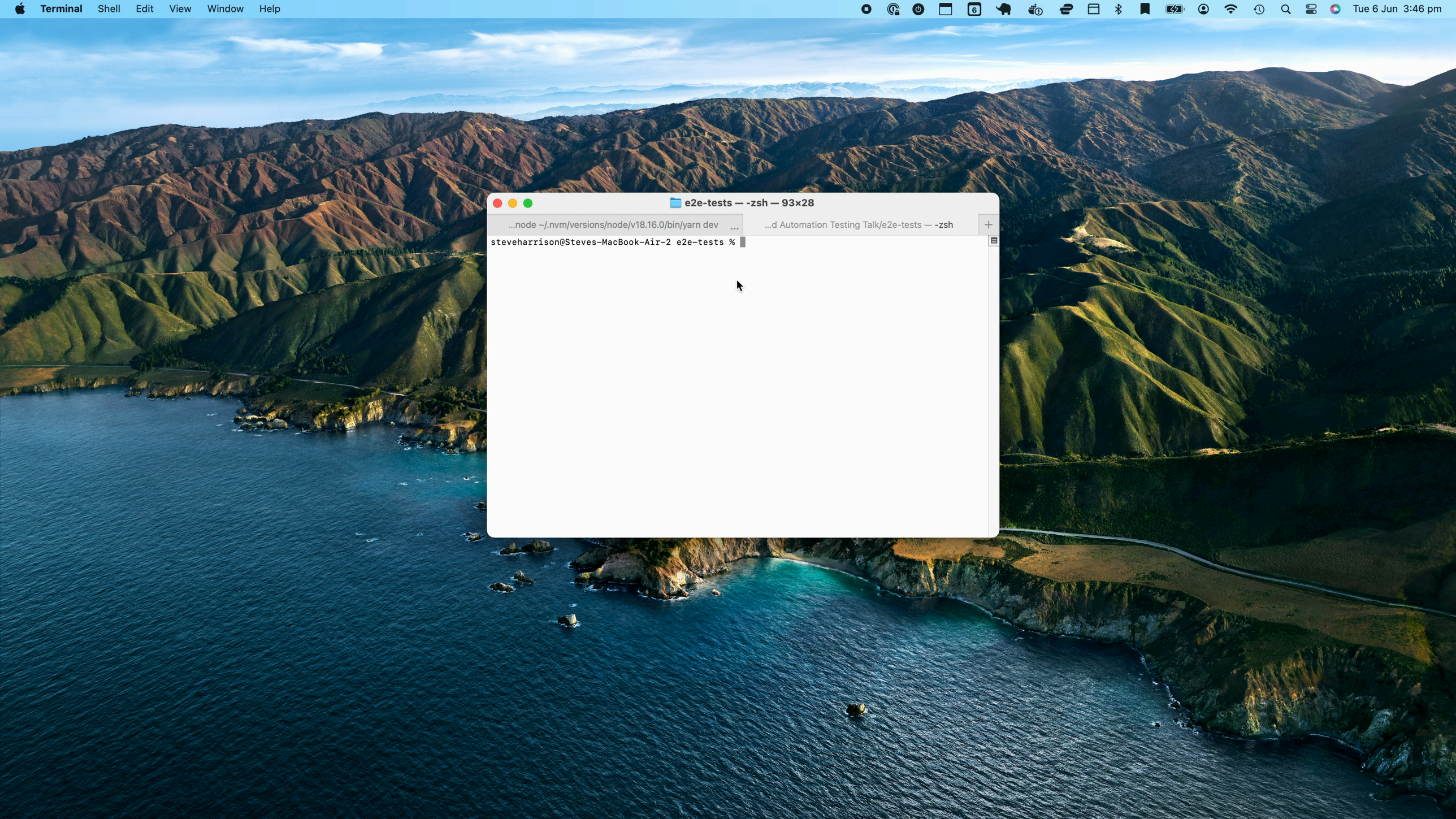
Create and control email accounts and phone numbers in any language or framework.


mailslurp.com
MailSlurp
Product
Guides
Documentation
Pricing
Support
Search ⌘ K
Sign in
Sign up
What is OTP and why would I use it?
Testing MFA and OTP
Playwright setup
- Create a test project
- Install playwright
- Running tests
Testing user sign up
- Load the create account page
- Use a test phone number
- Fill sign-up form
- Wait for and capture SMS
- Confirm user SMS OTP
- Sign in and view user data
Conclusion
Tags
sms
otp
testing
Wait for and capture SMS
Now we can use MailSlurp to wait for an expected SMS to arrive at our phone number and extract the verification code:
await page.click('[data-test="sign-up-create-account-button"]');
// wait for verification code
const sms = await mailslurp.waitController.waitForLatestSms({
waitForSingleSmsOptions: {
phoneNumberId: phone.id,
unreadOnly: true,
timeout: 30_000,
}
})
// extract the confirmation code (so we can confirm the user)
const code = /([0-9]{6})$/.exec(sms.body)?.[1]!!;
Confirm user SMS OTP
Use the code we extracted to fill the username field:
// enter confirmation code
await page.fill('[data-test="confirm-sign-up-confirmation-code-input"]', code);

Pricing

Provider	MailSlurp	Twilio	AWS
Plan	US$20/month	–	–
Dedicated Australian phone number	AU$8/month	US$6.5/month	–
SMS	US$0.10/SMS	US$0.0515/month	–


signup.test.js — e2e-tests
FOLDERS
e2e-tests
node_modules
page-objects
AuthPage.js
playwright-report
test-results
tests
signup.test.js-snapshots
signup.test.js
.gitignore
package-lock.json
package.json
playwright.config.js
test.mjs
signup.test.js
/Your My App verification code is (\d{6})<\/strong>\.<br \/><br \/>Please note
that this code will expire in 24 hours or until a new code is requested\./,
)?.[1];
await authPage.enterCodeField.fill(emailCode);
await authPage.verifyEmailCodeButton.click();
await expect(page.getByText('0483 907 225')).toBeVisible();
const sms = await mailslurp.waitController.waitForLatestSms({
waitForSingleSmsOptions: {
phoneNumberId: '6c4af933-53ff-4dd5-ab68-ea24f2fefc45',
timeout: 30_000,
unreadOnly: true
}
});
const mobileCode = sms.body.match(/Your My App verification code is (\d{6})\.\n\nPlease note
that this code will expire in 24 hours or until a new code is requested\./)?.[1];
await authPage.enterMobileCodeField.fill(mobileCode);
await authPage.verifyMobileCodeButton.click();
await expect(page.getByText('Account created')).toBeVisible();
await authPage.okGotItButton.click();
await authPage.openMenuButton.click();
await expect(authPage.myAccountMenuItem).toBeVisible();
// Clean up after ourselves
await authPage.myAccountMenuItem.click();
await page.getByText('Delete account').click();
await authPage.modalDeleteButton.click();
await expect(page.getByText('Your account has been deleted.')).toBeVisible();
});
});
Line 54, Column 39
Spaces: 2
JavaScript


Terminal
Shell
Edit
View
Window
Help
Tue 6 Jun 3:46 pm
e2e-tests — -zsh — 93×28
...node ~/.nvm/versions/node/v18.16.0/bin/yarn dev
...d Automation Testing Talk/e2e-tests — -zsh
steveharrison@Steves-MacBook-Air-2 e2e-tests %

Thank you

https://www.steveharrison.dev