CYPRESS VS PLAYWRIGHT

WORKSHOP

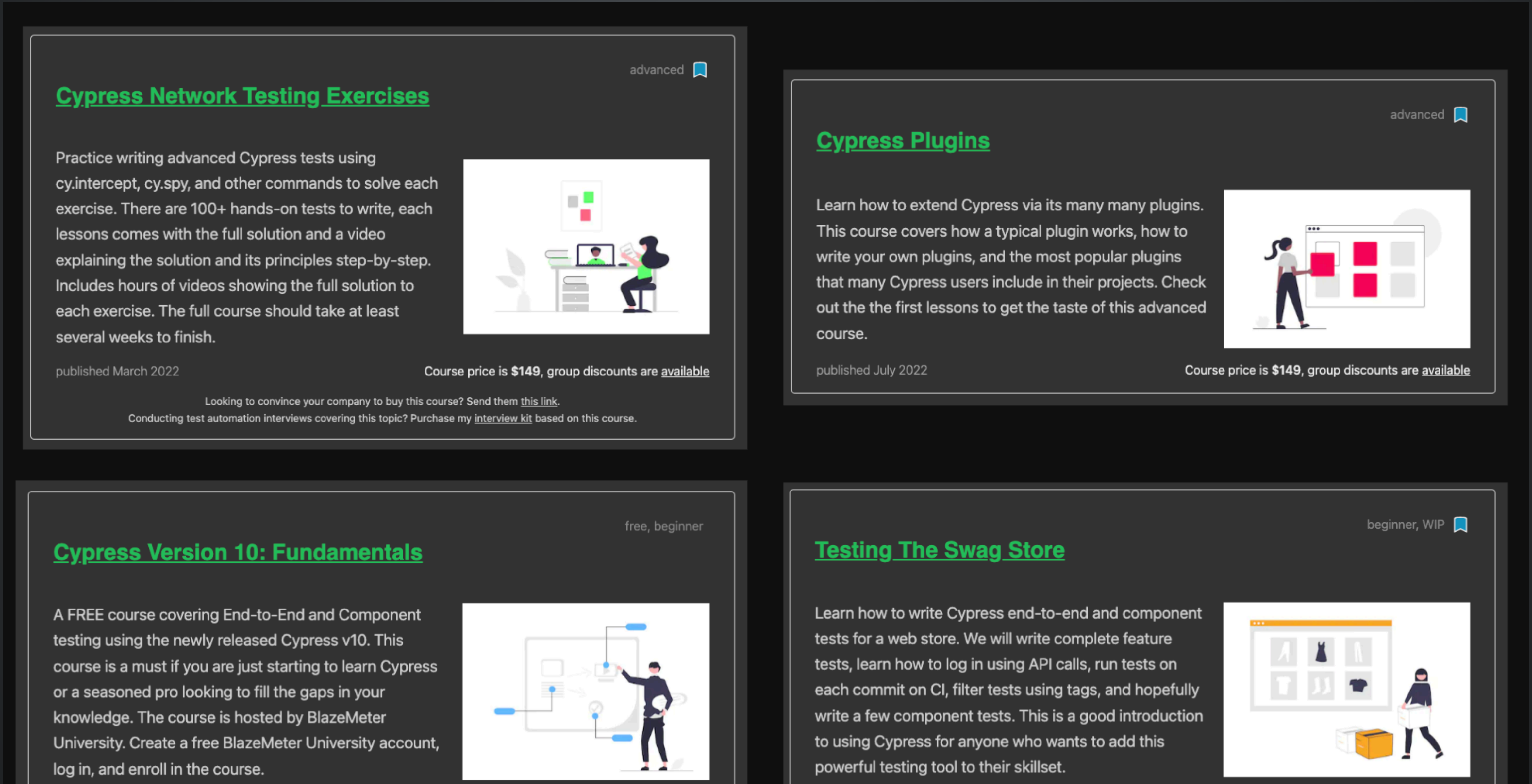
- github.com/bahmutov/cypress-workshop-cy-vs-pw

Jump to: 00-start, 01-basic, 02-adding-items, 03-completing-items, 04-test-ui, 05-hover, 06-network, 07-clock, 08-retries, 09-app-access, 10-ci, 11-component-tests, end

AUTHOR: GLEB BAHMUTOV, PHD

- Ex-VP of Engineering at Cypress
- Ex-Distinguished Engineer at Cypress
- actively using Cypress since 2016
- gleb.dev
- 🦋 bahmutov.bsky.social
- https://glebbahmutov.com/blog/tags/cypress/ 300+ Cypress blog posts
- https://www.youtube.com/glebbahmutov 700+ Cypress videos
- cypress.tips with links, search, my courses
- Cypress Tips monthly newsletter

cypress.tips/courses



advanced

Cypress Network Testing Exercises

Practice writing advanced Cypress tests using cy.intercept, cy.spy, and other commands to solve each exercise. There are 100+ hands-on tests to write, each lessons comes with the full solution and a video explaining the solution and its principles step-by-step. Includes hours of videos showing the full solution to each exercise. The full course should take at least several weeks to finish.

published March 2022

Course price is **$149**, group discounts are available

Looking to convince your company to buy this course? Send them this link.
Conducting test automation interviews covering this topic? Purchase my interview kit based on this course.

advanced

Cypress Plugins

Learn how to extend Cypress via its many many plugins. This course covers how a typical plugin works, how to write your own plugins, and the most popular plugins that many Cypress users include in their projects. Check out the the first lessons to get the taste of this advanced course.

published July 2022

Course price is **$149**, group discounts are available

free, beginner

Cypress Version 10: Fundamentals

A FREE course covering End-to-End and Component testing using the newly released Cypress v10. This course is a must if you are just starting to learn Cypress or a seasoned pro looking to fill the gaps in your knowledge. The course is hosted by BlazeMeter University. Create a free BlazeMeter University account, log in, and enroll in the course.

beginner, WIP

Testing The Swag Store

Learn how to write Cypress end-to-end and component tests for a web store. We will write complete feature tests, learn how to log in using API calls, run tests on each commit on CI, filter tests using tags, and hopefully write a few component tests. This is a good introduction to using Cypress for anyone who wants to add this powerful testing tool to their skillset.

CAN GLEB BE FAIR?

I will try.

WHAT WE ARE GOING TO COVER

- example TodoMVC
  - web app, data store, REST calls
- basic tests using Pw and Cy
- using JSON fixture files
- making HTTP requests
- network spying and stubbing, fixtures
- controlling the application clock
- assertions and retries
- component testing (React)
- any questions

EXAMPLE REPOS

Please clone these repositories, since we will be using through the workshop:

- https://github.com/bahmutov/cy-vs-pw-example-todomvc
- https://github.com/bahmutov/cy-vs-pw-ci-example
- https://github.com/bahmutov/taste-the-sauce-vite

SCHEDULE 🕰

1 Day

- 9:00 start
- 10:30 coffee break
- 12:30 lunch break
- 13:30 afternoon start
- 15:00 coffee break
- ~16:30 the end
- time for questions during the workshop and after each section

TEST TOPICS

- example TodoMVC
  - web app, data store, REST calls
- basic tests using Pw and Cy
- using JSON fixture files
- making HTTP requests
- network spying and stubbing, fixtures
- writing E2E tests practice time (end of network chapter)
  - "TodoMVC app"
- controlling the application clock
- assertions and retries
- component testing (React)
- writing E2E and component tests practice time

POLL 1 🗳️: HAVE YOU USED CYPRESS BEFORE?

- This is my first time
- Using for less than 1 month 👍
- Using it for less than 1 year 👍👍
- Using for longer than 1 year ❤️
- Using for longer than 2 years ❤️❤️

POLL 2 🗳️: HAVE YOU USED PLAYWRIGHT BEFORE?

- This is my first time
- Using for less than 1 month 👍
- Using it for less than 1 year 👍👍
- Using for longer than 1 year ❤️
- Using for longer than 2 years ❤️❤️

EXAMPLE APPLICATION

We will use at first the app in the repo bahmutov/cy-vs-pw-example-todomvc

- each exercise starts in the prepared branch: `a1`, `a2`, `a3`, etc
- `git checkout a1` for example
- `npm install`
- `npx playwright install` (if using Playwright in the current lesson)
- `npx cypress install` (if needed for Cypress)

HOW EFFICIENT LEARNING WORKS

1. I explain and show
2. We do together
3. You do and I help

Tip: this repository has everything to work through the test exercises.

bahmutov/cy-vs-pw-example-todomvc

🎓 This workshop covers a subset of lessons from my course Cypress vs Playwright

The repo has the starting code in multiple branches.

REQUIREMENTS

You will need:

- `git` to clone the example repo
- Node v20+ to install dependencies

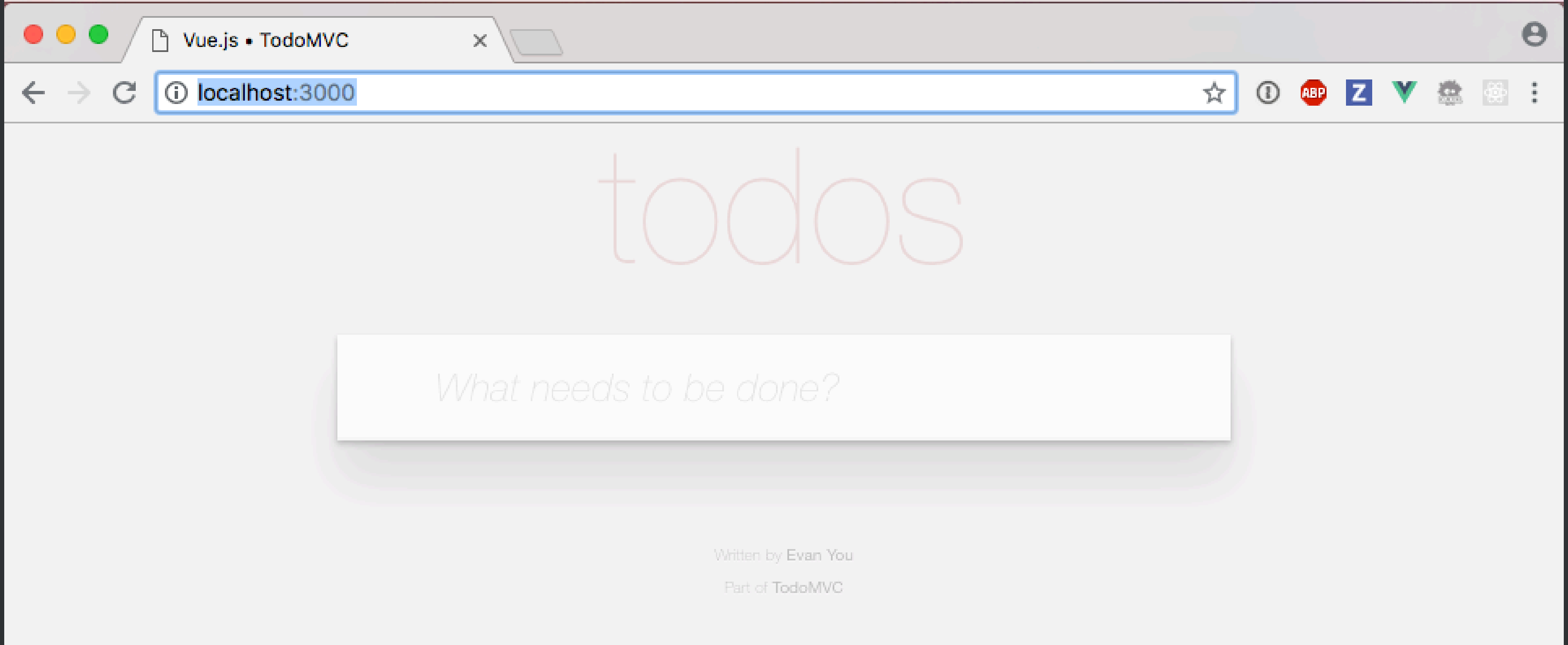
```
git clone git@github.com:bahmutov/cy-vs-pw-example-todomvc.git
cd cy-vs-pw-example-todomvc
git checkout <branch name>
npm install
```

THE APP `todomvc`

Let us look at the application.

- `cd cy-vs-pw-example-todomvc`
- `git checkout main`
- `npm install`
- `npm start`
- `open localhost:3000`

It is a regular TodoMVC application.



Vue.js • TodoMVC

localhost:3000

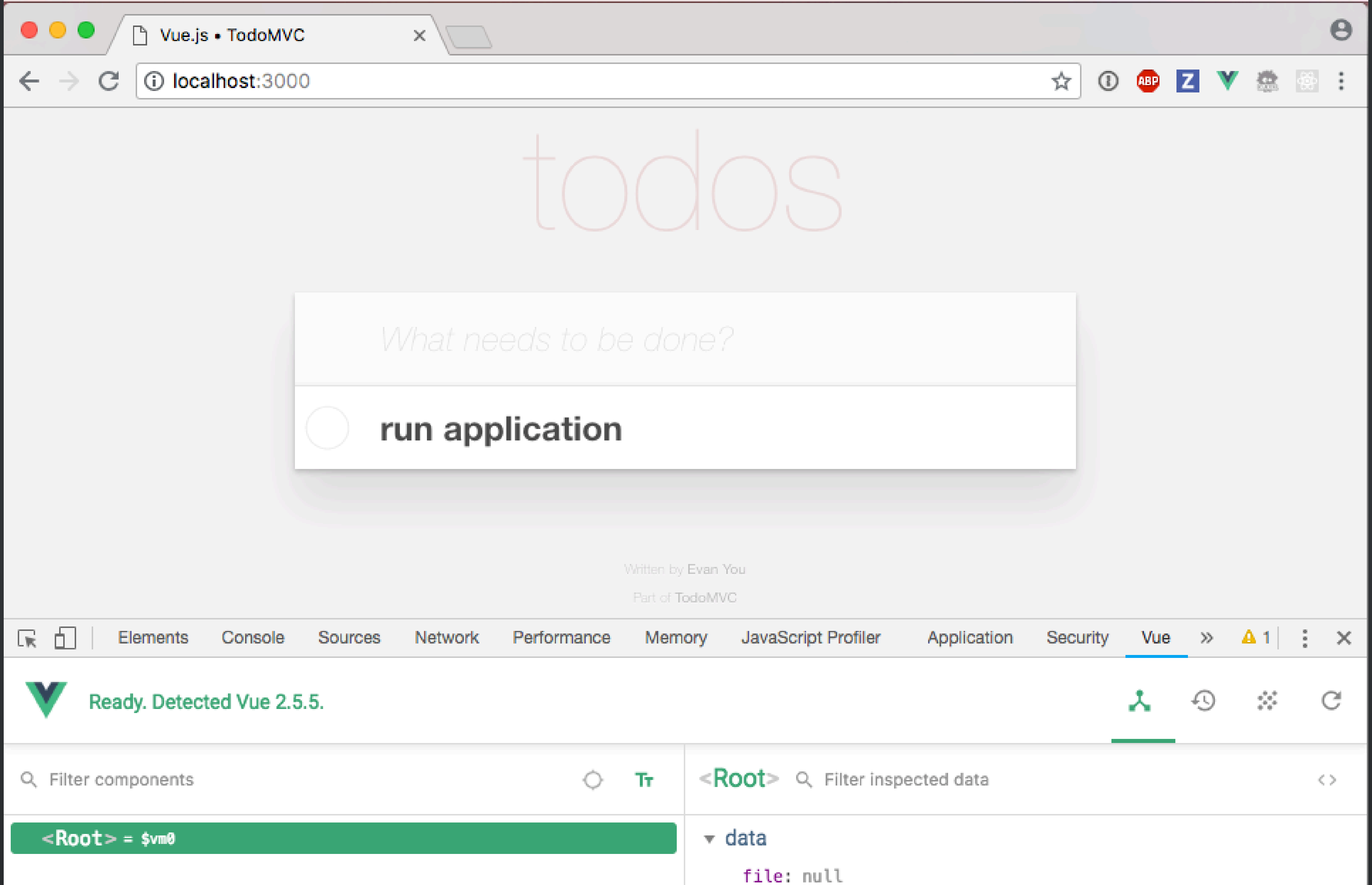
todos

What needs to be done?

Written by Evan You

Part of TodoMVC

If you have Vue DevTools plugin

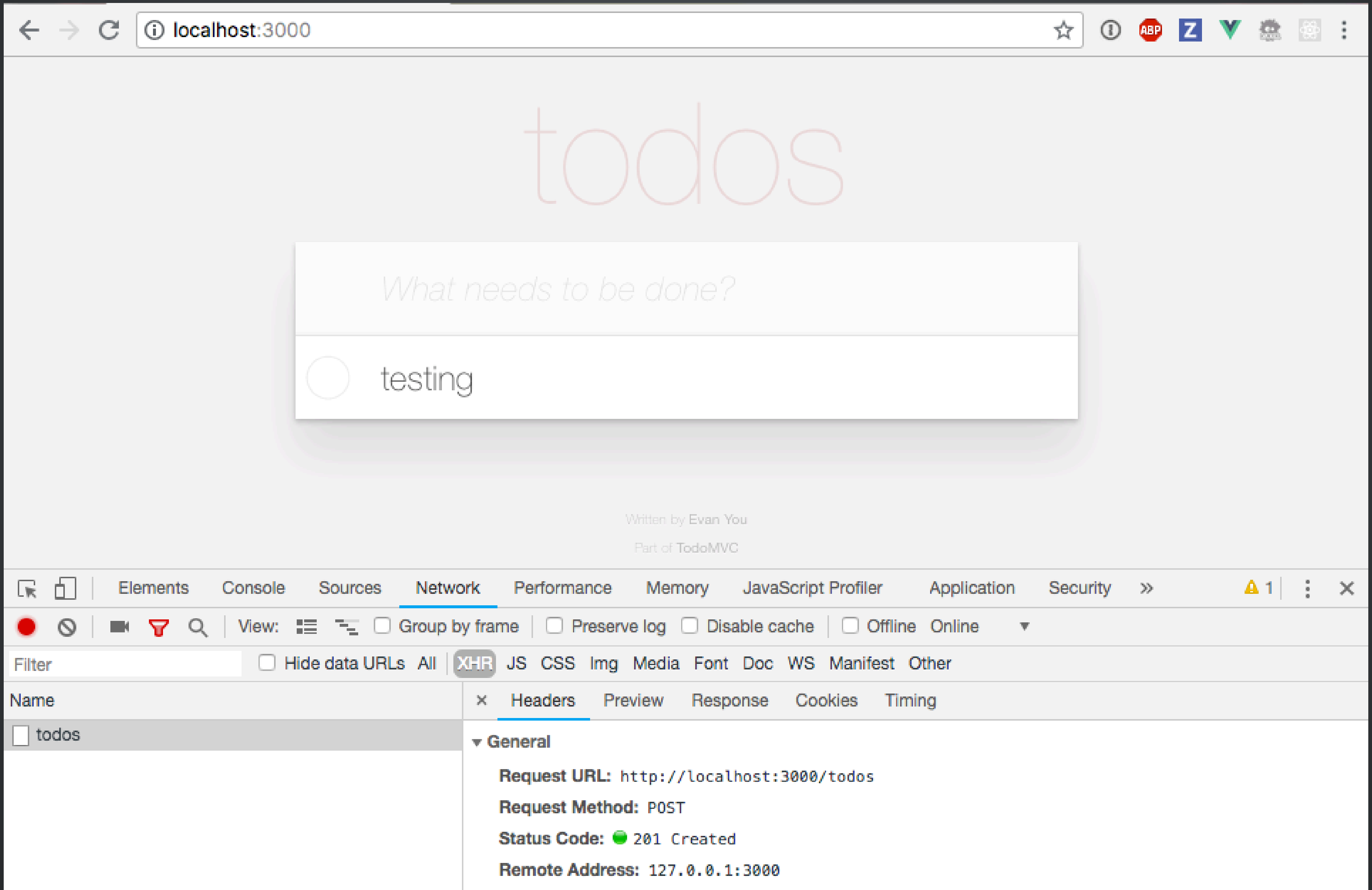


Ready. Detected Vue 2.5.5.

```
<Root> = $vm0
```

```
▾ data
    file: null
```

Look at XHR when using the app



localhost:3000
todos
What needs to be done?
testing
Written by Evan You
Part of TodoMVC
Elements Console Sources Network Performance Memory JavaScript Profiler Application Security
View: Group by frame Preserve log Disable cache Offline Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
Name
todos
Headers Preview Response Cookies Timing

General

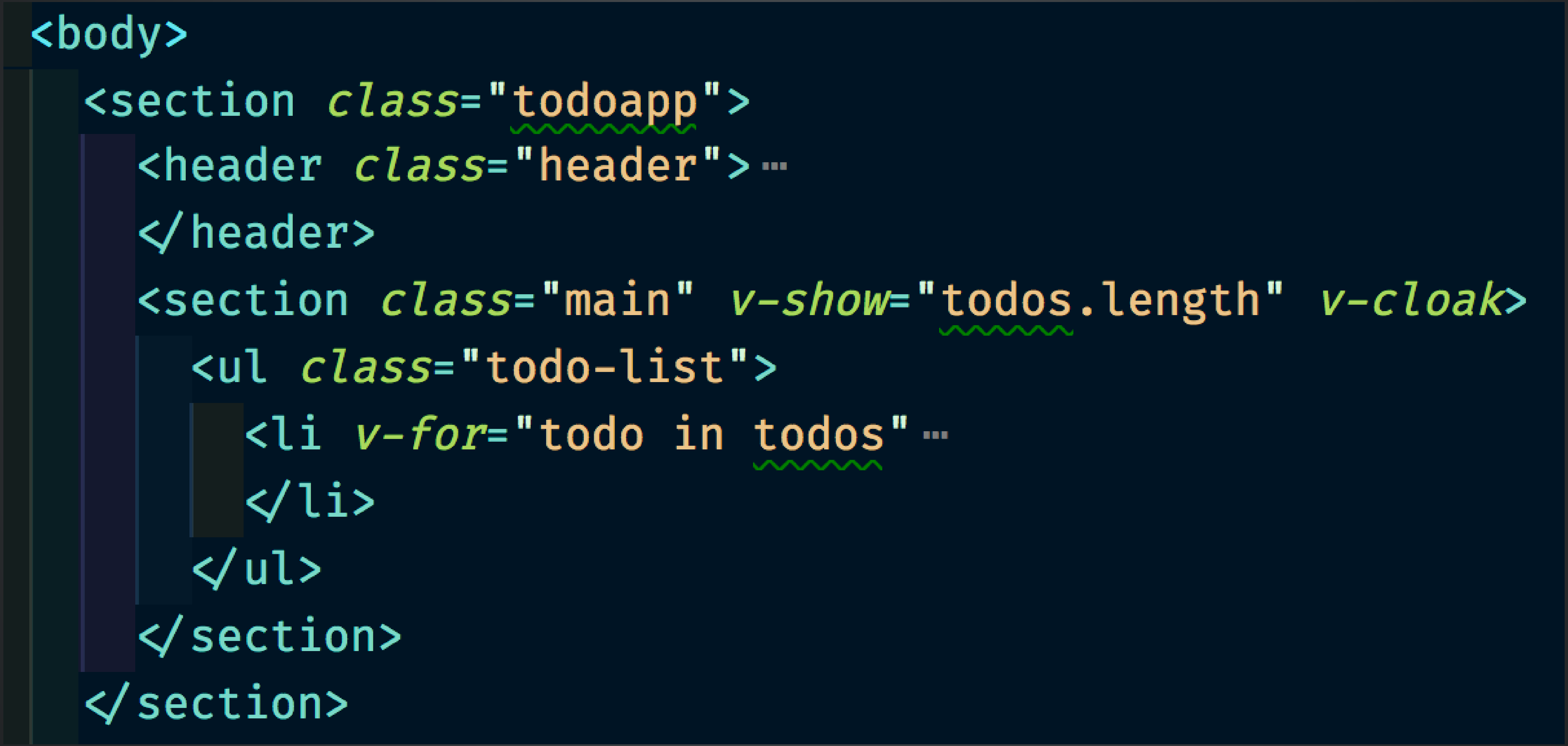
Request URL: `http://localhost:3000/todos`

Request Method: `POST`

Status Code: ● `201 Created`

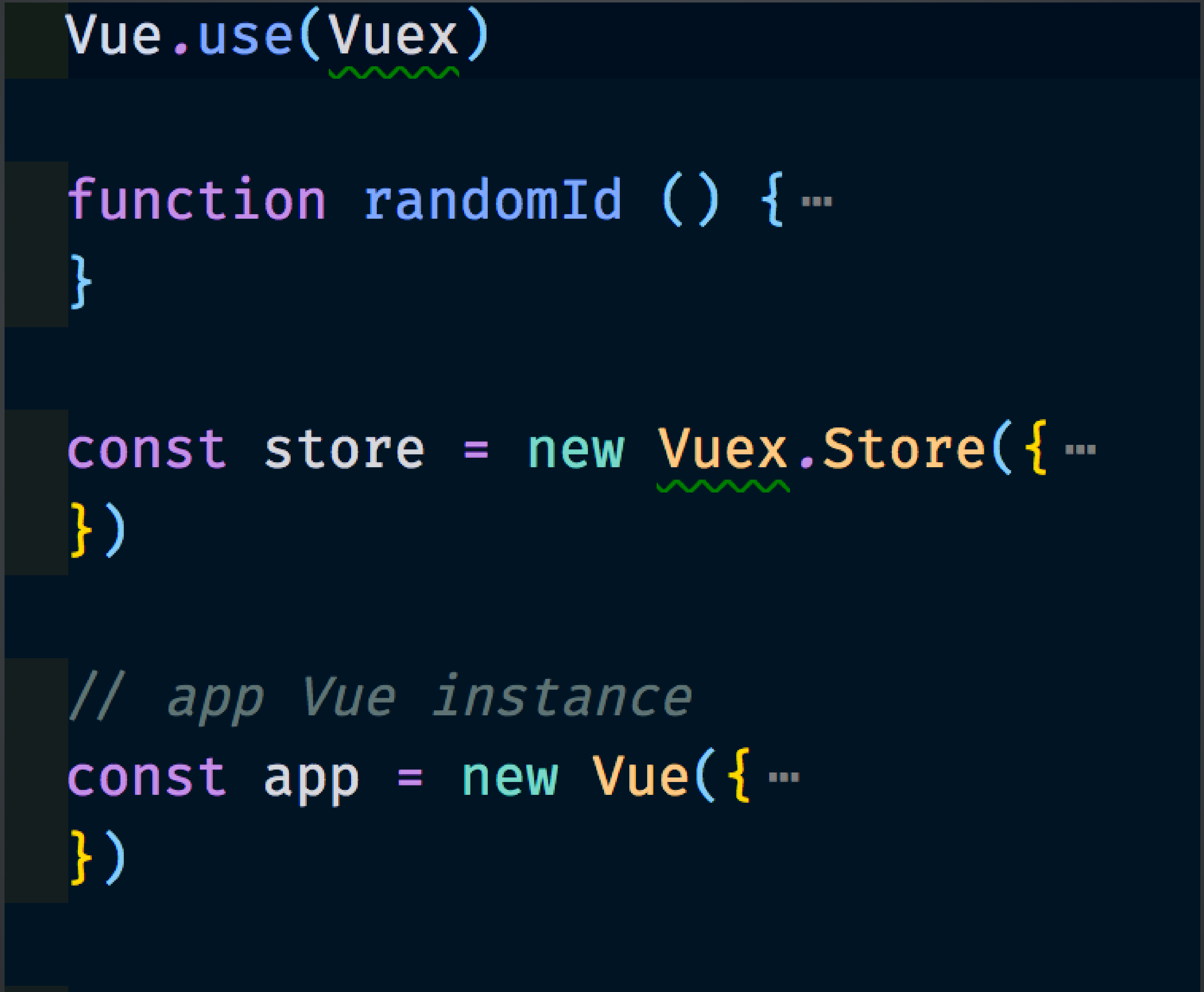
Remote Address: `127.0.0.1:3000`

Look at `todomvc/index.html` - main app DOM structure



```html
<body>
  <section class="todoapp">
    <header class="header">⋯
    </header>
    <section class="main" v-show="todos.length" v-cloak>
      <ul class="todo-list">
        <li v-for="todo in todos"⋯
        </li>
      </ul>
    </section>
  </section>
```

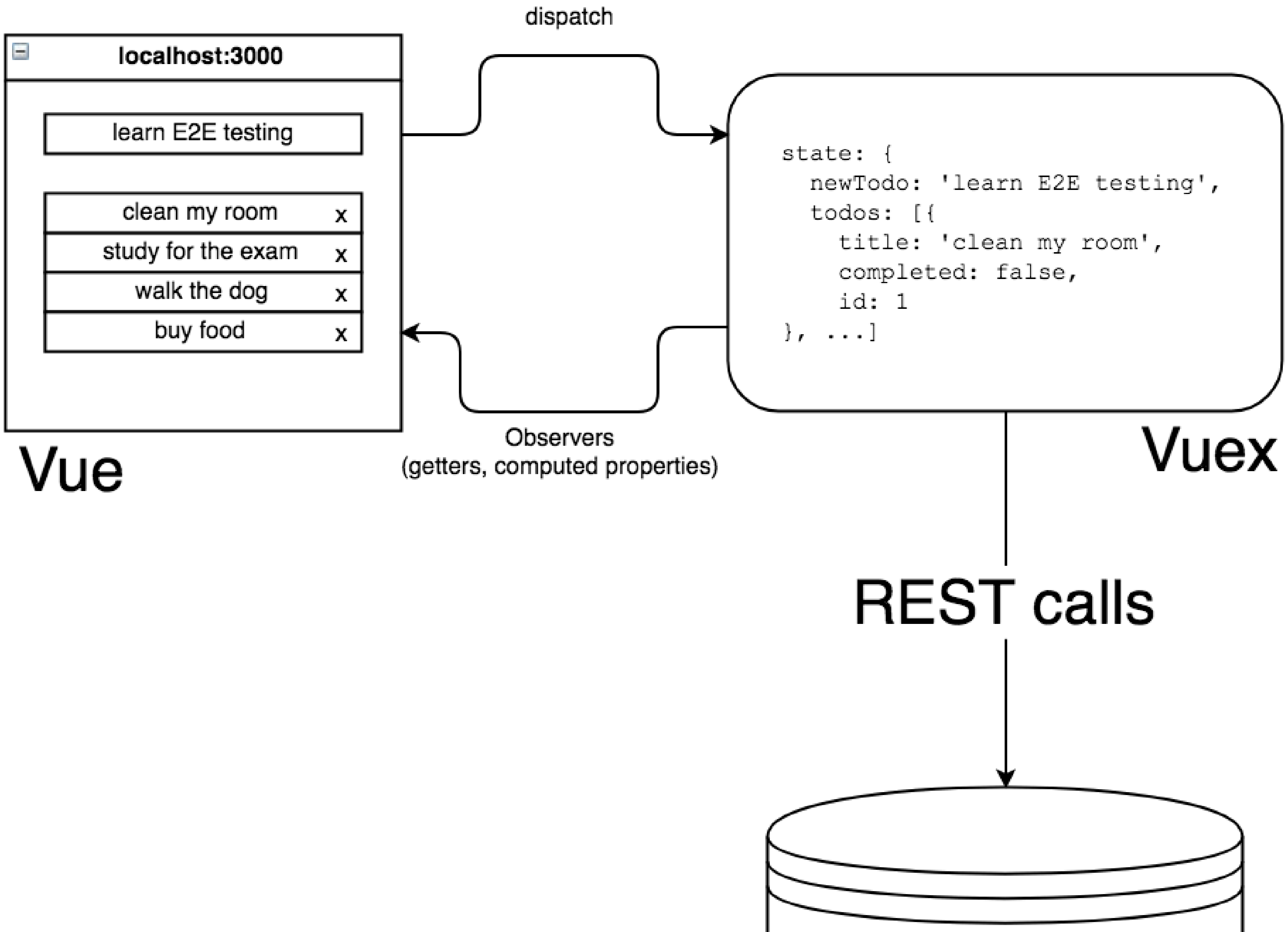
Look at todomvc/app.js



```
Vue.use(Vuex)

function randomId () {⋯
}

const store = new Vuex.Store({⋯
})

// app Vue instance
const app = new Vue({⋯
})
```

QUESTIONS

- what happens when you add a new Todo item?
- how does it get to the server?
- where does the server save it?
- what happens on start up?



```
state: {
  newTodo: 'learn E2E testing',
  todos: [{
    title: 'clean my room',
    completed: false,
    id: 1
}, ...]
```

This app has been coded and described in this blog post https://www.cypress.io/blog/2017/11/28/testing-vue-web-application-with-vuex-data-store-and-rest-backend/

END OF INTRODUCTION

➡️ Pick the next section or go to the 00-start chapter