Chapter 1: TypeScript Basics

```
function calculateTotalPrice(product, quantity, discount) {
  var priceWithoutDiscount = product.price * quantity;
  var discountAmount = priceWithoutDiscount * discount;
  return priceWithoutDiscount - discountAmount;
}

```

```
interface IProduct {
  name: string;
  unitPrice: number;
}

function calculateTotalPrice(product: IProduct, quantity: number, discount: number): number {
  var priceWithoutDiscount: number = product.price * quantity;
  var discountAmount: number = priceWithoutDiscount * discount;
  return priceWithoutDiscount - discountAmount;
}
```

```
function calculateTotalPrice(product: IProduct, quantity: number, discount: number): number {
  var priceWithoutDiscount: number = product. * quantity;
  var discountAmount: number = priceWithoutDi
  return priceWithoutDiscount - discountAmoun
}
```

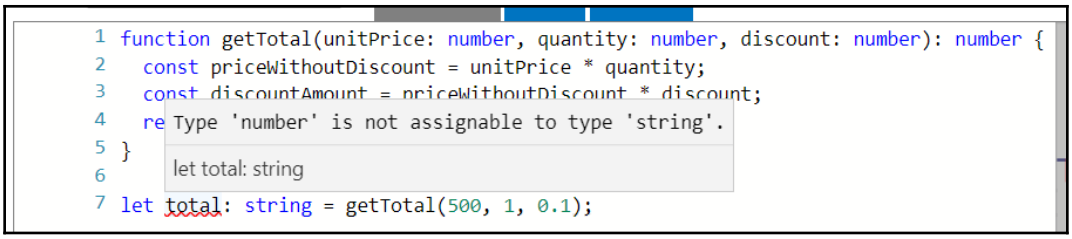
name
unitPrice (property) IProduct.unitPrice: number

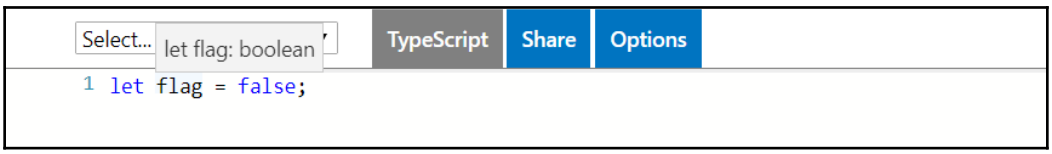
```
Type '"Table"' is not assignable to type 'number'.
let unitPrice: number
1
2 unitPrice = "Table";
```

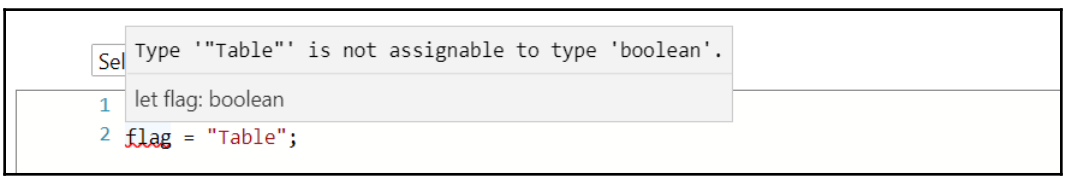
```
1 var unitPrice;
2 unitPrice = "Table";
3
```

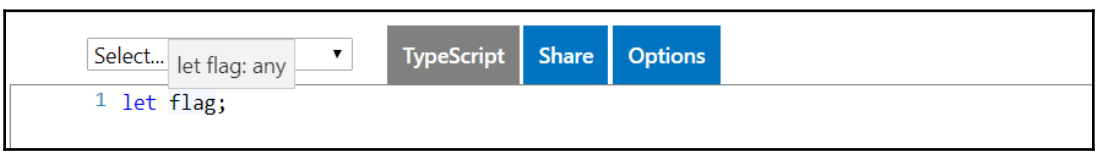
```
function getTotal(unitPrice: number, quantity: number, discount: number): number {
  const priceWithoutDiscount = unitPrice * quantity;
  const discountAmount = priceWithoutDiscount * discount;
  return priceWithoutDiscount - di
}

let total: string = getTotal(500, "one", 0.1);
```

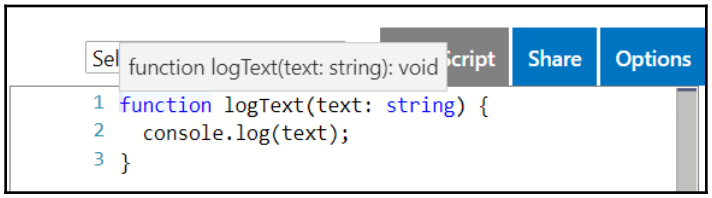
Argument of type '"one"' is not assignable to parameter of type 'number'.

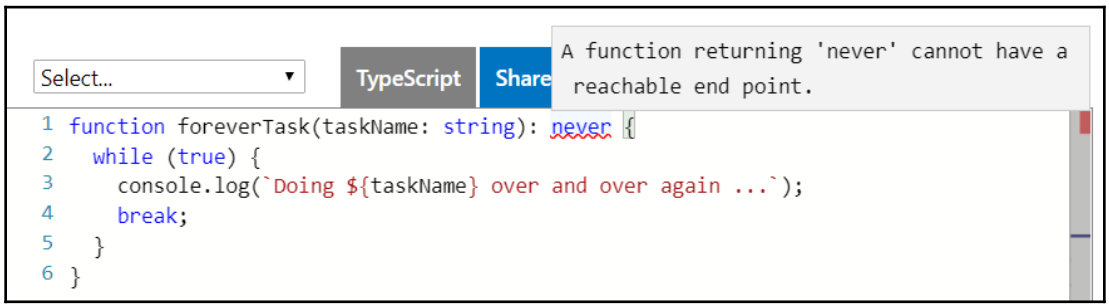

function getTotal(unitPrice: number, quantity: number, discount: number): number {
const priceWithoutDiscount = unitPrice * quantity;
Type 'number' is not assignable to type 'string'.
let total: string
}
let total: string = getTotal(500, 1, 0.1);

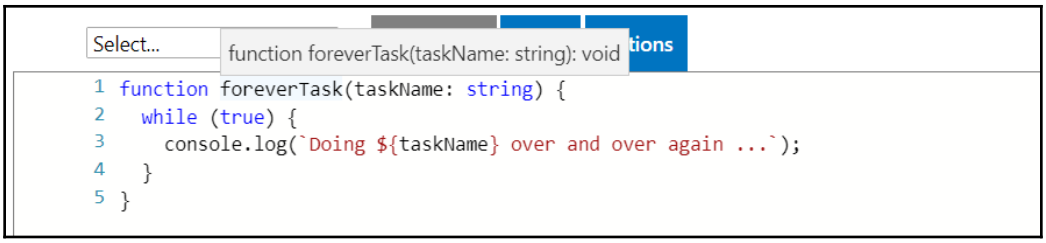

Select...
let flag: boolean
TypeScript
Share
Options
let flag = false;

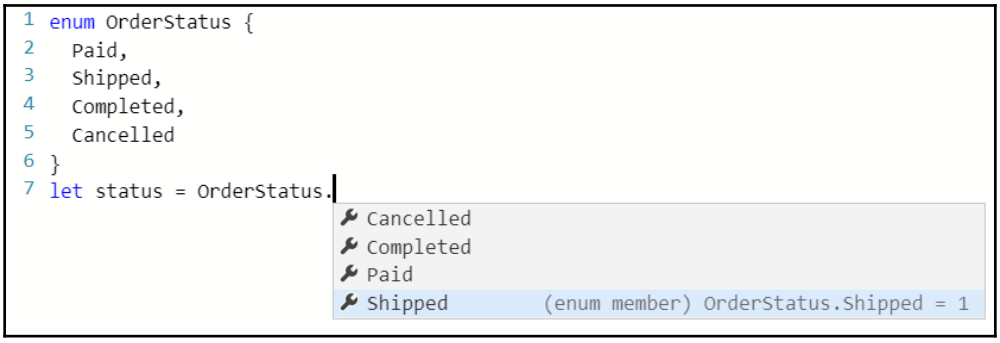

Type '"Table"' is not assignable to type 'boolean'.
let flag: boolean
flag = "Table";

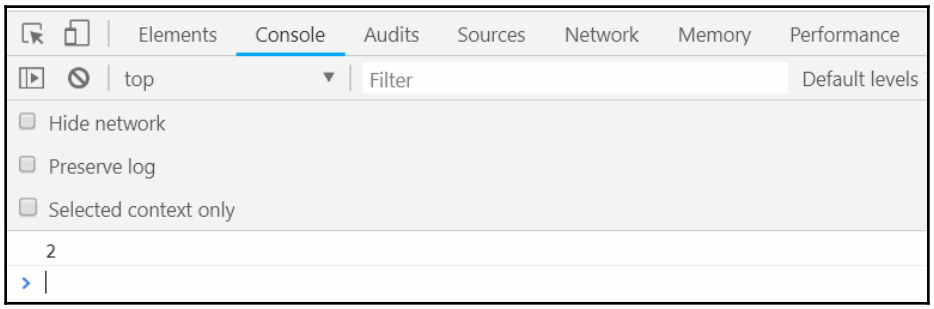

Select...
let flag: any
TypeScript
Share
Options
let flag;

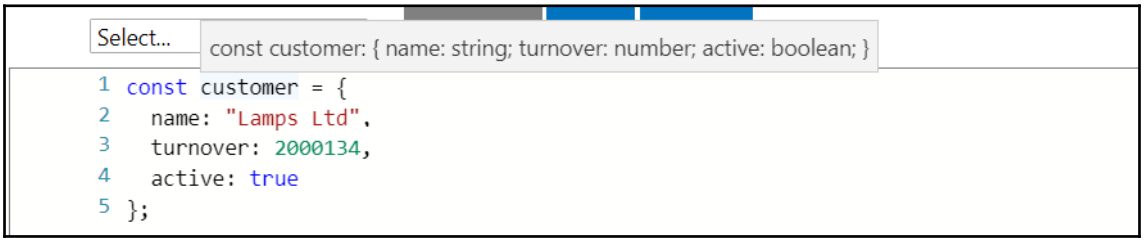

function logText(text: string): void
Share
Options
function logText(text: string) {
console.log(text);
}

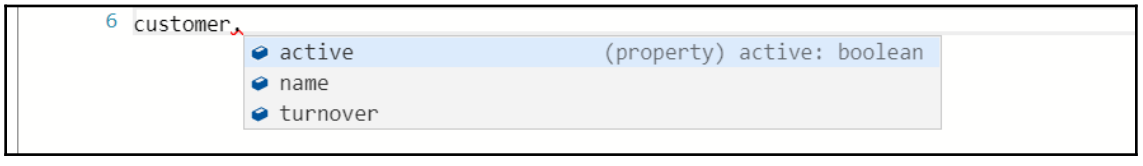

Select...
TypeScript
Share
A function returning 'never' cannot have a reachable end point.
function foreverTask(taskName: string): never {
while (true) {
console.log(`Doing ${taskName} over and over again ...`);
break;
}
}

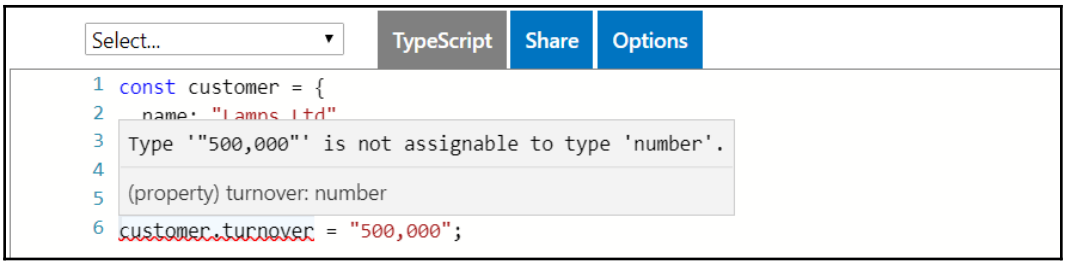

Select...
function foreverTask(taskName: string): void
tions
1 function foreverTask(taskName: string) {
2 while (true) {
3 console.log(`Doing ${taskName} over and over again ...`);
4 }
5 }

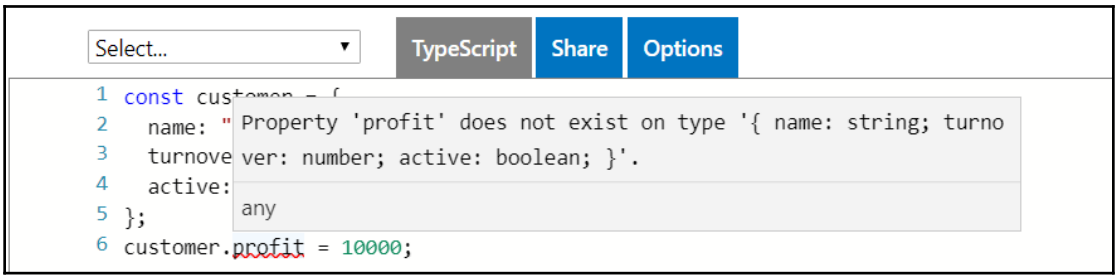

1 enum OrderStatus {
2 Paid,
3 Shipped,
4 Completed,
5 Cancelled
6 }
7 let status = OrderStatus.
Cancelled
Completed
Paid
Shipped (enum member) OrderStatus.Shipped = 1

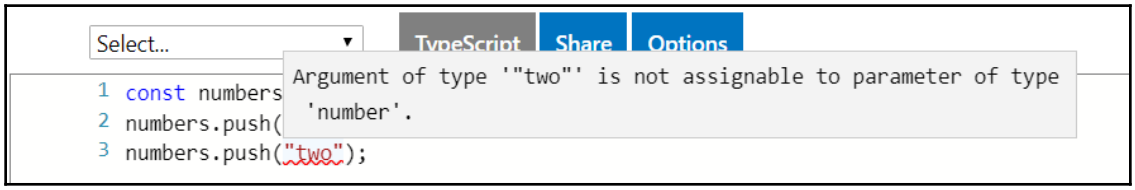

Elements
Console
Audits
Sources
Network
Memory
Performance
top
Filter
Default levels
Hide network
Preserve log
Selected context only
2
>

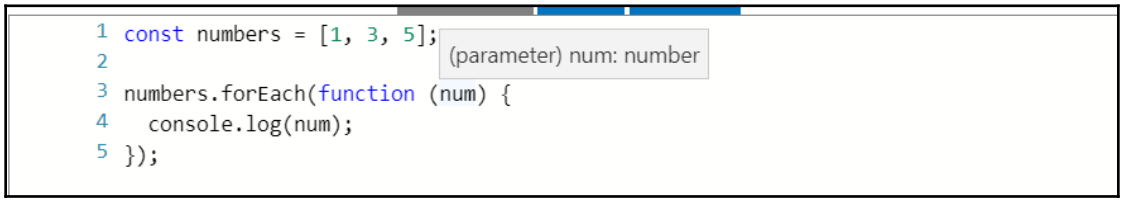

Select...
const customer: { name: string; turnover: number; active: boolean; }
1 const customer = {
2 name: "Lamps Ltd",
3 turnover: 2000134,
4 active: true
5 };

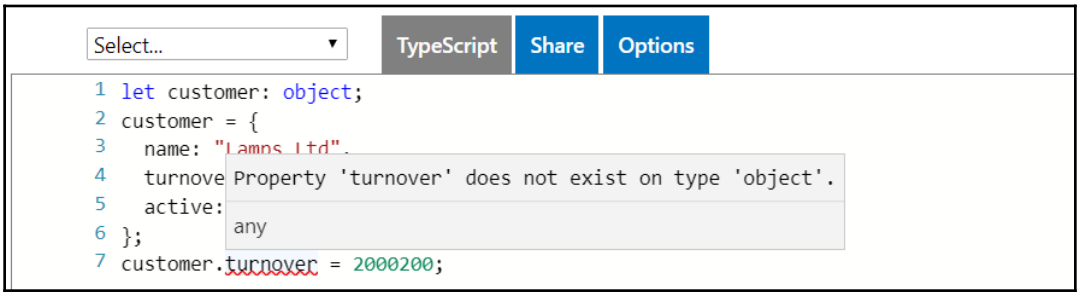

6 customer.
active (property) active: boolean
name
turnover

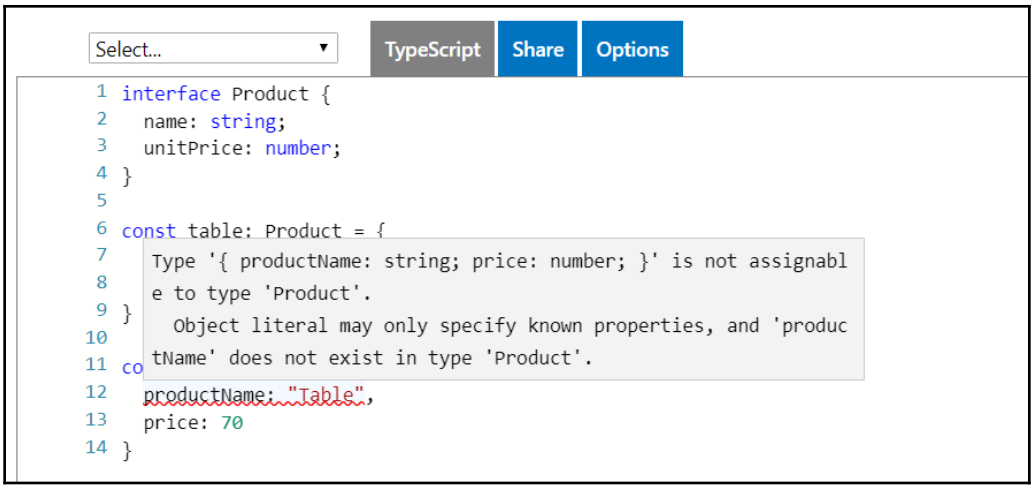

Select...
TypeScript
Share
Options
const customer = {
name: "Lamps Ltd"
Type '"500,000"' is not assignable to type 'number'.
(property) turnover: number
customer.turnover = "500,000";

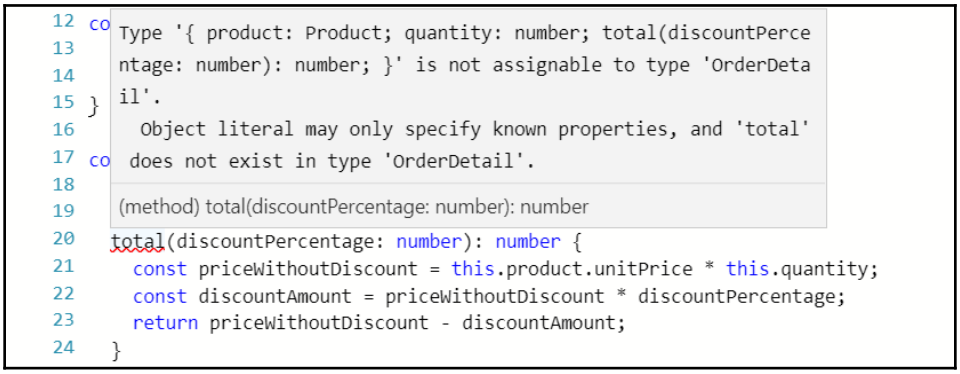

Select...
TypeScript
Share
Options
const cust
name: "
turnove
active:
};
customer.profit = 10000;
Property 'profit' does not exist on type '{ name: string; turno
ver: number; active: boolean; }'.
any


Select...
TypeScript
Share
Options
const numbers
numbers.push(
numbers.push("two");
Argument of type '"two"' is not assignable to parameter of type
'number'.


const numbers = [1, 3, 5];
numbers.forEach(function (num) {
console.log(num);
});
(parameter) num: number


Select...
TypeScript
Share
Options
let customer: object;
customer = {
name: "Lamps Ltd",
turnove
active:
};
customer.turnover = 2000200;
Property 'turnover' does not exist on type 'object'.
any


Select...
TypeScript
Share
Options
interface Product {
name: string;
unitPrice: number;
}
const table: Product = {
Type '{ productName: string; price: number; }' is not assignable to type 'Product'.
Object literal may only specify known properties, and 'productName' does not exist in type 'Product'.
productName: "Table",
price: 70
}


Type '{ product: Product; quantity: number; total(discountPercentage: number): number; }' is not assignable to type 'OrderDetail'.
Object literal may only specify known properties, and 'total' does not exist in type 'OrderDetail'.
(method) total(discountPercentage: number): number
total(discountPercentage: number): number {
const priceWithoutDiscount = this.product.unitPrice * this.quantity;
const discountAmount = priceWithoutDiscount * discountPercentage;
return priceWithoutDiscount - discountAmount;
}

```ts
interface Product {
  name: string;
  unitPrice: number;
}

interface OrderDetail {
  product: Product;
  quantity: number;
  getTotal(discount: number): number;
}

co
  [ts]
  Type '(discountPercentage: number) => string' is not assignable
   to type '(discount: number) => number'.
    Type 'string' is not assignable to type 'number'.

  • interfaces.ts(9, 3): The expected type comes from property 'getTotal' which
    is declared here on type 'OrderDetail'

  (method) getTotal(discountPercentage: number): string
};

co
  getTotal(discountPercentage: number): string {
    const priceWithoutDiscount = this.product.unitPrice * this.quantity;
    const discountAmount = priceWithoutDiscount * discountPercentage;
    return (priceWithoutDiscount - discountAmount).toString();
  }
};
```

```ts
  [ts]
  Type '(discountPercentage: string) => number' is not assignable
co  to type '(discount: number) => number'.
    Types of parameters 'discountPercentage' and 'discount' are i
  ncompatible.
};      Type 'number' is not assignable to type 'string'.

  • interfaces.ts(9, 3): The expected type comes from property 'getTotal' which
co  is declared here on type 'OrderDetail'

  (method) getTotal(discountPercentage: string): number
  getTotal(discountPercentage: string): number {
    const priceWithoutDiscount = this.product.unitPrice * this.quantity;
    const discountAmount = priceWithoutDiscount * parseInt(discountPercentage);
    return priceWithoutDiscount - discountAmount;
  }
};
```

```
interface Product {
  readonly name: string;
  unitPrice: number;
}

const table: Product = {
  name: "Table",
  unitPrice: 100
};

table.name = "Better Table";
```

Cannot assign to 'name' because it is a constant or a read-only property.

(property) Product.name: string

```
abstract class Product {
  name: string;
  unitPrice: number;
}

const bread = new Product();
```

Cannot create an instance of an abstract class.

```
abstract class Product {
  name: string;
  unitPrice: number;
  abstract delete(): void;
}

class Food extends Product {
  constructor(public bestBefore: Date) {
    super();
  }
}
```

Non-abstract class 'Food' does not implement inherited abstract member 'delete' from class 'Product'.

class Food

```
const orderDetail = new OrderDetail();
orderDetail.deleted = true;
```

Property 'deleted' is private and only accessible within class 'OrderDetail'.

(property) OrderDetail.deleted: boolean

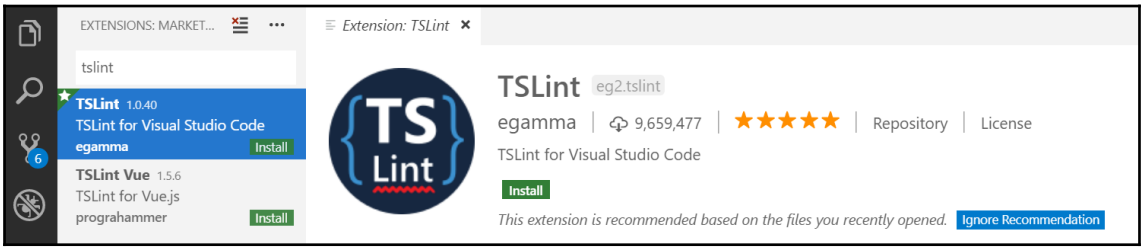
```
class OrderDetail {
  product: Product;
  quantity: number;

  static getTotal(discount: number): n
    const priceWithoutDiscount = this.product.unitPrice * this.quantity;
    const discountAmount = priceWithoutDiscount * discount;
    return priceWithoutDiscount - discountAmount;
  }
}
```

Property 'product' does not exist on type 'ty
peof OrderDetail'.

any

product.ts orderDetail.ts

```
1  class Order [ts] Cannot find name 'Product'.
2    product: Product;
3    quantity: number;
4    getTotal(discount: number): number {
5      const priceWithoutDiscount = this.product.unitPrice * this.quantity;
6      const discountAmount = priceWithoutDiscount * discount;
7      return priceWithoutDiscount - discountAmount;
8    }
9  }
```

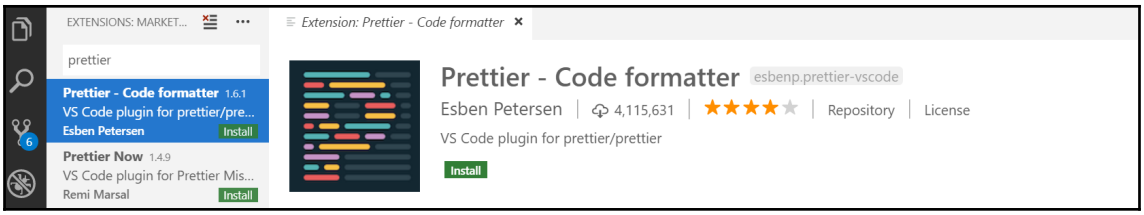


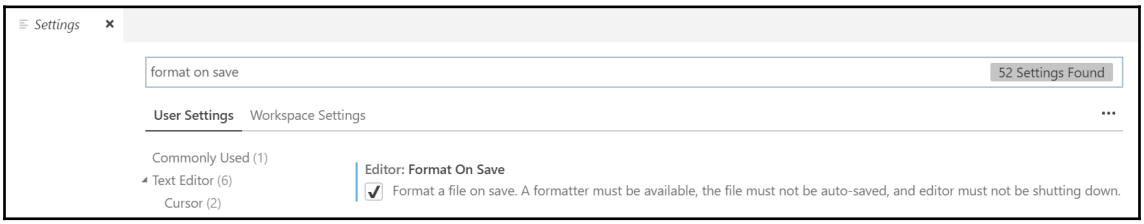
```
6   export class OrderDetail {
7     product: Product;
8     quantity: number;
9     getTotal(discount: number): number {
10      const priceWithoutDiscount = this.product.unitPrice * this.quantity;
11      const discountAmount = priceWithoutDiscount * discount;
12      return priceWithoutDiscount - discountAmount;
13    }
14  }
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

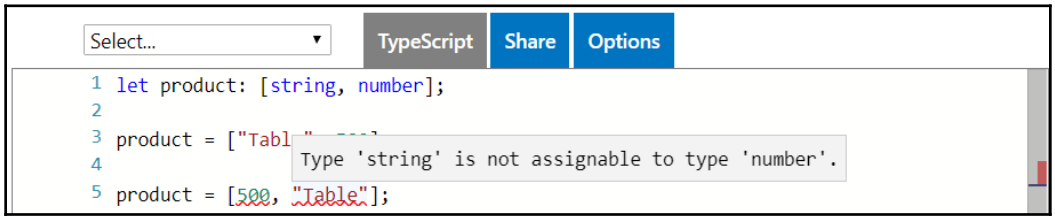
orderDetail.ts 3

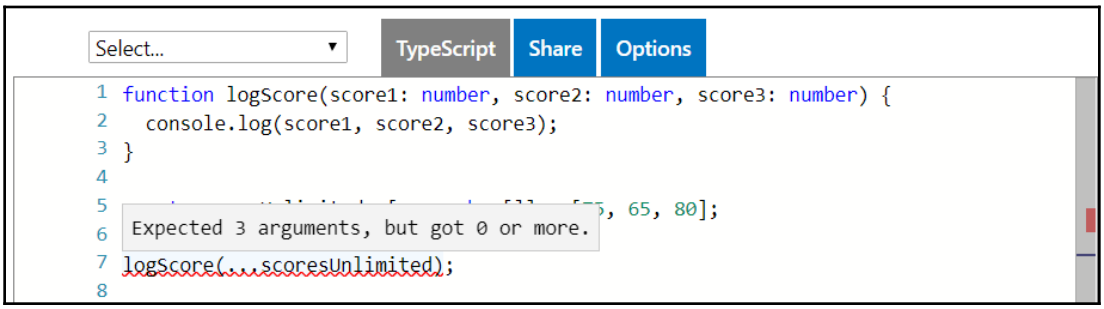
- [tslint] The class property 'product' must be marked either 'private', 'public', or 'protected' (member-access) (7, 3)
- [tslint] The class property 'quantity' must be marked either 'private', 'public', or 'protected' (member-access) (8, 3)
- [tslint] The class method 'getTotal' must be marked either 'private', 'public', or 'protected' (member-access) (9, 3)

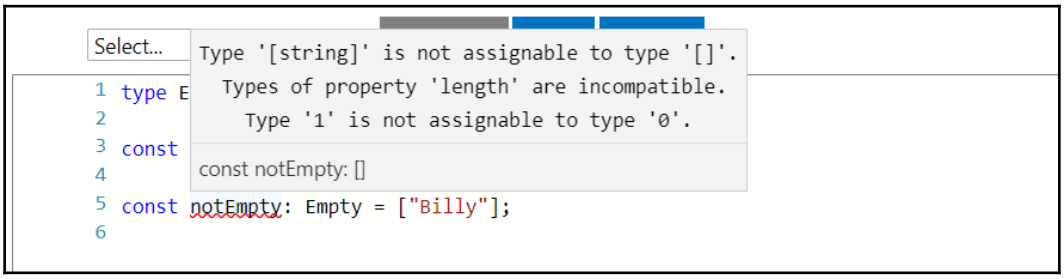


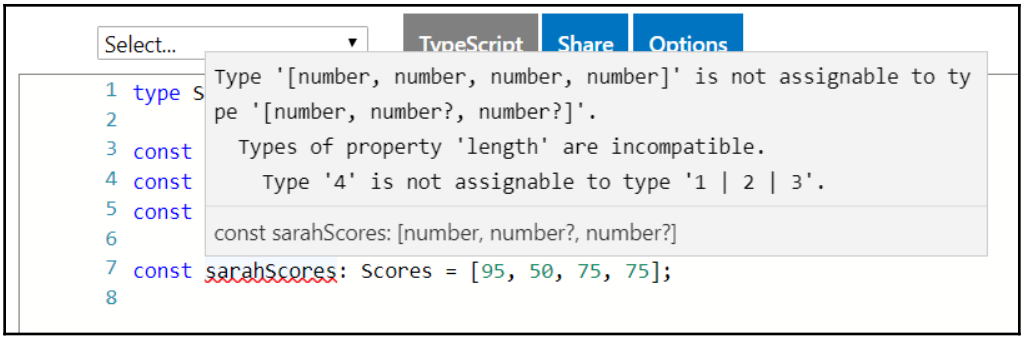


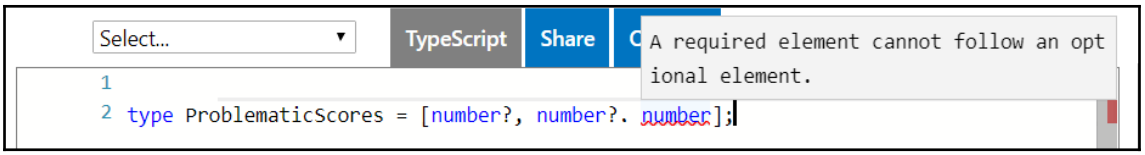
Chapter 2: What is New in TypeScript 3

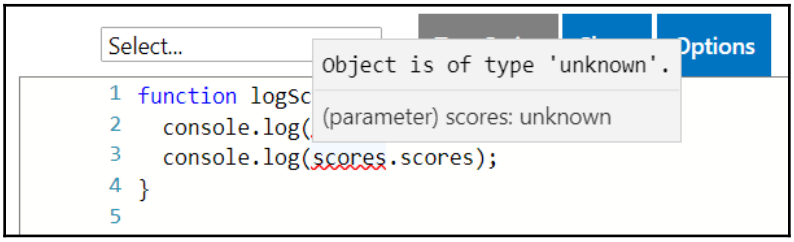


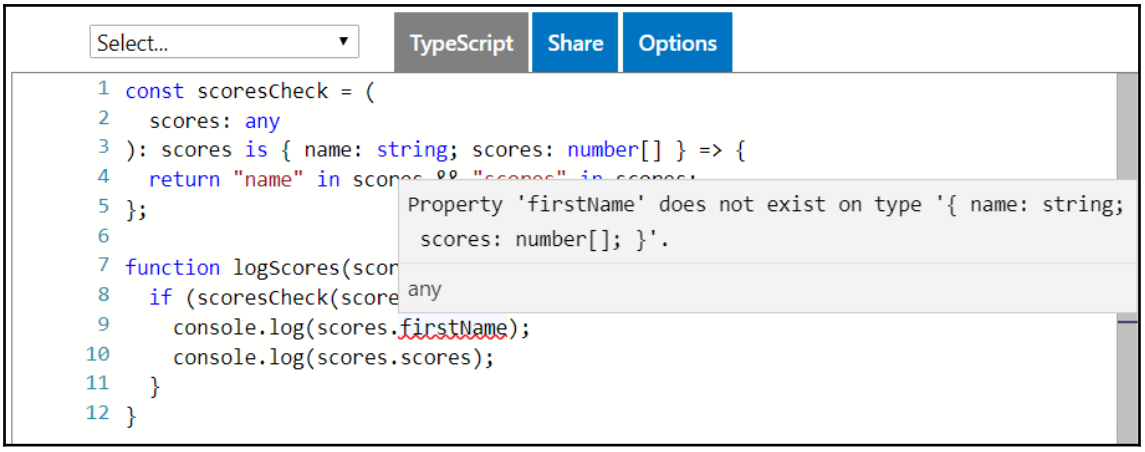


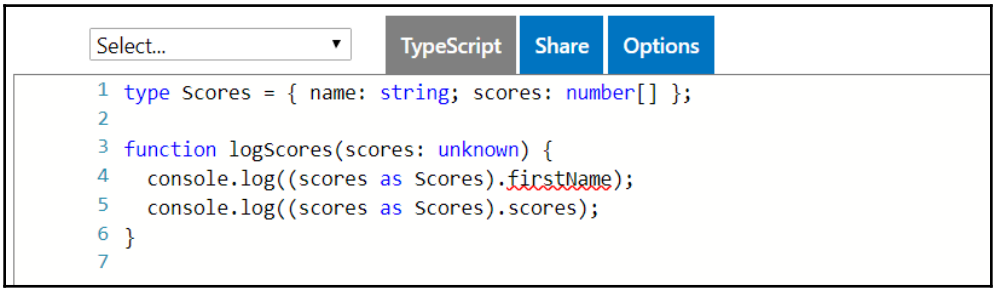





Select...
TypeScript
Share
A required element cannot follow an optional element.
1
2 type ProblematicScores = [number?, number?. number];


Select...
Options
Object is of type 'unknown'.
(parameter) scores: unknown
1 function logSc
2 console.log(
3 console.log(scores.scores);
4 }
5


Select...
TypeScript
Share
Options
1 const scoresCheck = (
2 scores: any
3): scores is { name: string; scores: number[] } => {
4 return "name" in scor
5 };
6
7 function logScores(scor
8 if (scoresCheck(score
9 console.log(scores.firstName);
10 console.log(scores.scores);
11 }
12 }
Property 'firstName' does not exist on type '{ name: string; scores: number[]; }'.
any


Select...
TypeScript
Share
Options
1 type Scores = { name: string; scores: number[] };
2
3 function logScores(scores: unknown) {
4 console.log((scores as Scores).firstName);
5 console.log((scores as Scores).scores);
6 }
7

person.ts

```
1  import { randomString } from "../../Shared/dist/utils";
2
3  class Person {
4    id: string;
5    name: string;
6    constructor() {
7      this.id = randomStrin();
8    }
9  }
10
```

Navigate to test/snapshot Alt+Ctrl+F12
Go to Definition F12
Peek Definition Alt+F12
Go to Type Definition
Find All References Shift+F12

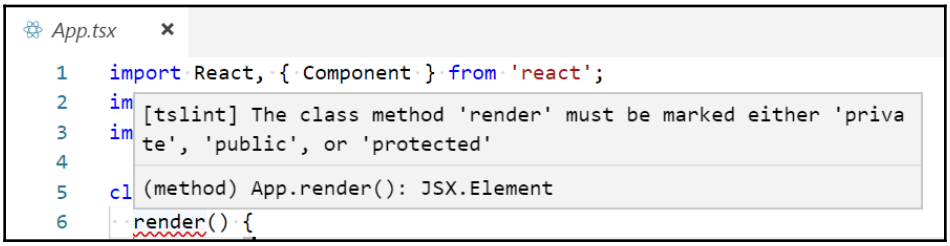
utils.d.ts

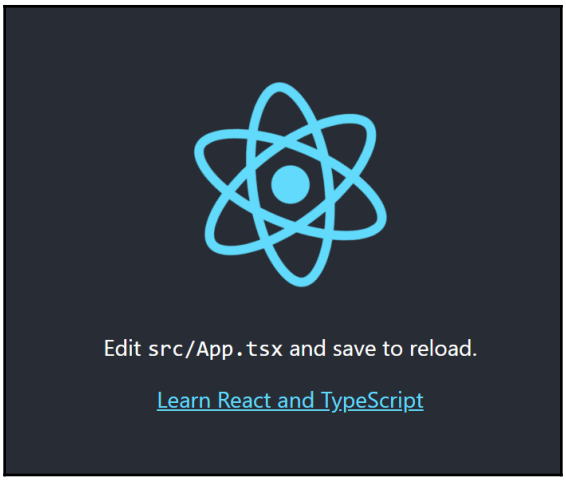
```
1  export declare function randomString(): string;
2
```

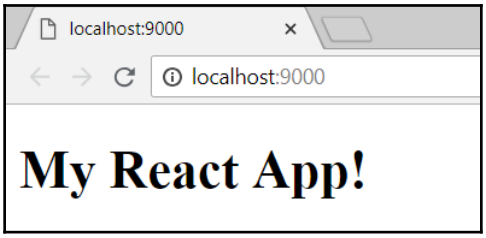
localhost:3000

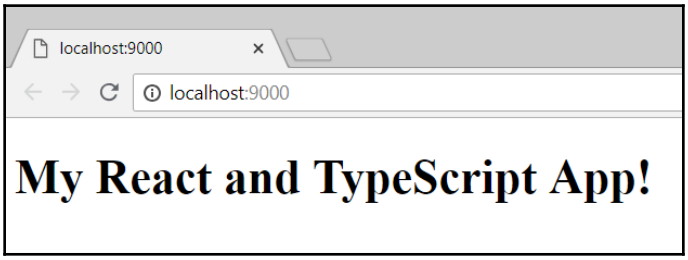
- Fred
- Jane
- Bob

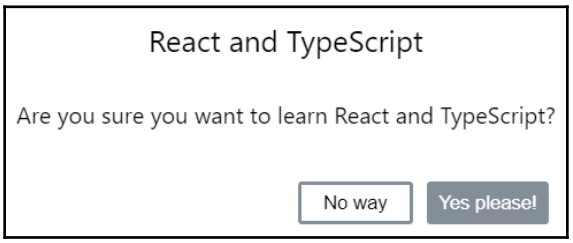
Chapter 3: Getting Started with React and TypeScript

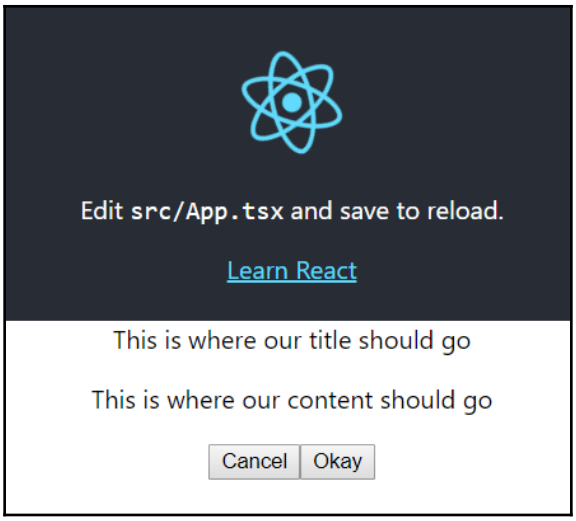


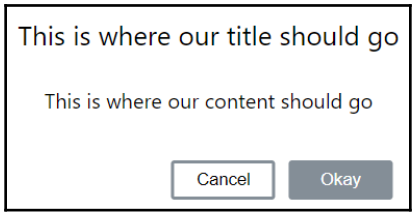

Edit src/App.tsx and save to reload.
Learn React and TypeScript


localhost:9000
localhost:9000
My React App!


localhost:9000
localhost:9000
My React and TypeScript App!


React and TypeScript
Are you sure you want to learn React and TypeScript?
No way
Yes please!





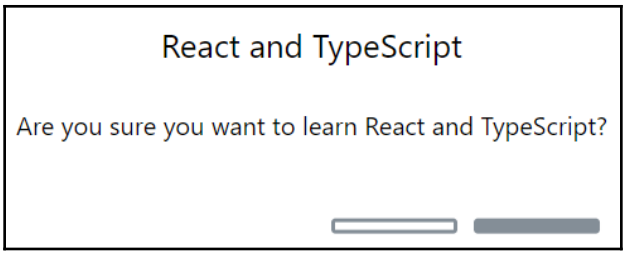

```
21          <img src={logo} className="App-logo" alt="logo" />
22          <h1 className="App-title">Welcome to React</h1>
23        </header>
24        <p className="App-intro">
25          To get started, edit <code>src/App.tsx</code> and save to reload.
26        </p>
27        <Confirm />
28      </div>
29    );
30  }
31  }
32
33  export default App;
```

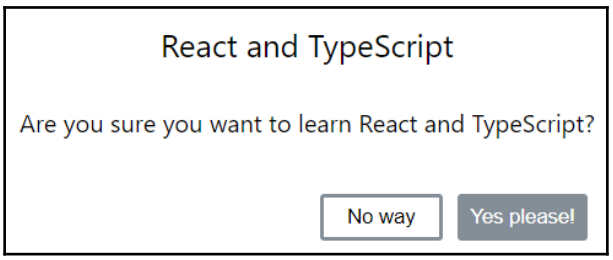
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Filter. Eg: text, **/*.ts, !**/node_modules/**

App.tsx confirm\src 1

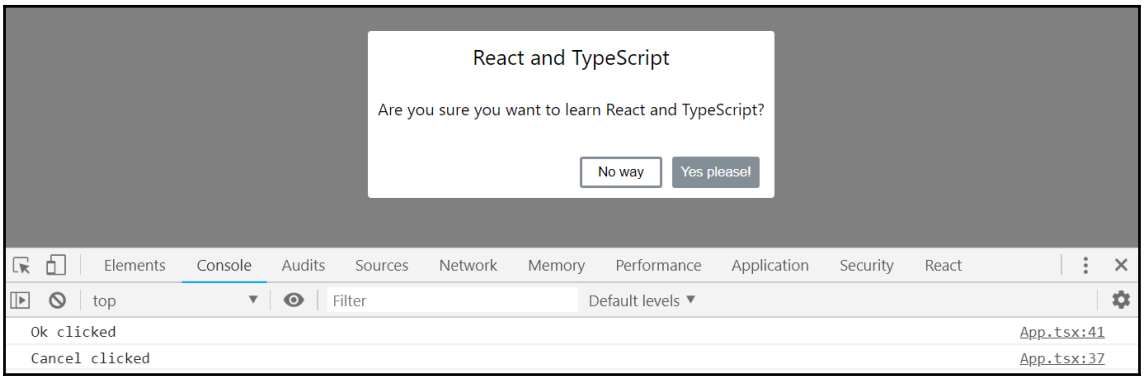
[ts] Type '{}' is not assignable to type 'Readonly<IProps>'. Property 'title' is missing in type '{}'. (27, 10)


React and TypeScript
Are you sure you want to learn React and TypeScript?


React and TypeScript
Are you sure you want to learn React and TypeScript?
No way
Yes please!

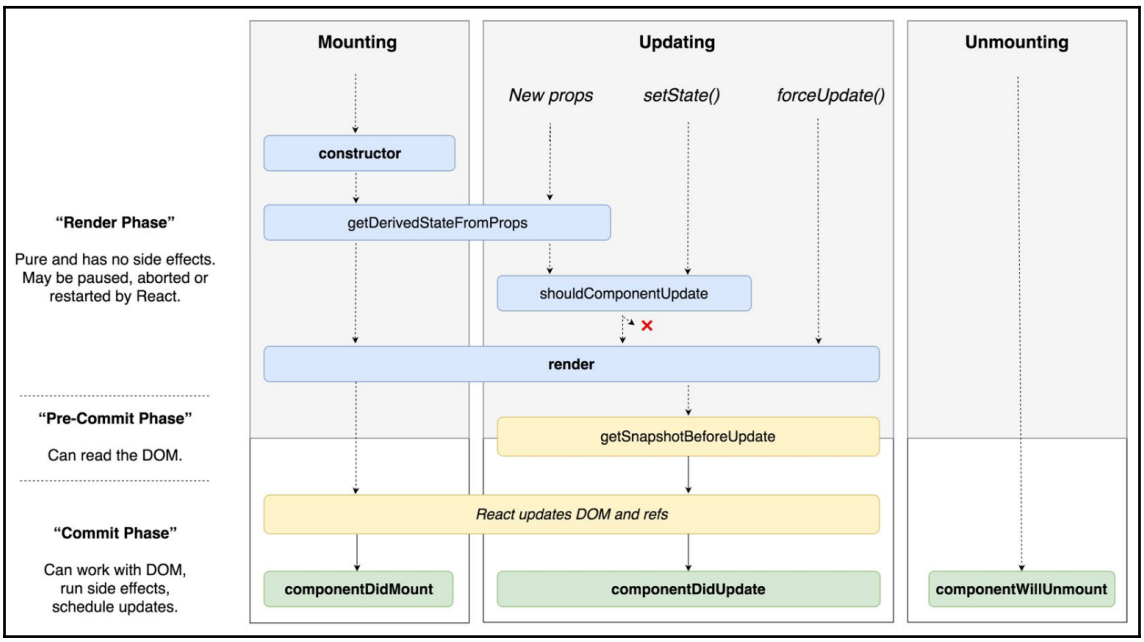

Elements
Console
Audits
Sources
Network
Memory
Performance
Application
Security
React
top
Filter
Default levels
Ok clicked
Confirm.tsx:42

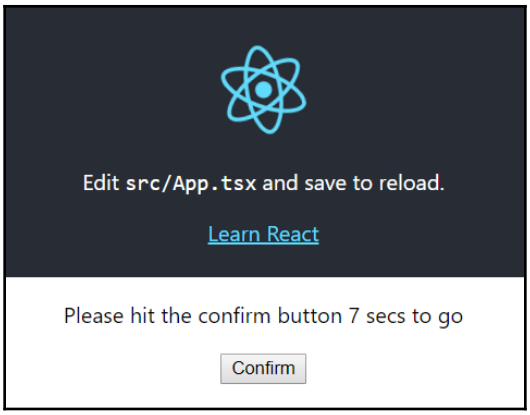

TypeError: Cannot read property 'props' of undefined
./src/Confirm.4.tsx.Confirm3.handleOkClick
C:/code/_temp/ts-react-confirm/confirm/src/Confirm.4.tsx:39
36 | }
37 |
38 | private handleOkClick() {
> 39 | console.log("Ok clicked", this.props);
40 | }
41 |
42 | private handleCancelClick() {
View compiled

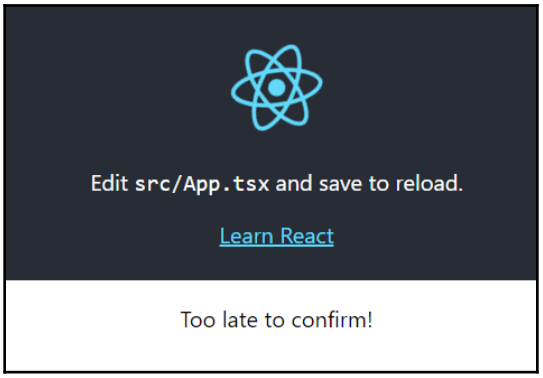


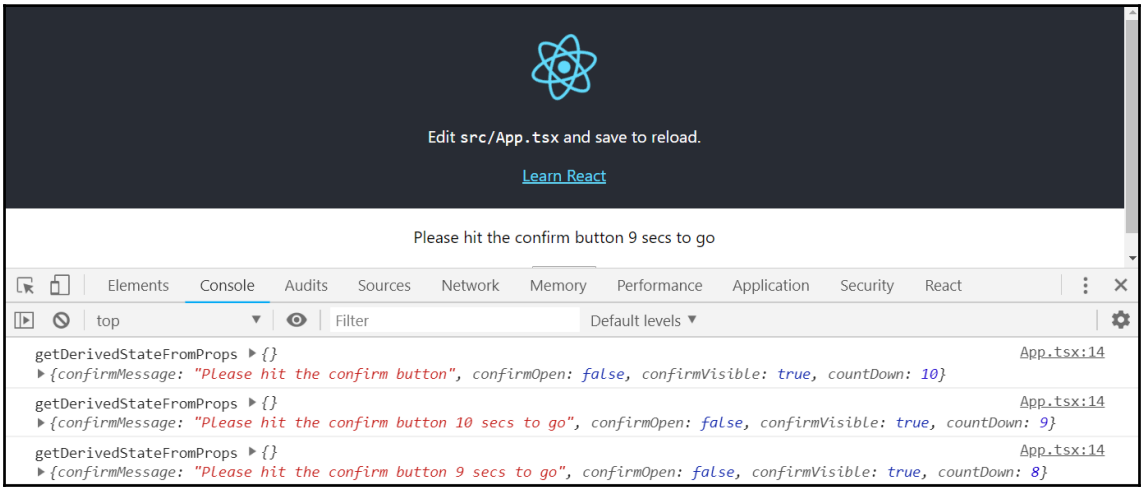

```
private handleCancelConfirmClick = () => {
  console.log
};

private handl
  this.state.confirmOpen = false;
};
```

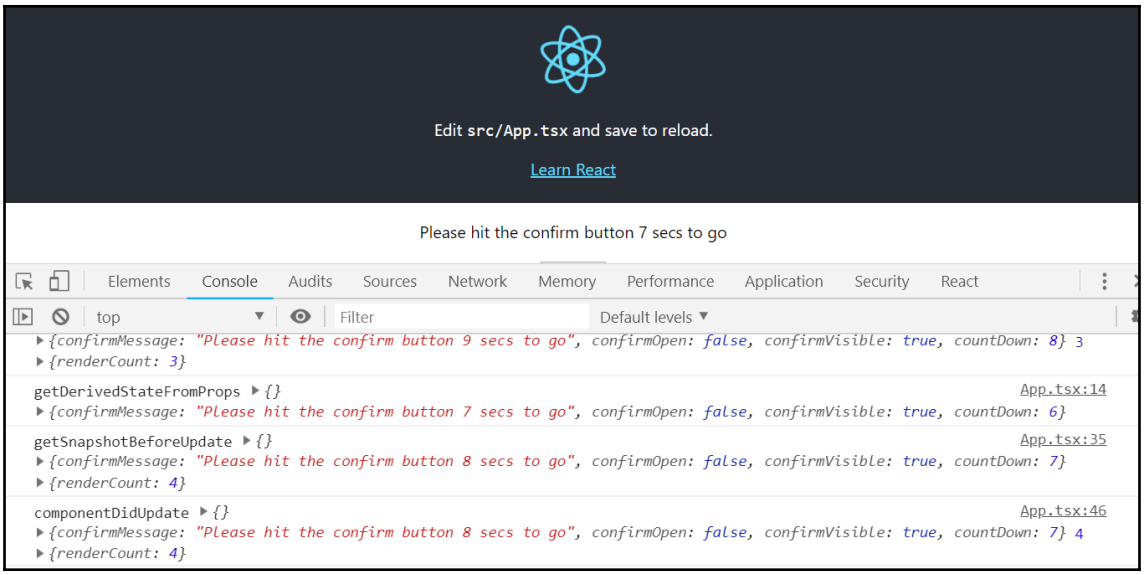
```
[ts] Cannot assign to 'confirmOpen' because it is a constant or
 a read-only property.
(property) confirmOpen: boolean
```

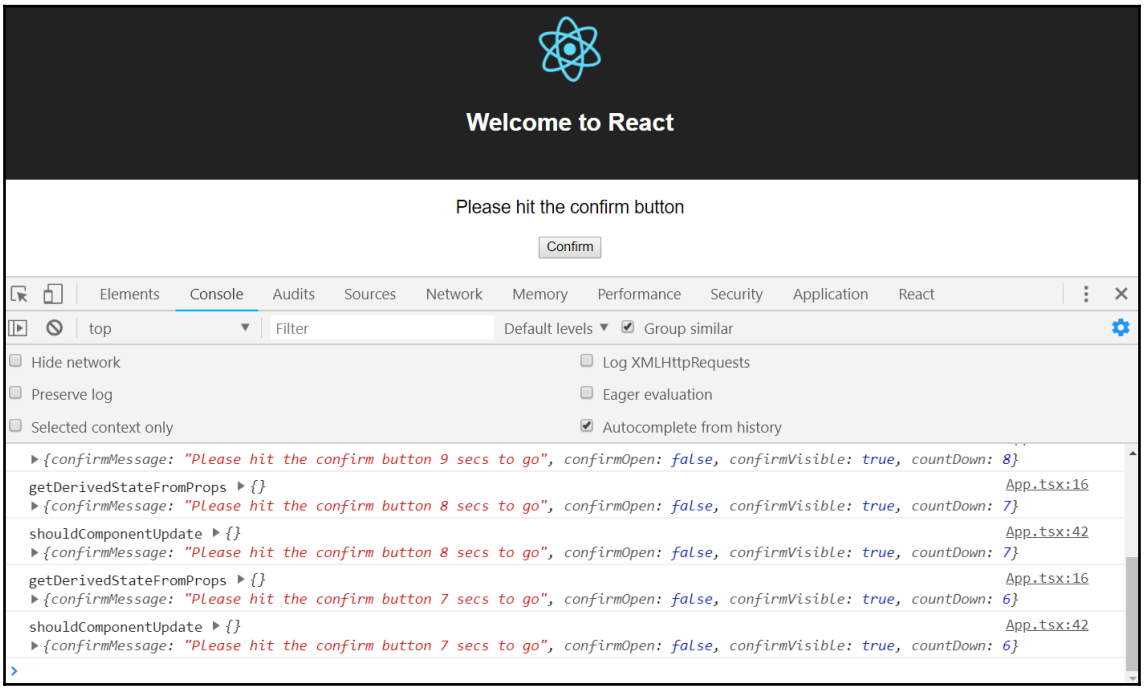



Edit src/App.tsx and save to reload.
Learn React
Please hit the confirm button 7 secs to go
Confirm

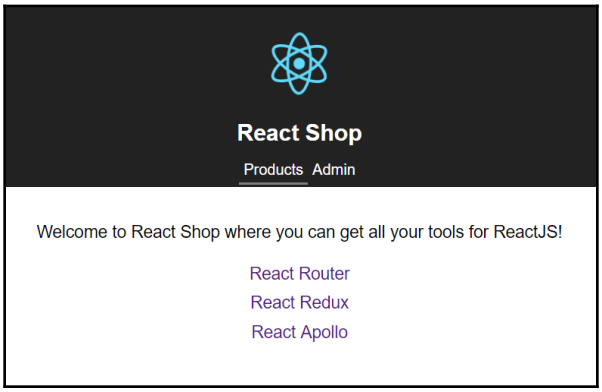

Edit src/App.tsx and save to reload.
Learn React
Too late to confirm!

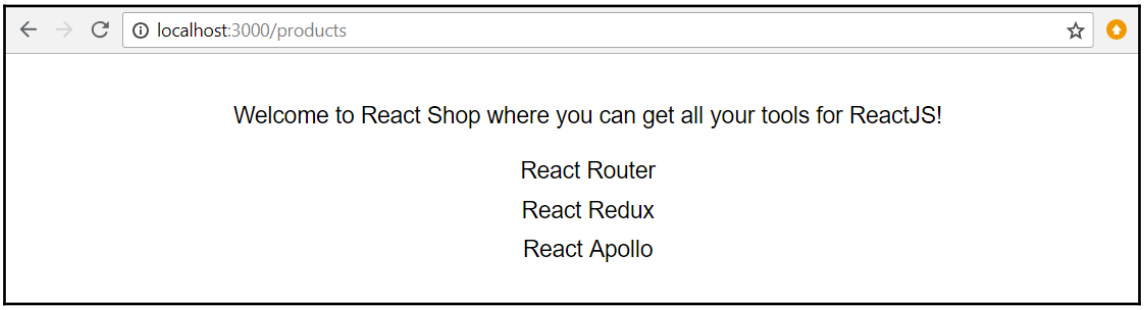

Edit src/App.tsx and save to reload.
Learn React
Please hit the confirm button 9 secs to go
Elements
Console
Audits
Sources
Network
Memory
Performance
Application
Security
React
top
Filter
Default levels
getDerivedStateFromProps {}
{confirmMessage: "Please hit the confirm button", confirmOpen: false, confirmVisible: true, countDown: 10}
App.tsx:14
getDerivedStateFromProps {}
{confirmMessage: "Please hit the confirm button 10 secs to go", confirmOpen: false, confirmVisible: true, countDown: 9}
App.tsx:14
getDerivedStateFromProps {}
{confirmMessage: "Please hit the confirm button 9 secs to go", confirmOpen: false, confirmVisible: true, countDown: 8}
App.tsx:14

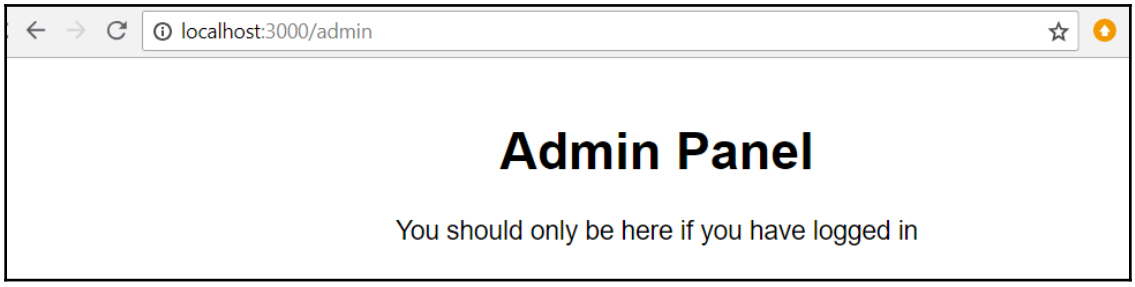

Edit src/App.tsx and save to reload.
Learn React
Please hit the confirm button 7 secs to go
Elements Console Audits Sources Network Memory Performance Application Security React
top Filter Default levels
{confirmMessage: "Please hit the confirm button 9 secs to go", confirmOpen: false, confirmVisible: true, countDown: 8} 3
{renderCount: 3}
getDerivedStateFromProps {} App.tsx:14
{confirmMessage: "Please hit the confirm button 7 secs to go", confirmOpen: false, confirmVisible: true, countDown: 6}
getSnapshotBeforeUpdate {} App.tsx:35
{confirmMessage: "Please hit the confirm button 8 secs to go", confirmOpen: false, confirmVisible: true, countDown: 7}
{renderCount: 4}
componentDidUpdate {} App.tsx:46
{confirmMessage: "Please hit the confirm button 8 secs to go", confirmOpen: false, confirmVisible: true, countDown: 7} 4
{renderCount: 4}

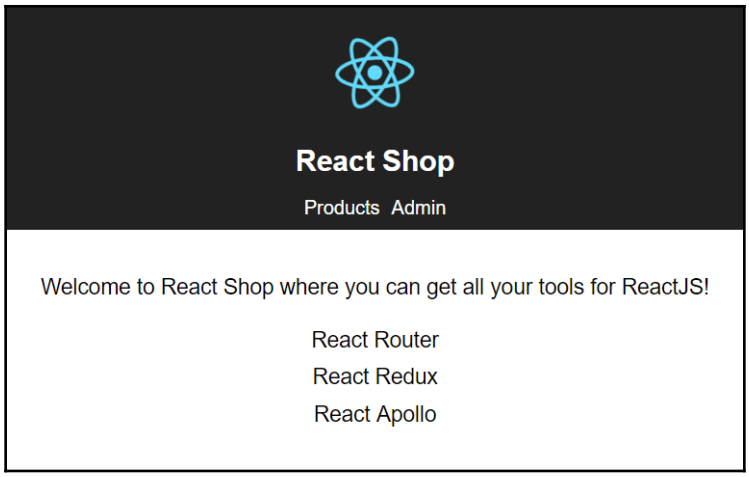

Welcome to React
Please hit the confirm button
Confirm
Elements Console Audits Sources Network Memory Performance Security Application React
top Filter Default levels Group similar
Hide network
Preserve log
Selected context only
Log XMLHttpRequests
Eager evaluation
Autocomplete from history
{confirmMessage: "Please hit the confirm button 9 secs to go", confirmOpen: false, confirmVisible: true, countDown: 8}
getDerivedStateFromProps {} App.tsx:16
{confirmMessage: "Please hit the confirm button 8 secs to go", confirmOpen: false, confirmVisible: true, countDown: 7}
shouldComponentUpdate {} App.tsx:42
{confirmMessage: "Please hit the confirm button 8 secs to go", confirmOpen: false, confirmVisible: true, countDown: 7}
getDerivedStateFromProps {} App.tsx:16
{confirmMessage: "Please hit the confirm button 7 secs to go", confirmOpen: false, confirmVisible: true, countDown: 6}
shouldComponentUpdate {} App.tsx:42
{confirmMessage: "Please hit the confirm button 7 secs to go", confirmOpen: false, confirmVisible: true, countDown: 6}

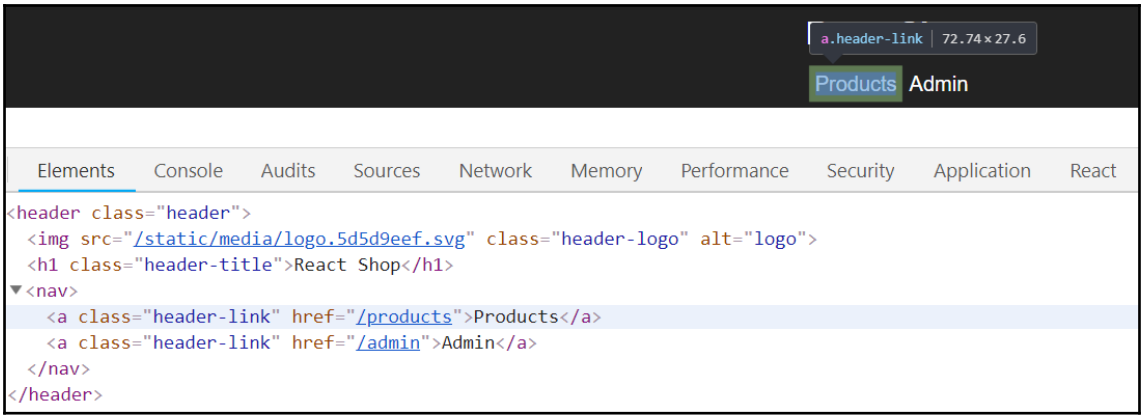
Chapter 4: Routing with React Router

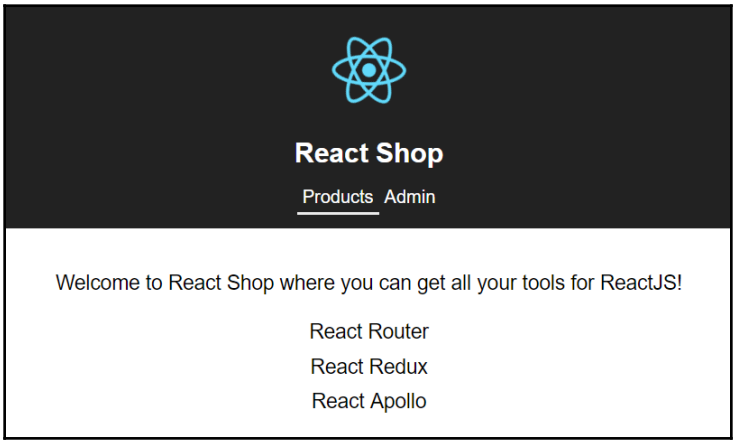


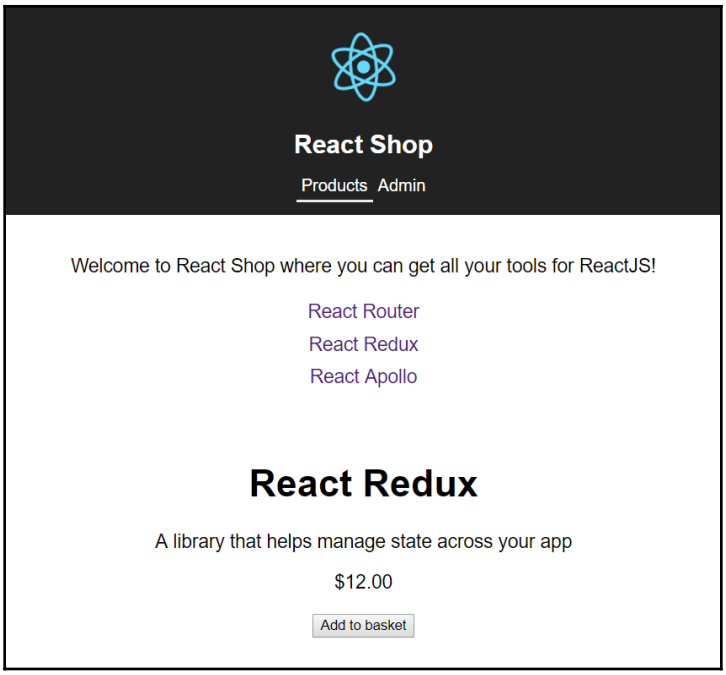


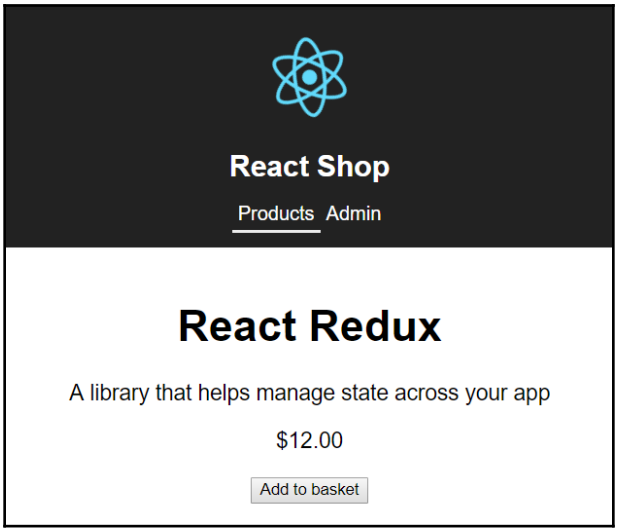


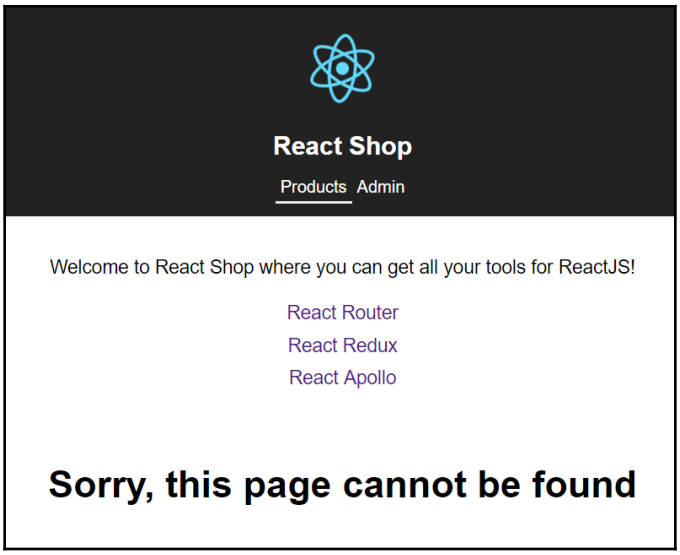

React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Router
React Redux
React Apollo

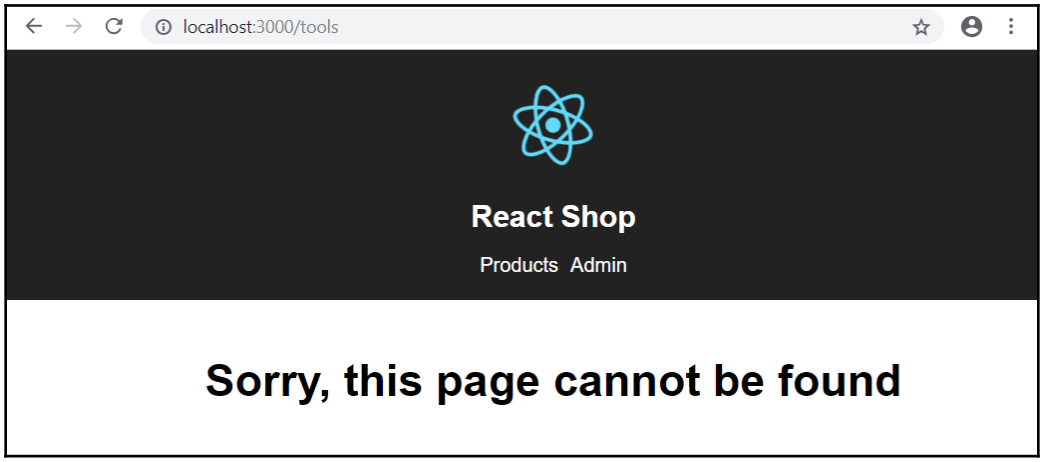

a.header-link | 72.74 × 27.6
Products Admin
Elements Console Audits Sources Network Memory Performance Security Application React
<header class="header">
<img src="/static/media/logo.5d5d9eef.svg" class="header-logo" alt="logo">
<h1 class="header-title">React Shop</h1>
<nav>
Products
Admin
</nav>
</header>

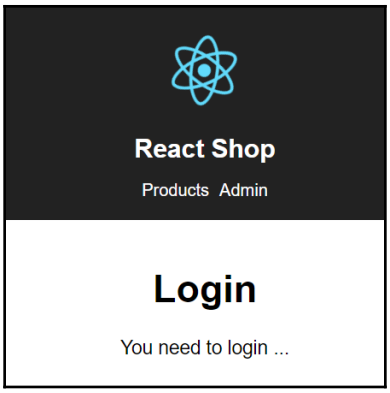

React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Router
React Redux
React Apollo

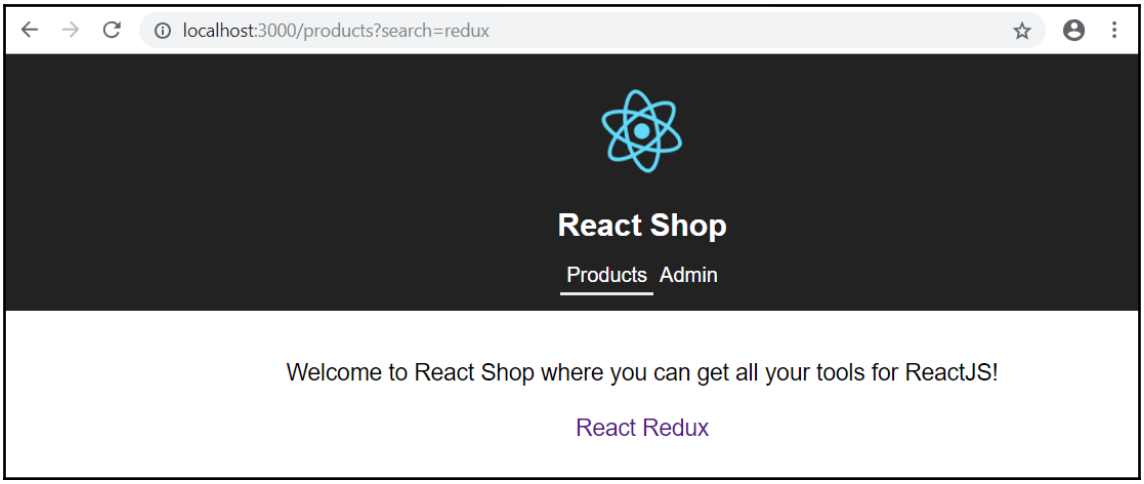

React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Router
React Redux
React Apollo
React Redux
A library that helps manage state across your app
$12.00
Add to basket

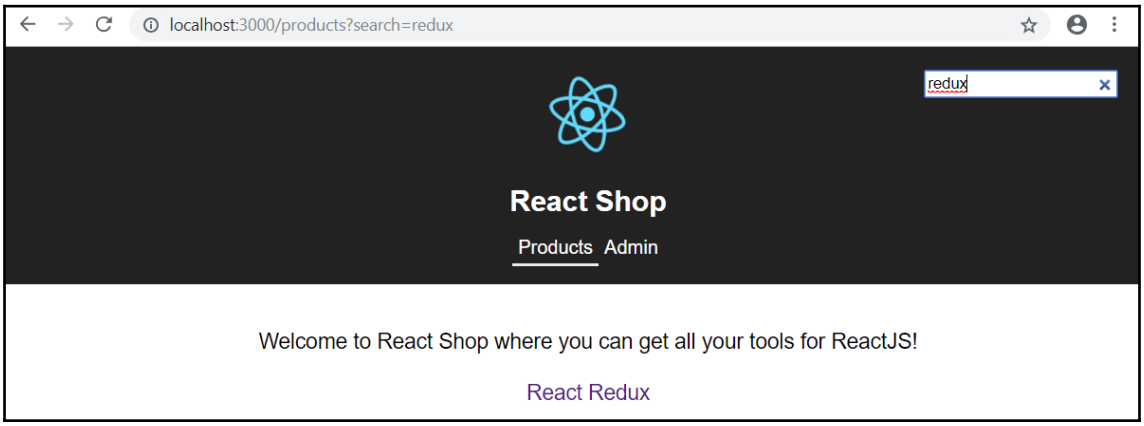

React Shop
Products Admin
React Redux
A library that helps manage state across your app
$12.00
Add to basket

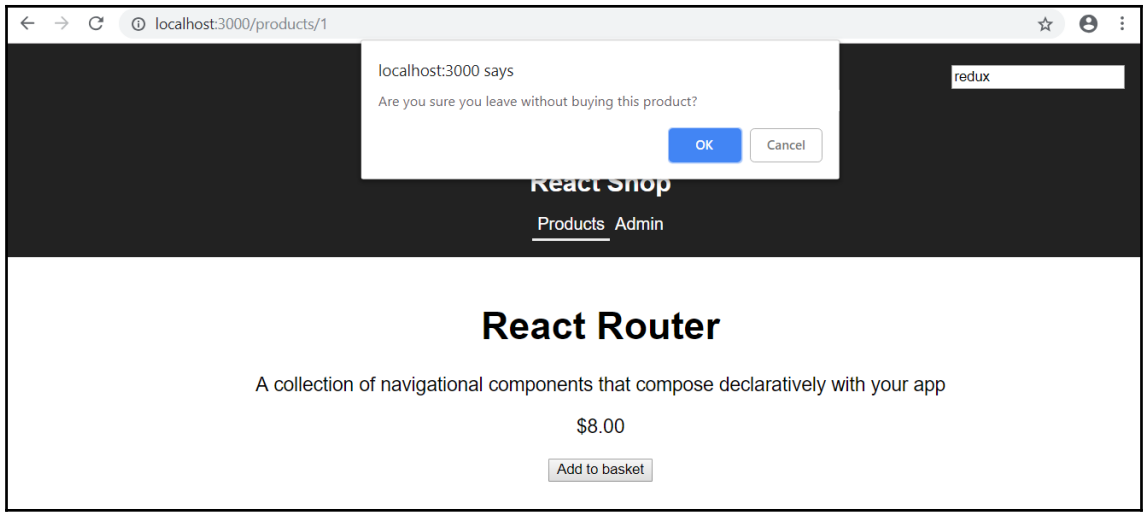

React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Router
React Redux
React Apollo
Sorry, this page cannot be found

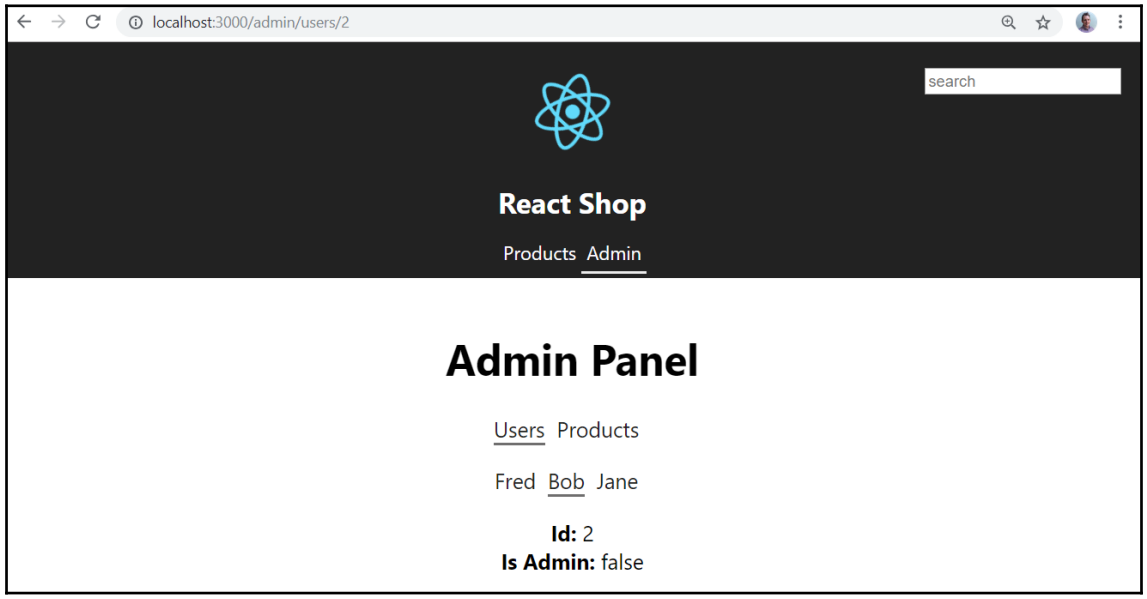

localhost:3000/tools
React Shop
Products Admin
Sorry, this page cannot be found


React Shop
Products Admin
Login
You need to login ...


localhost:3000/products?search=redux
React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Redux


localhost:3000/products?search=redux
redux
React Shop
Products Admin
Welcome to React Shop where you can get all your tools for ReactJS!
React Redux


localhost:3000/products/1
localhost:3000 says
Are you sure you leave without buying this product?
OK
Cancel
redux
Products Admin
React Router
A collection of navigational components that compose declaratively with your app
$8.00
Add to basket


localhost:3000/admin/users/2
search
React Shop
Products Admin
Admin Panel
Users Products
Fred Bob Jane
Id: 2
Is Admin: false

Admin Panel

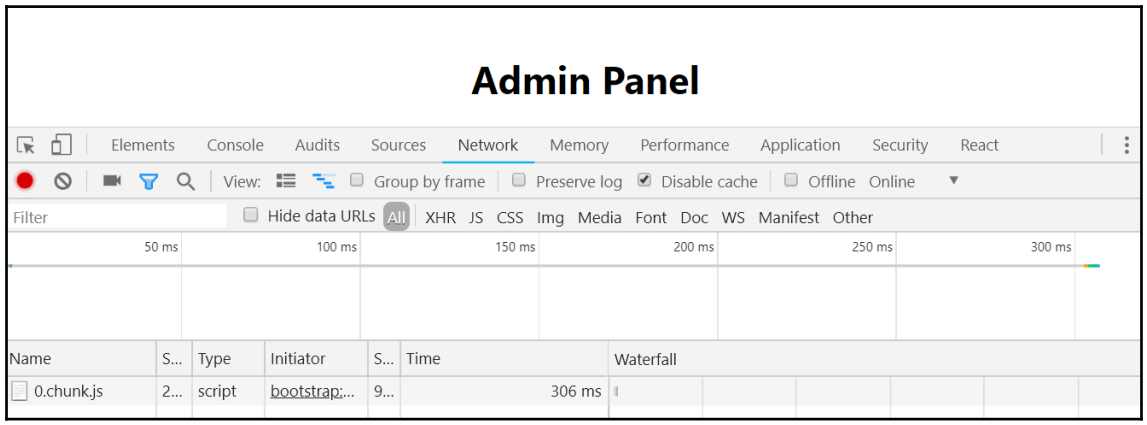
Users Products

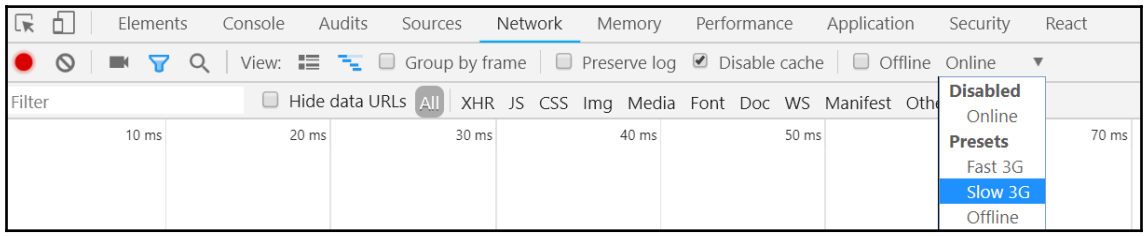
Some options to administer products

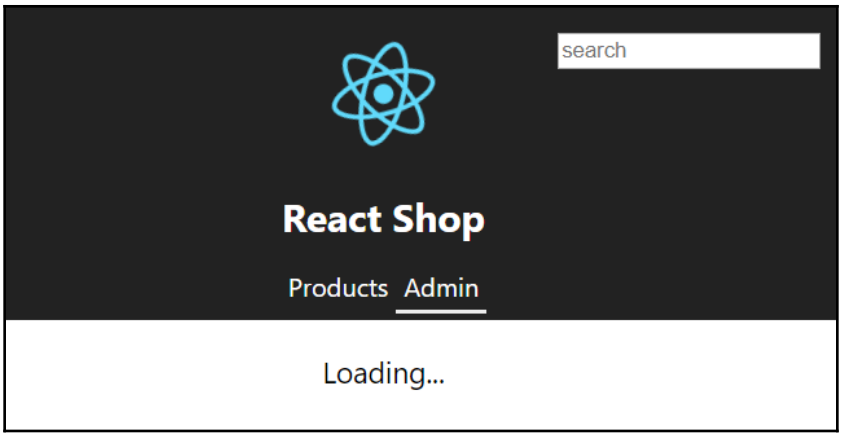
```
ReactDOM.render(<Routes />, document.getElementById("root") as HTMLElement);
```

```
[ts]
Type '{}' is not assignable to type 'Readonly<RouteComponentPro
ps<{}, StaticContext, any>>'.
  Property 'history' is missing in type '{}'.
```

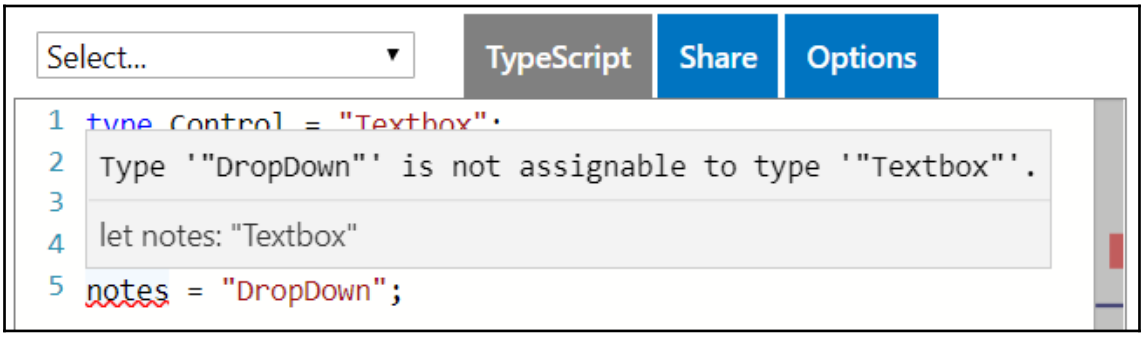
```
(alias) class Routes
import Routes
```

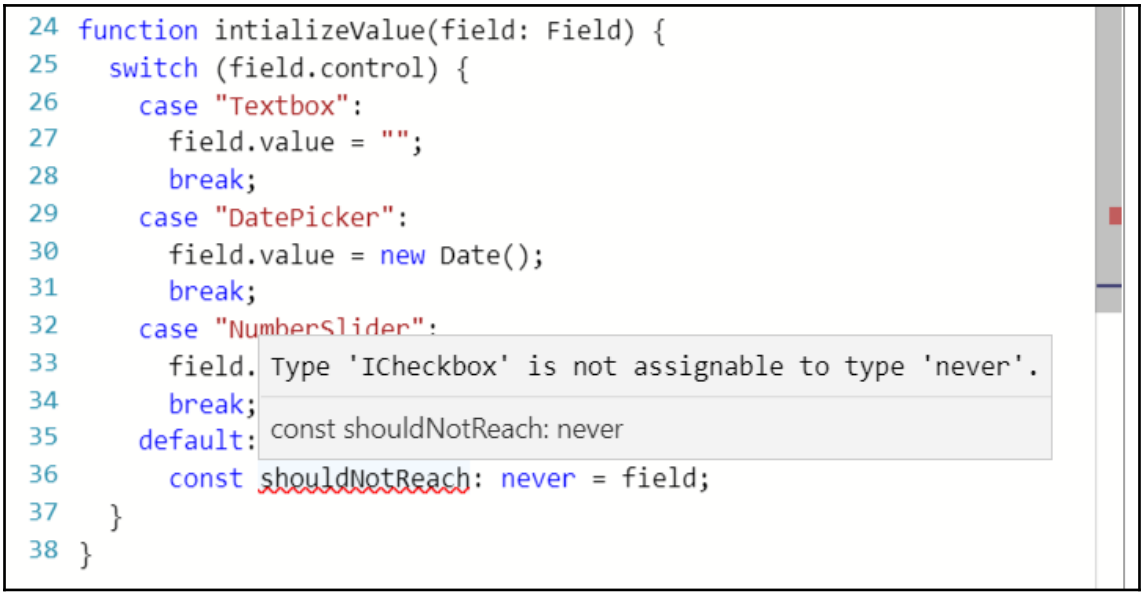





search
React Shop
Products
Admin
Loading...

Chapter 5: Advanced Types





```
type StringOrStringArray = string | string[];
function fi                                    tringArray): string {
  if (typeo                                    ) {
    return stringOrArray.substr(0, 1);
  } else {
    return stringOrArray[0];
  }
}
```

(parameter) stringOrArray: string

```
type StringOrStringArray = string | string[];
function first(stringOrArray: StringOrStringArray): string {
  if (typeof stringOrArray === "string") {
    return
  } else {
    return stringOrArray[0];
  }
}
```

(parameter) stringOrArray: string[]

```
type StringOr
function firs
  if (typeof
    return st
  } else if (typeof stringOrArray === "string[]") {
    return stringOrArray[0];
  } else {
    const shouldNotReach: never = stringOrArray;
  }
}
```

This condition will always return 'false' since the types '"string" | "number" | "boolean" | "symbol" | "undefined" | "object" | "function"' and '"string[]"' have no overlap.

```
14 function logName(pe
15   if (personOrCompa
16     console.log(`${personOrCompany.firstName} ${personOrCompany.surname}`);
17   } else {
18     console.log(personOrCompany.name);
19   }
20 }
```

(parameter) personOrCompany: Person

```
14 function logName(personOrCompany: PersonOrCompany) {
15   if (personOrCompany instanceof Person) {
16     console.log(`${personOrCompany.surname}`);
17   } else {
18     console.log(personOrCompany.name);
19   }
20 }
```

(parameter) personOrCompany: Company

```
12
13
14 getData<IPerson>("/people/1").then(person => console.log(person));
15
```

(parameter) person: IPerson

```
31
32
33
34
35 people.add({name: "Sally"});
36
```

```
Argument of type '{ name: string; }' is not assignable to param
eter of type 'IPerson'.
  Property 'id' is missing in type '{ name: string; }'.
```

```
30
31
32 const items = people.getList();
33
```

const items: IPerson[]

```
24
25                                  condense(stringOrArray: string | string[]):
26                                  string | string[]
27 const condensedText = condense("The c")
28 console.log("condensedText", condensedText);
29
```

```
25
26          const condensedText: string | string[]
27 const condensedText = condense("The c")
28 console.log("condensedText", condensedText);
```

```
28                                  ^ 1/2 condense(string: string): string
29
30 const moreCondensedText = condense("The")
31
```

```
28
29          const moreCondensedText: string
30 const moreCondensedText = condense("The")
31
```

```
1 interface IPerson {
2   id: number;
3   name: string;
4 }
5
6          type PersonProps = "id" | "name"
7 type PersonProps = keyof IPerson;
```

```
1 interface IPerson {
2   id: number;
3   name: string;
4   age: number;
5 }
6
7
8 type PersonProps = keyof IPerson;
```

type PersonProps = "id" | "name" | "age"

```
const addressField: Field<IPerson, "address"> = new Field();
```

Type '"address"' does not satisfy the constraint '"id" | "name" | "age"'.

```
14
15
16
17
18 idField.defaultValue = "2";
```

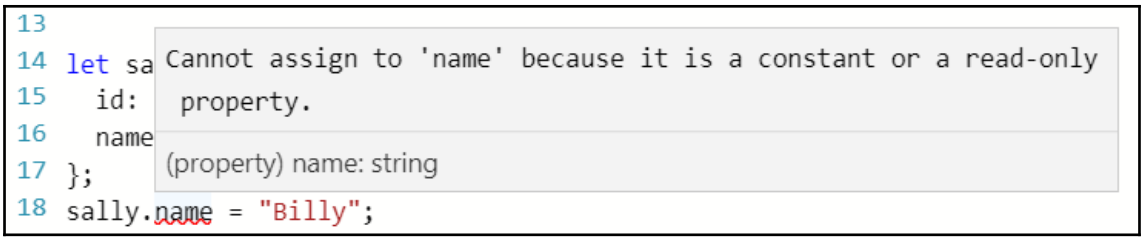
Type '"2"' is not assignable to type 'number'.

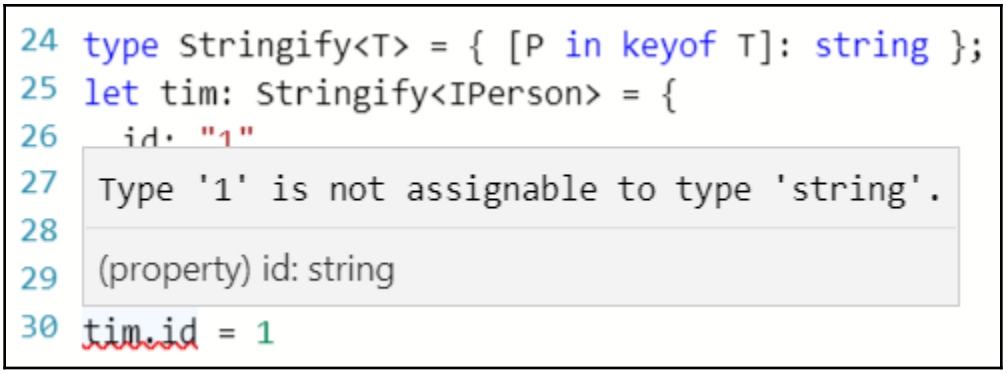
(property) Field<IPerson, "id">.defaultValue: number

```
6 type ReadonlyPerson = { readonly [P in keyof IPerson]: IPerson[P] };
7
8 let bi
9   id:
10  name
11 };
12 billy.name = "Sally";
```

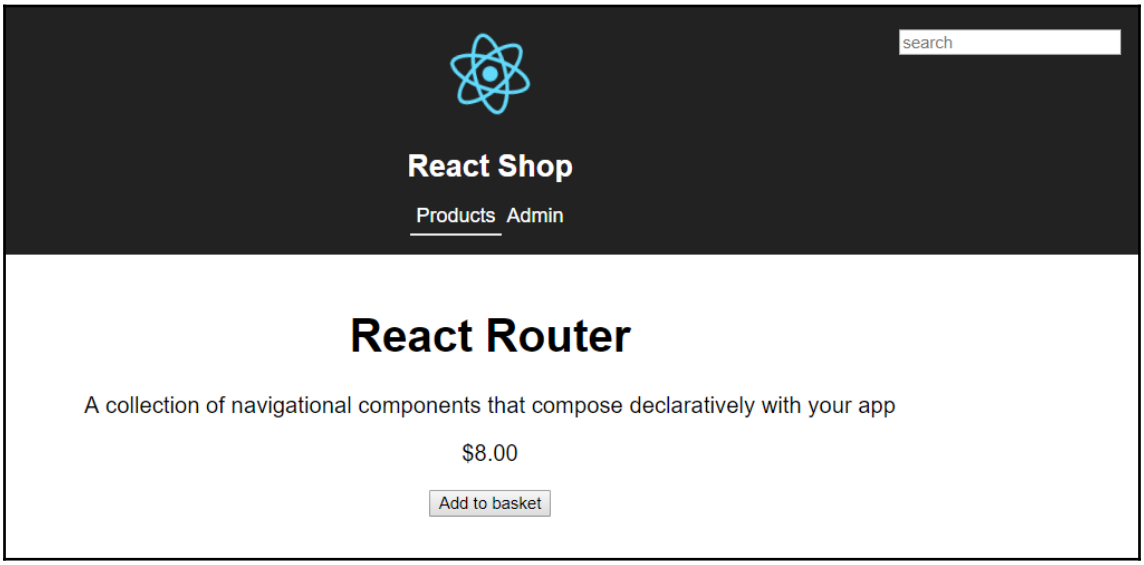
Cannot assign to 'name' because it is a constant or a read-only property.

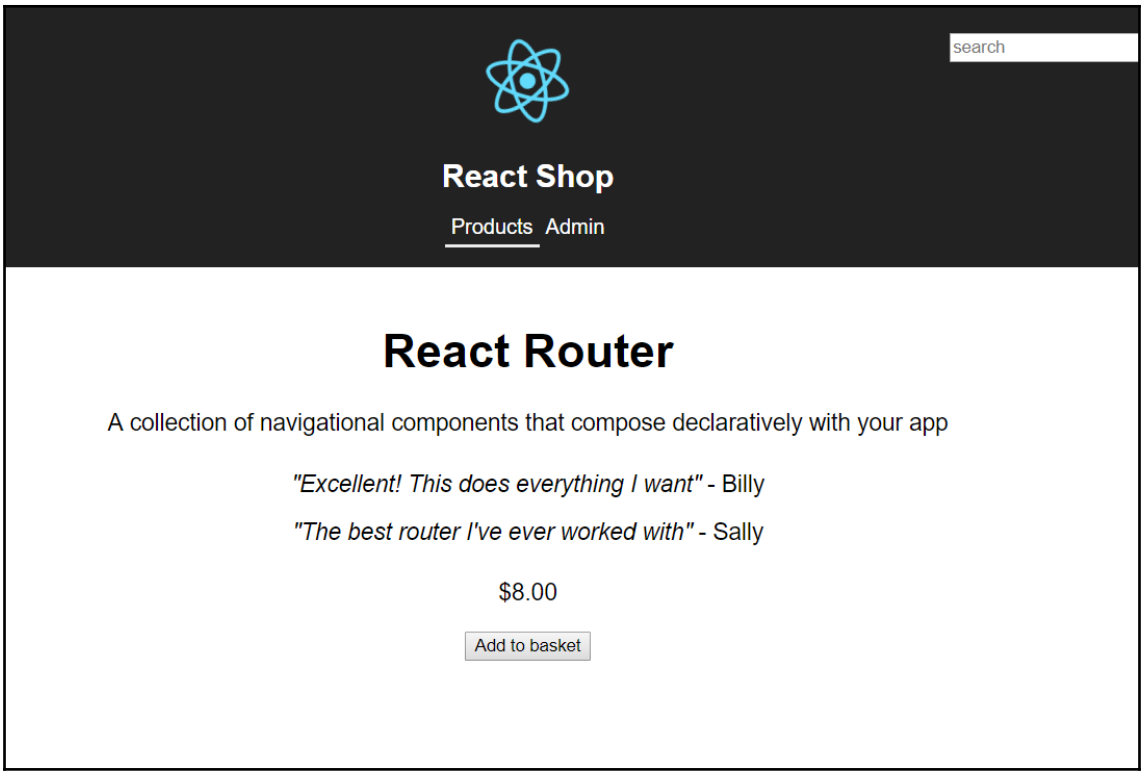
(property) name: string

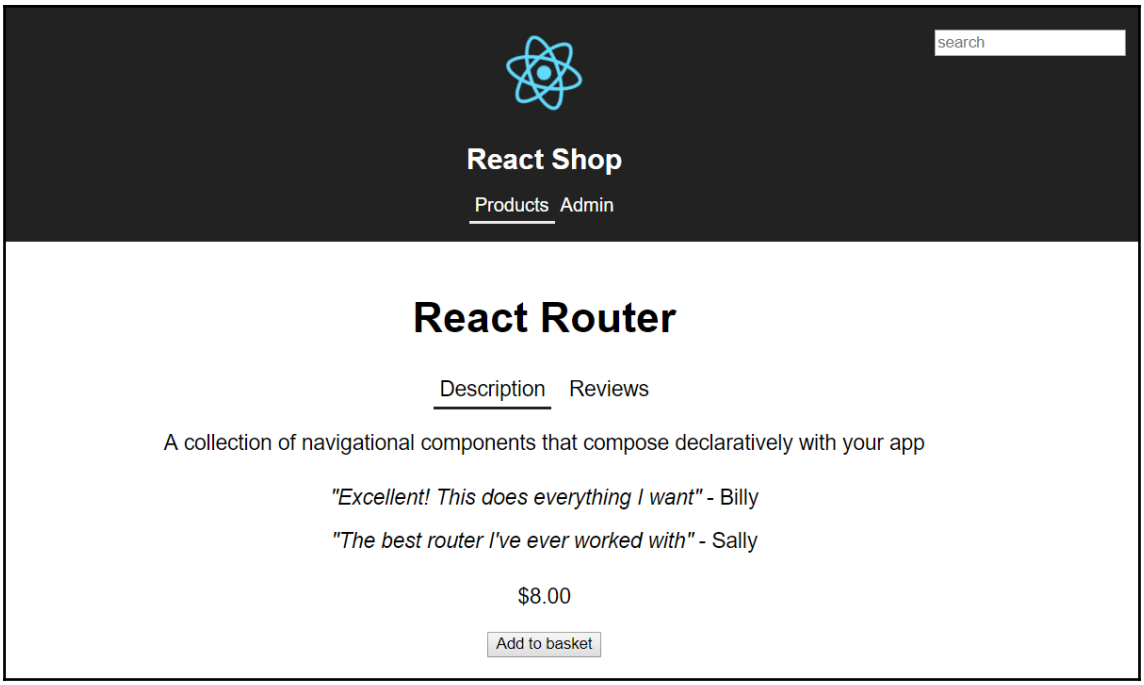

13
14 let sa
15 id:
16 name
17 };
18 sally.name = "Billy";
Cannot assign to 'name' because it is a constant or a read-only property.
(property) name: string

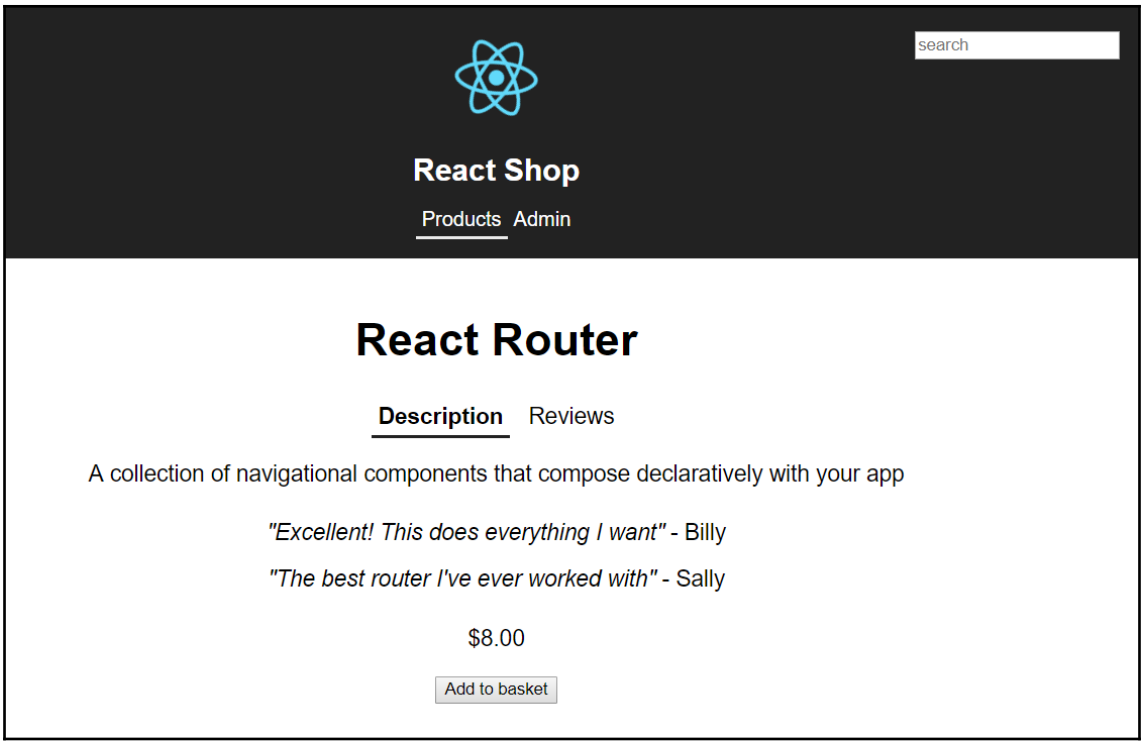

24 type Stringify<T> = { [P in keyof T]: string };
25 let tim: Stringify<IPerson> = {
26 id: "1"
27
28
29
30 tim.id = 1
Type '1' is not assignable to type 'string'.
(property) id: string

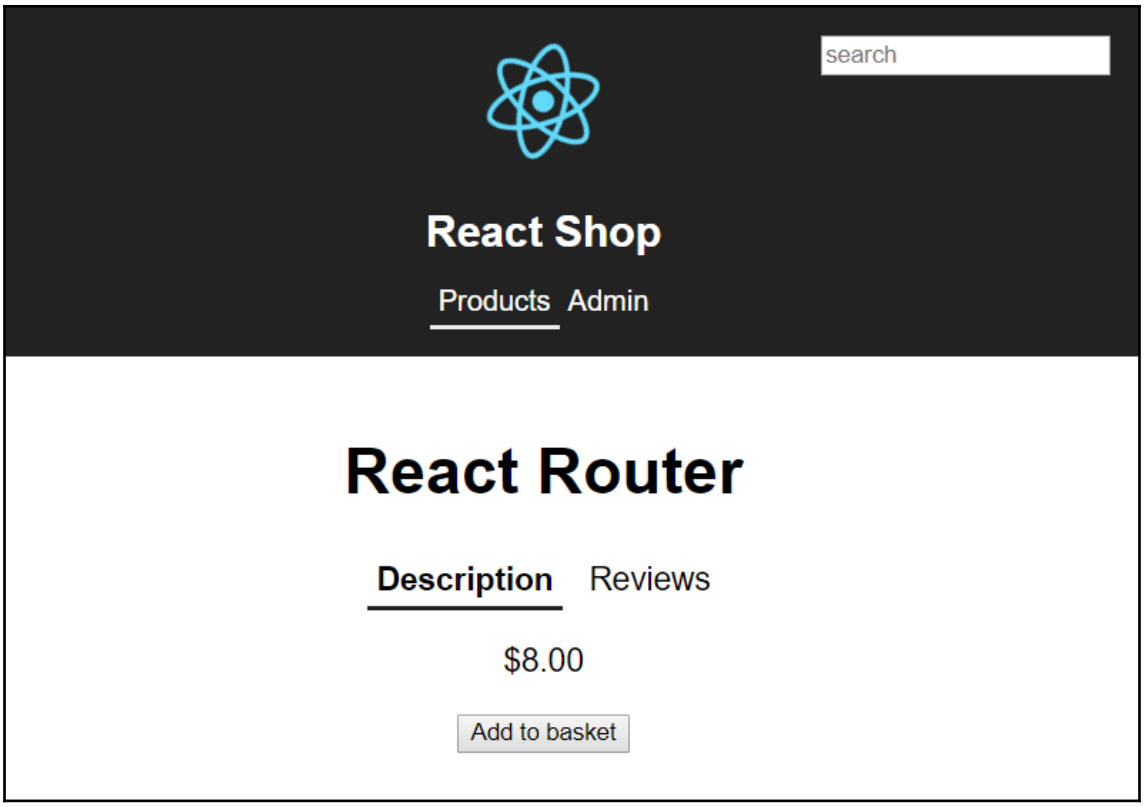
Chapter 6: Component Patterns

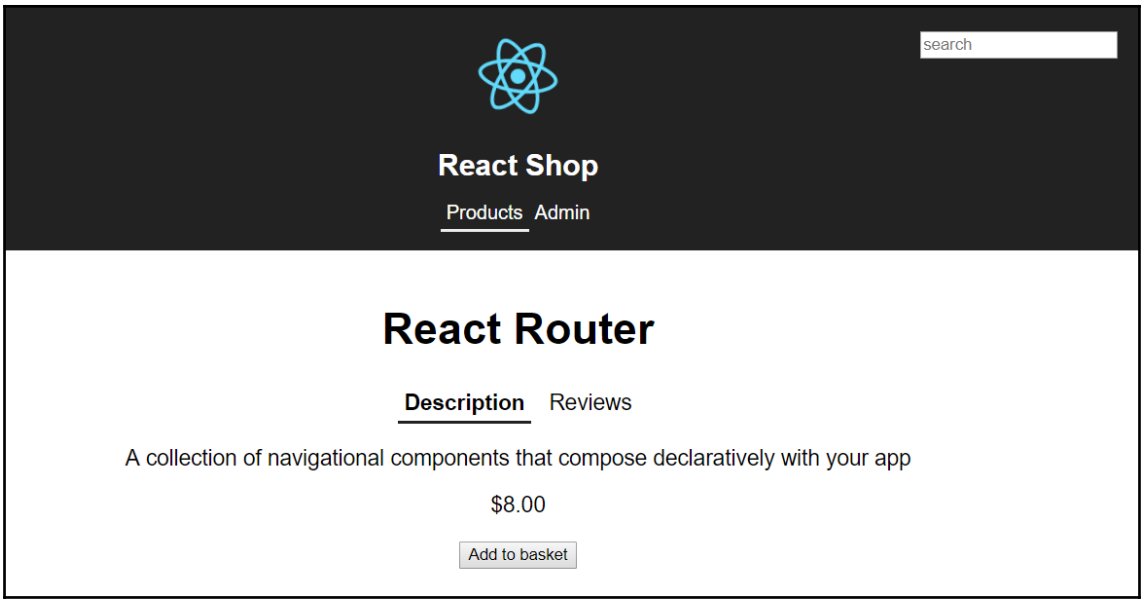


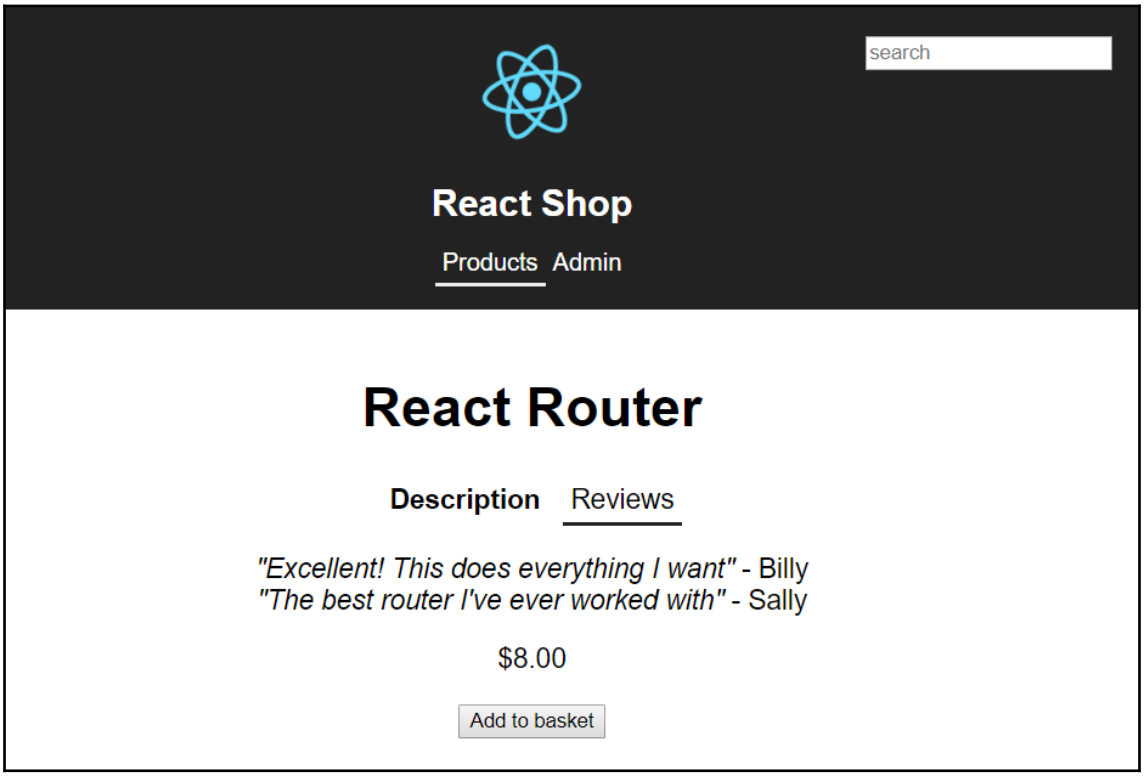

search
React Shop
Products Admin
React Router
A collection of navigational components that compose declaratively with your app
"Excellent! This does everything I want" - Billy
"The best router I've ever worked with" - Sally
$8.00
Add to basket

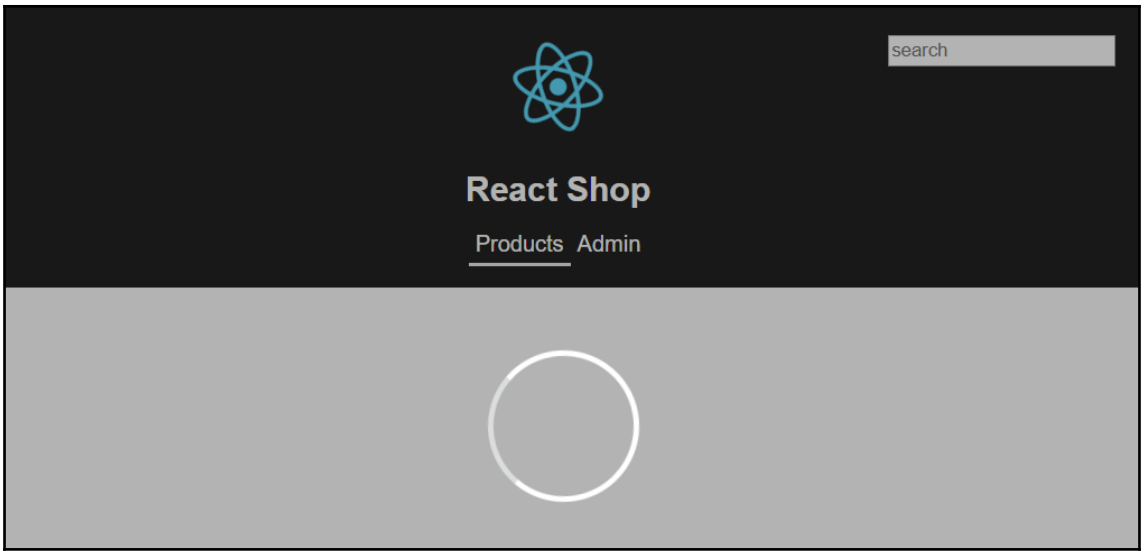

search
React Shop
Products Admin
React Router
Description Reviews
A collection of navigational components that compose declaratively with your app
"Excellent! This does everything I want" - Billy
"The best router I've ever worked with" - Sally
$8.00
Add to basket


search
React Shop
Products Admin
React Router
Description Reviews
A collection of navigational components that compose declaratively with your app
"Excellent! This does everything I want" - Billy
"The best router I've ever worked with" - Sally
$8.00
Add to basket

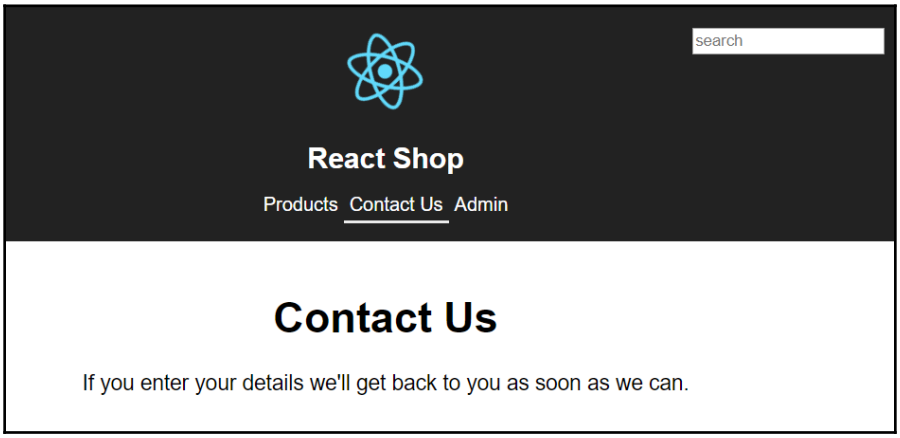

search
React Shop
Products
Admin
React Router
Description
Reviews
$8.00
Add to basket

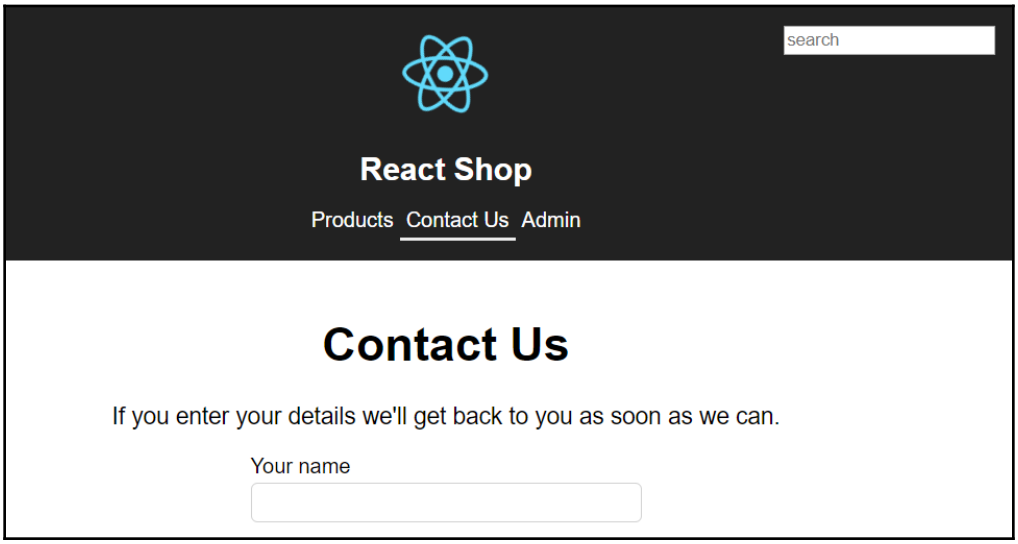

search
React Shop
Products Admin
React Router
Description Reviews
A collection of navigational components that compose declaratively with your app
$8.00
Add to basket

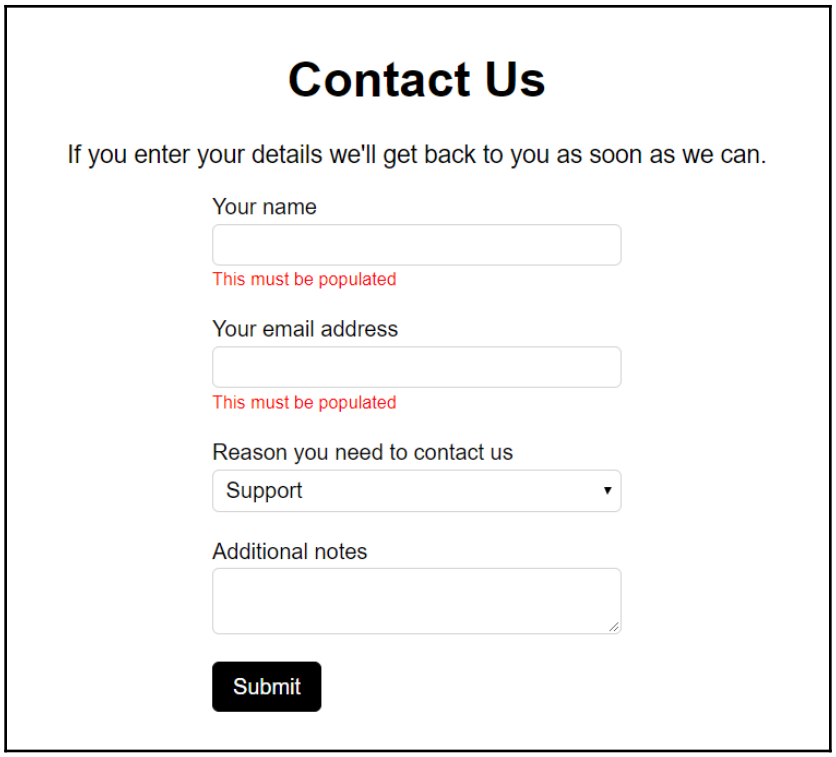

search
React Shop
Products Admin
React Router
Description Reviews
"Excellent! This does everything I want" - Billy
"The best router I've ever worked with" - Sally
$8.00
Add to basket

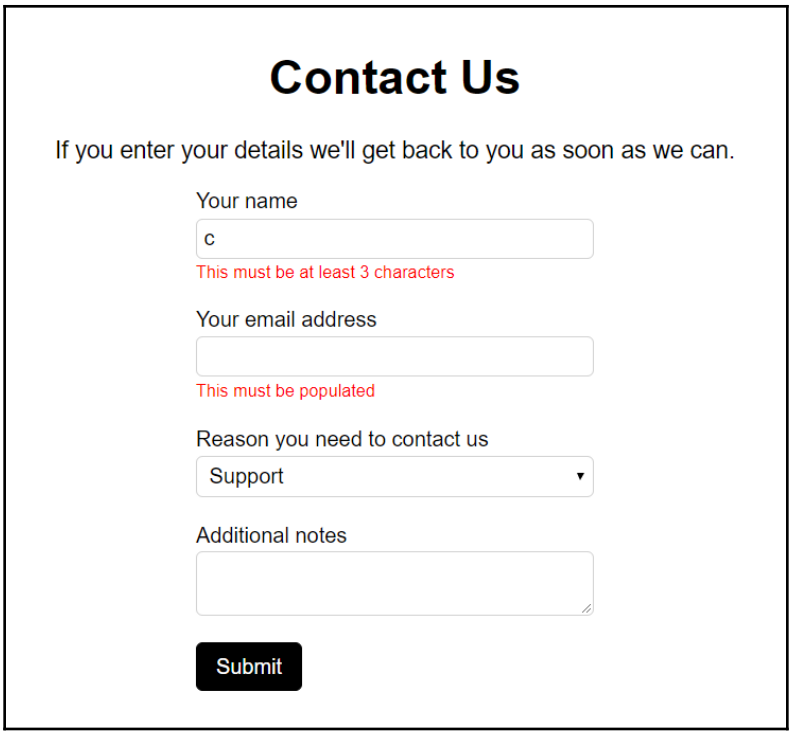

search
React Shop
Products
Admin

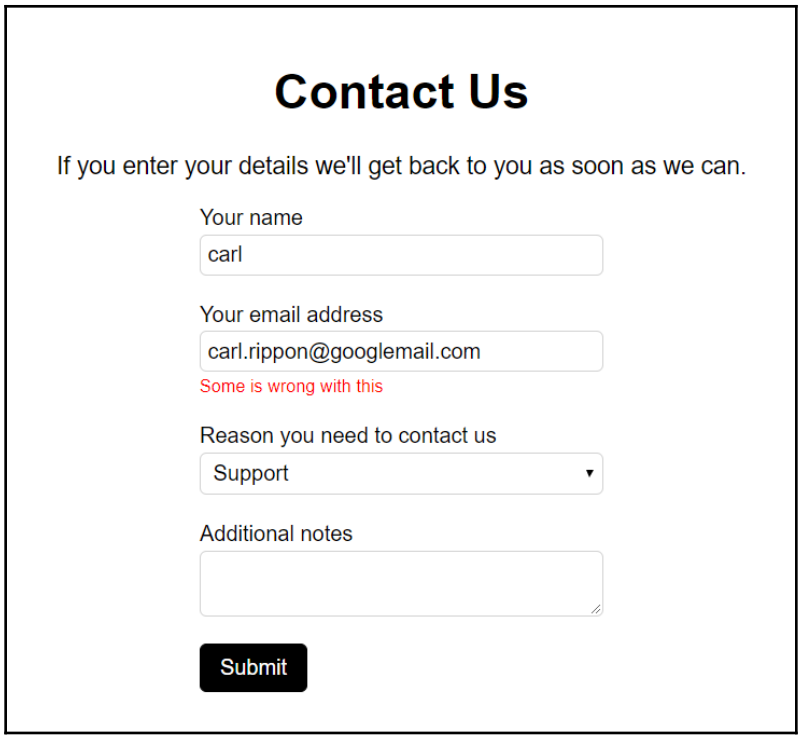
Chapter 7: Working with Forms

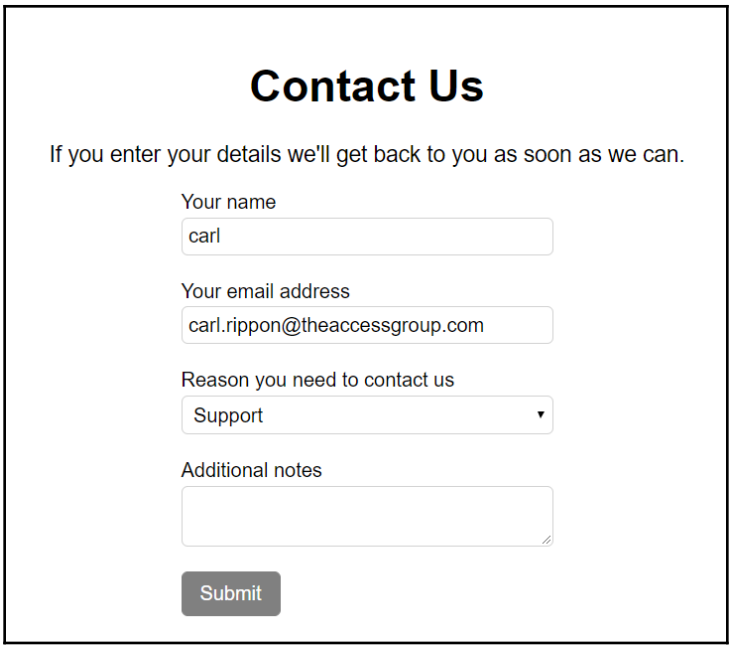




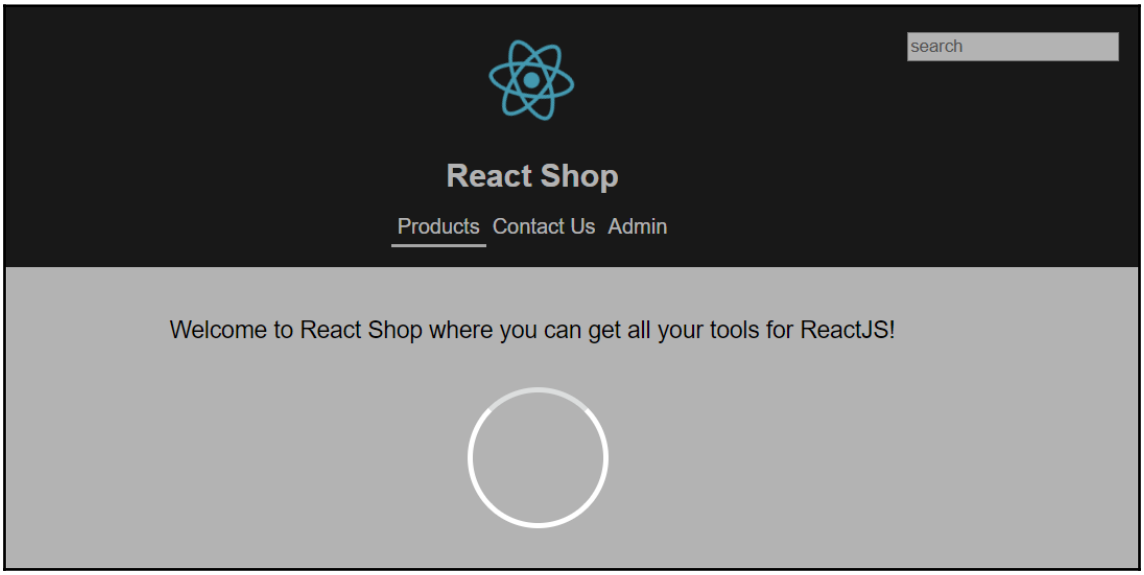

Contact Us
If you enter your details we'll get back to you as soon as we can.
Your name
This must be populated
Your email address
This must be populated
Reason you need to contact us
Support
Additional notes
Submit

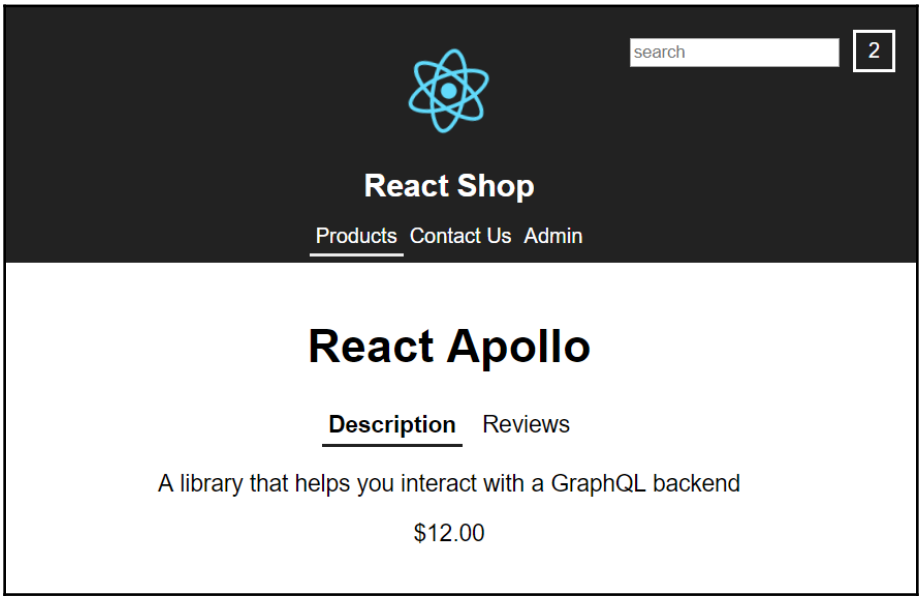

Contact Us
If you enter your details we'll get back to you as soon as we can.
Your name
c
This must be at least 3 characters
Your email address
This must be populated
Reason you need to contact us
Support
Additional notes
Submit

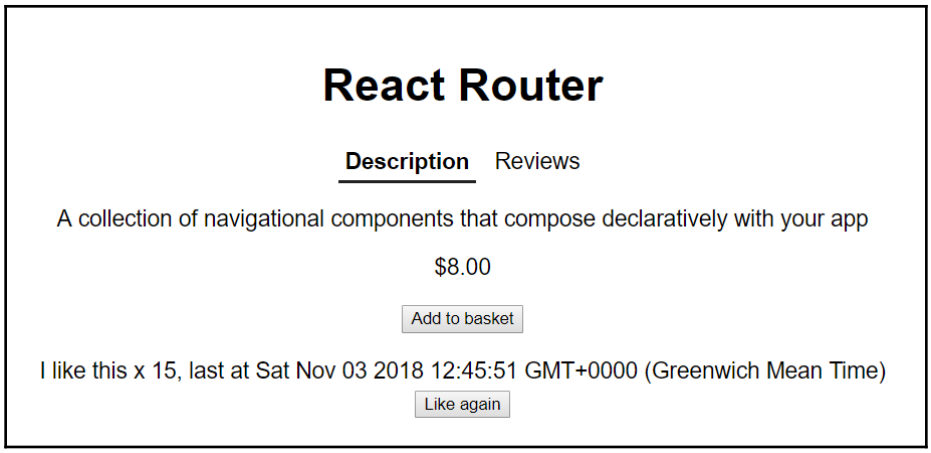

Contact Us
If you enter your details we'll get back to you as soon as we can.
Your name
carl
Your email address
carl.rippon@googlemail.com
Some is wrong with this
Reason you need to contact us
Support
Additional notes
Submit


Contact Us
If you enter your details we'll get back to you as soon as we can.
Your name
carl
Your email address
carl.rippon@theaccessgroup.com
Reason you need to contact us
Support
Additional notes
Submit

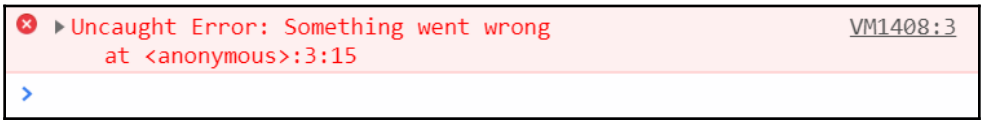
Chapter 8: React Redux

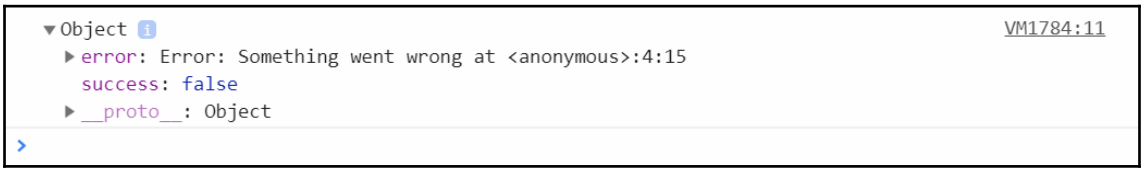


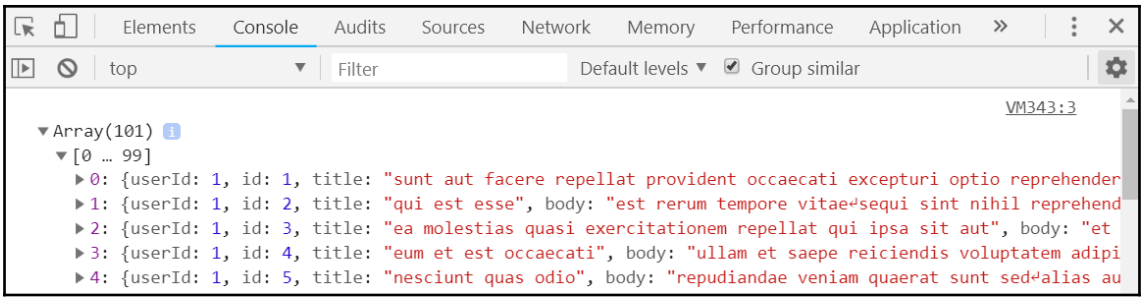



React Router
Description
Reviews
A collection of navigational components that compose declaratively with your app
$8.00
Add to basket
I like this x 15, last at Sat Nov 03 2018 12:45:51 GMT+0000 (Greenwich Mean Time)
Like again

Chapter 9: Interacting with RESTful APIs







sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

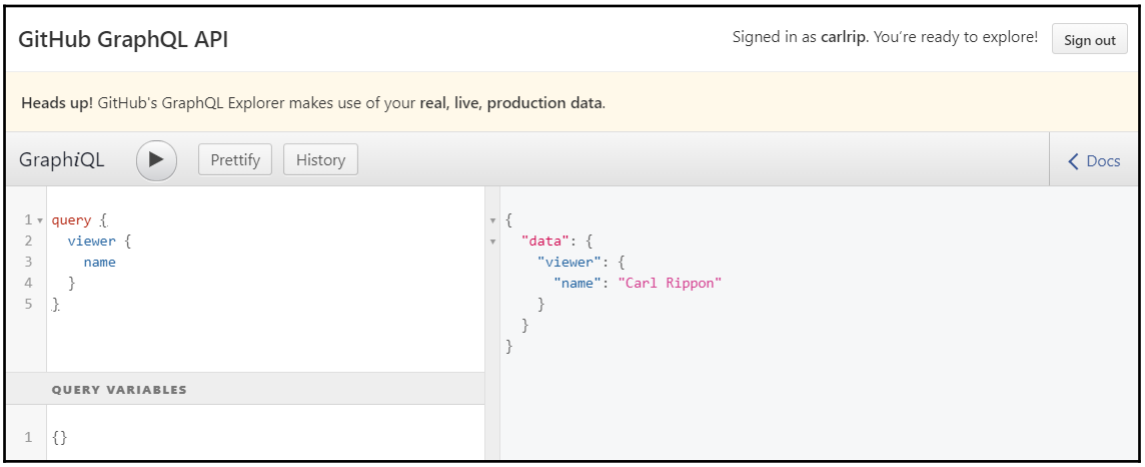
qui est esse

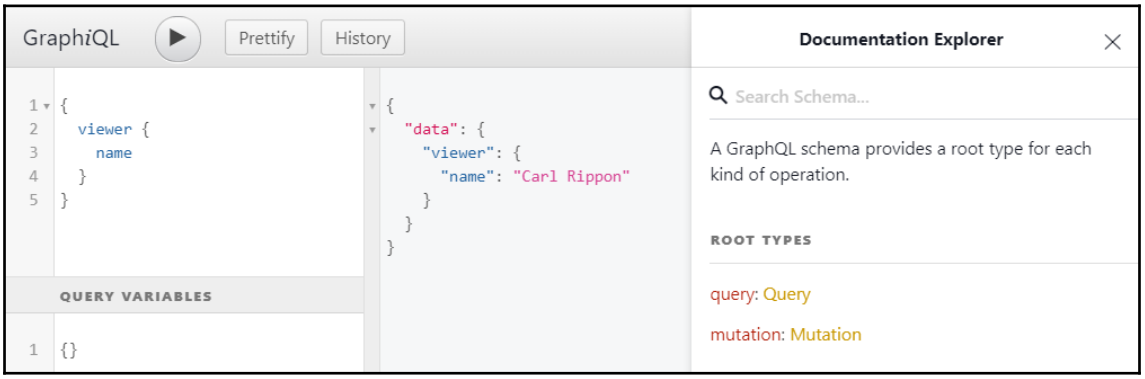
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

Chapter 10: Interacting with GraphQL APIs





```
query {
  repository (owner:"facebook", name:"react") {
    name
    description
    stargazers {
      totalCount
    }
  }
}
```

```
{
  "data": {
    "repository": {
      "name": "react",
      "description": "A declarative, efficient, and flexible JavaScript library for
building user interfaces.",
      "stargazers": {
        "totalCount": 115001
      }
    }
  }
}
```

```
query ($org: String!, $repo: String!) {
  repository (owner:$org, name:$repo) {
    name
    description
    stargazers {
      stars:totalCount
    }
    issues(last: 5) {
```

QUERY VARIABLES

```
{
  "org": "facebook",
  "repo": "react"
}
```

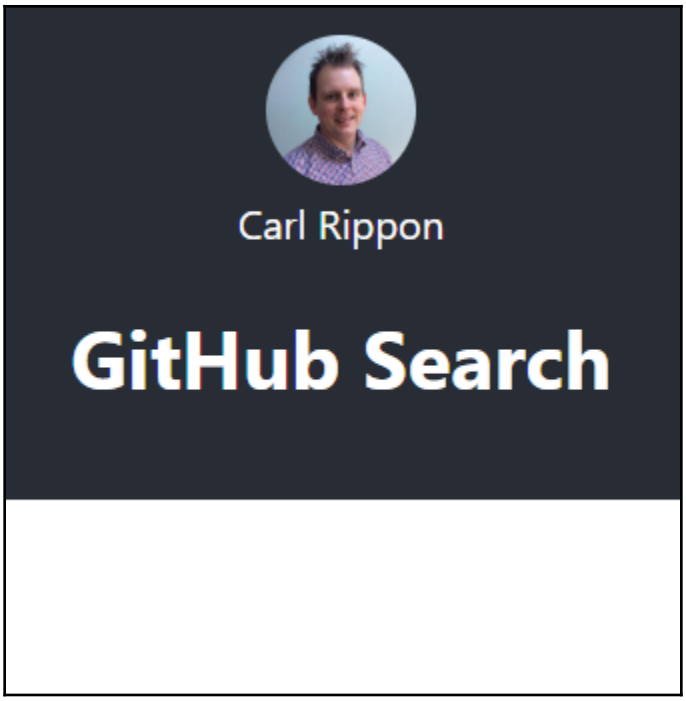
```
{
  "data": {
    "repository": {
      "name": "react",
      "description": "A declarative, efficient, and flexible JavaScript library
for building user interfaces.",
      "stargazers": {
        "stars": 115002
      },
      "issues": {
        "edges": [
          {
            "node": {
              "id": "MDU6SXNzdWUzNzg0NTg2MDc=",
              "title": "Tapping outside of an input on iOS does not fire onBlur
```

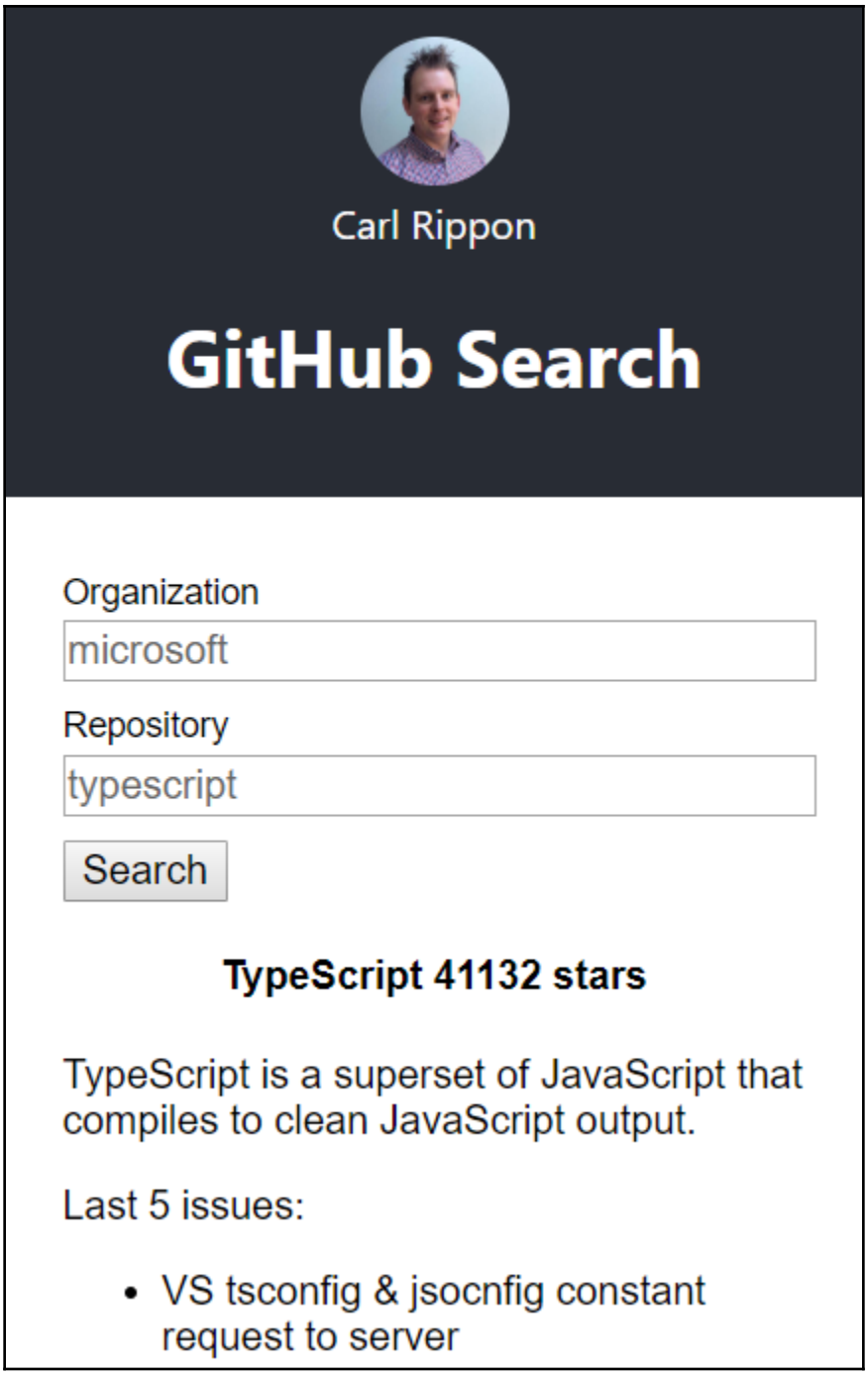
```
mutation ($repoId: ID!) {
  addStar(input: { starrableId: $repoId }) {
    starrable {
      stargazers {
        totalCount
      }
    }
  }
}
```

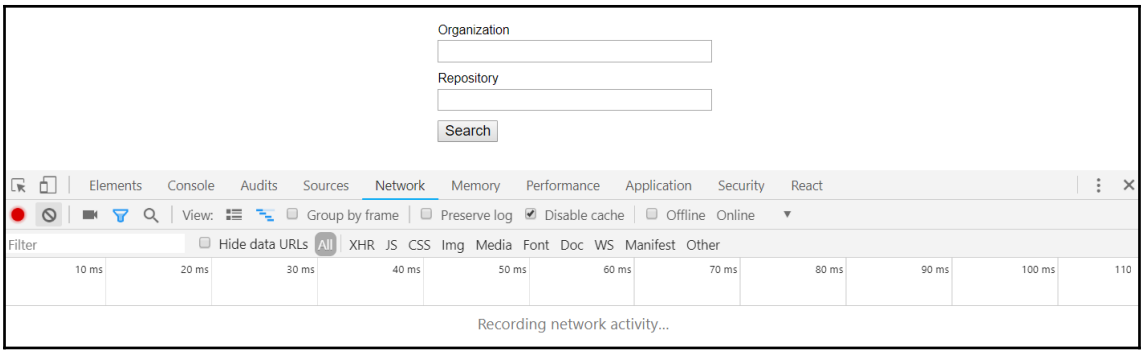
QUERY VARIABLES

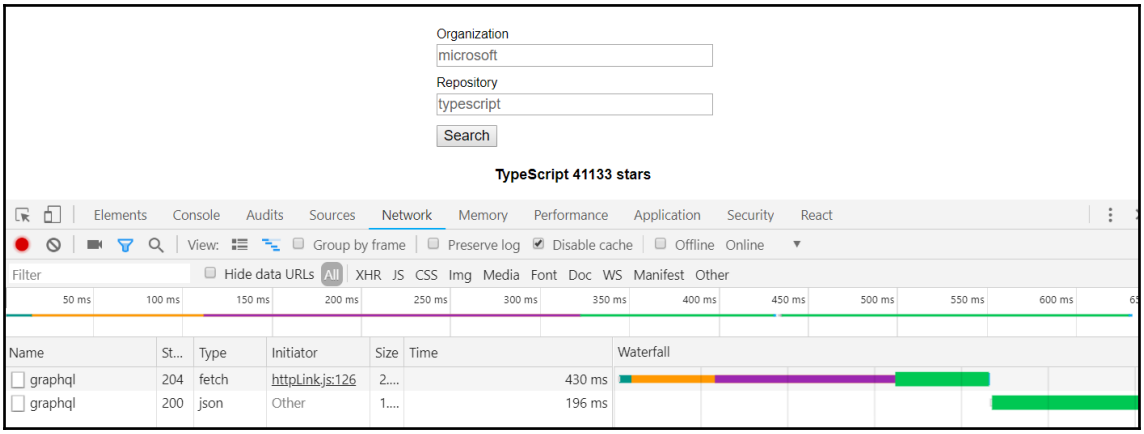
```
{
  "repoId": "MDEwOlJlcG9zaXRvcnkxMDI3MDI1MA=="
}
```

```
{
  "data": {
    "addStar": {
      "starrable": {
        "stargazers": {
          "totalCount": 115066
        }
      }
    }
  }
}
```


Carl Rippon
GitHub Search


Carl Rippon
GitHub Search
Organization
microsoft
Repository
typescript
Search
TypeScript 41132 stars
TypeScript is a superset of JavaScript that compiles to clean JavaScript output.
Last 5 issues:
VS tsconfig & jsocnfig constant request to server


Organization
Repository
Search
Elements
Console
Audits
Sources
Network
Memory
Performance
Application
Security
React
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
10 ms
20 ms
30 ms
40 ms
50 ms
60 ms
70 ms
80 ms
90 ms
100 ms
110
Recording network activity...


Organization
microsoft
Repository
typescript
Search
TypeScript 41133 stars
Elements
Console
Audits
Sources
Network
Memory
Performance
Application
Security
React
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
50 ms
100 ms
150 ms
200 ms
250 ms
300 ms
350 ms
400 ms
450 ms
500 ms
550 ms
600 ms
Name
St...
Type
Initiator
Size
Time
Waterfall
graphql
204
fetch
httpLink.js:126
2....
430 ms
graphql
200
json
Other
1....
196 ms

Chapter 11: Unit Testing with Jest

```
> react-scripts test
 PASS  src/Form.test.tsx
  √ When required called with title being an empty string,
an error should be 'This must be populated' (3ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.613s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press q to quit watch mode.
 › Press Enter to trigger a test run.
```

```
FAIL src/Form.test.tsx
  x When required called with title being an empty string, an error should be 'This must be pop
ulated' (19ms)

  ● When required called with title being an empty string, an error should be 'This must be pop
ulated'

    expect(received).toBe(expected) // Object.is equality

    Expected: "This must be populatedX"
    Received: "This must be populated"

       6 |   };
       7 |   const result = required("title", values);
    >  8 |   expect(result).toBe("This must be populatedX");
         |                  ^
       9 | });
      10 |

      at Object.toBe (src/Form.test.tsx:8:18)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        3.412s
Ran all test suites.
```

```
No failed test found.
Press `f` to quit "only failed tests" mode.

Watch Usage: Press w to show more.
```

```
Pattern Mode Usage
 › Press Esc to exit pattern mode.
 › Press Enter to filter by a filenames regex pattern.

 pattern › form
```

```
Active Filters: filename /form/

Pattern Mode Usage
 › Press Esc to exit pattern mode.
 › Press Enter to filter by a tests regex pattern.

 pattern › required
```

```
PASS src/Form.test.tsx
  required
    √ When required called with title being an empty string, an error should be 'This must be populated' (1ms
)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.388s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

```
PASS src/Form.test.tsx
PASS src/ContactUs.test.tsx

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        4.359s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

```
 › 1 snapshot written.

Snapshot Summary
 › 1 snapshot written from 1 test suite.

Test Suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:   1 written, 1 total
Time:        4.241s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

```
Received value does not match stored snapshot "ContactUs Renders okay 1".

- Snapshot
+ Received

@@ -1,89 +1,91 @@
  <div>
    <form
      class="form"
      novalidate=""
    >
-     <div
-       class="form-group"
-     >
-       <label
-         for="name"
+     <div>
+       <div
+         class="form-group"
        >
-         Your name
-       </label>
-       <input
-         id="name"
-         type="text"
-         value=""
-       />
-     </div>
```

```
Snapshot Summary
 › 1 snapshot updated from 1 test suite.

Test Suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:   1 updated, 1 total
Time:        2.901s, estimated 3s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

```
  App
    √ When page loads, posts are rendered (43ms)

---------------------------|----------|----------|----------|----------|-------------------|
File                       |  % Stmts | % Branch |  % Funcs |  % Lines | Uncovered Line #s |
---------------------------|----------|----------|----------|----------|-------------------|
All files                  |     8.74 |     4.35 |     10.2 |     8.74 |                   |
 App.tsx                   |    33.33 |    21.43 |    26.32 |    33.33 |... 71,177,180,181 |
 index.tsx                 |        0 |      100 |      100 |        0 |      1,2,3,4,5,12 |
 registerServiceWorker.ts  |        0 |        0 |        0 |        0 |... 11,118,119,120 |
 serviceWorker.ts          |        0 |        0 |        0 |        0 |... 31,138,139,140 |
---------------------------|----------|----------|----------|----------|-------------------|
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   1 passed, 1 total
Time:        3.97s
Ran all test suites.
```

All files

8.74% Statements 9/103 **4.35%** Branches 3/69 **10.2%** Functions 5/49 **8.74%** Lines 9/103

File		Statements		Branches		Functions		Lines	
App.tsx		33.33%	9/27	21.43%	3/14	26.32%	5/19	33.33%	9/27
index.tsx		0%	0/6	100%	0/0	100%	0/0	0%	0/6
registerServiceWorker.ts		0%	0/32	0%	0/23	0%	0/15	0%	0/32
serviceWorker.ts		0%	0/38	0%	0/32	0%	0/15	0%	0/38

```
35      public componentDidMount() {
36  1x    const cancelToken = axios.CancelToken;
37  1x    const cancelTokenSource = cancelToken.source();
38  1x    this.setState({ cancelTokenSource });
39  1x    axios
40          .get<IPost[]>("https://jsonplaceholder.typicode.com/posts", {
41            cancelToken: cancelTokenSource.token,
42            headers: {
43              "Content-Type": "application/json"
44            },
45            timeout: 5000
46          })
47          .then(response => {
48  1x        this.setState({ posts: response.data, loading: false });
49          })
50          .catch(ex => {
51            const error = axios.isCancel(ex)
52              ? "Request cancelled"
53              : ex.code === "ECONNABORTED"
54              ? "A timeout has occurred"
55              : ex.response.status === 404
56              ? "Resource not found"
57              : "An unexpected error has occurred";
58            this.setState({ error, loading: false });
59          });
60
61        // cancelTokenSource.cancel("User cancelled operation");
62      }
```