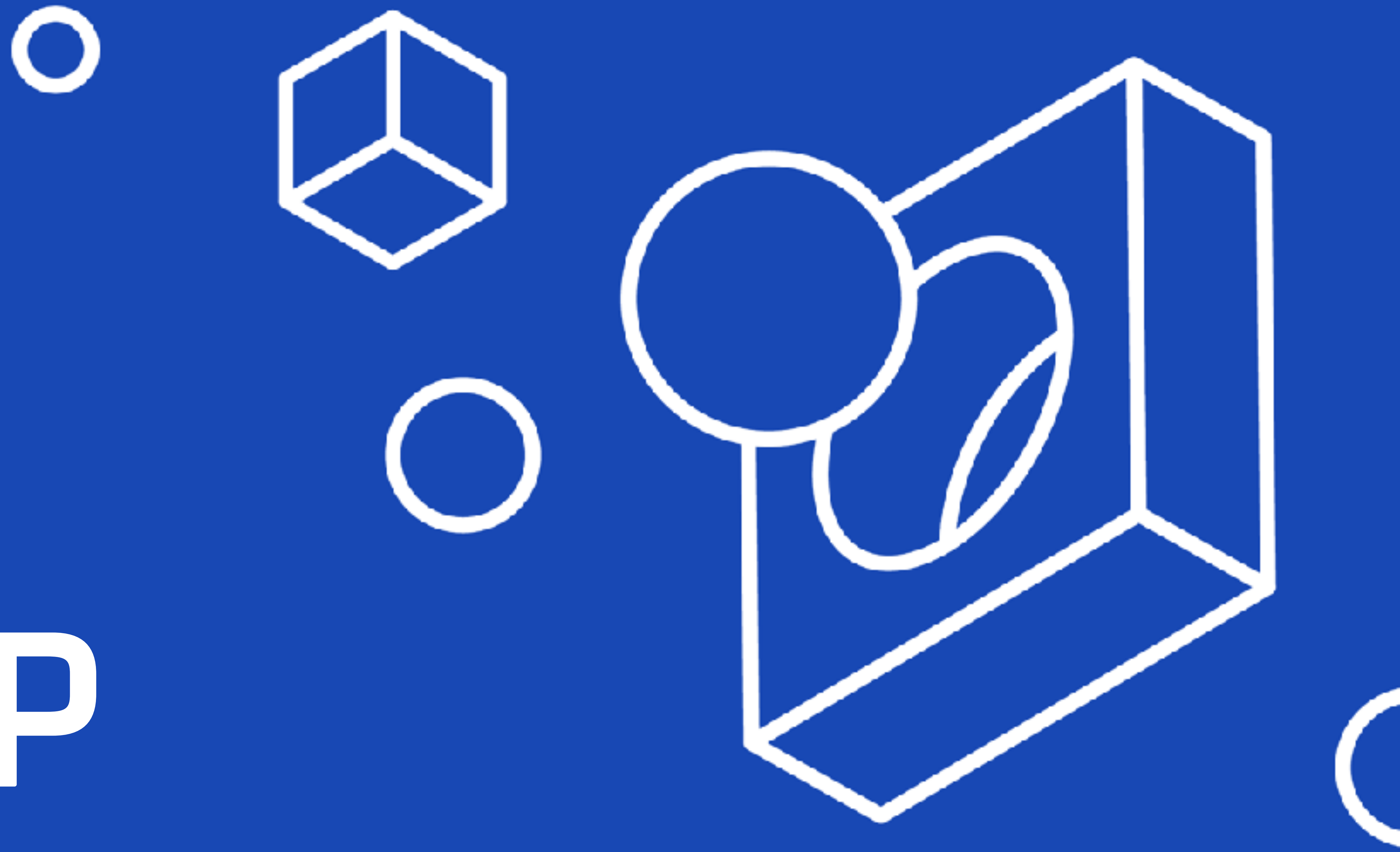
DEV SUMMIT 2018



WEB WORKSHOP

MATT SECCAFIEN
Front-end Developer

MICHELLE CHEN
Front-end Developer

INTRODUCTIONS

AGENDA

Introduce you to a number of foundational principles and best practises for web development at Shopify.

Demonstrate new tooling we've been working on help get you productive quickly.

Expose you to different parts of the Web stack and provide a base understanding of the individual purpose of each as well as how they work together.

WEB FOUNDATION PRINCIPLES

https://github.com/Shopify/web-foundation/tree/master/Principles

USER OVER TEAM OVER SELF

API OVER IMPLEMENTATION

EXPLICIT OVER AUTOMATIC

COMMUNITY OVER OWNERSHIP

DECLARATIVE OVER IMPERATIVE

ISOLATION OVER INTEGRATION

WEBGEN

A new tool we built initially for RnD Summit to help developers scaffold code quickly and easily.

```
yarn create shopify <COMMAND> <NAME>
```

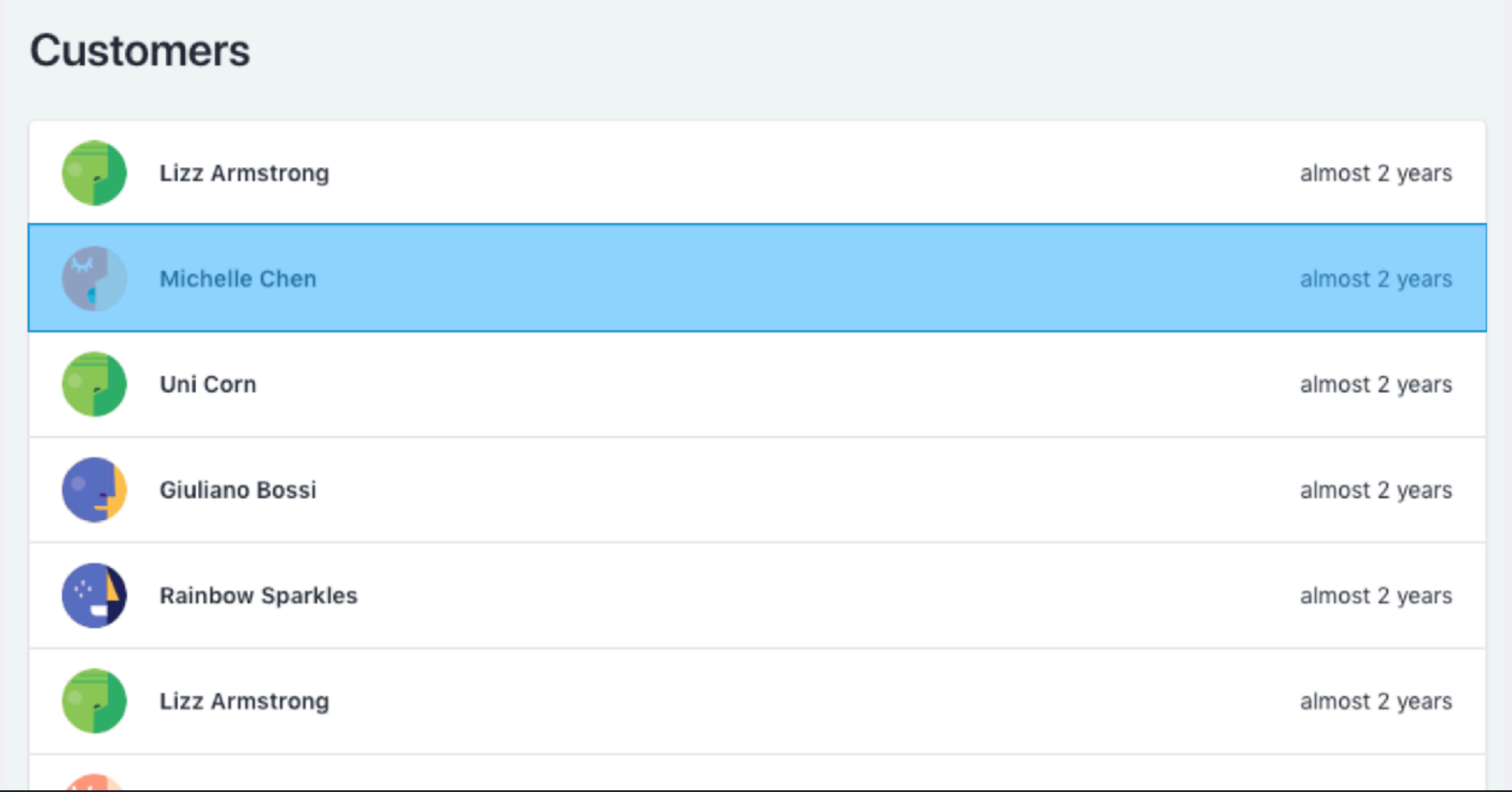
COMMANDS

```
app
component
section (coming soon)
polaris (coming soon)
```

```
yarn create shopify --help
```

APP STRUCTURE

What's included



Customers

Lizz Armstrong	almost 2 years
Michelle Chen	almost 2 years
Uni Corn	almost 2 years
Giuliano Bossi	almost 2 years
Rainbow Sparkles	almost 2 years
Lizz Armstrong	almost 2 years

WHAT ARE WE GOING TO BUILD?

https://invis.io/4DN0ENNZMGA#/309239104

CODE TIME

TYPESCRIPT

What it is and why we "all-in" on it at Shopify.

WHAT IS TYPESCRIPT?

Typescript is a strict syntactical superset of JavaScript that adds static typing to the language where types of the variables is known at runtime.

WHAT WE LOVE ABOUT TS

Easy to find compile time errors.

Tonnes of community support.

An additional layer of documentation.

"Typescript is hard. I hate it."

EVERY DEVELOPER

After ~1 week with TS

"How did we ever write code without Typescript, it is the Sh*t!"

EVERY DEVELOPER

After ~3 weeks with TS

EXERCISE: CUSTOMER SHOW

Use Webgen to create a new "function" component, name it "CustomerShow" and put it in the Customers section.

Add another Route to the Customers section for "/show" and configure it to point to this new component.

Consult the mock up for what Polaris components you will need and them to the "render" function within your new component

Take a break! If you finish early take a look at the react-form-state repo from quilt. We will be implementing this next.

https://github.com/Shopify/quilt/blob/master/packages/react-form-state

GRAPHQL

What it is and why we "all-in" on it at Shopify.

WHAT IS GRAPHQL?

GraphQL Schema Language is a shorthand notation to succinctly express the basic shape of your GraphQL schema and its type system.

PROBLEMS WITH REST APIS

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

FETCHING ASSOCIATED DATA OFTEN REQUIRES MULTIPLE HTTP CALLS

```
/products
/products/1
/products/1/variants
/products/1/variants/1
```

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

ENDPOINTS HAVE NO WAY OF KNOWING WHAT THE CLIENT IS USING

```
product: {
  unused_foo: “foo”,
  unused_bar: “bar”,
  images: […]
  variants: {
  // …more fields
  }
  // …more fields
}
```

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

WEAKLY TYPED AND RELIES ON DOCUMENTATION TO EXPRESS ITS CAPABILITIES AND DATA

```
product: {
  price: “$100.00”,
  createdAt: “2006-04…”
  // …more fields
```

SOLUTIONS WITH GRAPHQL

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

PASS A QUERY DOCUMENT TO A SINGLE ENPOINT

```
https://…/graphql
```

POST REQUEST

```
{
  "query": "...",
  "operationName": "...",
}
```

JSON RESPONSE

```
{
  "data": { ... },
  "errors": [ ... ]
}
```

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

CLIENTS DETERMINE THE SHAPE OF THE RESPONSE

QUERY

```
{
  shop {
    id
    name
  }
}
```

RESPONSE

```
{
  “shop”: {
    “id”: “123"
    “name”: “Snow Devil”
  }
}
```

MULTIPLE HTTP CALLS

COMMONLY OVER-FETCHING

POORLY DOCUMENTED

TYPED SCHEMA AND DOCUMENTATION LIVES SIDE-BY-SIDE WITH CODE

SCHEMA.GRAPHQL

```
type Product {
  price: Int!
  taxable: Boolean
  createdAt: DateTime!
  // … more fields
```

SCHEMA.GRAPHQL.D.TS

```
interface Product {
  price: Schema.Money;
  taxable: boolean;
  createdAt: Date;
  // … more fields
```

REVIEW

Performance
one end-point, no over-fetching

Declarative Data-fetching
client asks only for what it needs

Developer Experience
self documenting, strong type system

GRAPHIQL

Load GraphiQL for any shop by going to
https://app.shopify.com/services/internal/shops/YOUR_SHOP_ID/graphql

Get your shop's ID from
https://app.shopify.com/services/internal/shops

CODE TIME

EXERCISE: QUERY VARIABLES

Use React-Router query parameters to pass the customer ID to CustomerShow.

Back on CustomerIndex, add a select box that orders the customers in an alternate way.

EXERCISE: DELETE CUSTOMER

Combine everything we've learned so far to add the ability to delete customers to the app.

Add GraphQL Mutation for customer deletion.

Add this new GraphQL Mutation to the graphql HOC

Add the necessary UI to Trigger the GraphQL mutation.

RESOURCES

WebGen
https://github.com/Shopify/webgen

Web
https://github.com/Shopify/web

Web Foundation
https://github.com/Shopify/web-foundation

Quilt
https://github.com/Shopify/quilt

On Slack
#webgen
#web-foundation
#web-foundation-apps
#libaray-extraction

@cartogram
@michelle.chen

THANK YOU