Delivering Headless Commerce

Matt Glaman | mglaman | 🐦 @nmdmatt

Matt Glaman

Product Lead, Commerce Guys

Drupal Commerce co-maintainer

Drupal 8 Development Cookbook

ContribKanban.com

What is...

Headless Commerce

Headless Commerce…

Separates the customer experience layer from the core business application.

Headless Commerce...

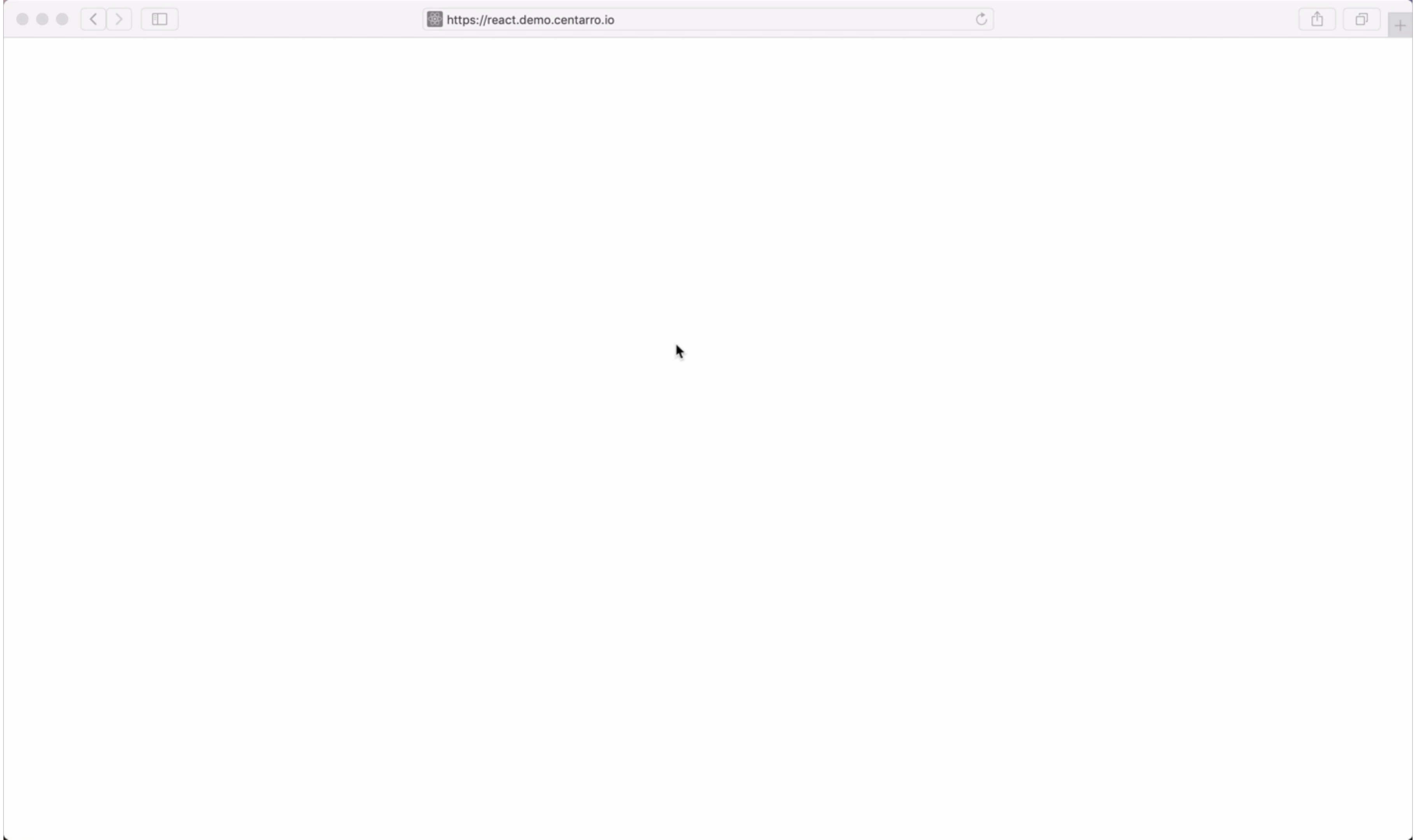
Supports any number of front end sites or apps from a single back end.

Headless Commerce...

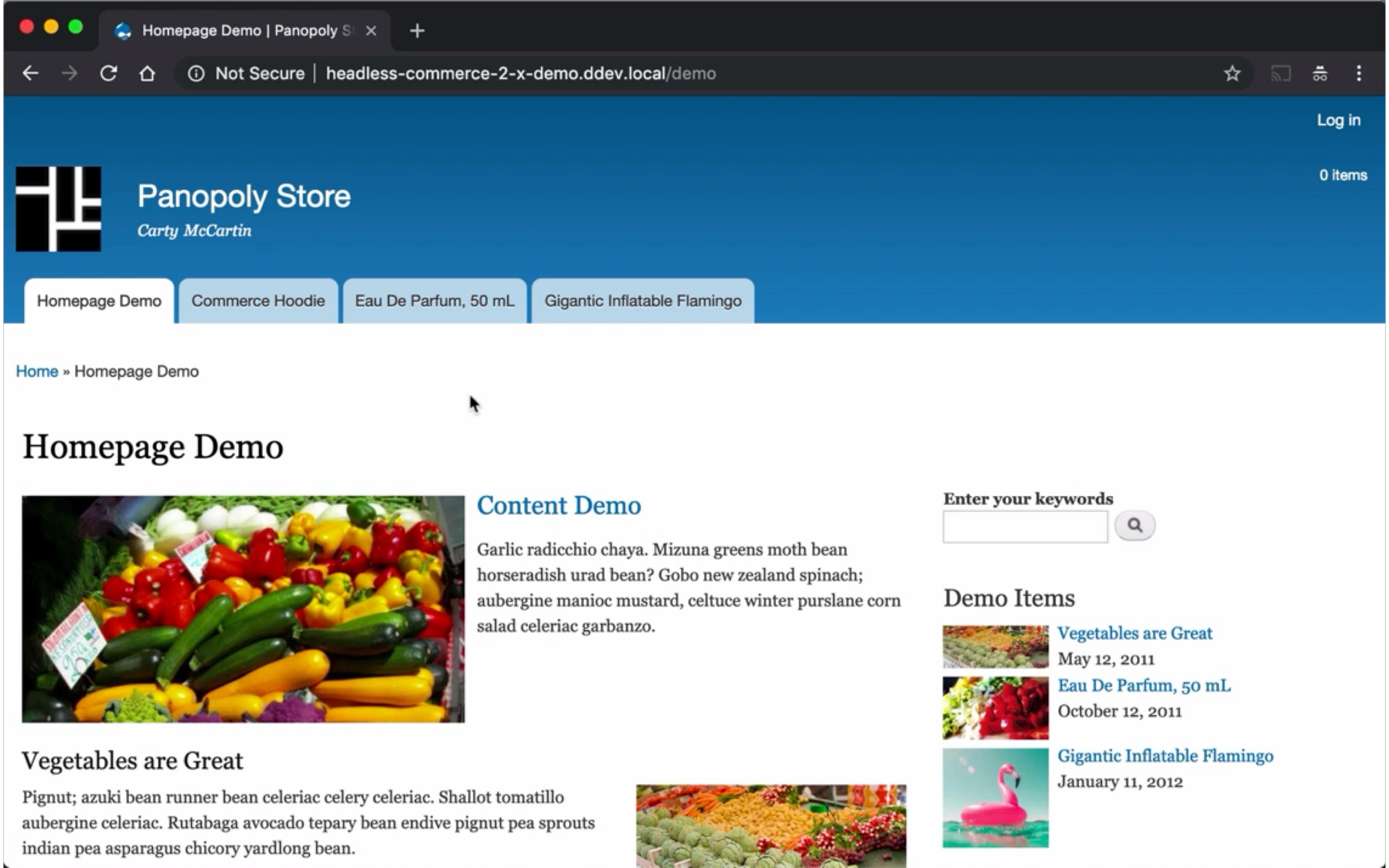
Evolve and scale parts of your commerce platform without impacting others.

For example…

React front end interacting with a Drupal Commerce API backend


https://react.demo.centarro.io

And… Panopoly?!?!
DRUPAL 7 ?!?$#@


Homepage Demo | Panopoly S
Not Secure | headless-commerce-2-x-demo.ddev.local/demo
Log in
0 items
Homepage Demo
Commerce Hoodie
Eau De Parfum, 50 mL
Gigantic Inflatable Flamingo
Home » Homepage Demo

Panopoly Store

Carty McCartin

Homepage Demo

Content Demo

Garlic radicchio chaya. Mizuna greens moth bean horseradish urad bean? Gobo new zealand spinach; aubergine manioc mustard, celtuce winter purslane corn salad celeriac garbanzo.

Vegetables are Great

Pignut; azuki bean runner bean celeriac celery celeriac. Shallot tomatillo aubergine celeriac. Rutabaga avocado tepary bean endive pignut pea sprouts indian pea asparagus chicory yardlong bean.

Enter your keywords

Demo Items

Vegetables are Great
May 12, 2011

Eau De Parfum, 50 mL
October 12, 2011

Gigantic Inflatable Flamingo
January 11, 2012

Challenges

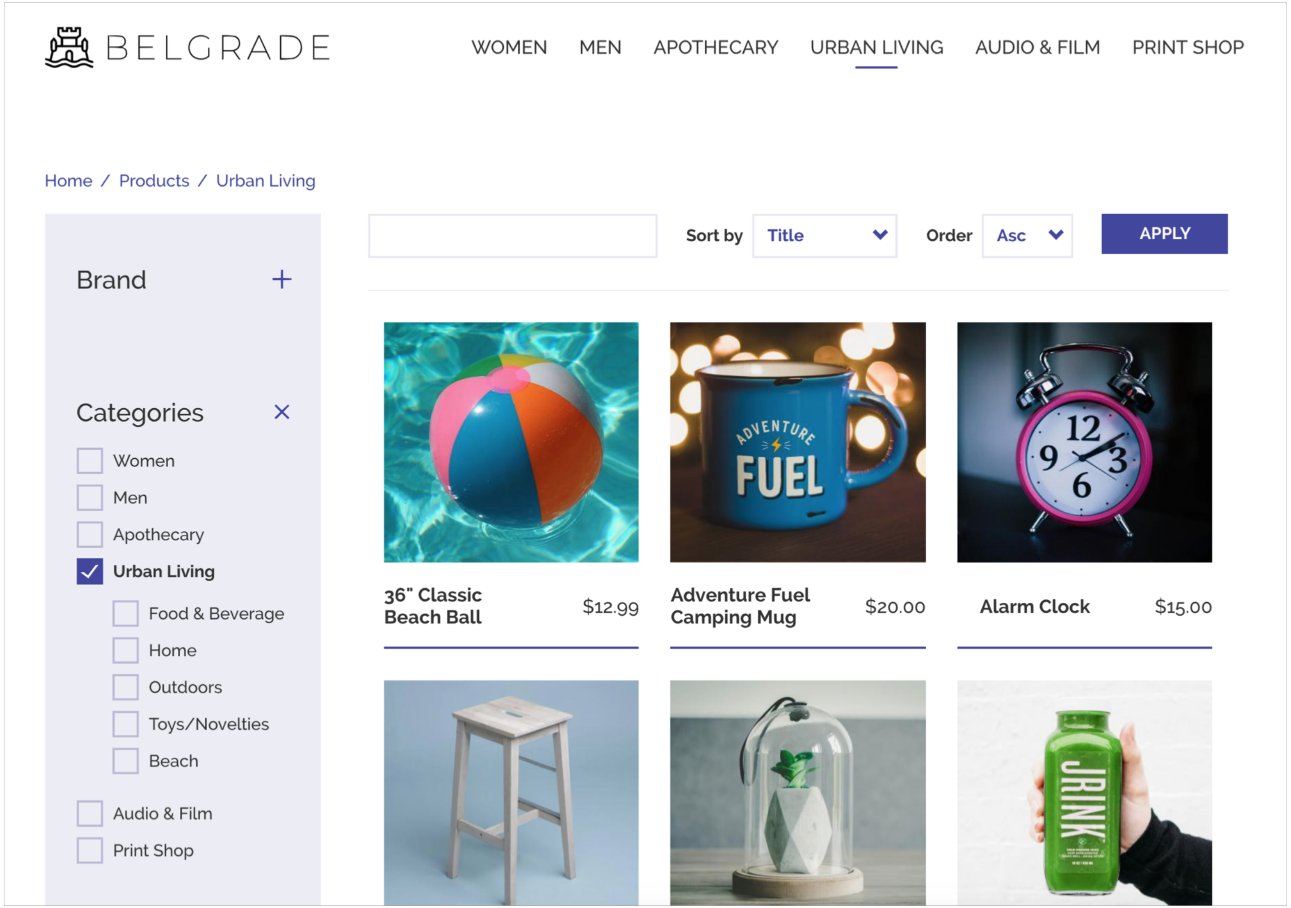
- Structured data model
- Multiple relationships and layers
- Data *vs* Presentation layer

Available options

- GraphQL
- JSON API (now in Drupal core!)
- RESTful Web Services* (Drupal core)
- JSON-RPC

* RESTful Web Services module does not provide any default schema, filtering, or collections but does allow building your own.

Catalogs



Home / Products / Urban Living

Sort by Title

Order Asc

Brand +

Categories ×

□ Women
□ Men
□ Apothecary
☑ Urban Living
  □ Food & Beverage
  □ Home
  □ Outdoors
  □ Toys/Novelties
  □ Beach
□ Audio & Film
□ Print Shop

36" Classic Beach Ball	$12.99
Adventure Fuel Camping Mug	$20.00
Alarm Clock	$15.00

Standard Drupal: Views + Search API

Catalog areas of concern

- Highly customized queries for products
- Powered by search indexes
- Utilize faceted searches for fine tuned browsing
- Variable pricing based on catalog context
- Product availability

{ json:api }

- Strong query capabilities
- Ability to include referenced entities and specify returned fields
- Declarative and discoverable
- *Does not* support mixed-bundle collections out of the box.
- *Could* integrate JSON API into Search API with some effort.

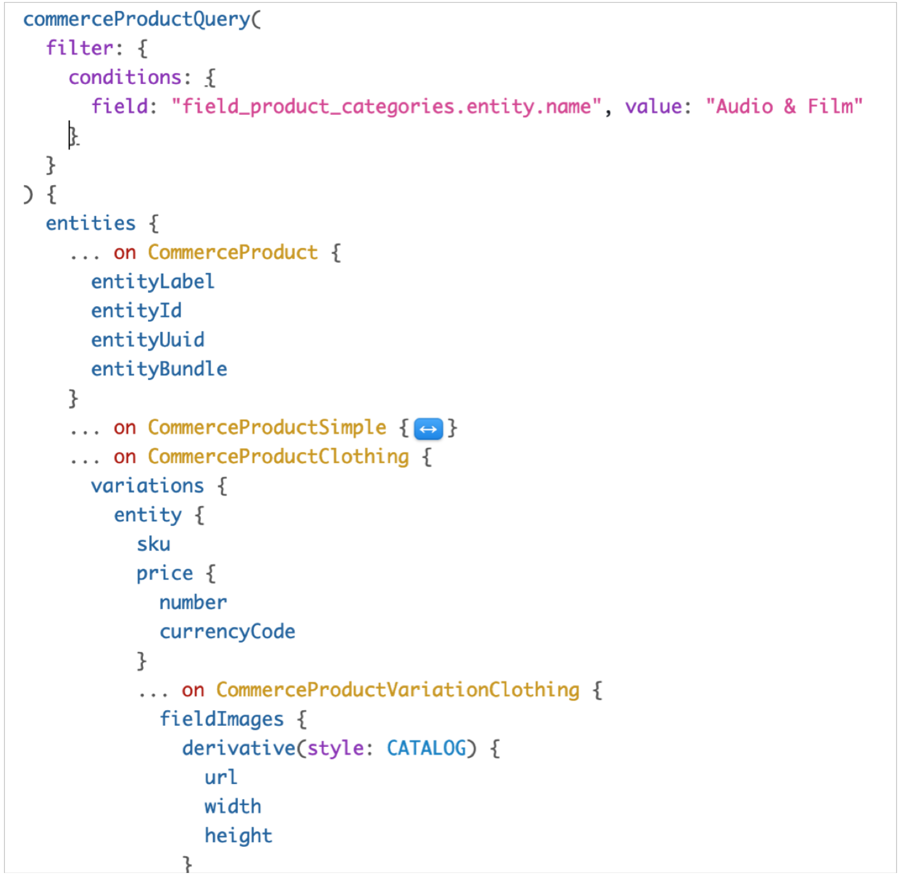
```
GET
/jsonapi/commerce_product/simple

GET
/jsonapi/commerce_product/clothing

GET
/jsonapi/commerce_product/
```

GraphQL

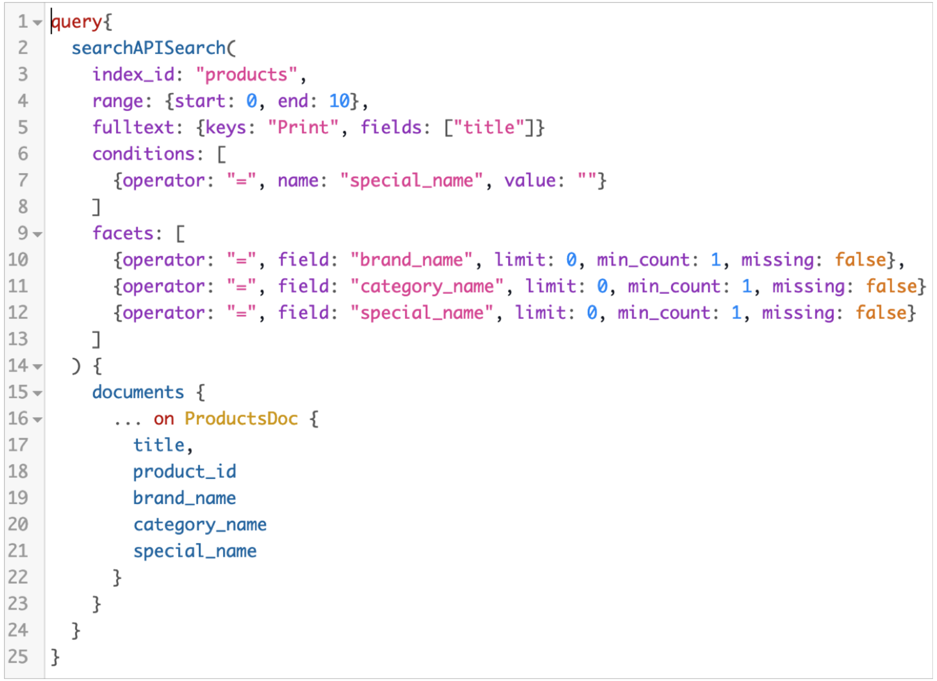
- Query-like language API
- Supports cross-bundle querying
- Use Graph*i*QL IDE to build the query
- Query can become *very large.*
- Reusable queries, as query maps are possible.



```
commerceProductQuery(
  filter: {
    conditions: {
      field: "field_product_categories.entity.name", value: "Audio & Film"
    }
  }
) {
  entities {
    ... on CommerceProduct {
      entityLabel
      entityId
      entityUuid
      entityBundle
    }
    ... on CommerceProductSimple {↔}
    ... on CommerceProductClothing {
      variations {
        entity {
          sku
          price {
            number
            currencyCode
          }
          ... on CommerceProductVariationClothing {
            fieldImages {
              derivative(style: CATALOG) {
                url
                width
                height
              }
```

GraphQL & Search API

- GraphQL has Search API integration through a contributed module
- Query support using full text searching
- Faceted search support
  (not Facets module)
- Only returns data in the index
- More like directly querying the search index with Drupal as a proxy.



```
query{
  searchAPISearch(
    index_id: "products",
    range: {start: 0, end: 10},
    fulltext: {keys: "Print", fields: ["title"]}
    conditions: [
      {operator: "=", name: "special_name", value: ""}
    ]
    facets: [
      {operator: "=", field: "brand_name", limit: 0, min_count: 1, missing: false},
      {operator: "=", field: "category_name", limit: 0, min_count: 1, missing: false}
      {operator: "=", field: "special_name", limit: 0, min_count: 1, missing: false}
    ]
  ) {
    documents {
      ... on ProductsDoc {
        title,
        product_id
        brand_name
        category_name
        special_name
      }
    }
  }
}
```

Product Display Pages

Product display page areas of concern

- Made up of marketing content and product data
- Representation of variations and their field values
- Product attribute display (ie: color swatches)
- Add to cart forms

{ json:api }

- JSON API is all about linking resources, which is useful for rich content

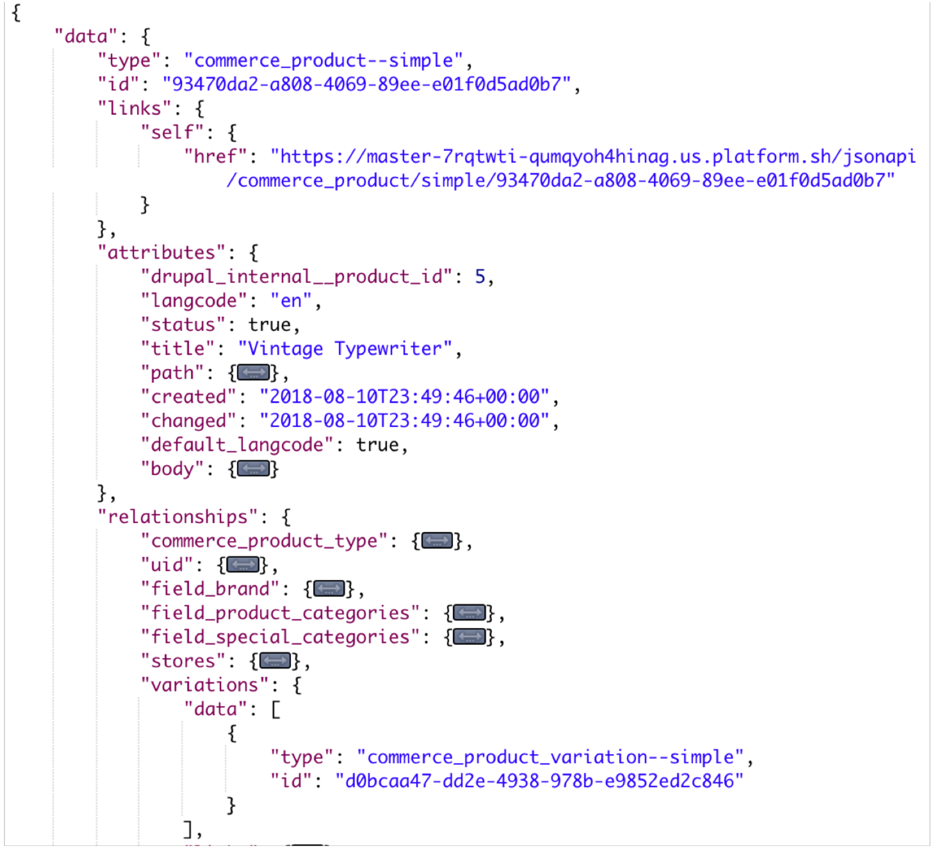
- Supports *includes* that will fetch linked resources automatically.

- Allows specifying specific fields to be returned

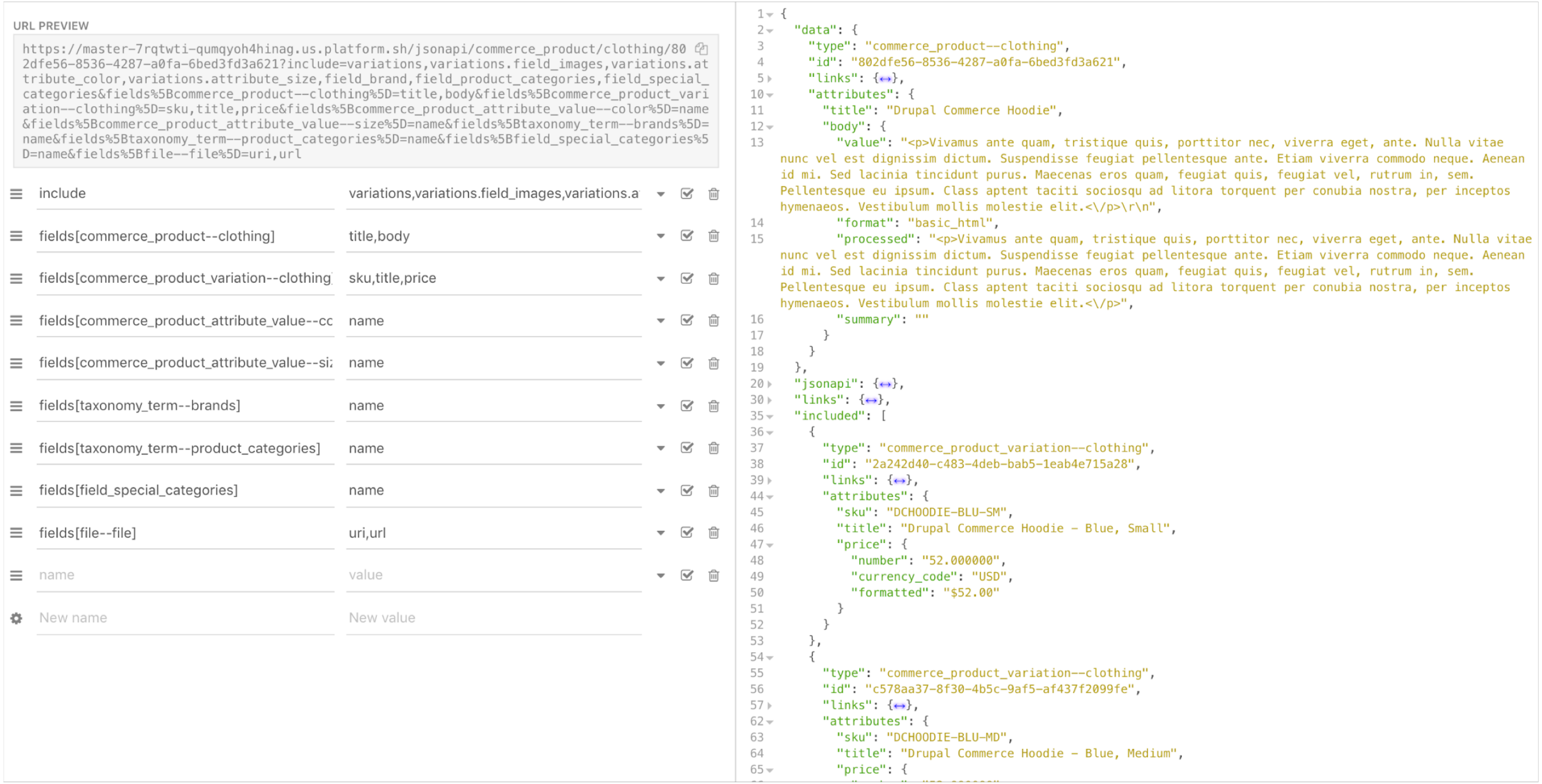
- *But*, routes require the bundle to be present

  /jsonapi/commerce_product/simple/93470da2...e01f0d5ad0b7

  /jsonapi/commerce_product/clothing/93470da2...e01f0d5ad0b7



```
{
    "data": {
        "type": "commerce_product--simple",
        "id": "93470da2-a808-4069-89ee-e01f0d5ad0b7",
        "links": {
            "self": {
                "href": "https://master-7rqtwti-qumqyoh4hinag.us.platform.sh/jsonapi
                    /commerce_product/simple/93470da2-a808-4069-89ee-e01f0d5ad0b7"
            }
        },
        "attributes": {
            "drupal_internal__product_id": 5,
            "langcode": "en",
            "status": true,
            "title": "Vintage Typewriter",
            "path": {↔},
            "created": "2018-08-10T23:49:46+00:00",
            "changed": "2018-08-10T23:49:46+00:00",
            "default_langcode": true,
            "body": {↔}
        },
        "relationships": {
            "commerce_product_type": {↔},
            "uid": {↔},
            "field_brand": {↔},
            "field_product_categories": {↔},
            "field_special_categories": {↔},
            "stores": {↔},
            "variations": {
                "data": [
                    {
                        "type": "commerce_product_variation--simple",
                        "id": "d0bcaa47-dd2e-4938-978b-e9852ed2c846"
                    }
                ],
```



URL PREVIEW

```
https://master-7rqtwti-qumqyoh4hinag.us.platform.sh/jsonapi/commerce_product/clothing/802dfe56-8536-4287-a0fa-6bed3fd3a621?include=variations,variations.field_images,variations.attribute_color,variations.attribute_size,field_brand,field_product_categories,field_special_categories&fields%5Bcommerce_product--clothing%5D=title,body&fields%5Bcommerce_product_variation--clothing%5D=sku,title,price&fields%5Bcommerce_product_attribute_value--color%5D=name&fields%5Bcommerce_product_attribute_value--size%5D=name&fields%5Btaxonomy_term--brands%5D=name&fields%5Btaxonomy_term--product_categories%5D=name&fields%5Bfield_special_categories%5D=name&fields%5Bfile--file%5D=uri,url
```

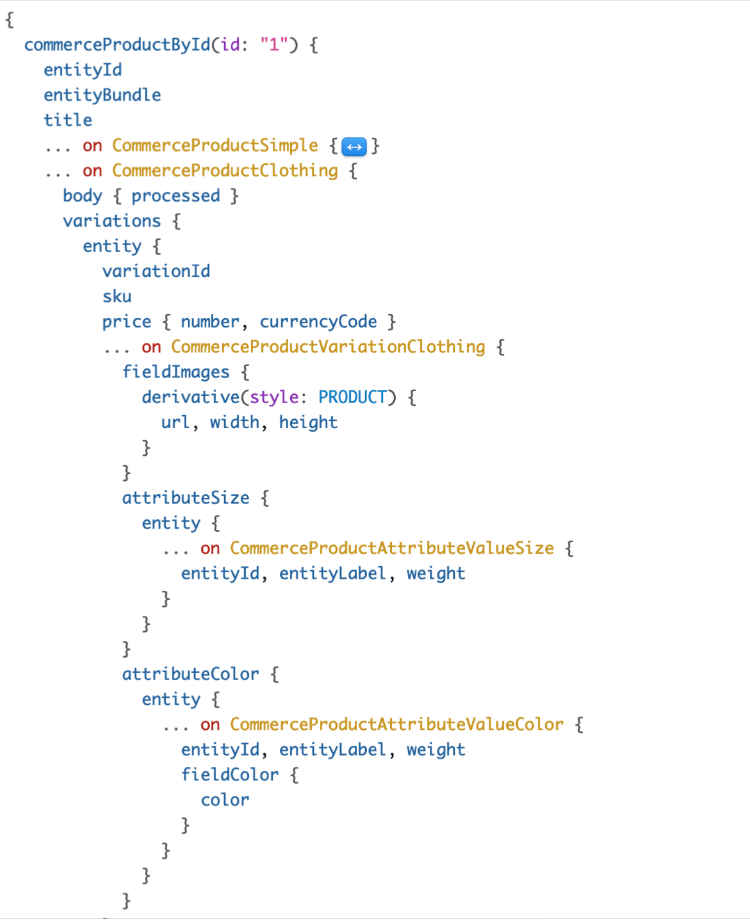
include	variations,variations.field_images,variations.a
fields[commerce_product--clothing]	title,body
fields[commerce_product_variation--clothing	sku,title,price
fields[commerce_product_attribute_value--cc	name
fields[commerce_product_attribute_value--si:	name
fields[taxonomy_term--brands]	name
fields[taxonomy_term--product_categories]	name
fields[field_special_categories]	name
fields[file--file]	uri,url
name	value
New name	New value

```
{
  "data": {
    "type": "commerce_product--clothing",
    "id": "802dfe56-8536-4287-a0fa-6bed3fd3a621",
    "links": {↔},
    "attributes": {
      "title": "Drupal Commerce Hoodie",
      "body": {
        "value": "<p>Vivamus ante quam, tristique quis, porttitor nec, viverra eget, ante. Nulla vitae
nunc vel est dignissim dictum. Suspendisse feugiat pellentesque ante. Etiam viverra commodo neque. Aenean
id mi. Sed lacinia tincidunt purus. Maecenas eros quam, feugiat quis, feugiat vel, rutrum in, sem.
Pellentesque eu ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
hymenaeos. Vestibulum mollis molestie elit.<\/p>\r\n",
        "format": "basic_html",
        "processed": "<p>Vivamus ante quam, tristique quis, porttitor nec, viverra eget, ante. Nulla vitae
nunc vel est dignissim dictum. Suspendisse feugiat pellentesque ante. Etiam viverra commodo neque. Aenean
id mi. Sed lacinia tincidunt purus. Maecenas eros quam, feugiat quis, feugiat vel, rutrum in, sem.
Pellentesque eu ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
hymenaeos. Vestibulum mollis molestie elit.<\/p>",
        "summary": ""
      }
    }
  },
  "jsonapi": {↔},
  "links": {↔},
  "included": [
    {
      "type": "commerce_product_variation--clothing",
      "id": "2a242d40-c483-4deb-bab5-1eab4e715a28",
      "links": {↔},
      "attributes": {
        "sku": "DCHOODIE-BLU-SM",
        "title": "Drupal Commerce Hoodie - Blue, Small",
        "price": {
          "number": "52.000000",
          "currency_code": "USD",
          "formatted": "$52.00"
        }
      }
    },
    {
      "type": "commerce_product_variation--clothing",
      "id": "c578aa37-8f30-4b5c-9af5-af437f2099fe",
      "links": {↔},
      "attributes": {
        "sku": "DCHOODIE-BLU-MD",
        "title": "Drupal Commerce Hoodie - Blue, Medium",
        "price": {
```

Sample JSON API request for a product and its variations

GraphQL

- Supports including referenced entities
- Allows specifying specific fields to be returned
- Dynamically handle values based on bundle schema that is available (Fragments)

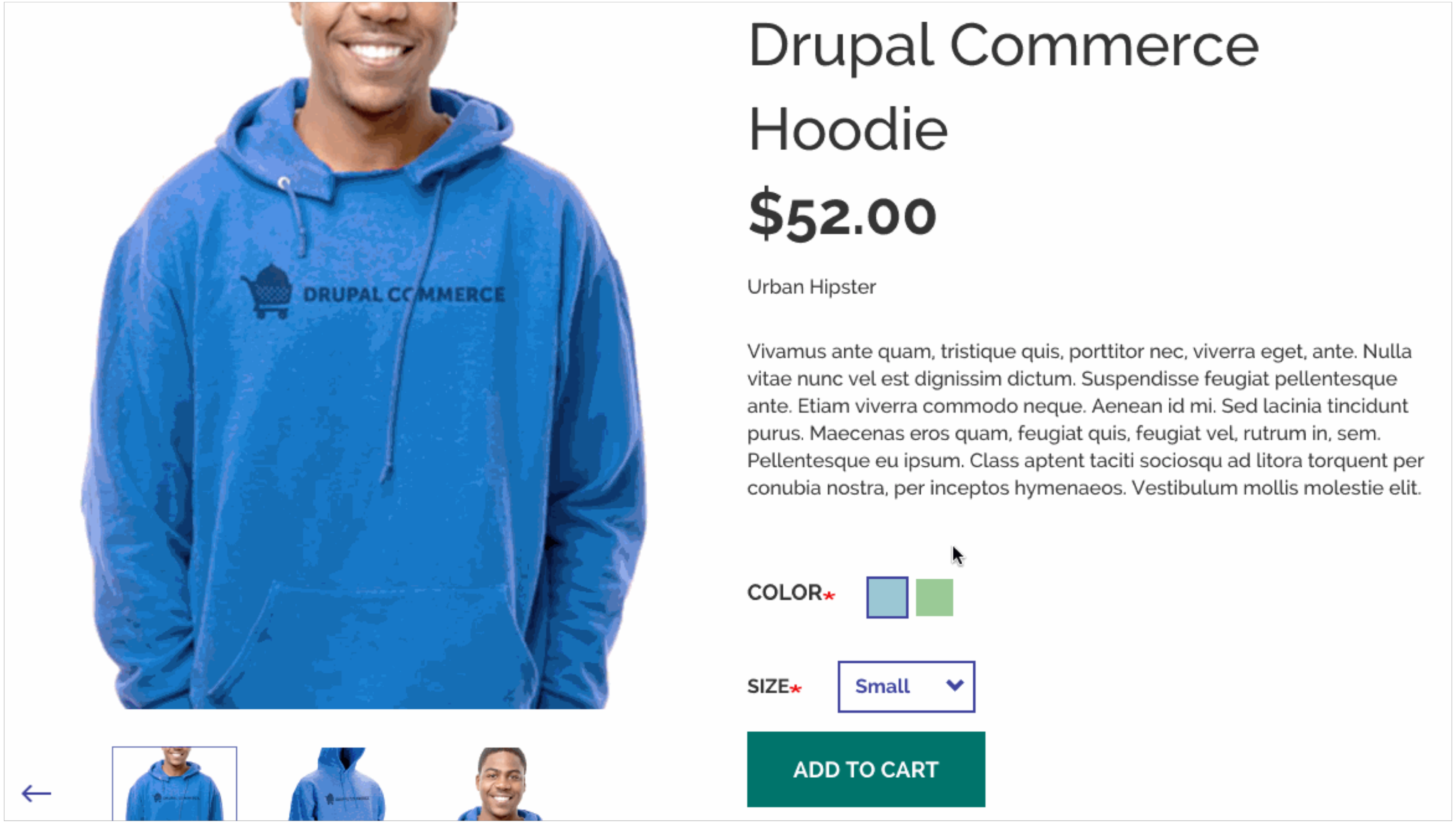


```
{
  commerceProductById(id: "1") {
    entityId
    entityBundle
    title
    ... on CommerceProductSimple {↔}
    ... on CommerceProductClothing {
      body { processed }
      variations {
        entity {
          variationId
          sku
          price { number, currencyCode }
          ... on CommerceProductVariationClothing {
            fieldImages {
              derivative(style: PRODUCT) {
                url, width, height
              }
            }
            attributeSize {
              entity {
                ... on CommerceProductAttributeValueSize {
                  entityId, entityLabel, weight
                }
              }
            }
            attributeColor {
              entity {
                ... on CommerceProductAttributeValueColor {
                  entityId, entityLabel, weight
                  fieldColor {
                    color
                  }
                }
              }
            }
```

Add to Cart Forms

Add to cart form areas of concern

- Attribute select lists vs. fancy formatting
- Identifying the currently selected product variation
- Custom order item fields
- Challenging to provide abstract or reusable solutions



Drupal Commerce Hoodie

$52.00

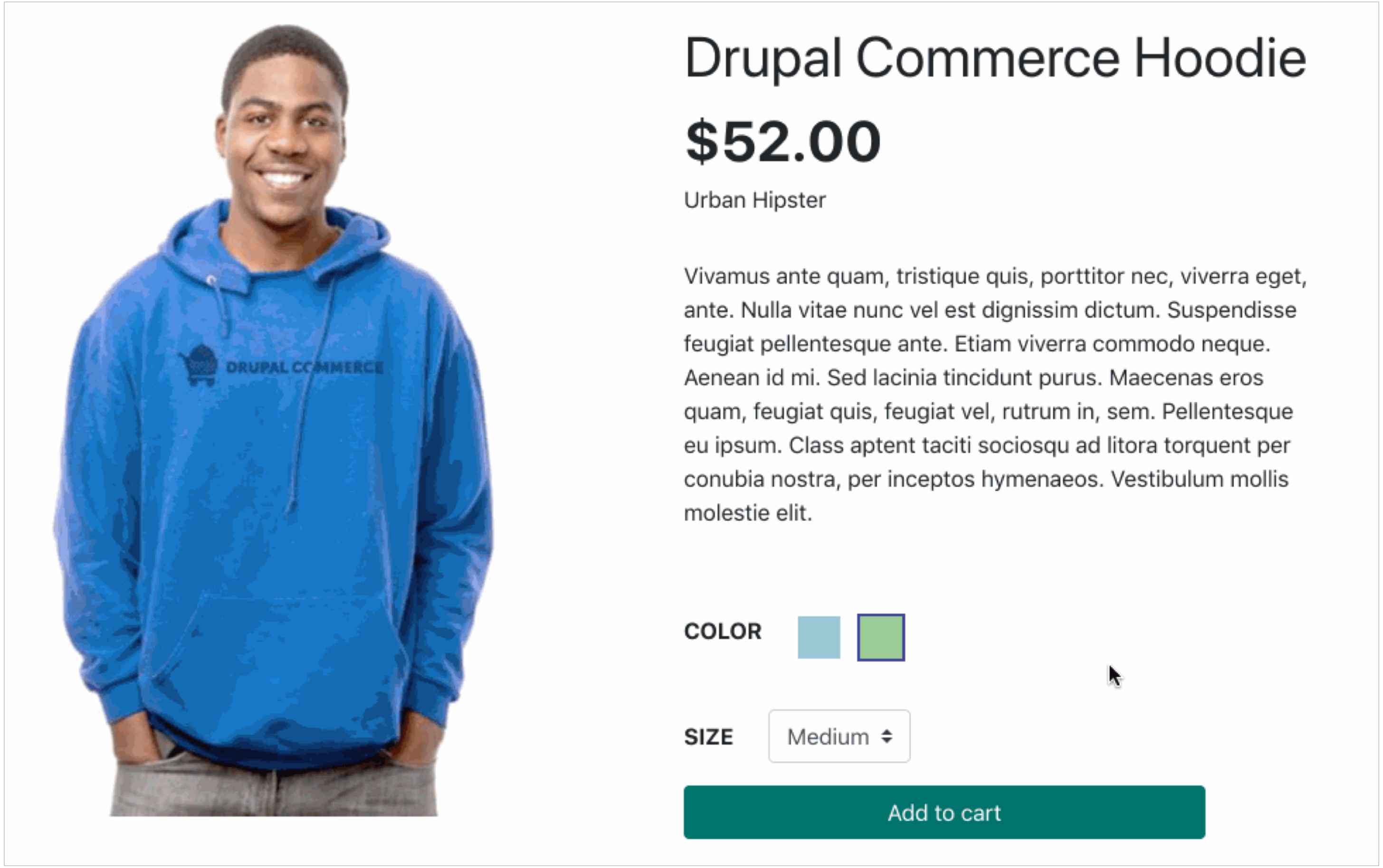
Urban Hipster

Vivamus ante quam, tristique quis, porttitor nec, viverra eget, ante. Nulla vitae nunc vel est dignissim dictum. Suspendisse feugiat pellentesque ante. Etiam viverra commodo neque. Aenean id mi. Sed lacinia tincidunt purus. Maecenas eros quam, feugiat quis, feugiat vel, rutrum in, sem. Pellentesque eu ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vestibulum mollis molestie elit.

COLOR* 🟦 🟩

SIZE* Small

jQuery + Backbone + Underscore



Drupal Commerce Hoodie

$52.00

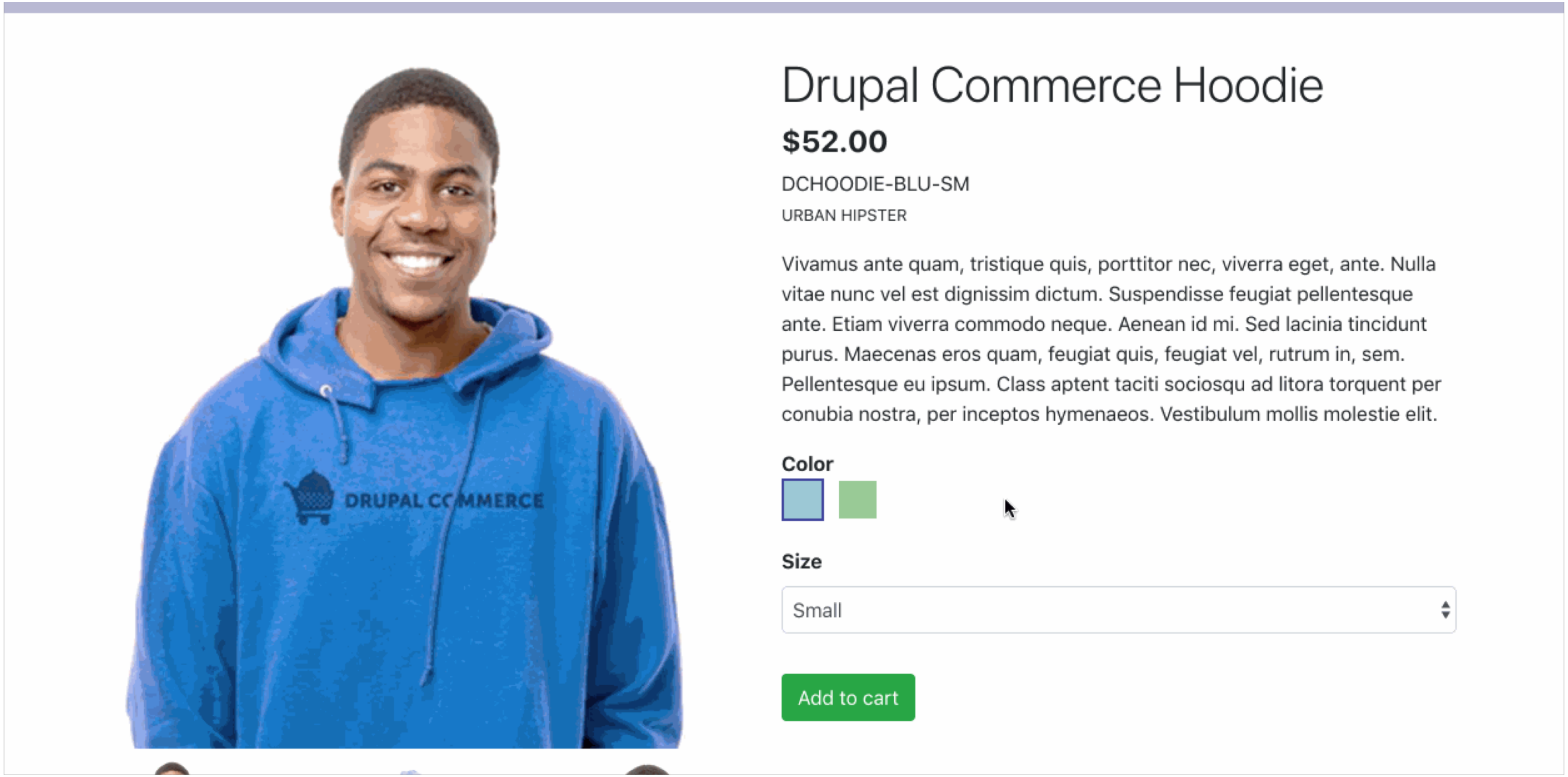
Urban Hipster

Vivamus ante quam, tristique quis, porttitor nec, viverra eget, ante. Nulla vitae nunc vel est dignissim dictum. Suspendisse feugiat pellentesque ante. Etiam viverra commodo neque. Aenean id mi. Sed lacinia tincidunt purus. Maecenas eros quam, feugiat quis, feugiat vel, rutrum in, sem. Pellentesque eu ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vestibulum mollis molestie elit.

COLOR 🟦 🟩

SIZE Medium

React + JSON API



Drupal Commerce Hoodie

$52.00

DCHOODIE-BLU-SM

URBAN HIPSTER

Vivamus ante quam, tristique quis, porttitor nec, viverra eget, ante. Nulla vitae nunc vel est dignissim dictum. Suspendisse feugiat pellentesque ante. Etiam viverra commodo neque. Aenean id mi. Sed lacinia tincidunt purus. Maecenas eros quam, feugiat quis, feugiat vel, rutrum in, sem. Pellentesque eu ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vestibulum mollis molestie elit.

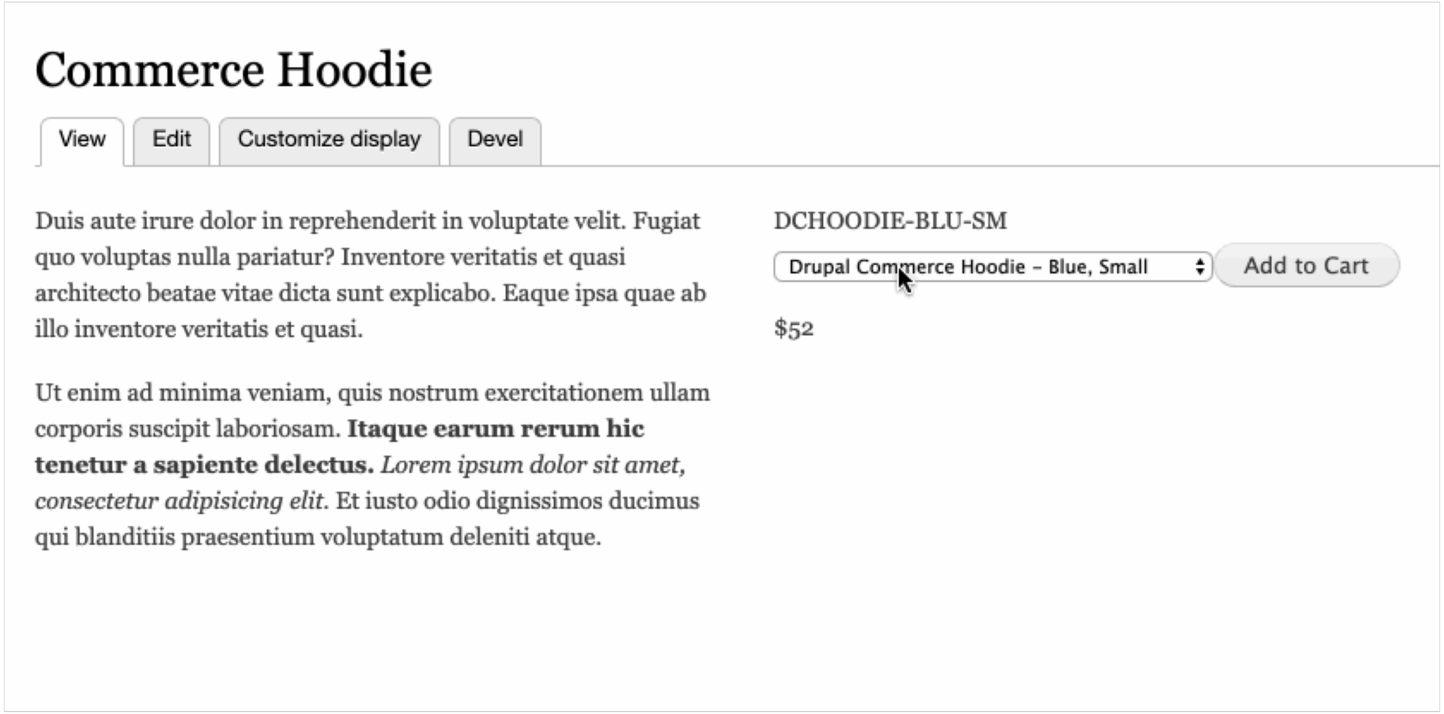
Color

🟦 🟩

Size

Small

GatsbyJS: React + GraphQL



Commerce Hoodie

Duis aute irure dolor in reprehenderit in voluptate velit. Fugiat quo voluptas nulla pariatur? Inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Eaque ipsa quae ab illo inventore veritatis et quasi.

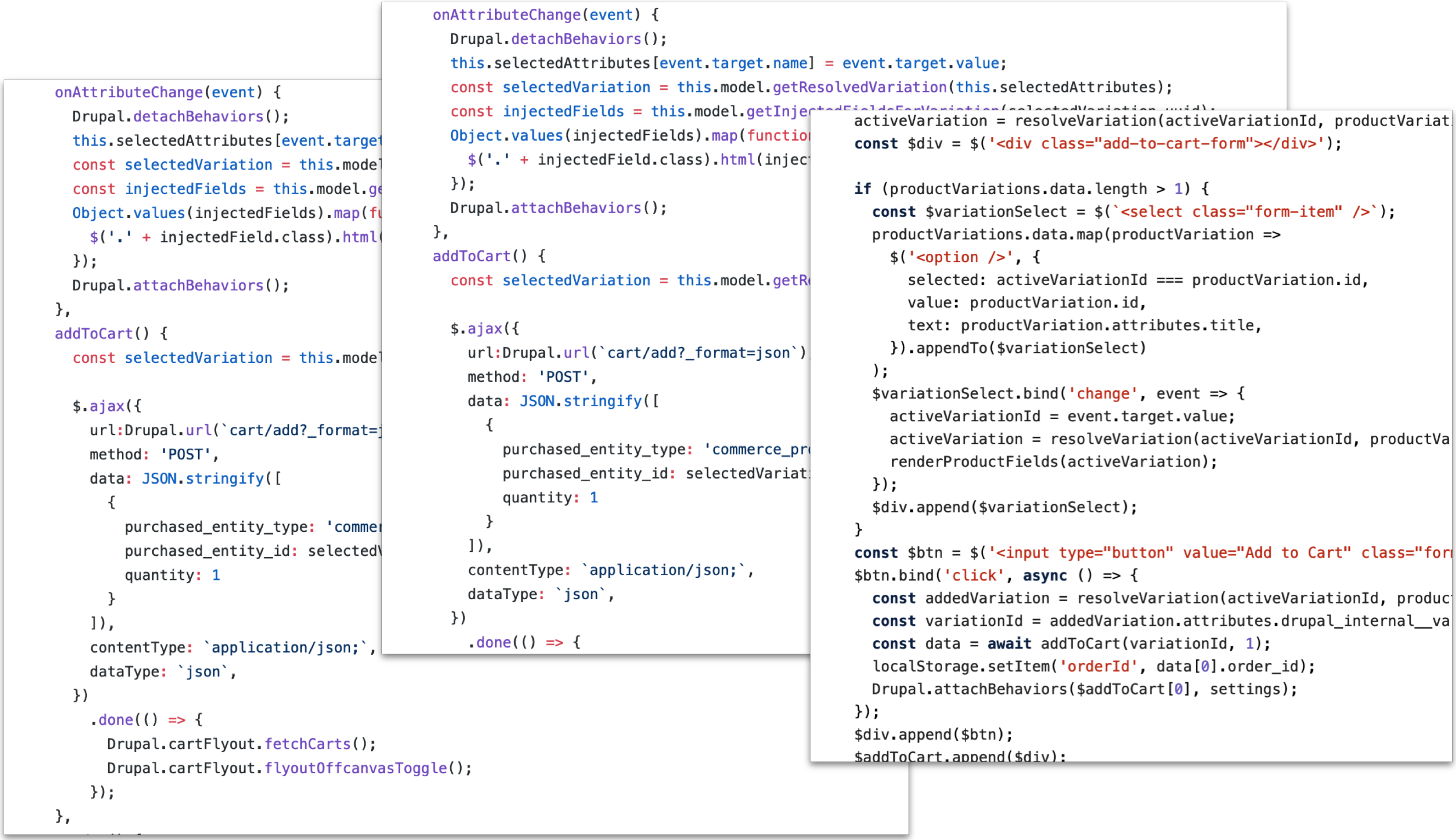
Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam. **Itaque earum rerum hic tenetur a sapiente delectus.** *Lorem ipsum dolor sit amet, consectetur adipisicing elit.* Et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque.

DCHOODIE-BLU-SM

$52

jQuery

The challenging part as framework maintainers is to provide abstract or reusable solutions for very purpose built frontends.



```
  onAttributeChange(event) {
    Drupal.detachBehaviors();
    this.selectedAttributes[event.target
    const selectedVariation = this.mode
    const injectedFields = this.model.ge
    Object.values(injectedFields).map(fu
      $('.' + injectedField.class).html(
    });
    Drupal.attachBehaviors();
  },
  addToCart() {
    const selectedVariation = this.mode

    $.ajax({
      url:Drupal.url(`cart/add?_format=j
      method: 'POST',
      data: JSON.stringify([
        {
          purchased_entity_type: 'comme
          purchased_entity_id: selectedV
          quantity: 1
        }
      ]),
      contentType: `application/json;`,
      dataType: `json`,
    })
      .done(() => {
        Drupal.cartFlyout.fetchCarts();
        Drupal.cartFlyout.flyoutOffcanvasToggle();
      });
  },
```

```
  onAttributeChange(event) {
    Drupal.detachBehaviors();
    this.selectedAttributes[event.target.name] = event.target.value;
    const selectedVariation = this.model.getResolvedVariation(this.selectedAttributes);
    const injectedFields = this.model.getInje
    Object.values(injectedFields).map(functio
      $('.' + injectedField.class).html(injec
    });
    Drupal.attachBehaviors();
  },
  addToCart() {
    const selectedVariation = this.model.getR

    $.ajax({
      url:Drupal.url(`cart/add?_format=json`)
      method: 'POST',
      data: JSON.stringify([
        {
          purchased_entity_type: 'commerce_pr
          purchased_entity_id: selectedVariat
          quantity: 1
        }
      ]),
      contentType: `application/json;`,
      dataType: `json`,
    })
    .done(() => {
```

```
activeVariation = resolveVariation(activeVariationId, productVariat
const $div = $('<div class="add-to-cart-form"></div>');

if (productVariations.data.length > 1) {
  const $variationSelect = $(`<select class="form-item" />`);
  productVariations.data.map(productVariation =>
    $('<option />', {
      selected: activeVariationId === productVariation.id,
      value: productVariation.id,
      text: productVariation.attributes.title,
    }).appendTo($variationSelect)
  );
  $variationSelect.bind('change', event => {
    activeVariationId = event.target.value;
    activeVariation = resolveVariation(activeVariationId, productVa
    renderProductFields(activeVariation);
  });
  $div.append($variationSelect);
}
const $btn = $('<input type="button" value="Add to Cart" class="for
$btn.bind('click', async () => {
  const addedVariation = resolveVariation(activeVariationId, produc
  const variationId = addedVariation.attributes.drupal_internal__va
  const data = await addToCart(variationId, 1);
  localStorage.setItem('orderId', data[0].order_id);
  Drupal.attachBehaviors($addToCart[0], settings);
});
$div.append($btn);
$addToCart.append($div);
```

Carts

Cart management areas of concern

- Belong to anonymous users and authenticated users
- Users can modify their carts, without fully managing the order
- Constraints on product availability
- :) Users shouldn’t be able to edit another user’s cart

Cart API: Cart Operations

- Cart API hides internals knowledge and provides a friendly interface.
- GET for collection of active carts or individual cart
- POST that creates the order item from a product and adds it to the proper cart.
- PATCH to bulk update quantities

```
POST /cart/add?_format=json
[
  { "purchased_entity_type": "commerce_product_variation",
    "purchased_entity_id": "6",
    "quantity": "1"
   }
]

PATCH /cart/{order}/items?_format=json
{
  3: {"quantity": 1},
  9: {"quantity": 2},
}
```

Cart API: Cart Tokens

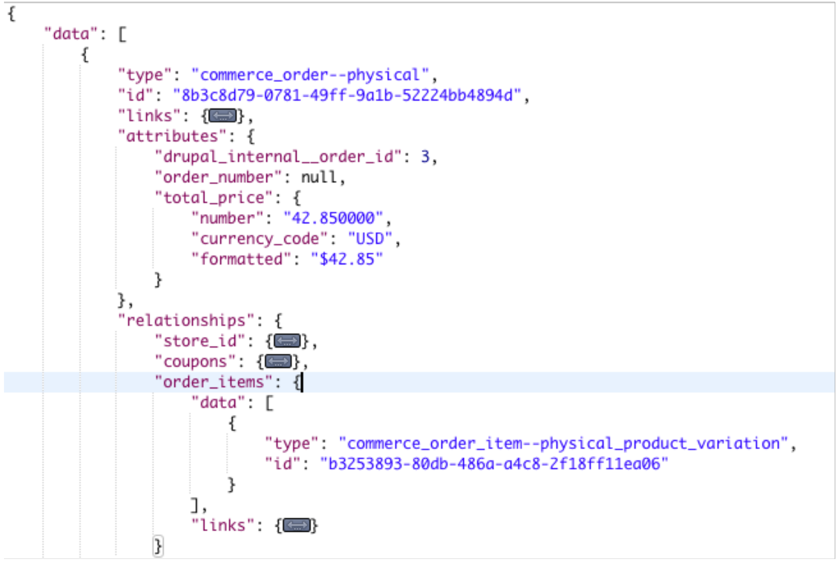
- PHP Sessions persist via cookies
- Cookies are opt-in when making CORS enabled AJAX requests
- Cookies are blocked on cross-origin CORS enabled AJAX requests
- Solution: a passable token as a query parameter or header.

```
GET /cart?_format=json
Accept: application/json
Commerce-Cart-Token: 123ABC

GET /cart?_format=json&cartToken=123ABC
Accept: application/json
```

JSON API

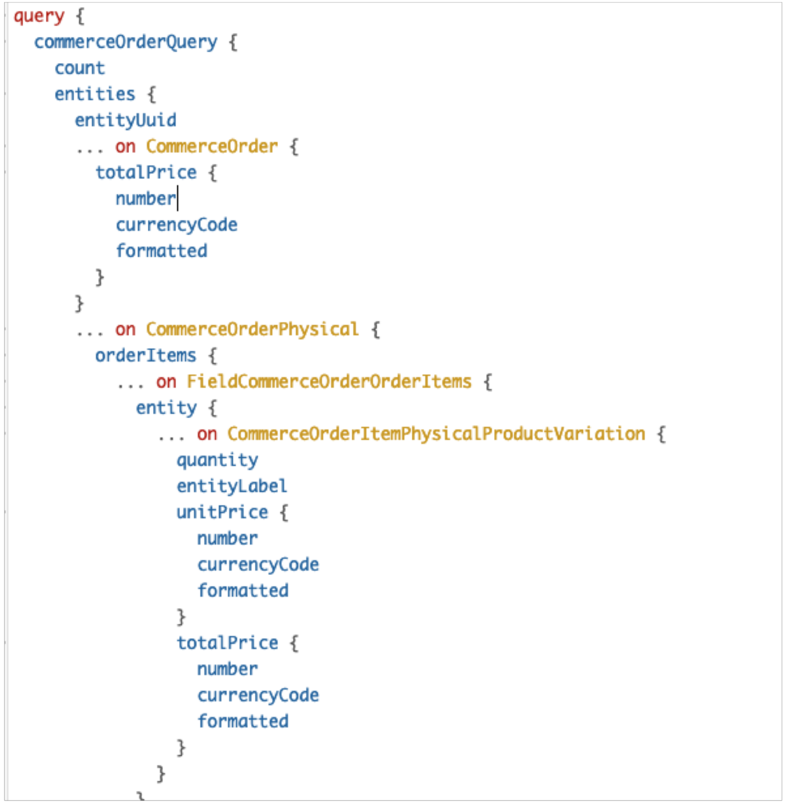
- Not out of the box, still requires Cart API
- Custom “Cart Collection” endpoint
- Adding a product to the cart requires knowledge of relationship internals, without a custom endpoint
- JSON API is now stable and in Drupal core, easier to integrate and extend!



```
{
    "data": [
        {
            "type": "commerce_order--physical",
            "id": "8b3c8d79-0781-49ff-9a1b-52224bb4894d",
            "links": {↔},
            "attributes": {
                "drupal_internal__order_id": 3,
                "order_number": null,
                "total_price": {
                    "number": "42.850000",
                    "currency_code": "USD",
                    "formatted": "$42.85"
                }
            },
            "relationships": {
                "store_id": {↔},
                "coupons": {↔},
                "order_items": {
                    "data": [
                        {
                            "type": "commerce_order_item--physical_product_variation",
                            "id": "b3253893-80db-486a-a4c8-2f18ff11ea06"
                        }
                    ],
                    "links": {↔}
                }
```

GraphQL

- Carts are restricted by query access, so only the requesting user's carts are returned
- Mutations are part of GraphQL Mutations module
- Requires writing GraphQL Mutation plugins

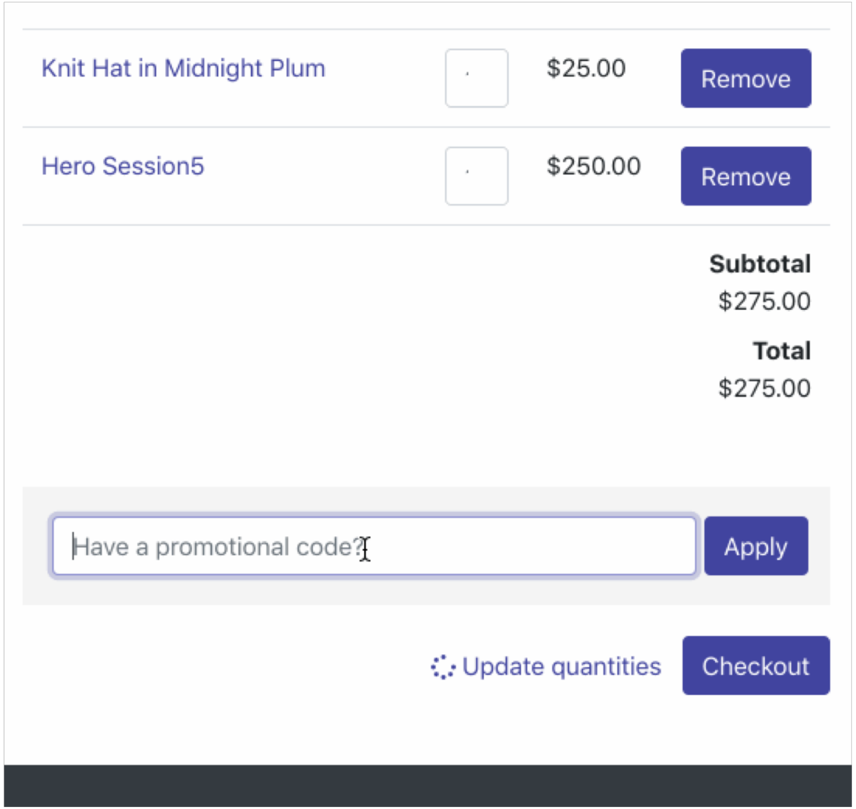


```
query {
  commerceOrderQuery {
    count
    entities {
      entityUuid
      ... on CommerceOrder {
        totalPrice {
          number
          currencyCode
          formatted
        }
      }
      ... on CommerceOrderPhysical {
        orderItems {
          ... on FieldCommerceOrderOrderItems {
            entity {
              ... on CommerceOrderItemPhysicalProductVariation {
                quantity
                entityLabel
                unitPrice {
                  number
                  currencyCode
                  formatted
                }
                totalPrice {
                  number
                  currencyCode
                  formatted
                }
              }
            }
```

Checkout & More

Coupons

- Coupon reference constraint to validate coupons when attached, better API level handling
- Coupon redemption endpoint via coupon code
- Limited to single coupon to avoid abuse



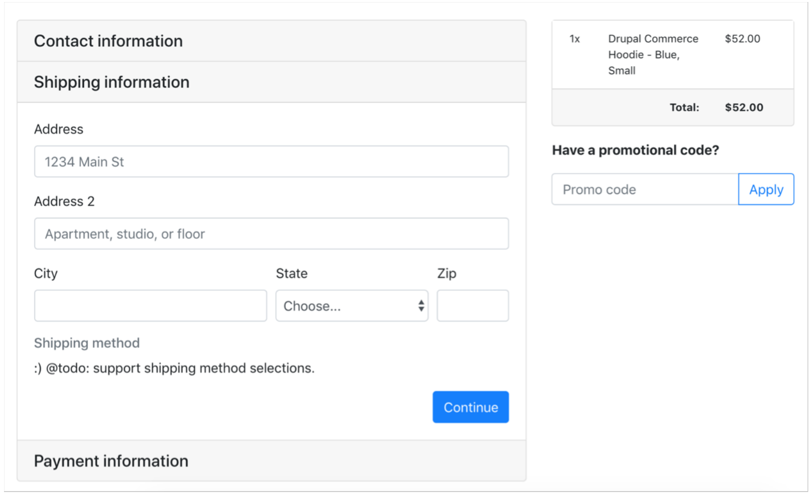
Knit Hat in Midnight Plum	$25.00
Hero Session5	$250.00

Subtotal
$275.00

Total
$275.00

Checkout over the API

- Experimented with a bespoke Checkout API for a GatsbyJS demo last year. Lessons learned and problems identified.
- Completing checkout is really just running the *place order* transition
- Problem is most of our validations are in the Checkout Form, not in data level constraints



Contact information

Shipping information

Address

1234 Main St

Address 2

Apartment, studio, or floor

City

State

Choose...

Zip

Shipping method

:) @todo: support shipping method selections.

Payment information

1x	Drupal Commerce Hoodie - Blue, Small	$52.00
	Total:	**$52.00**

Have a promotional code?

Promo code

To Headless,
And Beyond!

What's next? (in progress)

- `FormattedPriceItem` into Drupal Commerce and not just Cart API
  https://www.drupal.org/project/commerce/issues/3041802

- Improving query access and entity access
  https://www.drupal.org/project/commerce/issues/3041758

- Coupon validation constraint in Drupal Commerce
  https://www.drupal.org/project/commerce/issues/3041856

What's next? (in development)

- A computed base field on the order entity which returns order total summary
- Calculated pricing and product availability
- Coupon redemption (via coupon code)

Contribution Day

Saturday, March 23 – 10 am to 4 pm

Just bring a laptop.

Coffee will be available.

https://www.midcamp.org/2019/contribution-day

Questions?

matt@commmerceguys.com

Give it a spin!

https://react.demo.centarro.io/

https://gatsbyjs.demo.centarro.io/

https://demo.commercekickstart.com/