This is a public copy of “`y/why-graphql`”, an internal pitch deck created to promote the use of GraphQL at Yelp for our web and mobile clients.

(References to “`y/...`” are internal shortlinks)

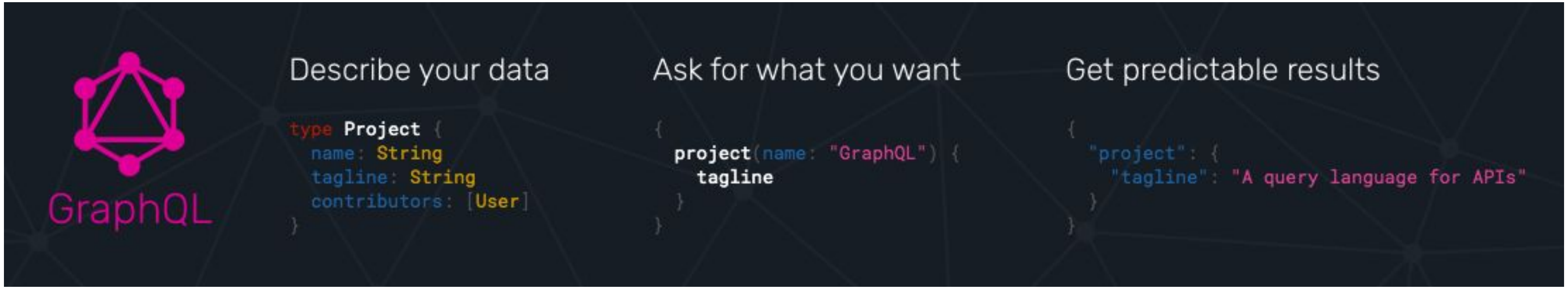
Published: April nth 2024

Why you should use GraphQL!

y/why-graphql

What is it?

GraphQL is a query language that makes it super straightforward to fetch data in our React, iOS and Android apps **declaratively.**



```
type Project {
  name: String
  tagline: String
  contributors: [User]
}
```

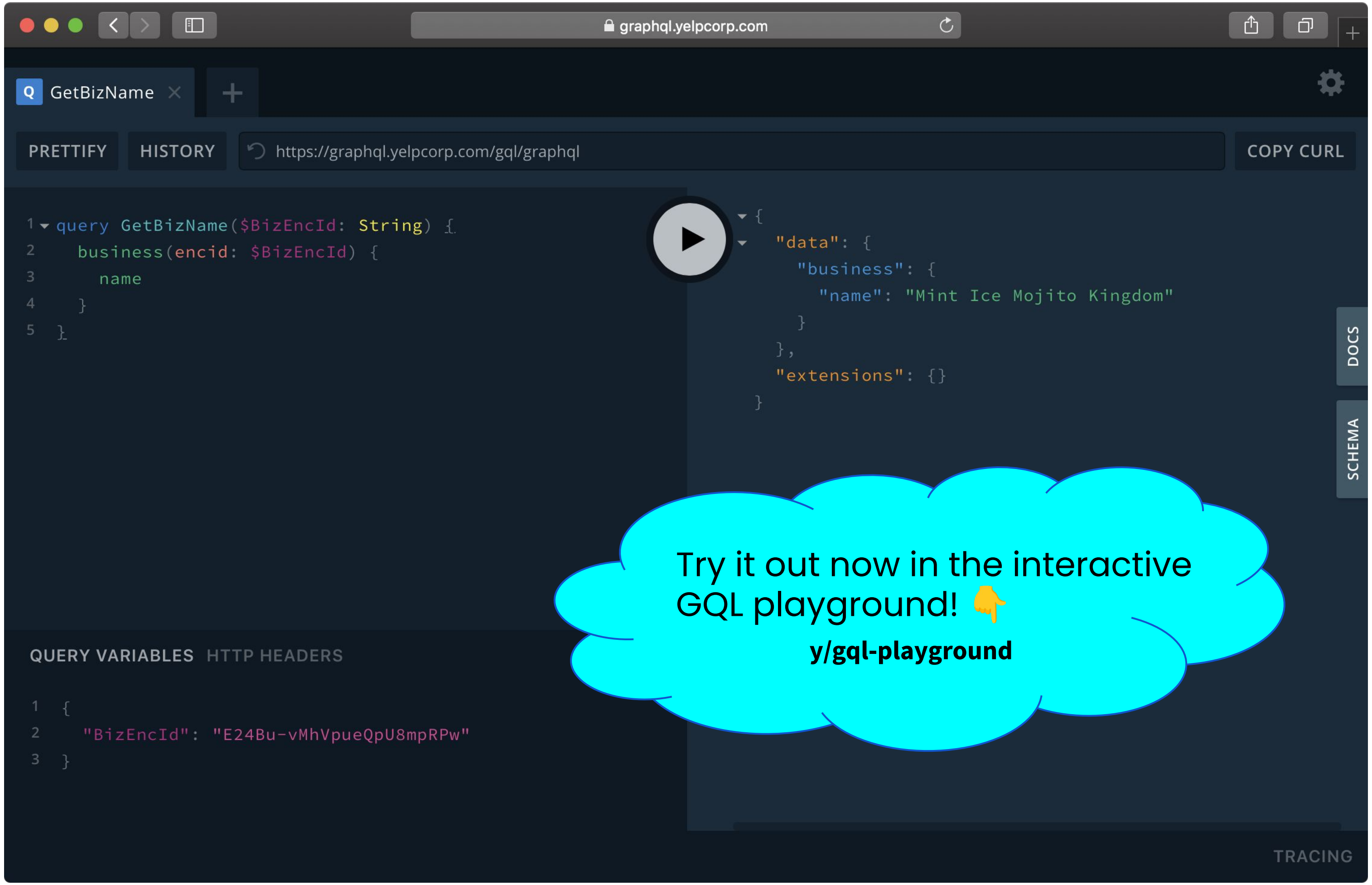
```
{
  project(name: "GraphQL") {
    tagline
  }
}
```

```
{
  "project": {
    "tagline": "A query language for APIs"
  }
}
```

https://graphql.org/

Used widely in industry by: shopify NBC PayPal lyft credit karma airbnb Microsoft IBM …

https://graphql.org/users/



https://graphql.yelpcorp.com/gql/graphql

```
query GetBizName($BizEncId: String) {
  business(encid: $BizEncId) {
    name
  }
}
```

```
{
  "data": {
    "business": {
      "name": "Mint Ice Mojito Kingdom"
    }
  },
  "extensions": {}
}
```

```
{
  "BizEncId": "E24Bu-vMhVpueQpU8mpRPw"
}
```

Try it out now in the interactive GQL playground! 👇

y/gql-playground

The Playground

“Classic” REST Service (@ Yelp) Problems:

Fetching data from multiple sources is hard.

1. Developers have to **write and maintain significant amounts of data-fetching code** for every feature.
   - ➢ If you need data from n services, you must write (at minimum) n network requests
   - ➢ Duplicated 2x: for the Mobile API (for iOS/Android), and in BFFs (for Python templates for web)
   - ➢ Discovering existing endpoints for a type is hard, leading to duplicated overlapping public endpoints
2. Return objects from REST endpoints are **untyped**, meaning runtime code is less safe.
3. Slower iteration time due to the complexities of fetching data mean **features take longer to ship**

See **y/cep980** for a full breakdown.

How Does GraphQL Help?

1. Developer Velocity (Improving Time to Market)

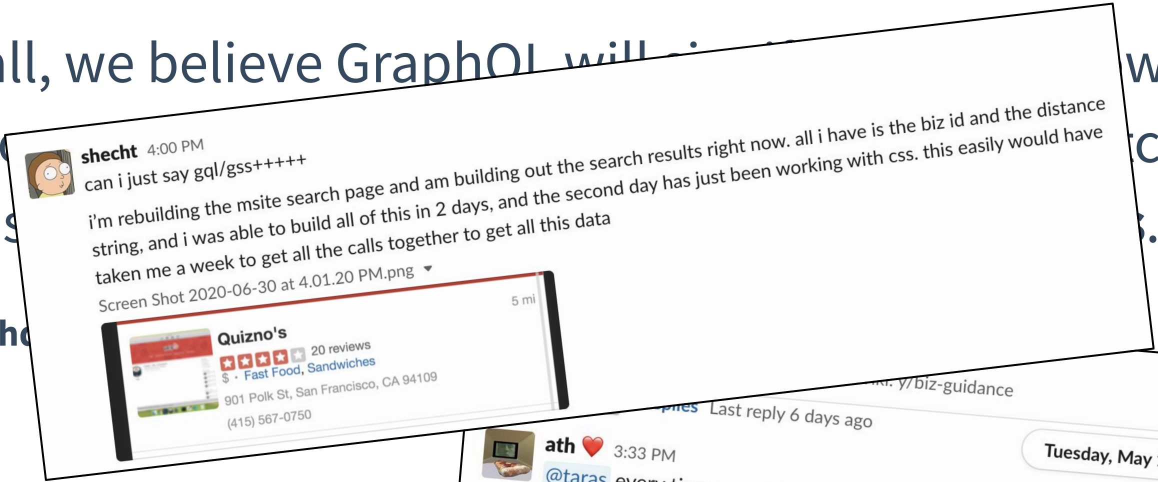
Overall, we believe GraphQL will significantly cut down on the time developers need to spend writing code related to fetching data. This means **faster iteration cycles and product releases**.

~ y/graphql-docs, 2019

1. Developer Velocity (Improving Time to Market)

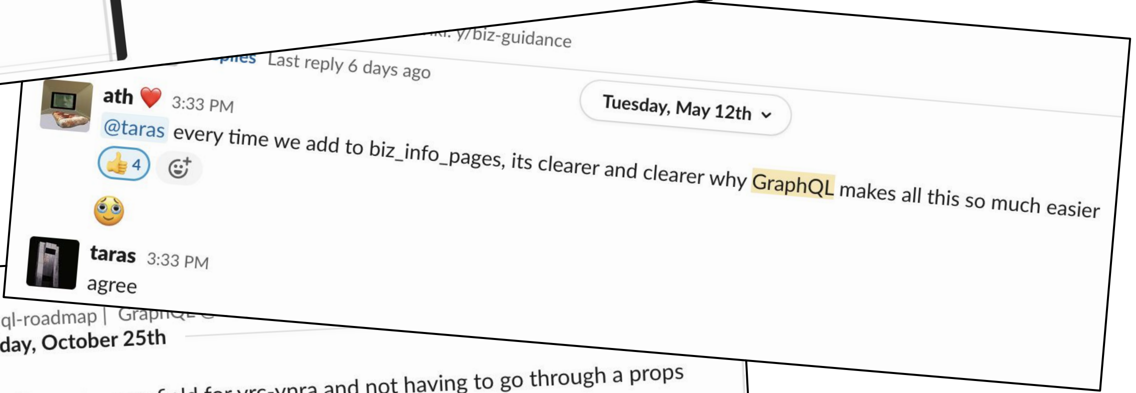
Overall, we believe GraphQL will ... wn on the time devel ... ching data. This means ... s.

~ y/graph



shecht 4:00 PM
can i just say gql/gss+++++
i'm rebuilding the msite search page and am building out the search results right now. all i have is the biz id and the distance string, and i was able to build all of this in 2 days, and the second day has just been working with css. this easily would have taken me a week to get all the calls together to get all this data
Screen Shot 2020-06-30 at 4.01.20 PM.png

Quizno's
5 mi
20 reviews
$ · Fast Food, Sandwiches
901 Polk St, San Francisco, CA 94109
(415) 567-0750

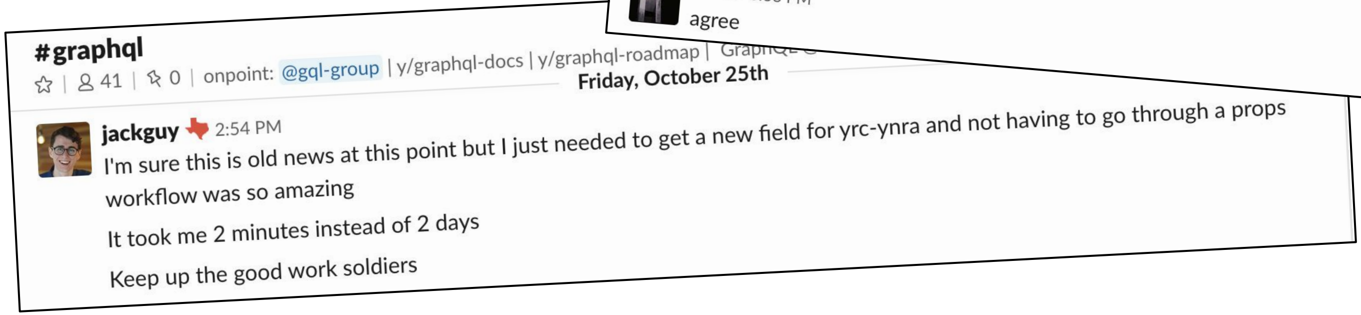


ki: y/biz-guidance
plies Last reply 6 days ago

Tuesday, May 12th

ath ❤️ 3:33 PM
@taras every time we add to biz_info_pages, its clearer and clearer why GraphQL makes all this so much easier
👍 4

taras 3:33 PM
agree

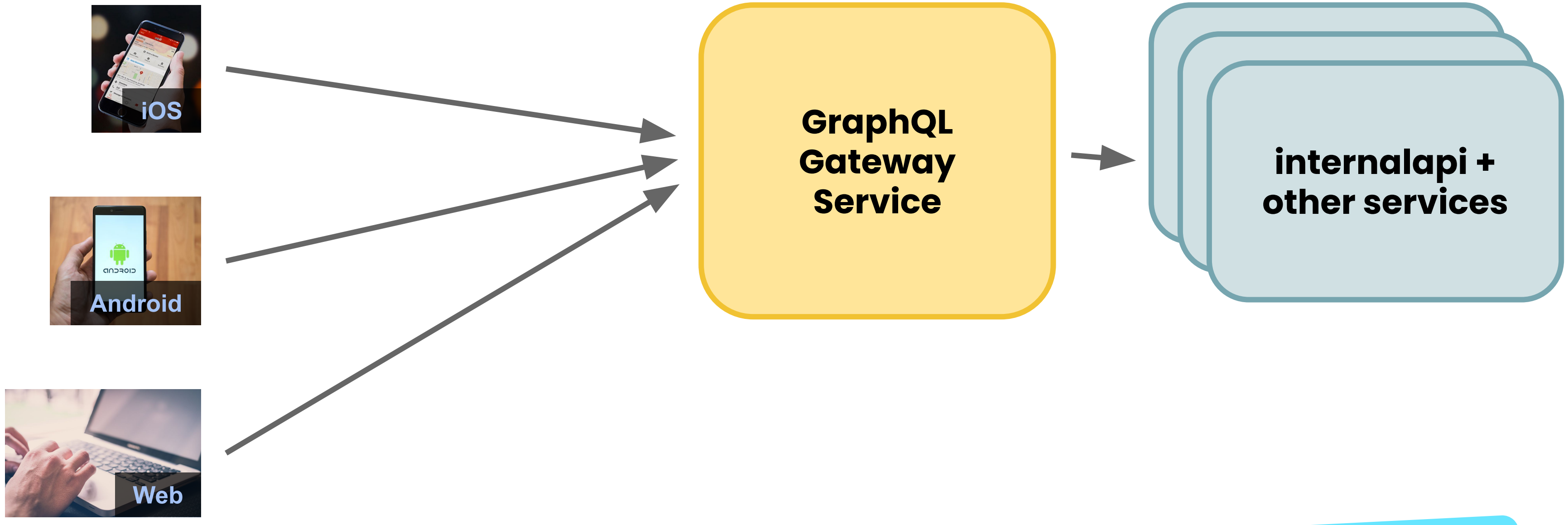


#graphql
☆ | 41 | 0 | onpoint: @gql-group | y/graphql-docs | y/graphql-roadmap | GraphQL

Friday, October 25th

jackguy 2:54 PM
I'm sure this is old news at this point but I just needed to get a new field for yrc-ynra and not having to go through a props workflow was so amazing
It took me 2 minutes instead of 2 days
Keep up the good work soldiers

2. Unified API Server for Consumer, Biz, Web, iOS, Android, BizApp

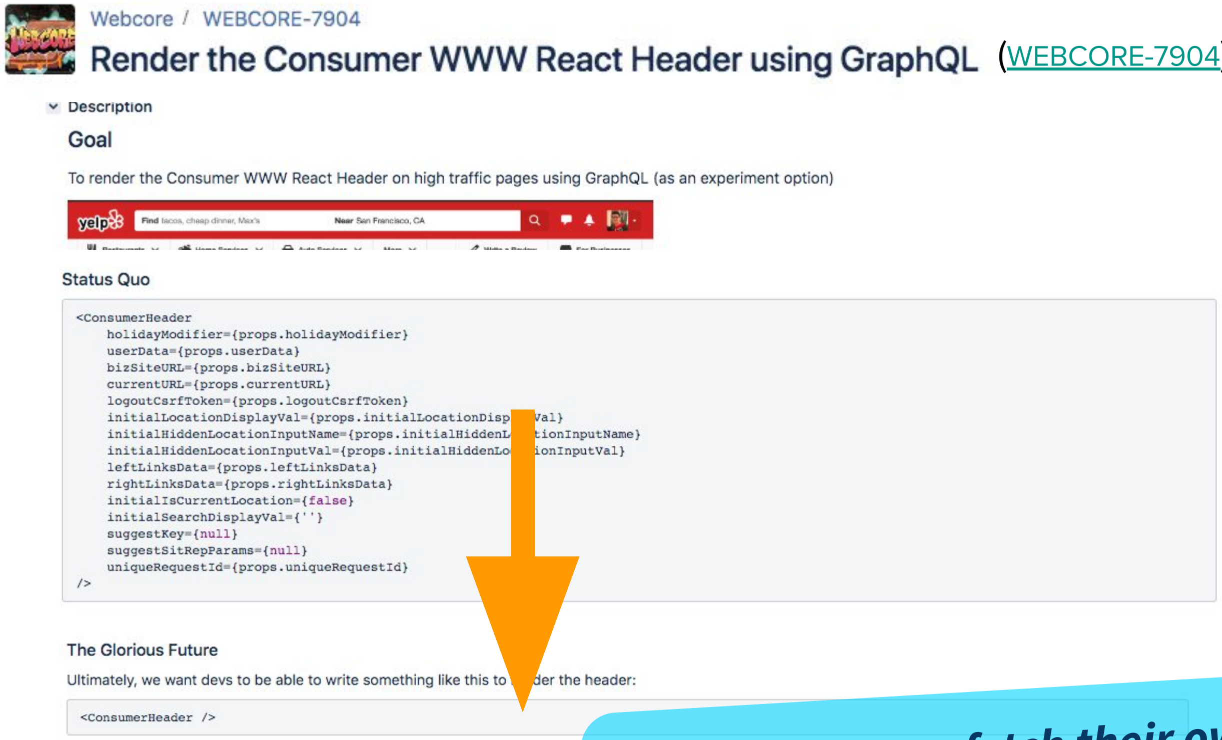


Shared reuse of backend logic for all our apps!

GGS can be called from anywhere - the web, iOS, Android and BizApp. Your team can write **shared backend logic** once and reuse it on all our client platforms!

3. Simplified Component API + Usage (Developer Velocity)

Example:



Webcore / WEBCORE-7904

Render the Consumer WWW React Header using GraphQL

(WEBCORE-7904)

Description

Goal

To render the Consumer WWW React Header on high traffic pages using GraphQL (as an experiment option)

Status Quo

```
<ConsumerHeader
    holidayModifier={props.holidayModifier}
    userData={props.userData}
    bizSiteURL={props.bizSiteURL}
    currentURL={props.currentURL}
    logoutCsrfToken={props.logoutCsrfToken}
    initialLocationDisp    Val}
    initialHiddenLocationInputName={props.initialHiddenL    tionInputName}
    initialHiddenLocationInputVal={props.initialHiddenLo    ionInputVal}
    leftLinksData={props.leftLinksData}
    rightLinksData={props.rightLinksData}
    initialIsCurrentLocation={false}
    initialSearchDisplayVal={''}
    suggestKey={null}
    suggestSitRepParams={null}
    uniqueRequestId={props.uniqueRequestId}
/>
```

The Glorious Future

Ultimately, we want devs to be able to write something like this to   der the header:

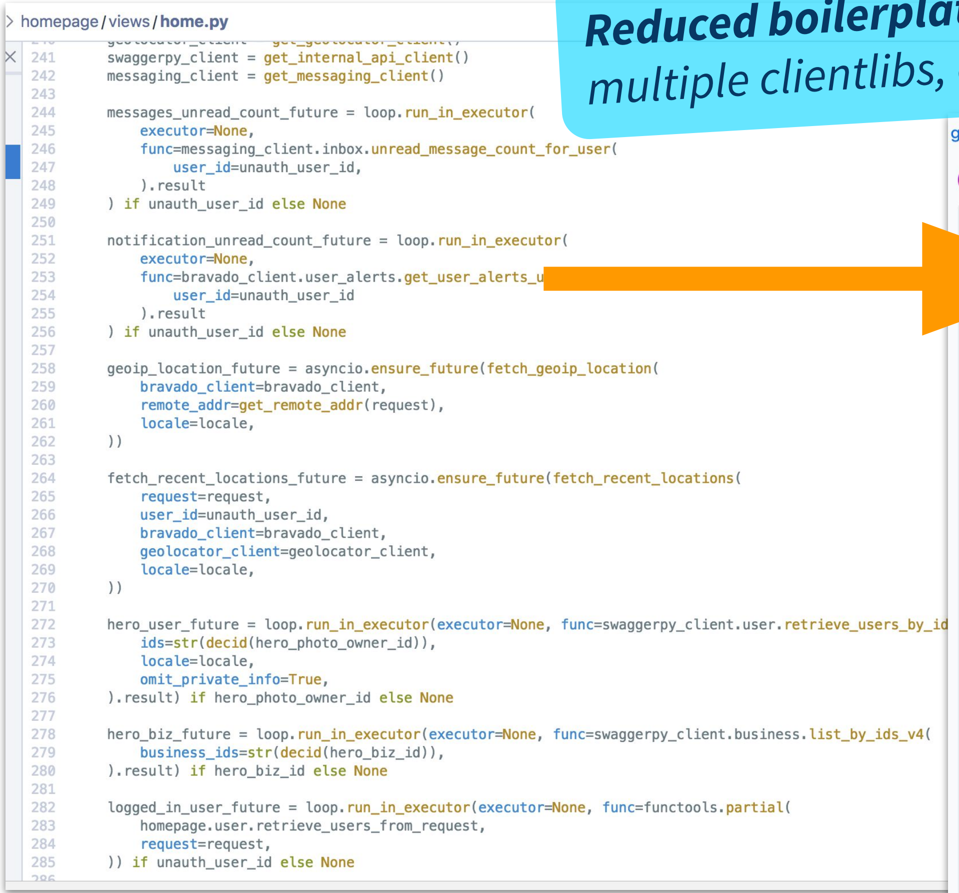
```
<ConsumerHeader />
```

No data props required!

Components can fetch their own data props! Wow! *No need to write props fetching code!*

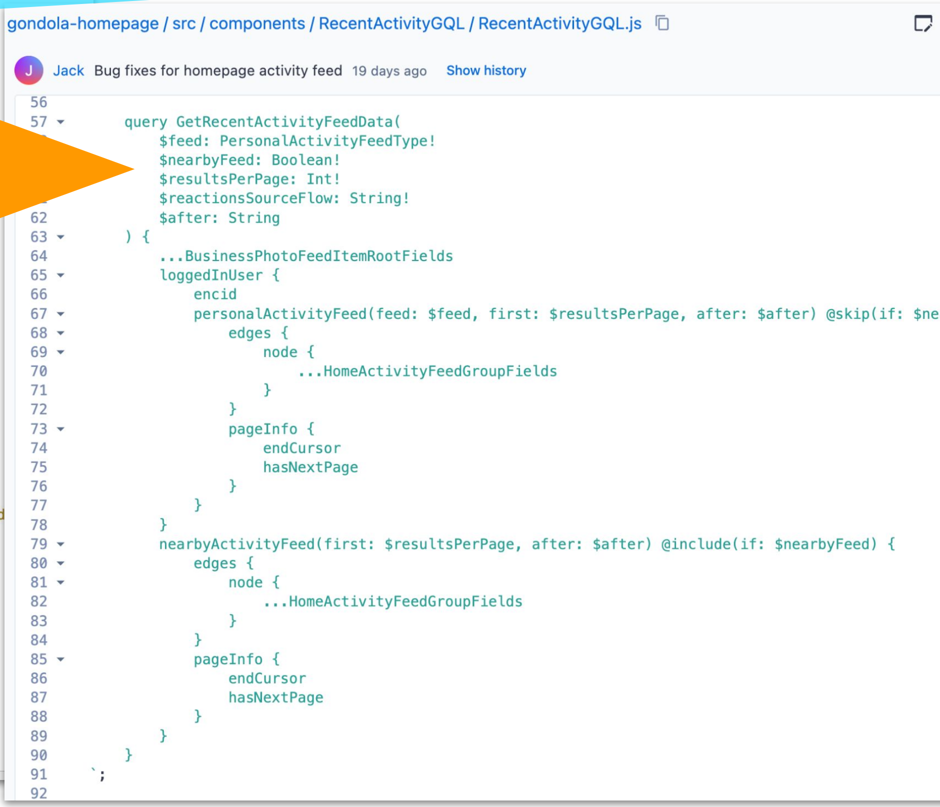
4. Simplified Data Fetching (Developer Experience + Happiness)

Reduced boilerplate *(e.g. setting up multiple clientlibs, asyncio loops)*



homepage / views / **home.py**

```python
    geolocator_client = get_geolocator_client()
    swaggerpy_client = get_internal_api_client()
    messaging_client = get_messaging_client()

    messages_unread_count_future = loop.run_in_executor(
        executor=None,
        func=messaging_client.inbox.unread_message_count_for_user(
            user_id=unauth_user_id,
        ).result
    ) if unauth_user_id else None

    notification_unread_count_future = loop.run_in_executor(
        executor=None,
        func=bravado_client.user_alerts.get_user_alerts_u
            user_id=unauth_user_id
        ).result
    ) if unauth_user_id else None

    geoip_location_future = asyncio.ensure_future(fetch_geoip_location(
        bravado_client=bravado_client,
        remote_addr=get_remote_addr(request),
        locale=locale,
    ))

    fetch_recent_locations_future = asyncio.ensure_future(fetch_recent_locations(
        request=request,
        user_id=unauth_user_id,
        bravado_client=bravado_client,
        geolocator_client=geolocator_client,
        locale=locale,
    ))

    hero_user_future = loop.run_in_executor(executor=None, func=swaggerpy_client.user.retrieve_users_by_id
        ids=str(decid(hero_photo_owner_id)),
        locale=locale,
        omit_private_info=True,
    ).result) if hero_photo_owner_id else None

    hero_biz_future = loop.run_in_executor(executor=None, func=swaggerpy_client.business.list_by_ids_v4(
        business_ids=str(decid(hero_biz_id)),
    ).result) if hero_biz_id else None

    logged_in_user_future = loop.run_in_executor(executor=None, func=functools.partial(
        homepage.user.retrieve_users_from_request,
        request=request,
    )) if unauth_user_id else None
```

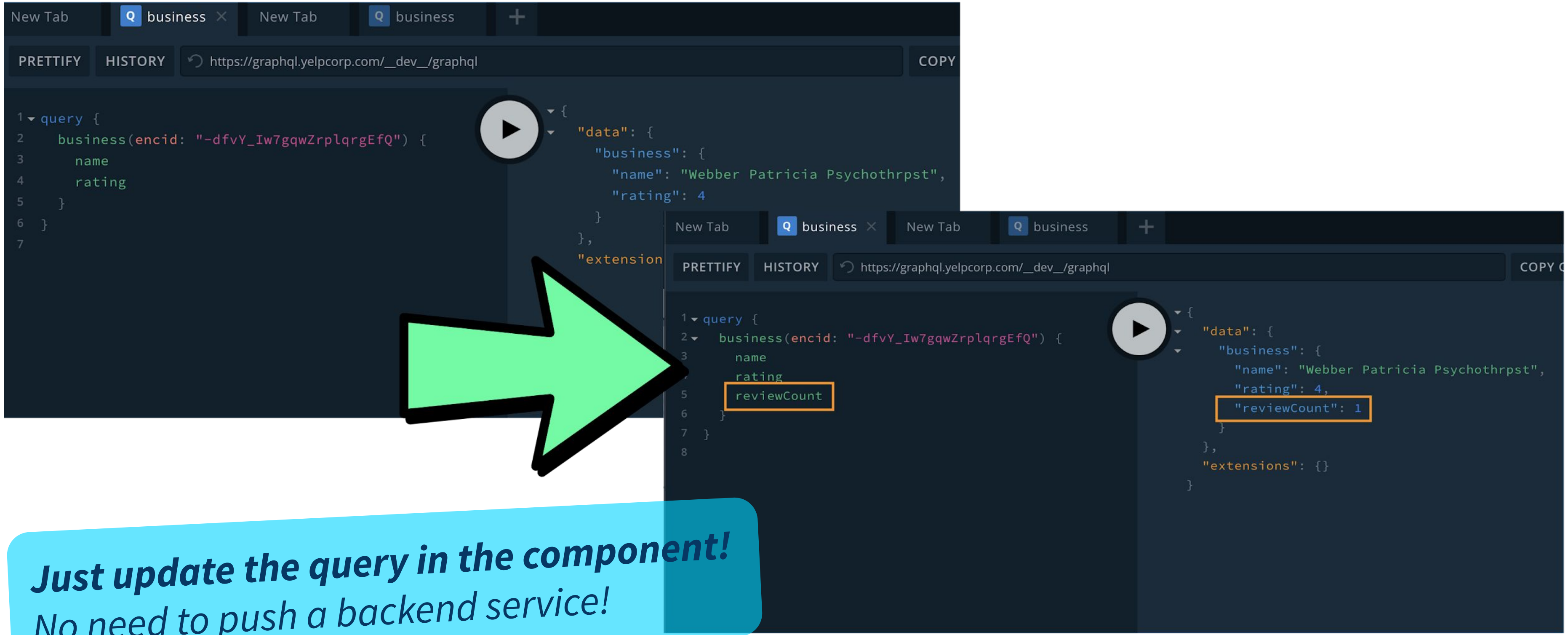


gondola-homepage / src / components / RecentActivityGQL / RecentActivityGQL.js

Jack Bug fixes for homepage activity feed 19 days ago Show history

```graphql

    query GetRecentActivityFeedData(
        $feed: PersonalActivityFeedType!
        $nearbyFeed: Boolean!
        $resultsPerPage: Int!
        $reactionsSourceFlow: String!
        $after: String
    ) {
        ...BusinessPhotoFeedItemRootFields
        loggedInUser {
            encid
            personalActivityFeed(feed: $feed, first: $resultsPerPage, after: $after) @skip(if: $ne
                edges {
                    node {
                        ...HomeActivityFeedGroupFields
                    }
                }
                pageInfo {
                    endCursor
                    hasNextPage
                }
            }
        }
        nearbyActivityFeed(first: $resultsPerPage, after: $after) @include(if: $nearbyFeed) {
            edges {
                node {
                    ...HomeActivityFeedGroupFields
                }
            }
            pageInfo {
                endCursor
                hasNextPage
            }
        }
    }
`;

```

5. Frontend Changes without Backend Changes (Improving **Time to Market** through quicker iteration times)



https://graphql.yelpcorp.com/__dev__/graphql

```
query {
  business(encid: "-dfvY_Iw7gqwZrplqrgEfQ") {
    name
    rating
  }
}
```

```
{
  "data": {
    "business": {
      "name": "Webber Patricia Psychothrpst",
      "rating": 4
    }
  },
  "extension
```

https://graphql.yelpcorp.com/__dev__/graphql

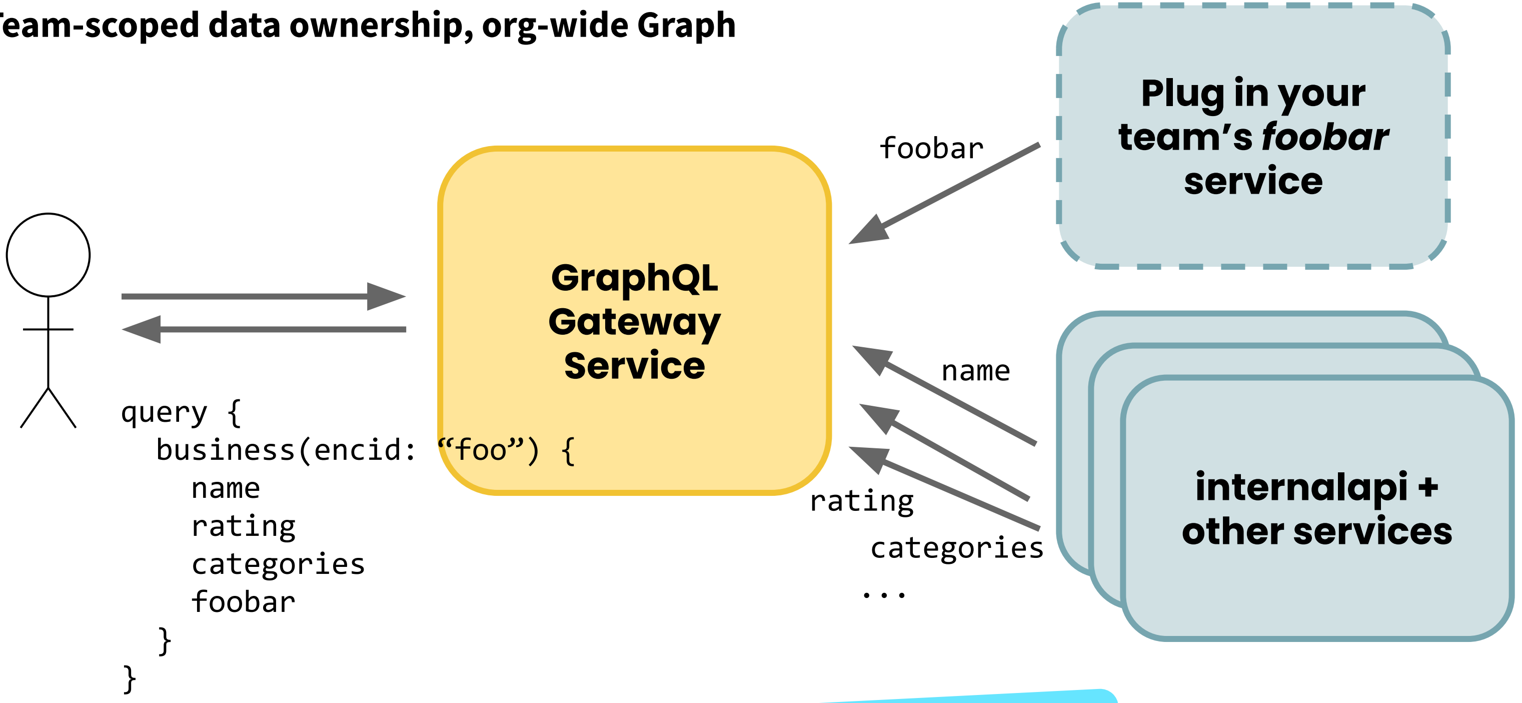
```
query {
  business(encid: "-dfvY_Iw7gqwZrplqrgEfQ") {
    name
    rating
    reviewCount
  }
}
```

```
{
  "data": {
    "business": {
      "name": "Webber Patricia Psychothrpst",
      "rating": 4,
      "reviewCount": 1
    }
  },
  "extensions": {}
}
```

Just update the query in the component!

No need to push a backend service!

6. Team-scoped data ownership, org-wide Graph


Plug in your team's *foobar* service
foobar
GraphQL Gateway Service
name
rating
categories
...
internalapi + other services

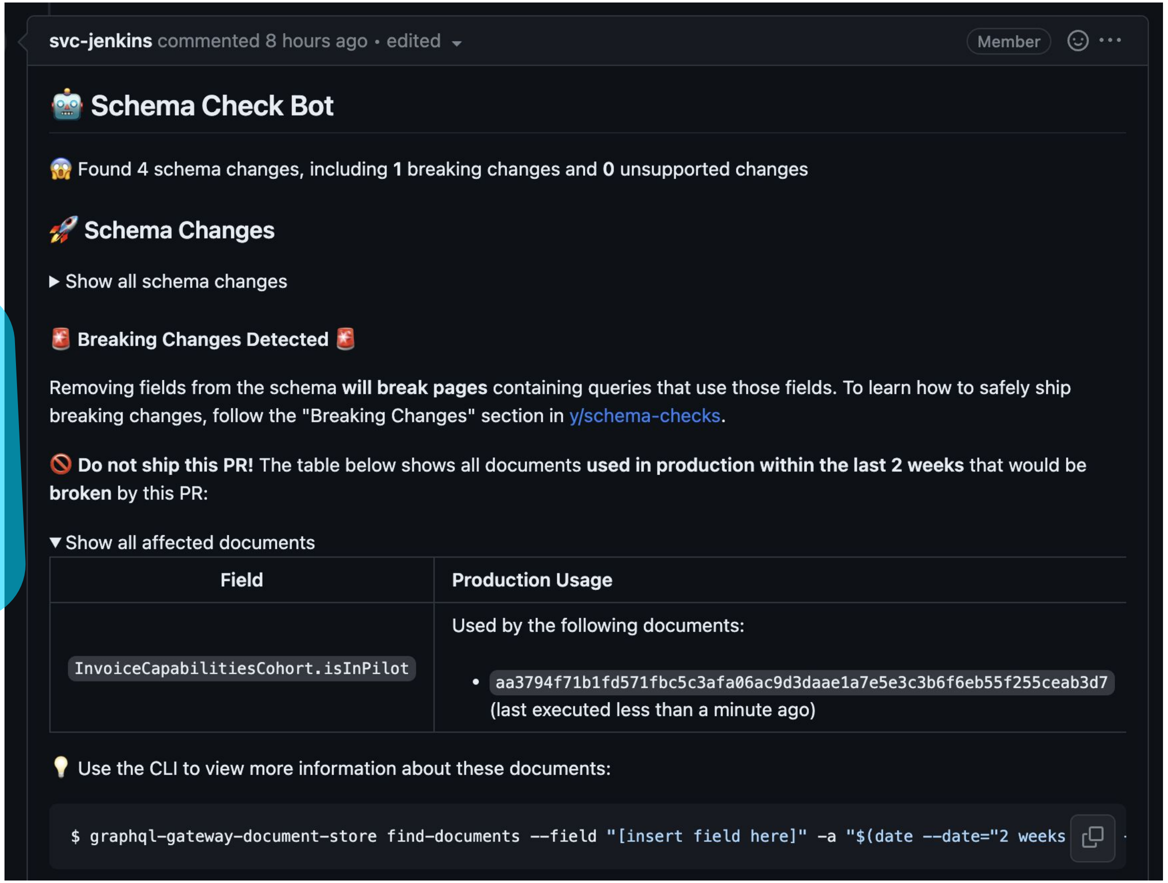
```
query {
  business(encid: “foo”) {
    name
    rating
    categories
    foobar
  }
}
```

Only manage the data your team owns!

Your page can fetch all the data it needs in **one** GQL query (including other team's data) without you having to worry about *how* it's all fetched. Just supply the encid of the entities you want - **GGS already knows how to fetch everything else!**

7. Zero-risk of breaking clients with breaking changes

Enforced safety for API changes. *Removed a field? Our Github PR bot prevents you deploying breaking changes that would break clients!*



svc-jenkins commented 8 hours ago • edited

Member

🤖 Schema Check Bot

😱 Found 4 schema changes, including **1** breaking changes and **0** unsupported changes

🚀 Schema Changes

▶ Show all schema changes

🚨 Breaking Changes Detected 🚨

Removing fields from the schema **will break pages** containing queries that use those fields. To learn how to safely ship breaking changes, follow the "Breaking Changes" section in y/schema-checks.

🚫 **Do not ship this PR!** The table below shows all documents **used in production within the last 2 weeks** that would be **broken** by this PR:

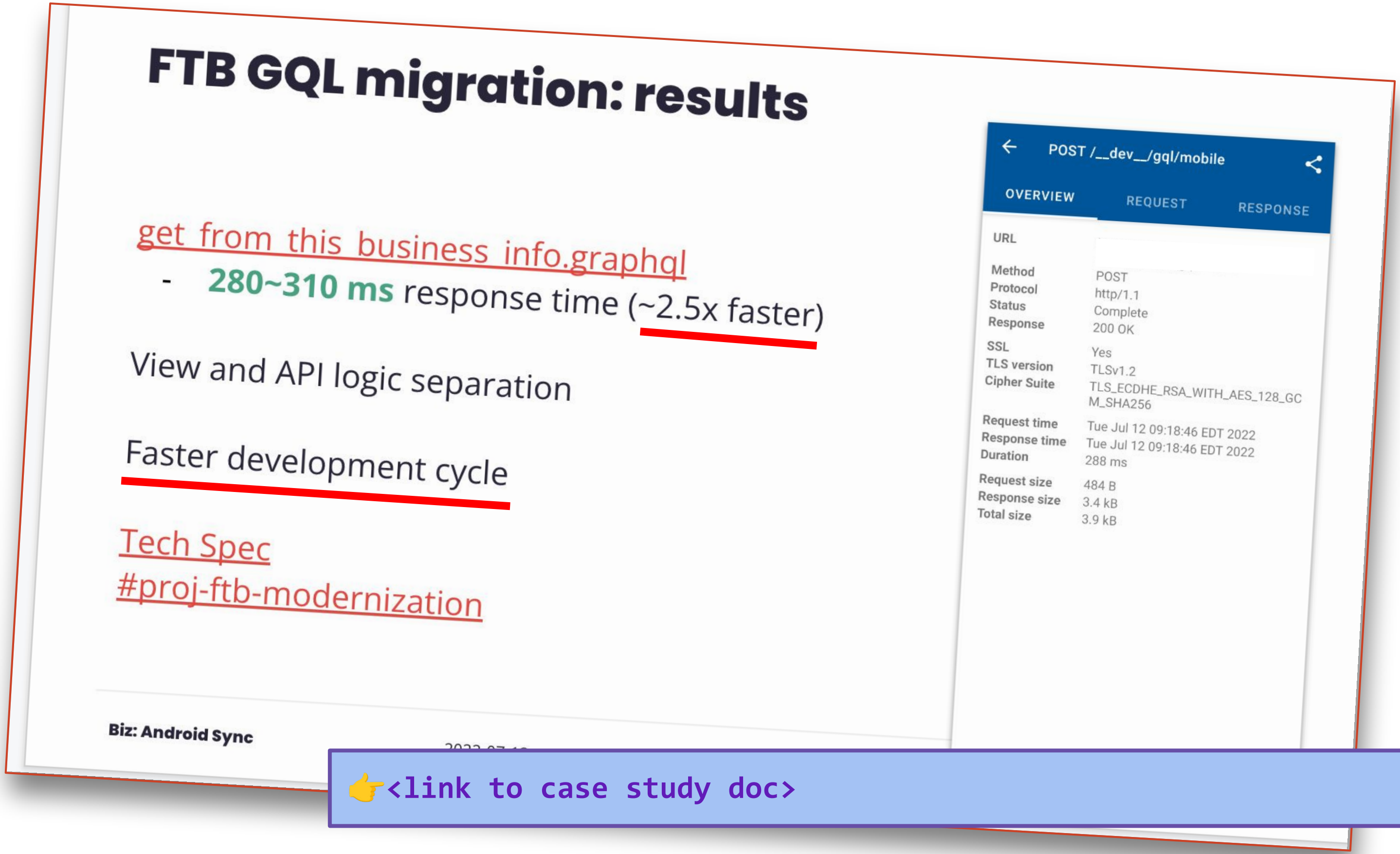
▼ Show all affected documents

Field	Production Usage
`InvoiceCapabilitiesCohort.isInPilot`	Used by the following documents: • `aa3794f71b1fd571fbc5c3afa06ac9d3daae1a7e5e3c3b6f6eb55f255ceab3d7` (last executed less than a minute ago)

💡 Use the CLI to view more information about these documents:

```
$ graphql-gateway-document-store find-documents --field "[insert field here]" -a "$(date --date="2 weeks
```

Case Study: “From This Business” product migration to GraphQL on Mobile



FTB GQL migration: results

get_from_this_business_info.graphql

- 280~310 ms response time (~2.5x faster)

View and API logic separation

Faster development cycle

Tech Spec

#proj-ftb-modernization

Biz: Android Sync

2022-07-1

POST /__dev__/gql/mobile

OVERVIEW REQUEST RESPONSE

URL	
Method	POST
Protocol	http/1.1
Status	Complete
Response	200 OK
SSL	Yes
TLS version	TLSv1.2
Cipher Suite	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
Request time	Tue Jul 12 09:18:46 EDT 2022
Response time	Tue Jul 12 09:18:46 EDT 2022
Duration	288 ms
Request size	484 B
Response size	3.4 kB
Total size	3.9 kB

👉<link to case study doc>

Next steps

What's in it for my team?

- Faster iteration cycles and product releases
- More delightful developer experience

We want you to try it out!

GraphQL is fully available for all developers at Yelp! If you're interested in seeing how GraphQL can save you time on your upcoming project, **please reach out in #graphql!**

Where can I learn more?

👉 **y/graphql-docs** 👈