

GraphQL

Gunnar Már Þorleifsson
Veftóftir og vængjaþytur, 8. maí 2019



GraphQL

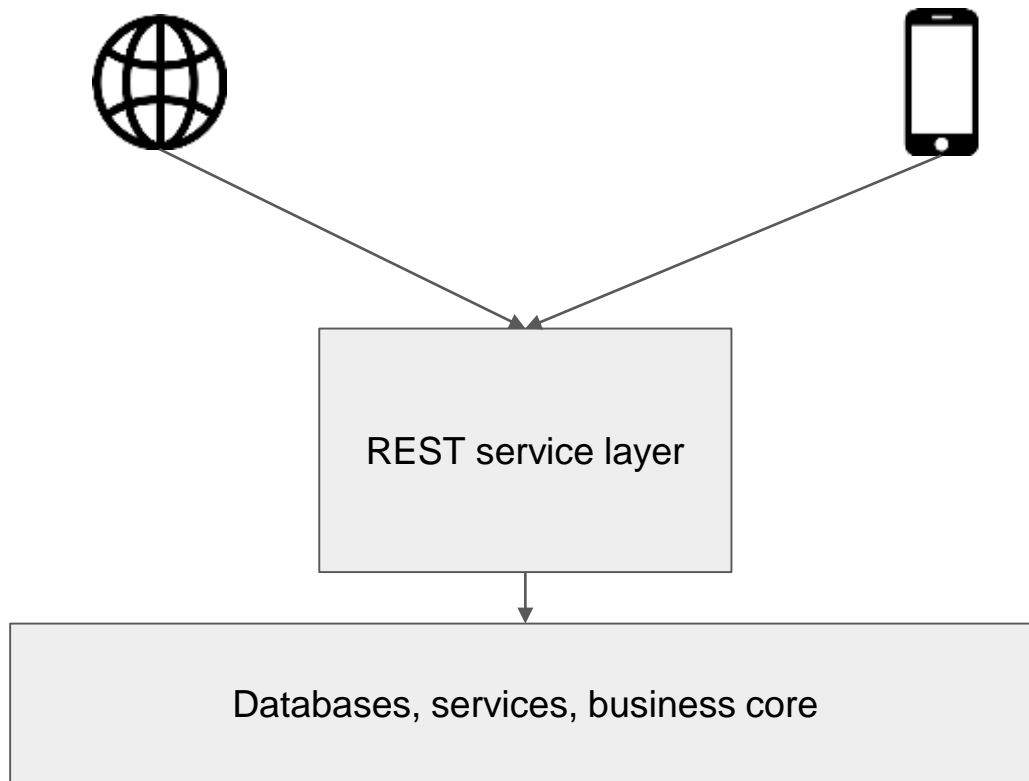
- Fyrirspurnarmál fyrir vefþjónustur
- Búið til af Facebook 🧐
- Open Source síðan 2018
 - GraphQL foundation undir Linux foundation
 - Stofnmeðlimir:
 - Airbnb, Apollo, Coursera, Facebook, Github, Shopify, Twitter ofl.



Hvers vegna?



REST vefþjónustur



REST Vefþjónustur

Að gefnu safni af færslum á vefþjóni á að vera til einsleitt URL og HTTP aðferðir sem er hægt að nota til að vinna með þetta safn af færslum.



REST Vefþjónustur

HTTP aðgerðir fyrir safn af bókum

URL	Aðferð	Aðgerð
/books	POST	Búa til færslu af bók í safni
/books	GET	Sækja safn af bókum
/books/3	GET	Sækja bók með ID 3 í safni
/books/3	PUT	Uppfæra bók með ID 3 í safni
/books/3	PATCH	Uppfæra ákveðin eigindi bókar með ID 3 í safni
/books/3	DELETE	Eyða bók með ID 3 í safni

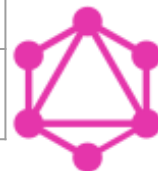


REST Vefþjónustur

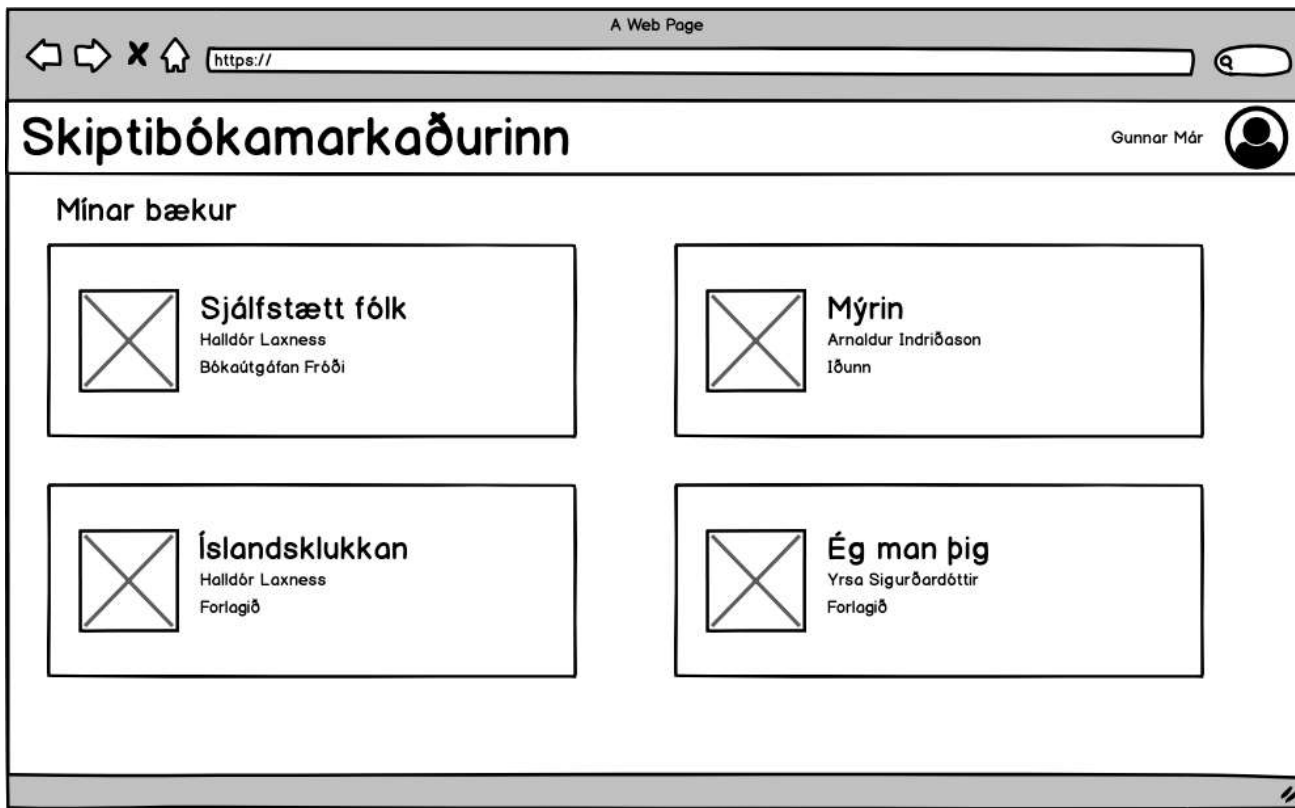
HTTP aðgerðir fyrir safn af notendum og vensl milli bóka og notenda

URL	Aðferð	Aðgerð
/users	POST	Búa til færslu af notanda í safni
/users	GET	Sækja lista af notendum
/users/16	GET	Sækja notanda 16
/users/16	PUT	Uppfæra notanda 16
/users/16	PATCH	Uppfæra eigindi notanda 16
/users/16	DELETE	Eyða notanda 16

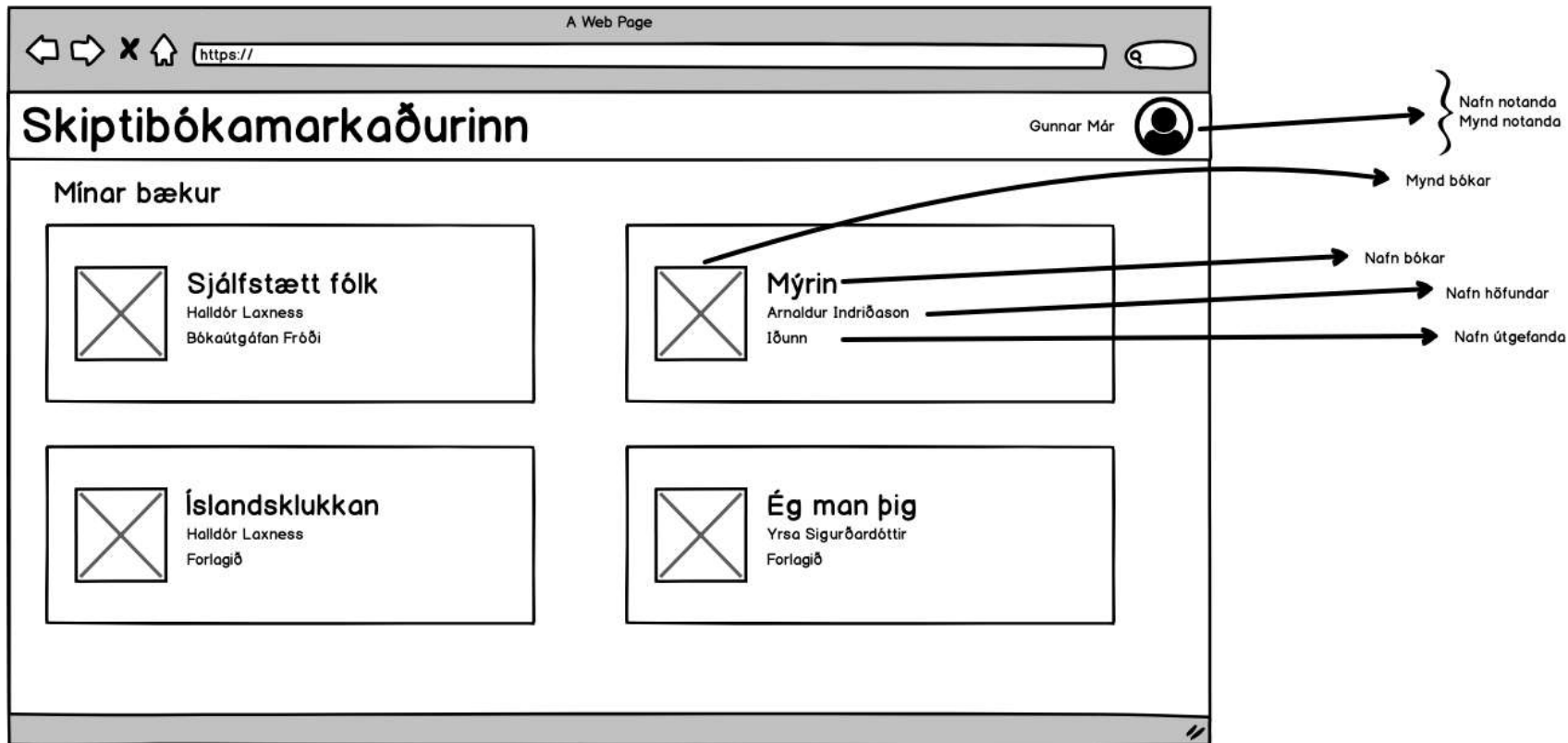
URL	Aðferð
/users/16/books	POST
/users/16/books	GET
/users/16/books/3	GET
/users/16/books/3	PUT
/users/16/books/3	PATCH
/books/16/books/3	DELETE



Vefviðmót fyrir skiptibókamarkað



Vefviðmót fyrir skiptibókamarkað



Gagnamódelið

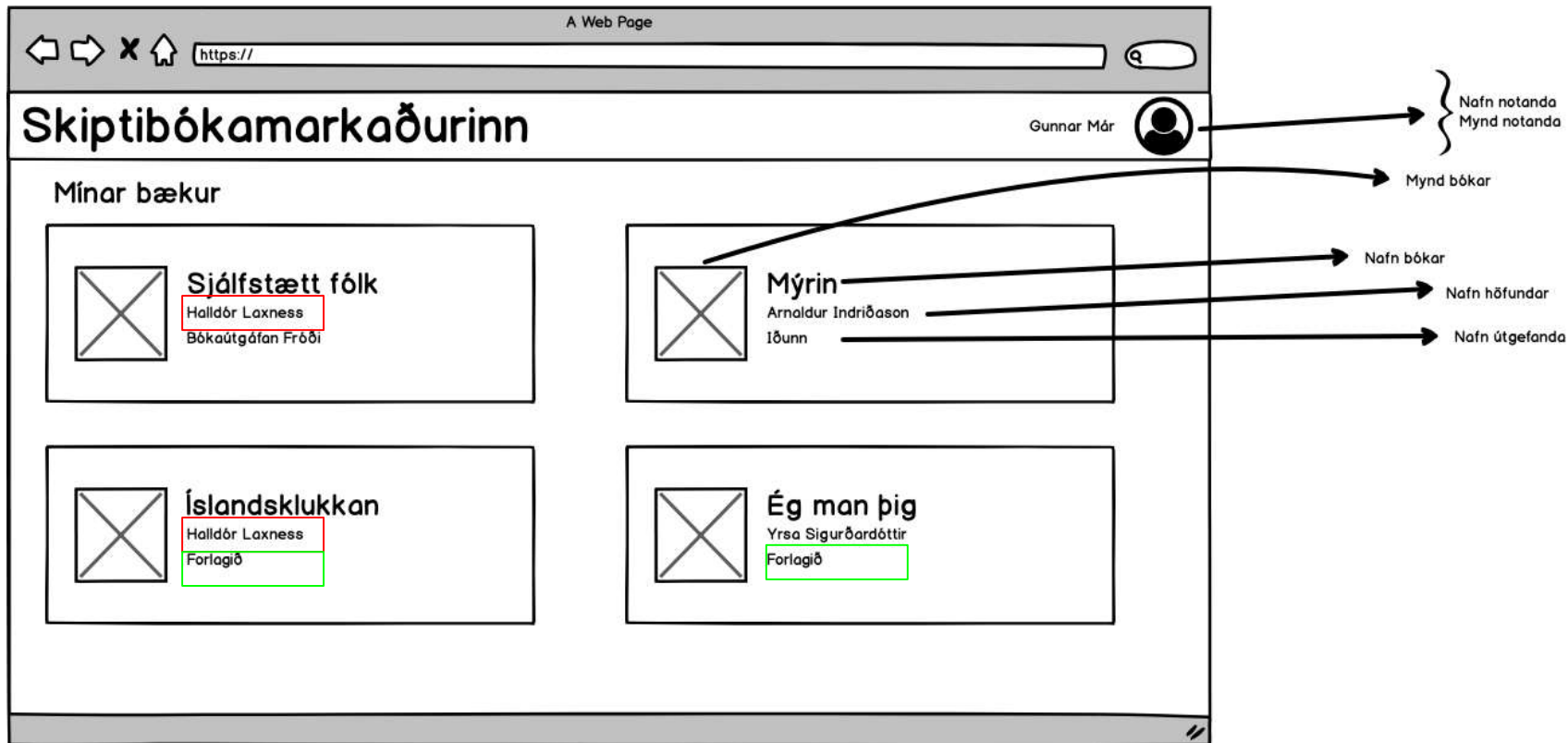
Book	
id	Id
name	String
image	String
author_id	Id
publisher_id	Id
....	

Author	
id	Id
name	String
image	String
...	

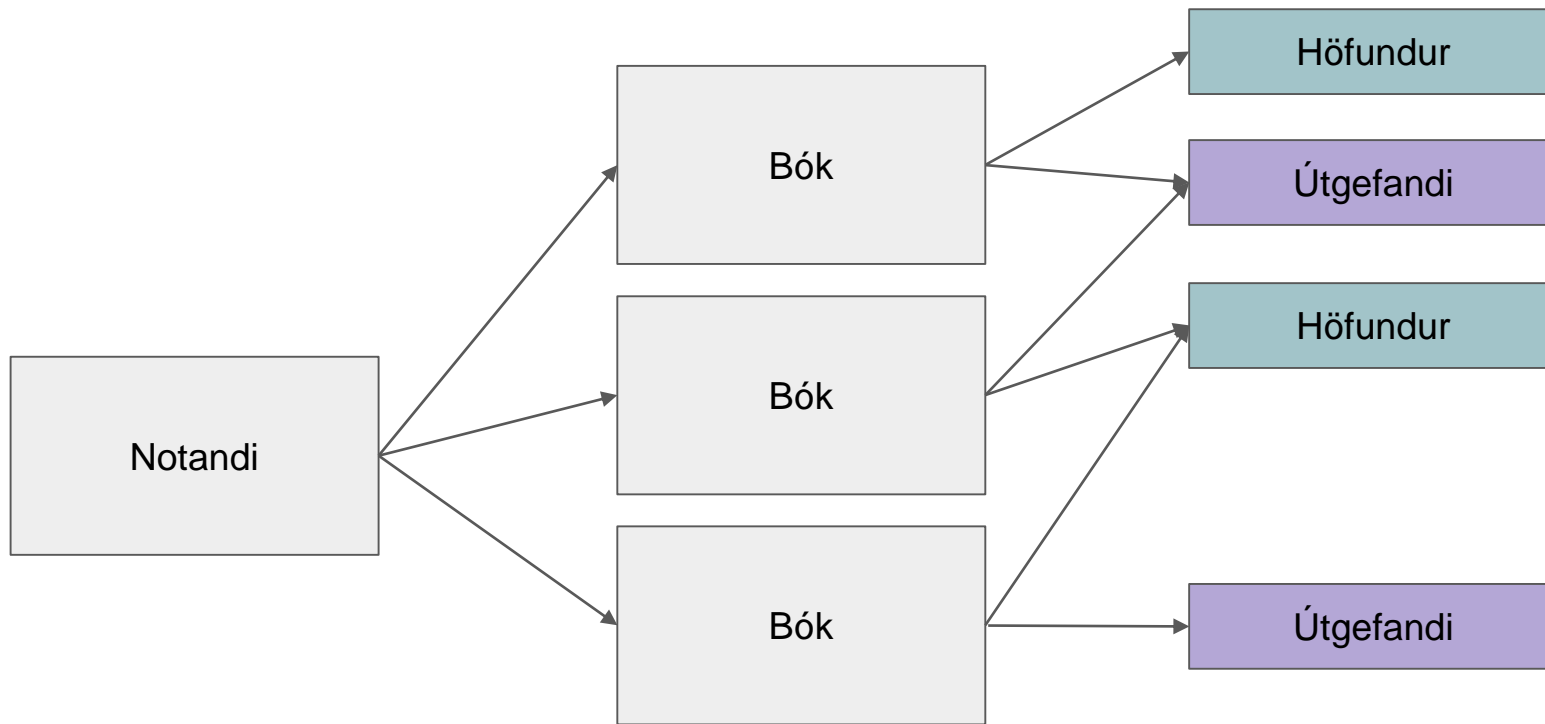
Publisher	
id	Id
name	String
description	String
...	



Vefviðmót fyrir skiptibókamarkað



Samband gagna

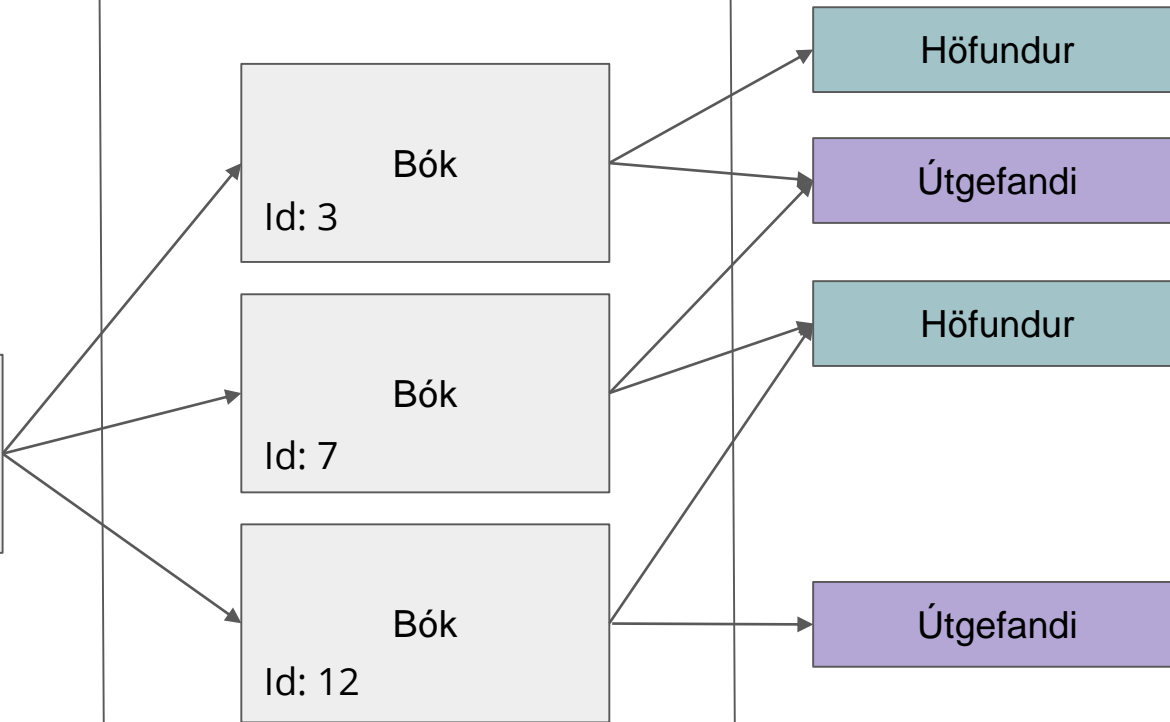
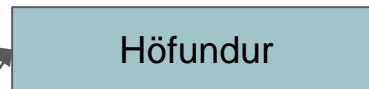
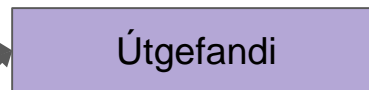
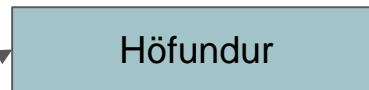
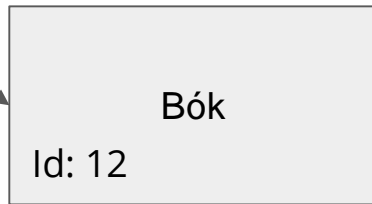
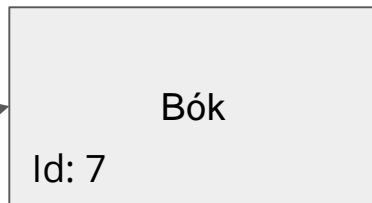
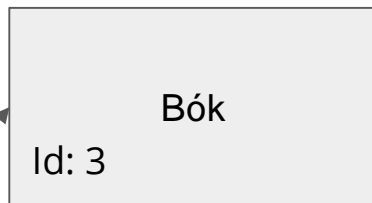


REST

/users/5

/users/5/books

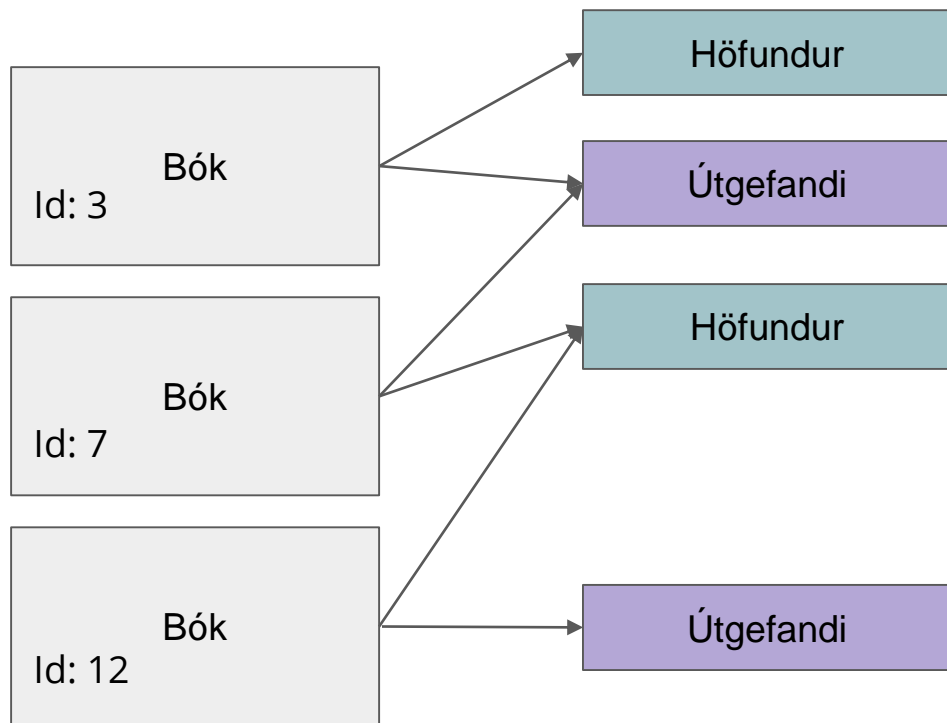
???



Tillaga 1

/users/5/books

/books/3/authors
/books/3/publishers
/books/7/authors
/books/7/publishers
/books/12/authors
/books/12/publishers

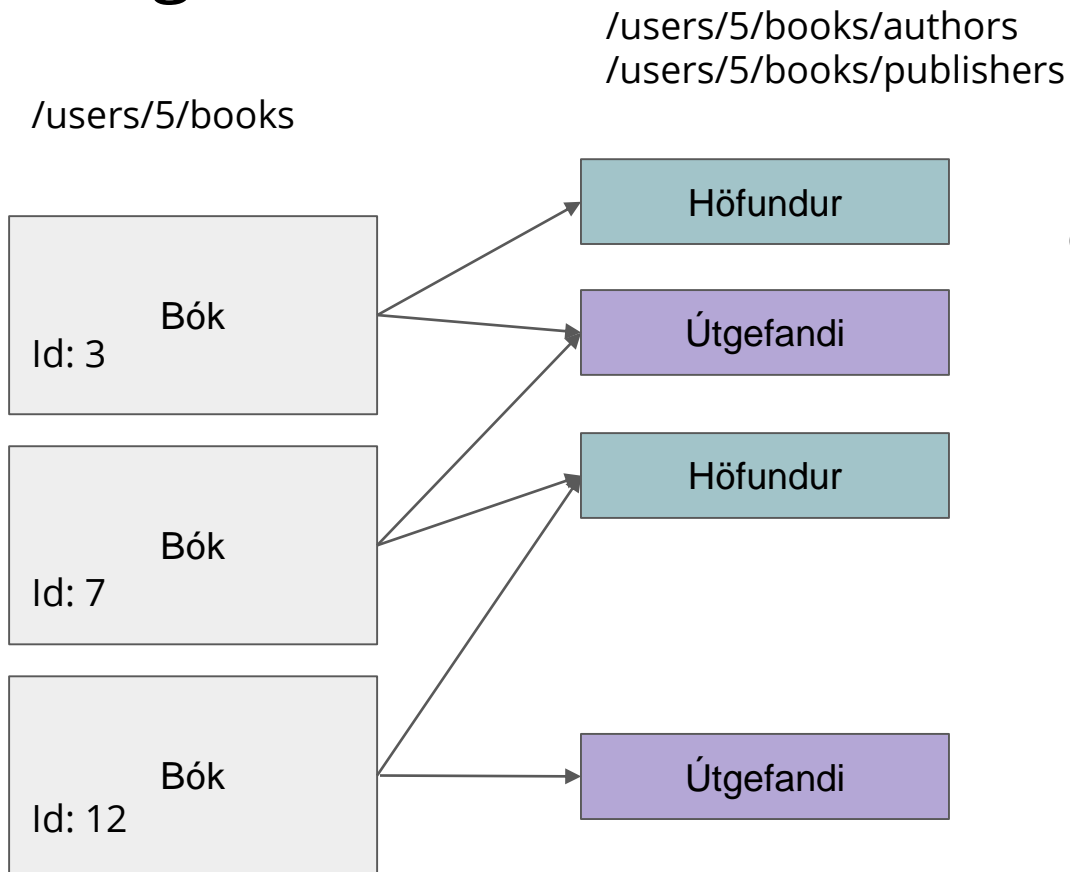


Fyrir hverja bók sækjum við höfund og útgefanda:

- Mörg http request fyrir eina síðu
- Flækir utanumhald gagna í client



Tillaga 2



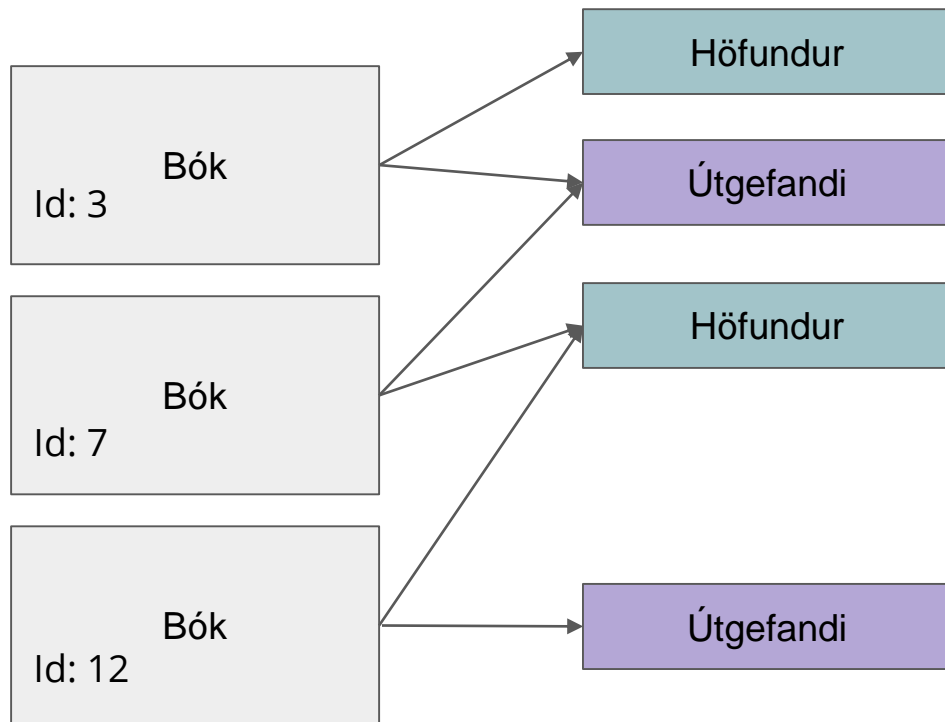
Búum til endapunkta fyrir alla höfunda og alla útgefendur á öllum bókum hjá notanda 5

- Byrjar aðeins að brjóta REST venjur
- Þurfum að pússla höfundum og útgefendum við bækur client megin
- Mjög sértækt, mikið customization
- Bakendaforritari þarf að búa til route-ið og fyrirspurnina
- Skalast illa með fleiri fíðusum



Tillaga 3

`/users/5/books_with_authors_and_publishers`



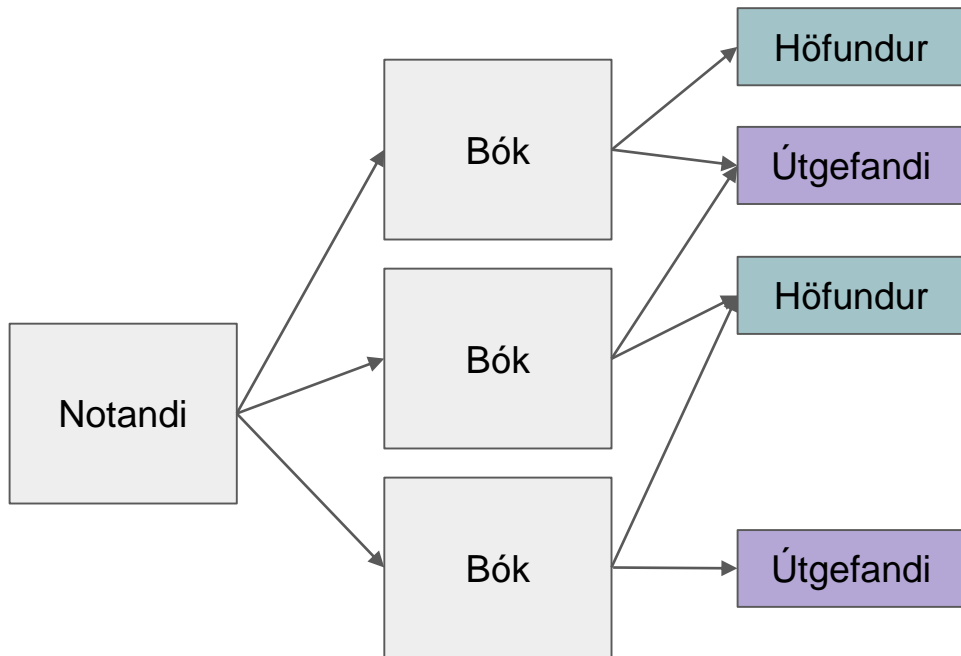
Búum til einn endapunkt sem sækir allar bækur með höfundum og útgefendum fyrir ákveðinn notanda:

- Gjörbrýtur REST venjur
- Gríðarlega sérstækt
- Nýr endapunktur fyrir hverja síðu eða feature



Allar tillögur

ALLTOF MIKIÐ AF GÖGNUM



```
{
  "id": 3,
  "name": "Gunnar Már Þorleifsson",
  "image": "http://cdn.some-company.com/assets/user-3.jpg",
  "address": "Gunnarsbraut",
  "city": "Reykjavík",
  "country": "Iceland",
  "type": "seller",
  "phonenumber": "7777777",
  "kennitala": "1234567890",
  "created_at": "2019-05-07T16:36:22.456Z",
  "last_updated_at": "2019-05-07T16:36:22.456Z",
  "books": [
    {
      "id": 3,
      "name": "Sjálfstætt fólk",
      "description": "Góð bok um sjálfstætt fólk...",
      "image": "http://cdn.some-company.com/assets/sjalfstætt-folk.jpg",
      "price": "1500",
      "currency": "ISK",
      "year_published": "1940",
      "version": "4",
      "review_count": 200,
      "created_at": "2019-05-07T16:36:22.456Z",
      "last_updated_at": "2019-05-07T16:36:22.456Z",
      "authors": [
        {
          "id": 12,
          "name": "Halldór Laxness",
          "description": "Góður rithöfundur",
          "image": "http://cdn.some-company.com/assets/halldur.jpg",
          "created_at": "2019-05-07T16:36:22.456Z",
          "last_updated_at": "2019-05-07T16:36:22.456Z"
        }
      ]
    },
    {
      "id": 17,
      "name": "Bókautgáfan Fróði",
      "description": "Mefur gefið út bækur í aldanna rás",
      "image": "http://cdn.some-company.com/assets/frodi.jpg",
      "created_at": "2019-05-07T16:36:22.456Z",
      "last_updated_at": "2019-05-07T16:36:22.456Z"
    }
  ]
}
}
```

Allar tillögur

ALLTOF MIKIÐ AF GÖGNUM

```
{
  "name": "Gunnar Már Þorleifsson",
  "image": "http://cdn.some-company.com/assets/user-3.jpg",
  "books": [
    {
      "name": "Sjálfstætt fólk",
      "image": "http://cdn.some-company.com/assets/sjalfstaett-folk.jpg",
      "authors": [
        {
          "name": "Halldór Laxness"
        }
      ],
      "publishers": [
        {
          "name": "Bókautgáfan Fróði"
        }
      ]
    }
  ]
}
```

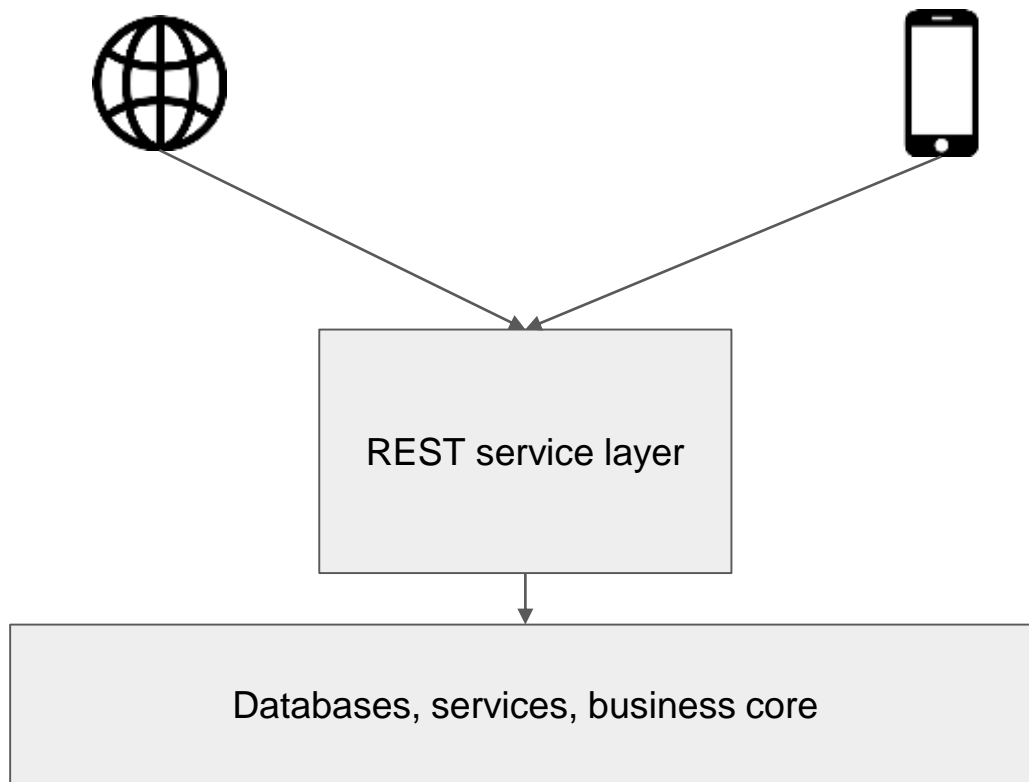
```
{
  "id": 3,
  "name": "Gunnar Már Þorleifsson",
  "image": "http://cdn.some-company.com/assets/user-3.jpg",
  "address": "Gunnarsbraut",
  "city": "Reykjavík",
  "country": "Iceland",
  "type": "seller",
  "phonenumber": "7777777",
  "kennitala": "1234567890",
  "created_at": "2019-05-07T16:36:22.456Z",
  "last_updated_at": "2019-05-07T16:36:22.456Z",
  "books": [
    {
      "id": 3,
      "name": "Sjálfstætt fólk",
      "description": "Góð bok um sjálfstætt fólk...",
      "image": "http://cdn.some-company.com/assets/sjalfstaett-folk.jpg",
      "price": "1500",
      "currency": "ISK",
      "year_published": "1940",
      "version": "4",
      "review_count": 200,
      "created_at": "2019-05-07T16:36:22.456Z",
      "last_updated_at": "2019-05-07T16:36:22.456Z",
      "authors": [
        {
          "id": 12,
          "name": "Halldór Laxness",
          "description": "Góður rithöfundur",
          "image": "http://cdn.some-company.com/assets/halldor.jpg",
          "created_at": "2019-05-07T16:36:22.456Z",
          "last_updated_at": "2019-05-07T16:36:22.456Z"
        }
      ],
      "publishers": [
        {
          "id": 17,
          "name": "Bókautgáfan Fróði",
          "description": "Mefur gefið út bækur í aldanna rás",
          "image": "http://cdn.some-company.com/assets/frodi.jpg",
          "created_at": "2019-05-07T16:36:22.456Z",
          "last_updated_at": "2019-05-07T16:36:22.456Z"
        }
      ]
    }
  ]
}
```

Engin augljós leið

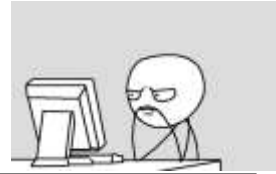
- RESTful arkitektúr flækist í praktík þegar þú ert með mikil vensl
- Annað hvort of mikið af http fyrirspurnum
- Eða alltof sérstækir endapunktur
- Og brot á REST venjum
- Alltof mikið af gögnum send í clientinn



Skoðum þetta aftur



Nær raunveruleikanum



Innri starfsmanna vefur

Markaðs og söluvefur

Mínar síður

Gamall söluvefur

Gömul native öpp

Í þróun: Nýtt cross-platform app

Order Service

User Service

Business Service #1

Business Service #2

Business Service #3 (gömul og brýtur venjur)

CMS



Product catalog



{ "snake_case" : "data" }

{ "camelCase" : "data" }

{ "PascalCase" : "data" }



CRM/ERP/Domain specific system

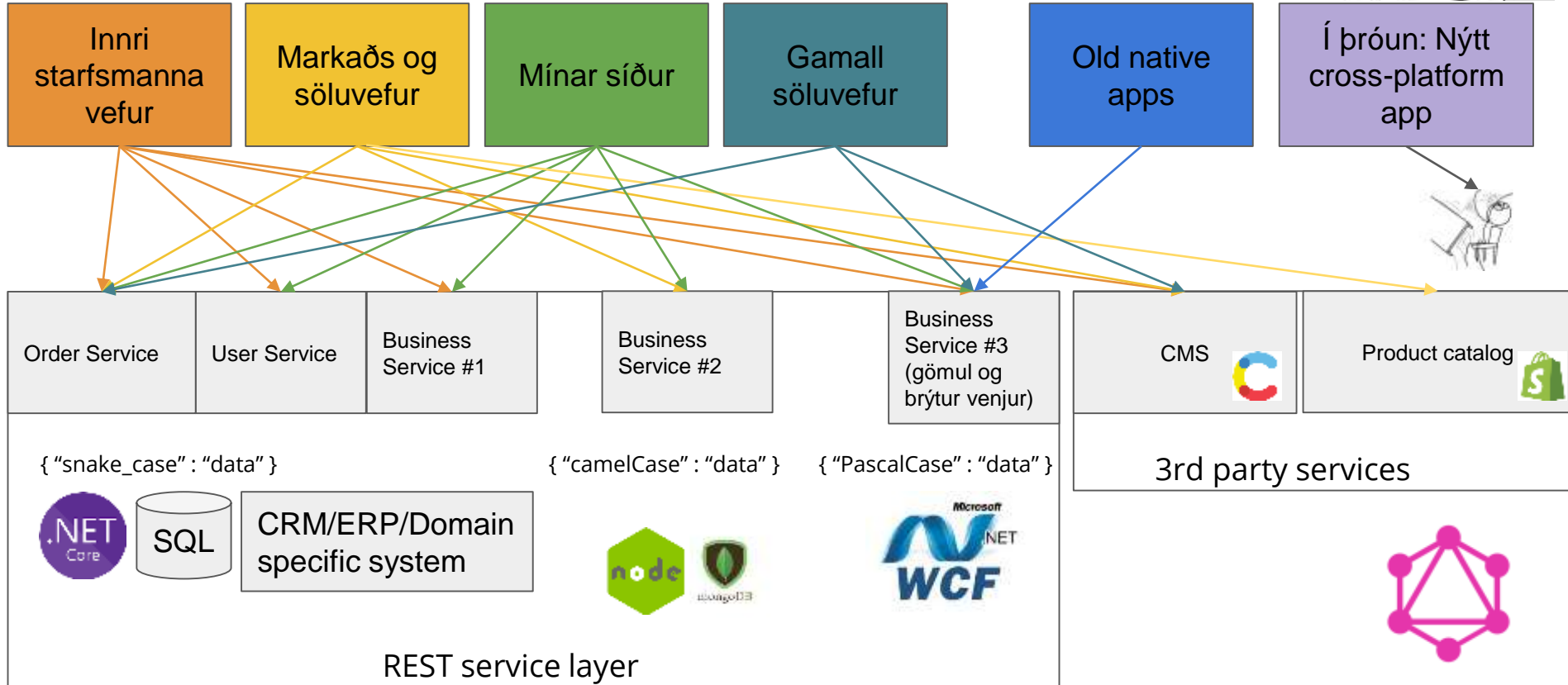


3rd party services



REST service layer

Raunveruleikinn er ekki alltaf fallegur



Hvers vegna:

- RESTful architecture flækist í praktík þegar þú ert með mikil vensl
- Alltof mikið af gögnum send í clientinn
- Dreif þjónustulag flækir client-a og hægir á nýpróun
- Mistækt þjónustulag færir mikið af legacy upp í clienta



Hvað?



Hvað er GraphQL?

A query language for your API

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.

Lýsir gögnum

- Öflugt type system
- Schema fyrir öll gögn

```
demo.graphql x
1  type Query {
2    user: User
3  }
4
5  type User {
6    id: ID
7    name: String
8    image: String
9    books: [Book]
10 }
11
12 type Book {
13   id: ID
14   name: String
15   image: String
16   author: Author
17   publisher: Publisher
18 }
19
20 type Author {
21   id: ID
22   name: String
23   image: String
24 }
25
26 type Publisher {
27   id: ID
28   name: String
29   description: String
30 }
31
32 schema {
33   query: Query
34 }
35
36
```

```
demo_schema.graphql x
1  {
2    user {
3      id
4      name
5      image
6      books {
7        id
8        name
9        image
10       author {
11         id
12         name
13         image
14       }
15       publisher {
16         id
17         name
18         description
19       }
20     }
21   }
22 }
23
```



Sækir gögn með Query

- Sæktu þau gögn sem þú þarft og fáðu það sem þú baðst um
- Sækir margar resource-a í einni fyrirspurn

```
demo_query.graphql x
1 {
2   user (id: "5") {
3     name
4     image
5     books {
6       name
7       image
8       author {
9         name
10      }
11     publisher {
12       name
13     }
14   }
15 }
16 }
17 }
```

```
demo_data.json x
1 {
2   "data": {
3     "name": "Gunnar Már Þorleifsson",
4     "image": "http://cdn.some-company.com/assets/user-3.jpg",
5     "books": [
6       {
7         "name": "Sjálftætt fólk",
8         "image": "http://cdn.some-company.com/assets/sjalfstaett-folk.jpg",
9         "author": {
10           "name": "Halldór Laxness"
11         },
12         "publisher": {
13           "name": "Bókafélagið Frósi"
14         }
15       },
16       {
17         "name": "Myrin",
18         "image": "http://cdn.some-company.com/assets/myrin.jpg",
19         "author": {
20           "name": "Arnaldur Indriðason"
21         },
22         "publisher": {
23           "name": "Íðunn"
24         }
25       }
26     ]
27   }
28 }
29 }
```



Uppfærir gögn

- Mutation
- Framkallaðu breytingar á gögnum og fáðu uppfærð gögn til baka

```
32 schema {
33   query: Query
34   mutation: Mutation
35 }
36
37 type Mutation {
38   bookCreate(userId: ID!, input: BookCreateInput!): BookCreatePayload
39 }
40
41 input BookCreateInput {
42   name: String!
43   image: String
44   authorId: ID!
45   publisherId: ID!
46 }
47
48 type BookCreatePayload {
49   book: Book
50   error: Error
51 }
52
53 type Error {
54   message: String!
55 }
56
```



Uppfærir gögn

- Mutation
- Framkallaðu breytingar á gögnum og fáðu uppfærð gögn til baka

```
demo_mutation.graphql x
1  mutation {
2    bookCreate(userId: "5", input: {
3      name: "Íslandsklukkan",
4      image: "some.url/bla",
5      authorId: 12,
6      publisherId: 17
7    }) {
8      book {
9        name
10       image
11       author {
12         name
13       }
14       publisher {
15         name
16       }
17     }
18     error {
19       message
20     }
21   }
22 }
23 }
```

```
demo_mutation_data.json x
1  {
2    "data": {
3      "book": {
4        "name": "Íslandsklukkan",
5        "image": "some.url/bla",
6        "author": {
7          "name": "Halldór Laxness"
8        },
9        "publisher":
10       {
11         "name": "Bókaútgáfan Fróði"
12       }
13     },
14     "error": null
15   }
16 }
17 }
```

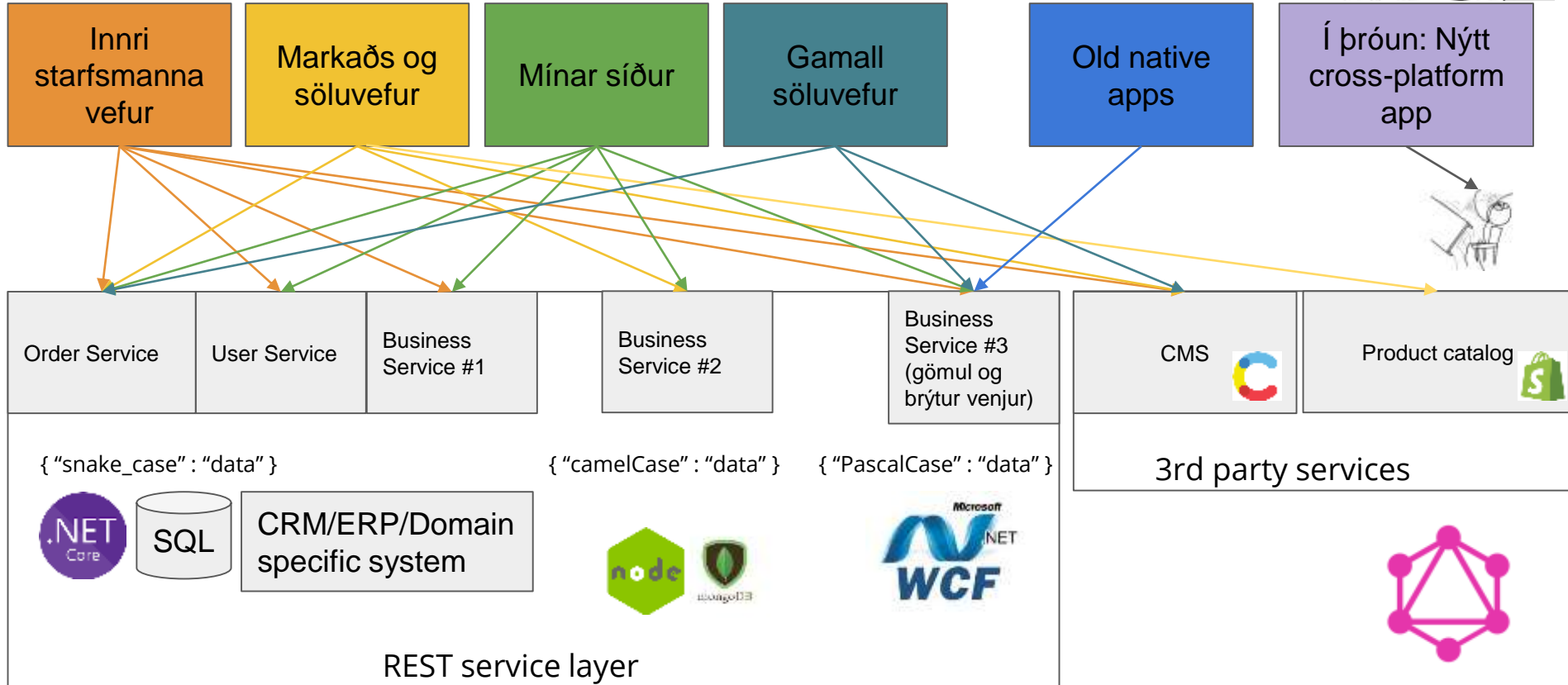
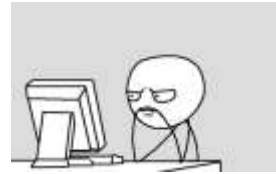


Einfaldar aðgengi clienta

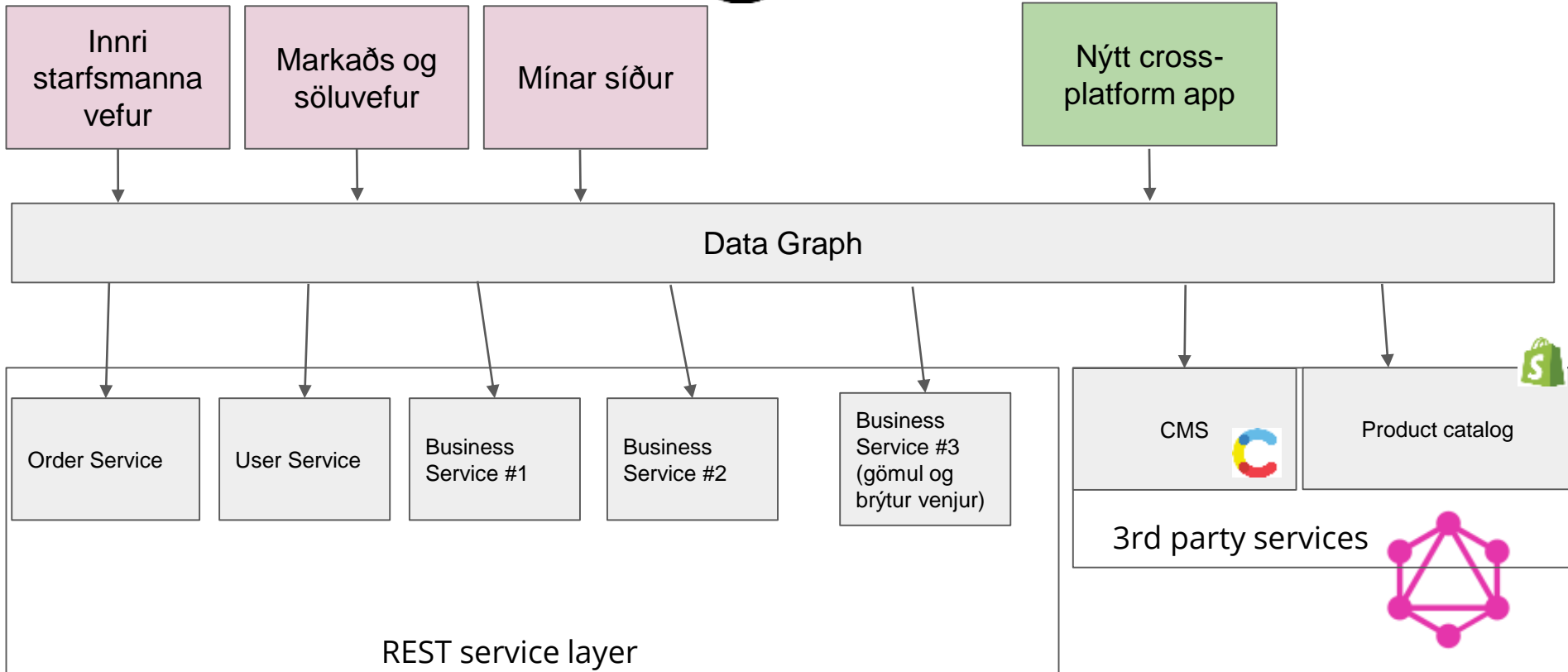
- API gateway fyrir sett af clientum
- Staðlar gögn úr mismunandi áttum t.d. PascalCase vs. camelCase vs. snake_case.



Rauveruleikinn er ekki alltaf fallegur



Nýi raunveruleikinn



Hvernig?



GraphQL server

- Útfærslur á vefþjóni til fyrir fjölda mála
 - JavaScript
 - Haskell
 - Python
 - Ruby
 - Java
 - C#
 - Scala
 - Go
 - PHP
 - ...ofl

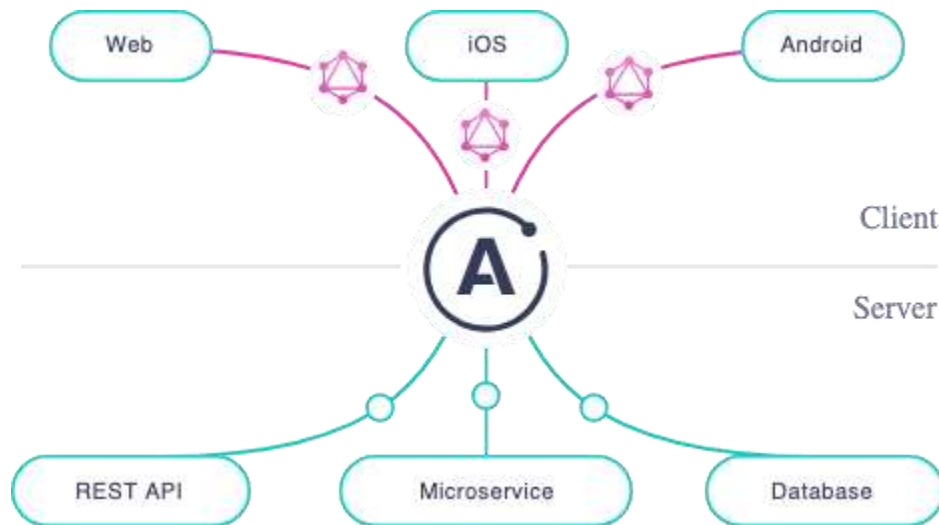


Apollo

- Apollo server
 - Útfærsla á GraphQL server fyrir node.js
- Apollo client
 - Library fyrir framenda til að nota GraphQL þjónustu
 - JavaScript:
 - React
 - Angular
 - Vue
 - Meteor
 - Ember
 - Native:
 - Swift -> iOS
 - Java -> Android



Apollo Server



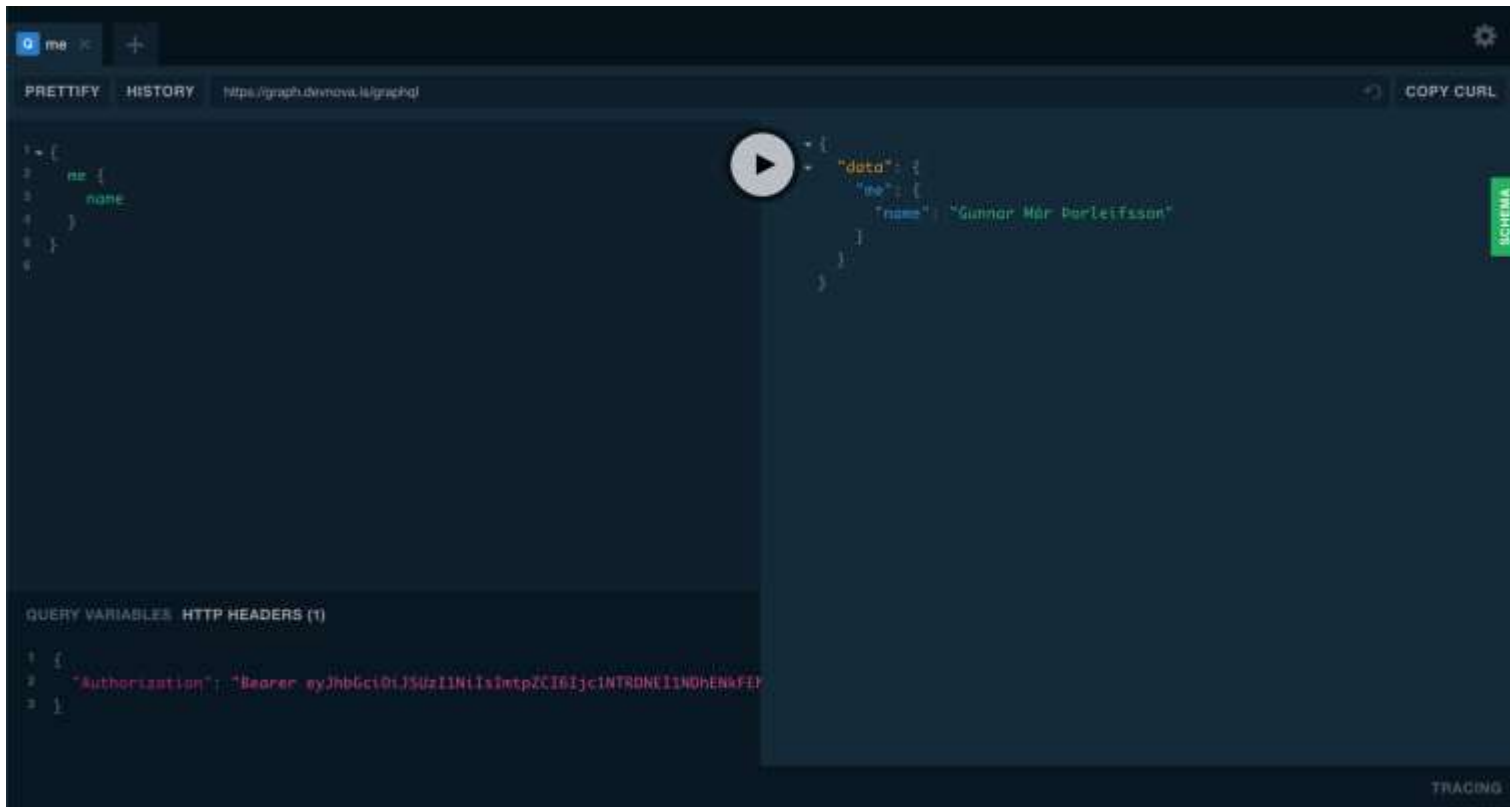
Apollo Server

```
3 import { ApolloServer } from 'apollo-server';
4
5 import typeDefs from 'schema/Query.graphql';
6
7 import {
8   userRepository,
9   bookRepository,
10  authorRepository,
11  publisherRepository,
12 } from 'repositories';
13
14 const resolvers = {
15   Query: {
16     user: (_, { id }) => userRepository.getUserById(id),
17   },
18   User: {
19     books: ({ userId }) => bookRepository.getBooksByUser(userId),
20   },
21   Book: {
22     author: ({ authorId }) => authorRepository.getAuthor(authorId),
23     publisher: ({ publisherId }) => publisherRepository.getPublisher(publisherId),
24   },
25 };
26
27 const server = new ApolloServer({ typeDefs, resolvers });
28
29 server.listen(3000, () => {
30   console.info('Server ready at localhost:3000');
31 });
32
```

REST API,
MICROSERVICE,
DATABASE,
JSON,
MOCK

```
1 type Query {
2   user: User
3 }
4
5 type User {
6   id: ID
7   name: String
8   image: String
9   books: [Book]
10 }
11
12 type Book {
13   id: ID
14   name: String
15   image: String
16   author: Author
17   publisher: Publisher
18 }
19
20 type Author {
21   id: ID
22   name: String
23   image: String
24 }
25
26 type Publisher {
27   id: ID
28   name: String
29   description: String
30 }
31
32 schema {
33   query: Query
34 }
35
36
```

Playground



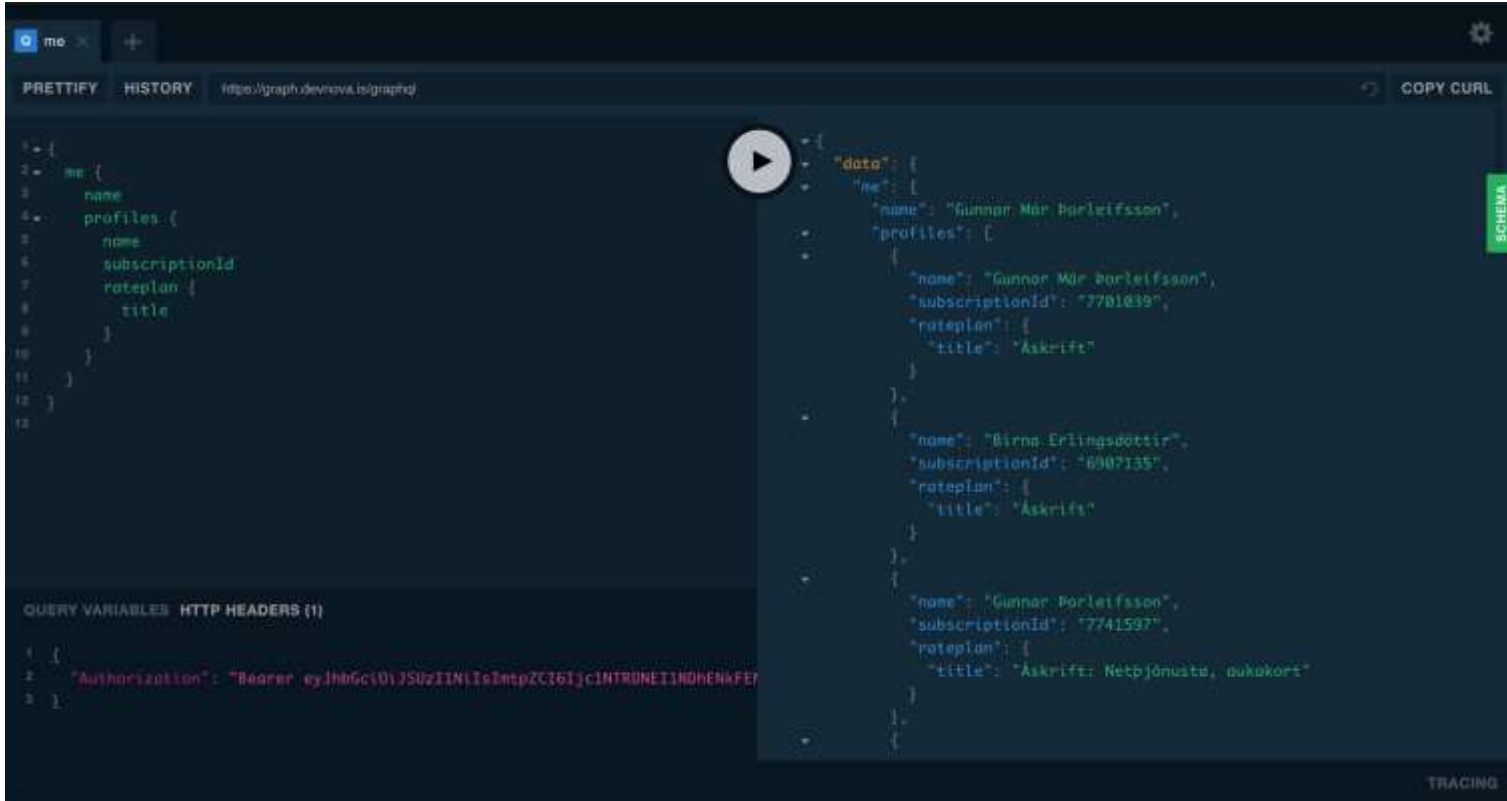
The screenshot displays a GraphQL Playground interface with a dark theme. At the top, there are tabs for "PRETTIFY" and "HISTORY", and a "COPY CURL" button on the right. The main area is split into three sections:

- Query Editor:** Contains the query `1 {`
`2 me {`
`3 name`
`4 }`
`5 }`
- Response:** Shows the JSON result: `{`
 `"data": {`
 `"me": {`
 `"name": "Gunnar Már Þorleifsson"`
 `}`
 `}`
`}`
- Query Variables / HTTP Headers:** Shows the headers: `1 {`
`2 "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IWRONlki11NDhENkFEP`
`3 }`

A play button icon is visible in the center of the response area. A "SCHEMA" button is on the right edge, and "TRACING" is at the bottom right.



Playground



The screenshot shows a GraphQL Playground interface with a dark theme. At the top, there are tabs for "PRETTIFY", "HISTORY", and "COPY CURL". The main area is split into three sections:

- Query:** A GraphQL query is entered in the left pane:

```
1 {
2   me {
3     name
4     profiles {
5       name
6       subscriptionId
7       rateplan {
8         title
9       }
10    }
11  }
12 }
```
- Response:** The JSON response is shown in the right pane:

```
{
  "data": {
    "me": [
      {
        "name": "Gunnar Már Þorleifsson",
        "profiles": [
          {
            "name": "Gunnar Már Þorleifsson",
            "subscriptionId": "7701039",
            "rateplan": {
              "title": "Áskrift"
            }
          },
          {
            "name": "Birna Erlingsdóttir",
            "subscriptionId": "6907135",
            "rateplan": {
              "title": "Áskrift"
            }
          },
          {
            "name": "Gunnar Þorleifsson",
            "subscriptionId": "7741597",
            "rateplan": {
              "title": "Áskrift: Netþjónusta, aukakort"
            }
          }
        ]
      }
    ]
  }
}
```
- Query Variables:** The bottom section shows the query variables and HTTP headers:

```
1 {
2   "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IHR5cGU6Ij09IiwiaWF0Ij09Ij09"
3 }
```

A play button icon is visible in the center of the interface. A "SCHEMA" button is on the right side, and "TRACING" is at the bottom right corner.



Playground

The screenshot displays the GraphQL Playground interface with the following components:

- Left Panel (Query Editor):** Contains a query:

```
1 {
2   me {
3     name
4     profiles {
5       name
6       subscriptionId
7       rateplan {
8         title
9       }
10    }
11  }
12 }
```
- Schema Panel:** A search bar "Search the schema ..." and a list of queries including `helloWorld: String!`, `novelNumbers(...): (String)`, `me: Account`, `accountBySsn(...): Account`, `profiles(...): (Profile)`, `subscriptions(...): ProfilePayload!`, `offers(...): OffersPayload!`, `redeemedOffers(...): RedeemedOffersPayload!`, `tones(...): TonesPayload!`, `popularTones: PopularTonesPayload!`, `refills(...): (Refill!)`, `availableRefills(...): (RefillCategory!)`, `refillCategory(...): RefillCategory!`, `version: Version!`, `rateplans: (Rateplan)`, and `authentication: Credentials`.
- Type Details Panel (Account):** Shows the definition:

```
type Account {
  sub: ID!
  ssn: ID!
  name: String!
  email: String
  streetAddress: String
  postalCode: String
  city: String
  msisdn: String
  profiles(...): (Profile)
  subscriptions(...): ProfilePayload!
  invoices: (Invoice)
  claims: (Claim)
  transactions(...): AccountTransactionPayload!
  tones(...): TonesPayload!
  paymentMethod: PaymentMethod
  redeemedOffers: RedeemedOffersPayload!
}
```
- Type Details Panel (Rateplan):** Shows the definition:

```
type Rateplan {
  id: ID!
  title: String!
  shortTitle: String!
  typeId: String!
  price: Float!
  description: String
  isPrepaid: Boolean!
  receiptText: String
  isVip: Boolean
  includedDataAmountInMb: Int
  availableRefills: (RefillCategory)
  availableServicepacks: (ServicepackCategory)
  availableRateplans: (RateplanCategory)
}
```
- Bottom Panel:** Includes sections for "QUERY VARIABLES" and "HTTP HEADERS".



Apollo Client

- Client sem sér um að framkvæma Query og Mutation
- Sér um alla gagnaumsýslu og uppfærslu á viðmóti
- Client side caching
- Data subscription, polling, optimistic updates osfrv.
- Fyrir React, Angular, Vue og miklu fleira:
 - Query component
 - Mutation component



Apollo client: React

App.js

```
1 import React from 'react';
2 import { render } from 'react-dom';
3 import { ApolloClient } from 'apollo-client';
4 import { ApolloProvider } from 'react-apollo';
5
6 import Books from 'components/books';
7
8 const client = new ApolloClient({
9   uri: 'https://my-graphql-server.com',
10 });
11
12 const App = () => (
13   <ApolloProvider client={client}>
14     <h1>Skiptibókamarkaður</h1>
15     <Books userId="5" />
16   </ApolloProvider>
17 );
18
19 render(<App />, document.getElementById('root'));
```

Books.js

```
1 import React from 'react';
2 import { Query } from 'react-apollo';
3
4 import BOOKS_QUERY from './Books.graphql';
5
6 const Books = () => {
7   <Query query={BOOKS_QUERY}>
8     {({ loading, error, data }) => {
9       if (loading) {
10         return <p>Loading...</p>;
11       }
12       if (error) {
13         return <p>Error!</p>;
14       }
15
16       const { name, image, books } = data;
17
18       return (
19         <section>
20           <h2>{name}</h2>
21           <img src={image} alt={name} />
22           <ul>
23             {books.map(({ name: bookTitle, author, publisher }) => (
24               <li>{bookTitle} - {author.name} - {publisher.name}</li>
25             ))}
26           </ul>
27         </section>
28       );
29     }
30   </Query>
31 };
32
33 export default Books;
```

Books.graphql


```
query Books {
  user (id: "5") {
    name
    image
    books {
      name
      author {
        name
      }
      publisher {
        name
      }
    }
  }
}
```




A Web Page

https://


Skiptibókamarkaðurinn

Gunnar Már 


Mínar bækur




Sjálfstætt fólk
Halldór Laxness
Bókaútgáfan Fróði



Mýrin
Arnaldur Indriðason
Iðunn



Íslandsklukkan
Halldór Laxness
Forlagið



Ég man þig
Yrsa Sigurðardóttir
Forlagið



Takk



Ítarefni

- <https://graphql.github.io/learn/>
- <https://www.apollographql.com/docs/apollo-server/>
- <https://www.apollographql.com/docs/react/>

