

Learning GraphQL and Relay

Chapter 1: An Introduction to GraphQL and Relay

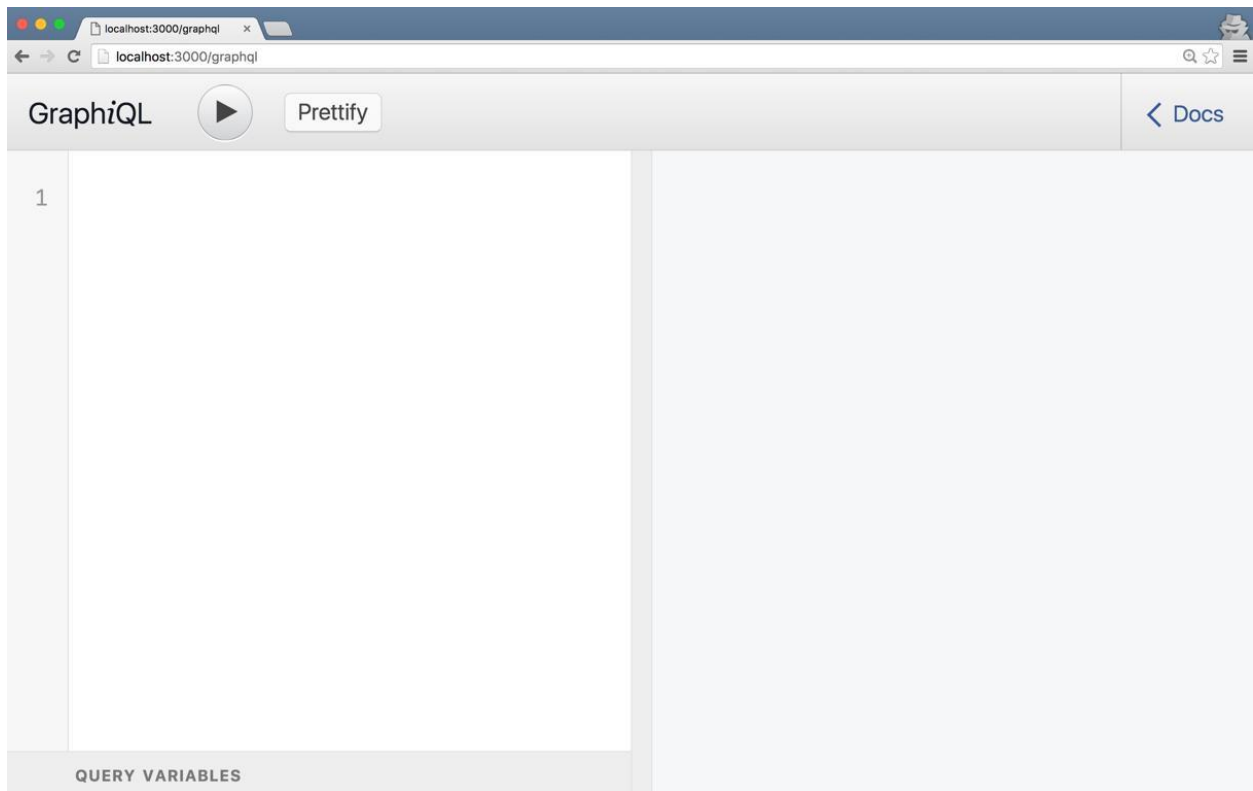
```
GraphiQL ▶ Prettify
```

```
1 {
2   person(personID: 4) {
3     name
4     birthYear
5     homeworld {
6       name
7     }
8     filmConnection {
9       films {
10        title
11      }
12    }
13  }
14 }
15
```

```
{
  "data": {
    "person": {
      "name": "Darth Vader",
      "birthYear": "41.9BBY",
      "homeworld": {
        "name": "Tatooine"
      }
    },
    "filmConnection": {
      "films": [
        {
          "title": "A New Hope"
        },
        {
          "title": "The Empire Strikes Back"
        },
        {
          "title": "Return of the Jedi"
        },
        {
          "title": "Revenge of the Sith"
        }
      ]
    }
  }
}
```



```
{  
  - data: {  
    usersCount: 2  
  }  
}
```



GraphiQL



Prettify

1
2
3

```
{  
  |  
}
```

- hello
- diceRoll
- usersCount
- __schema
- __type
- String Self descriptive.

GraphiQL

localhost:3000/graphql?query=%7B%0A%20%20hello%0A%20%20diceRoll(count%3A%204)%0A%2...

GraphiQL ▶ Prettify < Docs

```
1 {
2   hello
3   diceRoll(count: 4)
4   usersCount
5 }
```

```
{
  "data": {
    "hello": "World",
    "diceRoll": [
      3,
      6,
      4,
      2
    ],
    "usersCount": 2
  }
}
```

QUERY VARIABLES

Chapter 2: The Query Language

GraphiQL ▶ Prettify < Docs

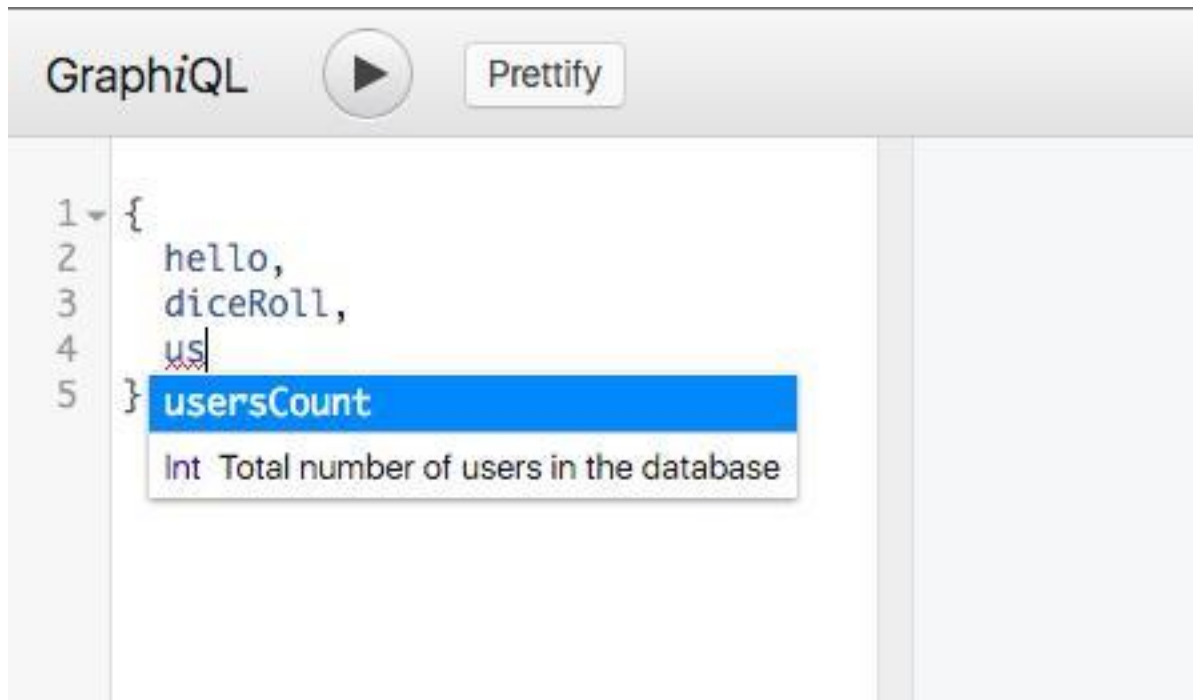
```
1 query PersonInfo($personId: ID) {
2   person(personID: $personId) {
3     name
4   }
5 }
6
```

QUERY VARIABLES

```
1 {
2   "personId": 4
3 }
```

```
{
  "data": {
    "person": {
      "name": "Darth Vader"
    }
  }
}
```

Chapter 3: The GraphQL Schema





```

1 query IntrospectionQuery {
2   __schema {
3     queryType { name }
4     mutationType { name }
5     subscriptionType { name }
6     types {
7       ...FullType
8     }
9     directives {
10      name
11      description
12      locations
13      args {
14        ...InputValue
15      }
16    }
17  }
18 }
19 fragment FullType on __Type {
20   kind
21   name
22   description
23   fields(includeDeprecated: true) {↔}
24 }
25 inputFields {
26   ...InputValue
27 }
28 interfaces {
29   ...TypeRef
30 }
31 enumValues(includeDeprecated: true) {↔}
32 possibleTypes {
33   ...TypeRef
34 }
35 }
36 fragment InputValue on __InputValue {↔}
37 fragment TypeRef on __Type {↔}

```

QUERY VARIABLES

```

{
  "data": {
    "__schema": {
      "queryType": {
        "name": "RootQuery"
      },
      "mutationType": null,
      "subscriptionType": null,
      "types": [
        {
          "kind": "OBJECT",
          "name": "RootQuery",
          "description": null,
          "fields": [
            {
              "name": "hello",
              "description": null,
              "args": [],
              "type": {
                "kind": "SCALAR",
                "name": "String",
                "ofType": null
              },
              "isDeprecated": false,
              "deprecationReason": null
            },
            {
              "name": "diceRoll",
              "description": "**Simulate** a dice roll determined by count",
              "args": [
                {
                  "name": "count",
                  "description": null,
                  "type": {
                    "kind": "SCALAR",
                    "name": "Int",
                    "ofType": null
                  },
                  "defaultValue": null
                }
              ],
              "type": {
                "kind": "LIST",
                "name": null,
                "ofType": {

```



```

1 query {
2   lastQuote
3 }

```

QUERY VARIABLES

```

{
  "data": {
    "lastQuote": "If opportunity doesn't knock, build a door."
  }
}

```

GraphiQL



Prettify

< Docs

```
1 mutation {  
2   addQuote(  
3     body: "Try to be a rainbow in someone's cloud."  
4   )  
5 }  
6
```

```
{  
  "data": {  
    "addQuote": "Try to be a rainbow in someone's cloud."  
  }  
}
```

QUERY VARIABLES

GraphiQL



Prettify

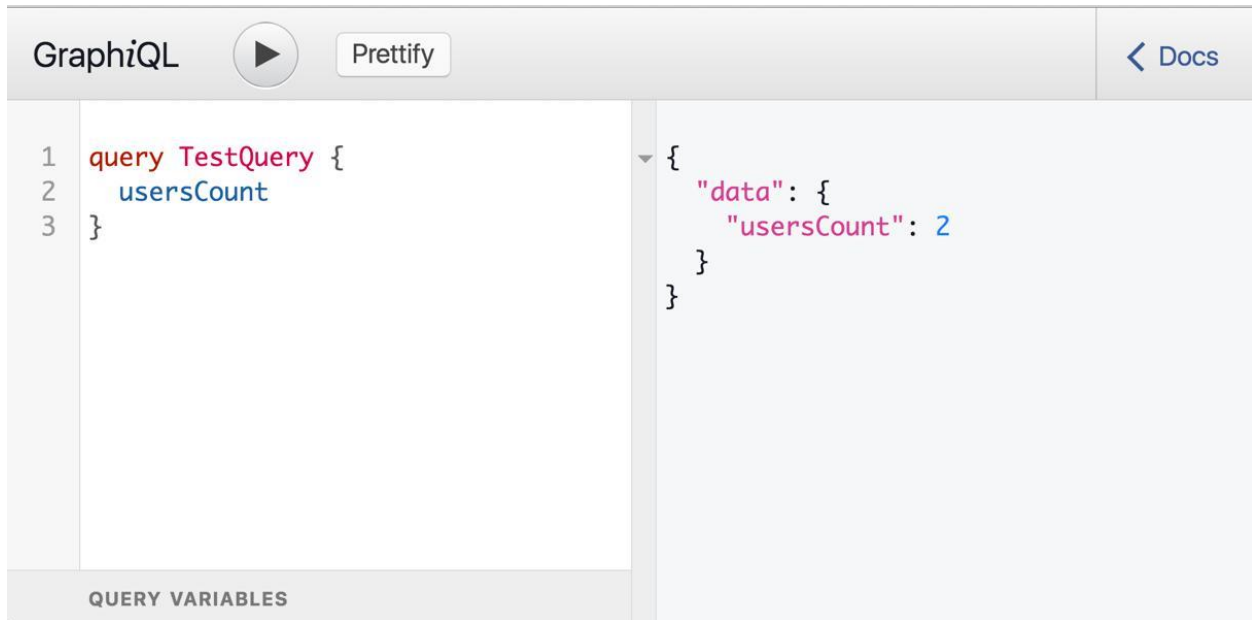
< Docs

```
1 query afterMutation {  
2   lastQuote  
3 }
```

```
{  
  "data": {  
    "lastQuote": "Try to be a rainbow in someone's cloud."  
  }  
}
```

QUERY VARIABLES

Chapter 4: Configuring React Applications to Use Relay



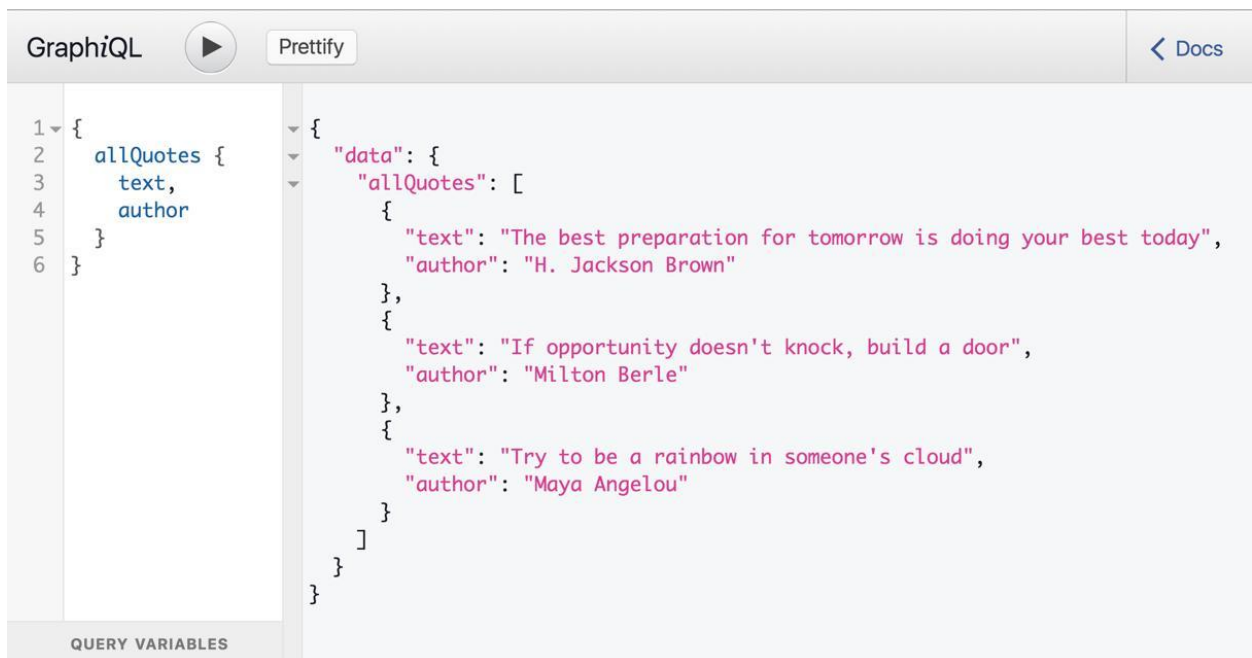
The screenshot shows the GraphQL IDE interface. At the top, there is a header with the text "GraphiQL", a play button icon, a "Prettify" button, and a "Docs" link. The main area is split into two panes. The left pane contains a query:

```
1 query TestQuery {  
2   usersCount  
3 }
```

The right pane shows the JSON response:

```
{  
  "data": {  
    "usersCount": 2  
  }  
}
```

At the bottom of the interface, there is a section labeled "QUERY VARIABLES" which is currently empty.



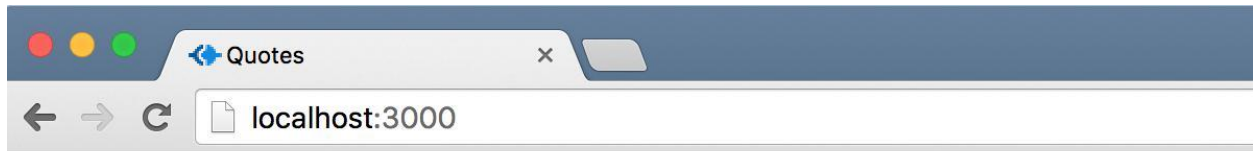
The screenshot shows the GraphQL IDE interface. At the top, there is a header with the text "GraphiQL", a play button icon, a "Prettify" button, and a "Docs" link. The main area is split into two panes. The left pane contains a query:

```
1 {  
2   allQuotes {  
3     text,  
4     author  
5   }  
6 }
```

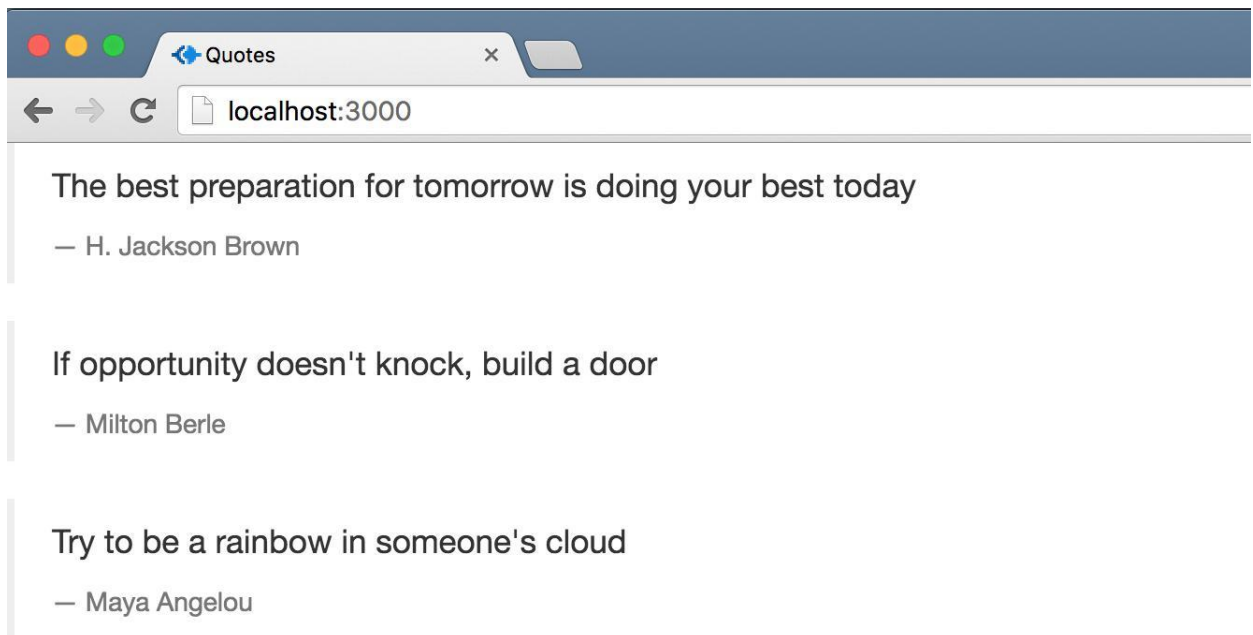
The right pane shows the JSON response:

```
{  
  "data": {  
    "allQuotes": [  
      {  
        "text": "The best preparation for tomorrow is doing your best today",  
        "author": "H. Jackson Brown"  
      },  
      {  
        "text": "If opportunity doesn't knock, build a door",  
        "author": "Milton Berle"  
      },  
      {  
        "text": "Try to be a rainbow in someone's cloud",  
        "author": "Maya Angelou"  
      }  
    ]  
  }  
}
```

At the bottom of the interface, there is a section labeled "QUERY VARIABLES" which is currently empty.



Hello World



The best preparation for tomorrow is doing your best today

— H. Jackson Brown

If opportunity doesn't knock, build a door

— Milton Berle

Try to be a rainbow in someone's cloud

— Maya Angelou

Elements Console Sources Network Timeline Profiles Resources Security Audits React

Elements Relay

Trace React Updates Highlight Search Use Regular Expressions

```
<RelayRootContainer Component=ContainerConstructor() route=AppRoute{...}>
  <RelayRenderer Container=ContainerConstructor() queryConfig=AppRoute{...} environ...
    <RelayReadyStateRenderer Container=ContainerConstructor() environment=Relay...
      <StaticContainer shouldUpdate=true>
        <Relay(QuotesLibrary)>
          <QuotesLibrary ref="component" relay={applyUpdate: bound applyUpdate(),...
            <div className="quotes-list">
              <Relay(Quote) key="5792d98d6f0d8c2df95167c4" quote={id: "5792d98d6f0...
                <Quote ref="component" quote={id: "5792d98d6f0d8c2df95167c4", text...
                  <blockquote>
                    <p>The best preparation for tomorrow is doing your best today</p>
                    <footer>H. Jackson Brown</footer>
                  </blockquote>
                </Quote>
              </Relay(Quote)>
              <Relay(Quote) key="5792d98d6f0d8c2df95167c5" quote={id: "5792d98d6f0...
              <Relay(Quote) key="5792d98d6f0d8c2df95167c6" quote={id: "5792d98d6f0...
            </div>
          </QuotesLibrary>
        </Relay(QuotesLibrary)>
      </StaticContainer>
    </RelayReadyStateRenderer>
  </RelayRenderer>
</RelayRootContainer>
```

<QuotesLibrary> (\$r in the console)

Ref "component"

Props

- ▶ relay: {...}

State

- ▼ allQuotes: Array[3]
 - ▶ 0: {...}
 - ▶ 1: {...}
 - ▶ 2: {...}

Chapter 5: Making GraphQL Queries Relay-Compliant

```
GraphiQL ▶ Prettify < Docs
```

```
1 query IntrospectionQuery {
2   __schema {
3     queryType {
4       name
5     }
6     mutationType {
7       name
8     }
9     subscriptionType {
10      name
11    }
12    types {
13      ...FullType
14    }
15    directives {
16      name
17      description
18      locations
19      args {
20        ...InputValue
21      }
22    }
23  }
24 }
25
26 fragment FullType on __Type { ... }
27
28 fragment InputValue on __InputValue { ... }
29
30 fragment TypeRef on __Type { ... }
31 }
```

```
{
  "data": {
    "__schema": {
      "queryType": {
        "name": "RootQuery"
      },
      "mutationType": null,
      "subscriptionType": null,
      "types": [
        {
          "kind": "OBJECT",
          "name": "RootQuery",
          "description": null,
          "fields": [
            {
              "name": "allQuotes",
              "description": "A list of the quotes in the database",
              "args": [],
              "type": {
                "kind": "LIST",
                "name": null,
                "ofType": {
                  "kind": "OBJECT",
                  "name": "Quote",
                  "ofType": null
                }
              }
            }
          ],
          "isDeprecated": false,
          "deprecationReason": null
        }
      ],
      "inputFields": null,
      "interfaces": [],
      "enumValues": null,
      "scalarTypes": null
    }
  }
}
```

```
Console ×
```

```
bundle.js:146
Object {children: Array[3], fieldName: "allQuotes", kind: "Query", metadata: Object, name: "AllQuotes"...}
  children: Array[3]
    0: Object
      fieldName: "id"
      kind: "Field"
      metadata: Object
      type: "String"
      __proto__: Object
    1: Object
    2: Object
      length: 3
      __proto__: Array[0]
      fieldName: "allQuotes"
      kind: "Query"
      metadata: Object
      name: "AllQuotes"
      type: "Quote"
      __proto__: Object
```

GraphiQL ▶ Prettify < Docs

```
1 {
2   allQuotes {
3     text,
4     author
5   }
6 }
```

```
{
  "data": {
    "allQuotes": [
      {
        "text": "The best preparation for tomorrow is doing your best today",
        "author": "H. Jackson Brown"
      },
      {
        "text": "If opportunity doesn't knock, build a door",
        "author": "Milton Berle"
      },
      {
        "text": "Try to be a rainbow in someone's cloud",
        "author": "Maya Angelou"
      }
    ]
  }
}
```

QUERY VARIABLES

GraphiQL ▶ Prettify < Docs

```
1 {
2   quotesLibrary {
3     allQuotes {
4       text
5       author
6     }
7   }
8 }
```

```
{
  "data": {
    "quotesLibrary": {
      "allQuotes": [
        {
          "text": "The best preparation for tomorrow is doing your best today",
          "author": "H. Jackson Brown"
        },
        {
          "text": "If opportunity doesn't knock, build a door",
          "author": "Milton Berle"
        },
        {
          "text": "Try to be a rainbow in someone's cloud",
          "author": "Maya Angelou"
        }
      ]
    }
  }
}
```

QUERY VARIABLES

The best preparation for tomorrow is doing your best today

— H. Jackson Brown

If opportunity doesn't knock, build a door

— Milton Berle

Try to be a rainbow in someone's cloud

— Maya Angelou

5 requests | 1.5 KB transfer...

```

query: "query QuotesLibrary { quotesLibrary { ...F1 } } fra
variables: {}

```

GraphQL ▶ Prettify < Docs

```

1 query QuotesLibrary {
2   quotesLibrary {
3     ...F1
4   }
5 }
6
7 fragment F0 on Quote {
8   text
9   author
10  id
11 }
12
13 fragment F1 on QuotesLibrary {
14   allQuotes {
15     id
16     ...F0
17   }
18 }
19

```

```

{
  "data": {
    "quotesLibrary": {
      "allQuotes": [
        {
          "id": "574d12b94ec08043094840a2",
          "text": "The best preparation for tomorrow is doing
your best today",
          "author": "H. Jackson Brown"
        },
        {
          "id": "574d12b94ec08043094840a3",
          "text": "If opportunity doesn't knock, build a door",
          "author": "Milton Berle"
        },
        {
          "id": "574d12b94ec08043094840a4",
          "text": "Try to be a rainbow in someone's cloud",
          "author": "Maya Angelou"
        }
      ]
    }
  }
}

```

QUERY VARIABLES



```

1 {
2   quotesLibrary {
3     quotesConnection(first: 2) {
4       edges {
5         node {
6           id
7           text
8           author
9         }
10      }
11    }
12  }
13 }
14

```

QUERY VARIABLES

```

{
  "data": {
    "quotesLibrary": {
      "quotesConnection": {
        "edges": [
          {
            "node": {
              "id": "574d12b94ec08043094840a2",
              "text": "The best preparation for tomorrow
is doing your best today",
              "author": "H. Jackson Brown"
            }
          },
          {
            "node": {
              "id": "574d12b94ec08043094840a3",
              "text": "If opportunity doesn't knock,
build a door",
              "author": "Milton Berle"
            }
          }
        ]
      }
    }
  }
}

```



```

1 {
2   quotesLibrary {
3     quotesConnection(first: 1) {
4       pageInfo {
5         hasNextPage
6       }
7       edges {
8         cursor
9         node {
10          id
11          text
12          author
13        }
14      }
15    }
16  }
17 }

```

QUERY VARIABLES

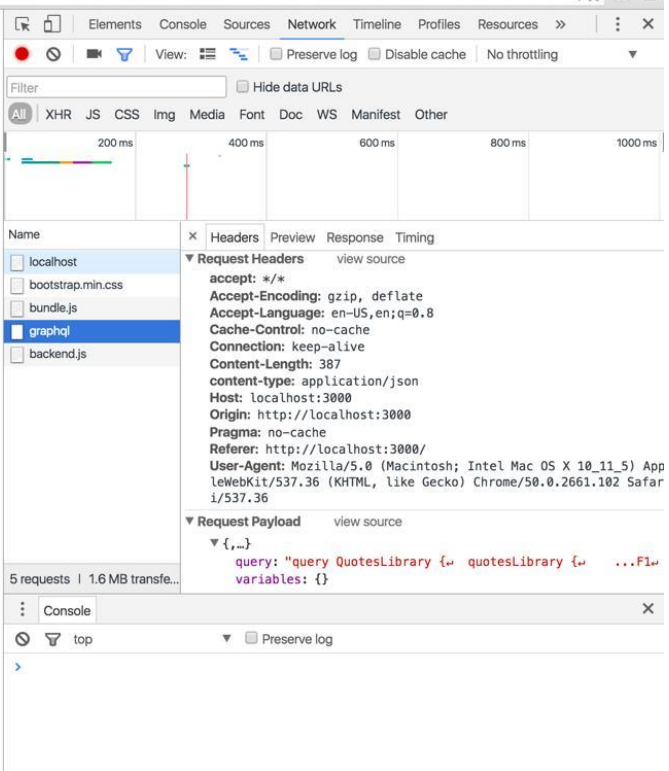
```

{
  "data": {
    "quotesLibrary": {
      "quotesConnection": {
        "pageInfo": {
          "hasNextPage": true
        },
        "edges": [
          {
            "cursor": "YXJyYXljb25uZWNOaW9uOjA=",
            "node": {
              "id": "574d12b94ec08043094840a2",
              "text": "The best preparation for tomorrow
is doing your best today",
              "author": "H. Jackson Brown"
            }
          }
        ]
      }
    }
  }
}

```

The best preparation for tomorrow is doing your best today
— H. Jackson Brown

If opportunity doesn't knock, build a door
— Milton Berle



The screenshot shows a web browser window with the URL localhost:3000. The page content consists of two quotes. The browser's developer tools are open to the Network tab, showing a list of resources. The selected resource is a GraphQL request. The request headers are expanded, showing the following details:

- accept: */*
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.8
- Cache-Control: no-cache
- Connection: keep-alive
- Content-Length: 387
- content-type: application/json
- Host: localhost:3000
- Origin: http://localhost:3000
- Pragma: no-cache
- Referer: http://localhost:3000/
- User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36

The request payload is also expanded, showing the following JSON:

```
{  
  "query": "query QuotesLibrary {  
    quotesLibrary {  
      ...Fl  
    }  
  }  
}
```


Chapter 6: Relay Variables and Object Identification

GraphQL

```
1 {
2   quotesLibrary {
3     quotesConnection(searchTerm: "best") {
4       edges {
5         node {
6           text
7         }
8       }
9     }
10  }
11 }
```

Prettify

```
{
  "data": {
    "quotesLibrary": {
      "quotesConnection": {
        "edges": [
          {
            "node": {
              "text": "The best preparation for tomorrow is doing
your best today"
            }
          }
        ]
      }
    }
  }
}
```

QUERY VARIABLES

Quotes

localhost:3000

best

The best preparation for tomorrow is doing your best today

— H. Jackson Brown

Elements Relay

- Trace React updates
- RelayRootContainer Component=ContainerConstructor() route=AppRo
- RelayRenderer Container=ContainerConstructor() queryConfig=App
- RelayReadyStateRenderer Container=ContainerConstructor() env
- StaticContainer shouldUpdate=true
- Relay(QuotesLibrary) library={ __dataID__: "client:7997966",
 <div className="quotes-library">
 <SearchForm searchAction=debounced()>
 <form className="navbar-form" role="search">

RelayRootContainer RelayRenderer RelayReadyStateRenderer
StaticContainer Relay(QuotesLibrary)

Search by Component Name

Console

top Preserve log

```
queryData: {}
library: {
  __dataID__: "client:7997966481"
}
quotesConnection: {
  __dataID__: "client:7997966486_first(100),searchTerm(best)"
  edges: Array[1]
  relayProp: {}
}
Context
  relay: RelayEnvironment[...]
  route: AppRoute[...]
```

do

The best preparation for tomorrow
is doing your best today

— H. Jackson Brown

If opportunity doesn't knock, build
a door

— Milton Berle

The screenshot shows the Network tab of a browser's developer tools. It displays two requests, both of type 'fetch' and status '200'. The first request has a size of 630 B and a time of 5 ms. The second request has a size of 454 B and a time of 5 ms. The timeline below the table shows the duration of each request. The console below the network tab is currently empty.

Name	Status	Type	Initiator	Size	Time	Timeline - Start Time
graphql	200	fetch	fetchWithRet...	630 B	5 ms	
graphql	200	fetch	fetchWithRet...	454 B	5 ms	

GraphiQL



Prettify

< Docs

```
1 {  
2   quotesLibrary {  
3     quotesConnection {  
4       edges {  
5         node {  
6           text,  
7           likesCount  
8         }  
9       }  
10    }  
11  }  
12 }
```

```
{  
  "data": {  
    "quotesLibrary": {  
      "quotesConnection": {  
        "edges": [  
          {  
            "node": {  
              "text": "The best preparation for tomorrow is  
doing your best today",  
              "likesCount": 9  
            }  
          },  
          {  
            "node": {  
              "text": "If opportunity doesn't knock, build  
a door",  
              "likesCount": 2  
            }  
          },  
          {  
            "node": {  
              "text": "Try to be a rainbow in someone's  
cloud",  
              "likesCount": 8  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```

QUERY VARIABLES

Search..

The best preparation
for tomorrow is doing
your best today

— H. Jackson Brown

If opportunity doesn't
knock, build a door

— Milton Berle

Try to be a rainbow in
someone's cloud

— Maya Angelou

The screenshot displays the Chrome DevTools Network tab. A request to `http://localhost:3000/graphql` is selected. The headers section shows the following details:

- `accept: */*`
- `Accept-Encoding: gzip, deflate`
- `Accept-Language: en-US,en;q=0.8,ar;q=0.6`
- `Connection: keep-alive`
- `Content-Length: 192`
- `content-type: application/json`
- `Host: localhost:3000`
- `Origin: http://localhost:3000`
- `Referer: http://localhost:3000/`
- `User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36`

The Request Payload section shows a GraphQL query:

```
{...}
query: "query OneQuote($id_0:ID!) {
  node(id:$id_0) {
    ...F0
  }
}
frag
variables: {id_0: "57281cc43a88dba20500f9c2"}"
```

The Console tab shows the following error messages:

- `POST http://localhost:3000/graphql 400 (Bad Request)` - `fetchWithRetries.js:54`
- `Warning: fetchWithRetries: HTTP error, retrying.` - `warning.js:44`
- `POST http://localhost:3000/graphql 400 (Bad Request)` - `fetchWithRetries.js:54`
- `Warning: fetchWithRetries: HTTP error, retrying.` - `warning.js:44`
- `POST http://localhost:3000/graphql 400 (Bad Request)` - `fetchWithRetries.js:54`
- `Warning: fetchWithRetries: Still no successful response after 3 retries, giving up.` - `RelayPendingQueryTracker.js:242`



```
1 query OneQuote($id_0: ID!) {
2   node(id: $id_0) {
3     ...F0
4   }
5 }
6
7 fragment F0 on Quote {
8   likesCount @include(if: true)
9   id
10 }
11
```

```
{
  "errors": [
    {
      "message": "Unknown type \"ID\".",
      "locations": [
        {
          "line": 1,
          "column": 23
        }
      ]
    },
    {
      "message": "Cannot query field \"node\" on type \"RootQuery\".",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ]
    }
  ]
}
```

QUERY VARIABLES



```
1 {
2   quotesLibrary {
3     id
4   }
5 }
```

```
{
  "data": {
    "quotesLibrary": {
      "id": "UXVvdGVzTGlicmFyeTo="
    }
  }
}
```

GraphiQL ▶ Prettify < Docs

```
1 {
2   quotesLibrary {
3     quotesConnection {
4       edges {
5         node {
6           id
7         }
8       }
9     }
10  }
11 }
```

```
{
  "data": {
    "quotesLibrary": {
      "quotesConnection": {
        "edges": [
          {
            "node": {
              "id": "UXVvdGU6NTcyODFjYzQzYTg4ZGJhMjA1MdBmOWMy"
            }
          },
          {
            "node": {
              "id": "UXVvdGU6NTcyODFjYzQzYTg4ZGJhMjA1MdBmOWMz"
            }
          },
          {
            "node": {
              "id": "UXVvdGU6NTcyODFjYzQzYTg4ZGJhMjA1MdBmOWM0"
            }
          }
        ]
      }
    }
  }
}
```

QUERY VARIABLES

GraphiQL ▶ Prettify < Docs

```
1 {
2   node(id: "UXVvdGU6NTcyODFjYzQzYTg4ZGJhMjA1MdBmOWMy") {
3     ... on Quote {
4       text
5       author
6     }
7   }
8 }
```

```
{
  "data": {
    "node": {
      "text": "The best preparation for tomorrow
is doing your best today",
      "author": "H. Jackson Brown"
    }
  }
}
```

Search..

The best preparation for tomorrow is doing your best today

— H. Jackson Brown

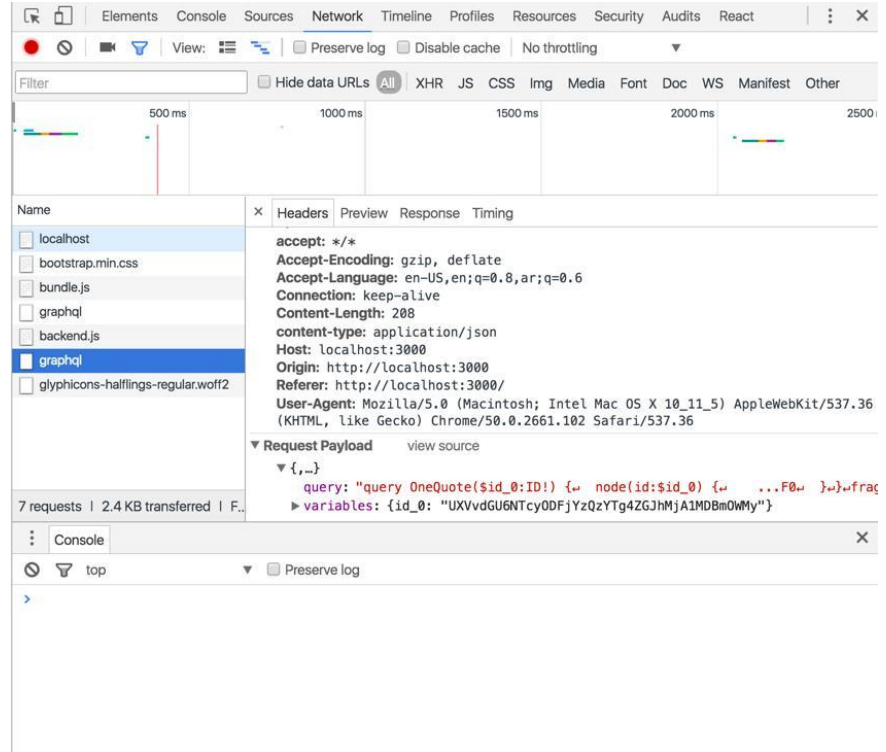
2 

If opportunity doesn't knock, build a door

— Milton Berle

Try to be a rainbow in someone's cloud

— Maya Angelou



The screenshot shows the Chrome DevTools Network tab. The top navigation bar includes Elements, Console, Sources, Network, Timeline, Profiles, Resources, Security, Audits, and React. The Network tab is active, displaying a list of requests. The selected request is from localhost:3000. The request details are shown in the right pane, including headers and a request payload. The headers include: accept: */*, Accept-Encoding: gzip, deflate, Accept-Language: en-US,en;q=0.8,ar;q=0.6, Connection: keep-alive, Content-Length: 208, content-type: application/json, Host: localhost:3000, Origin: http://localhost:3000, Referer: http://localhost:3000/, and User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36. The request payload is a GraphQL query with variables: {id_0: "UXVvdGU6NTcyODFjYzQzYTg4ZGJhMjA1MDBmOVMw"}.

7 requests | 2.4 KB transferred | F...

Console

top

Preserve log

Chapter 7: Relay Mutations

GraphiQL ▶ Prettify < Docs

```
1 mutation ThumbsUP($input: ThumbsUpMutationInput!) {
2   thumbsUp(input: $input) {
3     clientMutationId
4     quote {
5       likesCount
6     }
7   }
8 }
9
```

```
{
  "data": {
    "thumbsUp": {
      "clientMutationId": "1",
      "quote": {
        "likesCount": 1
      }
    }
  }
}
```

QUERY VARIABLES

```
1 {
2   "input": {
3     "clientMutationId": "1",
4     "quoteId": "UXVvdGU6NTc0ZDEyYjk0ZWw0DA0MzA5NDg0MGEy"
5   }
6 }
7
```

GraphiQL ▶ Prettify < Docs

```
1 {
2   node(id: "UXVvdGU6NTc0ZDEyYjk0ZWw0DA0MzA5NDg0MGEy")
3     .. on Quote {
4       text
5       likesCount
6     }
7 }
8 }
9 |
```

```
{
  "data": {
    "node": {
      "text": "The best preparation for tomorrow is doing
your best today",
      "likesCount": 3
    }
  }
}
```

Search...

The best preparation for tomorrow is doing your best today

— H. Jackson Brown

5

If opportunity doesn't knock, build a door

— Milton Berle

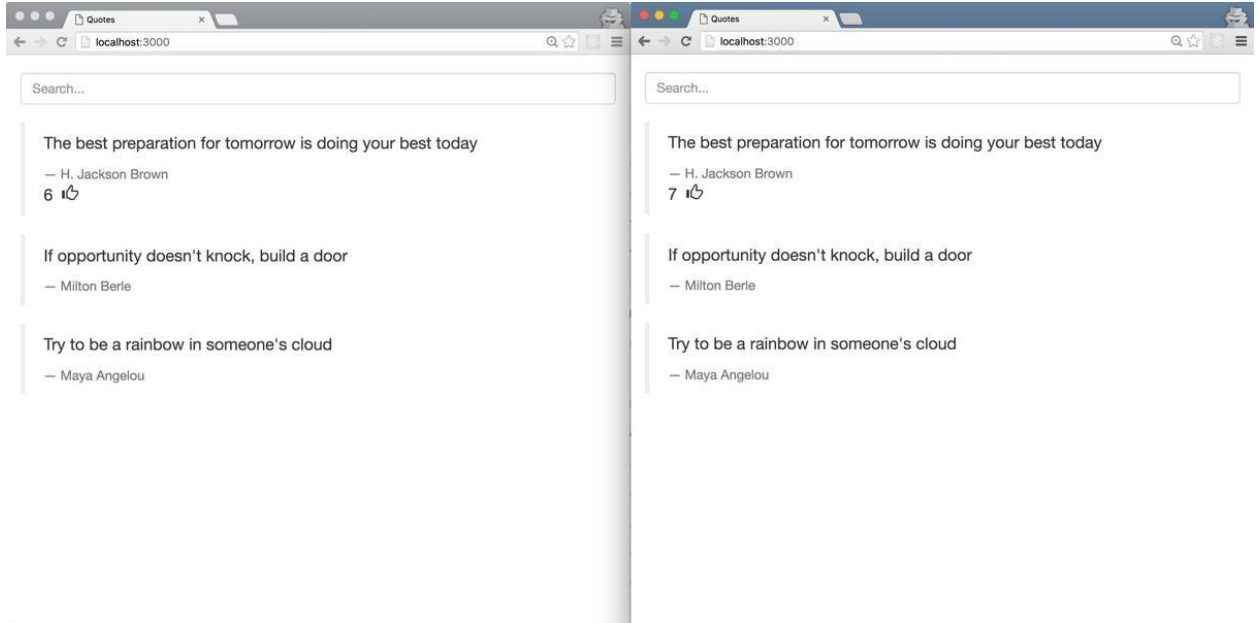
Try to be a rainbow in someone's cloud

— Maya Angelou

The screenshot shows the Chrome DevTools interface. The Network tab is selected, displaying a request to localhost. The request headers include 'Accept-Encoding: gzip, deflate', 'Accept-Language: en-US,en;q=0.8', and 'Content-Type: application/json'. The request payload is a GraphQL mutation: `mutation Thumb($input: ThumbUpMutationInput!) { thumbUp(input: $input) { clientMutationId, ...F3 } }`. The Console tab shows the response, which is a JSON object containing the mutation result and relay metadata. The response includes fields like `__typename`, `clientMutationId`, and `likesCount`.

The screenshot shows a web browser window with the URL localhost:3000. The page has a search bar at the top. Below it, there are three quote cards. The first card contains the quote 'The best preparation for tomorrow is doing your best today' by H. Jackson Brown, with a '5' and a share icon. The second card contains 'If opportunity doesn't knock, build a door' by Milton Berle. The third card contains 'Try to be a rainbow in someone's cloud' by Maya Angelou.

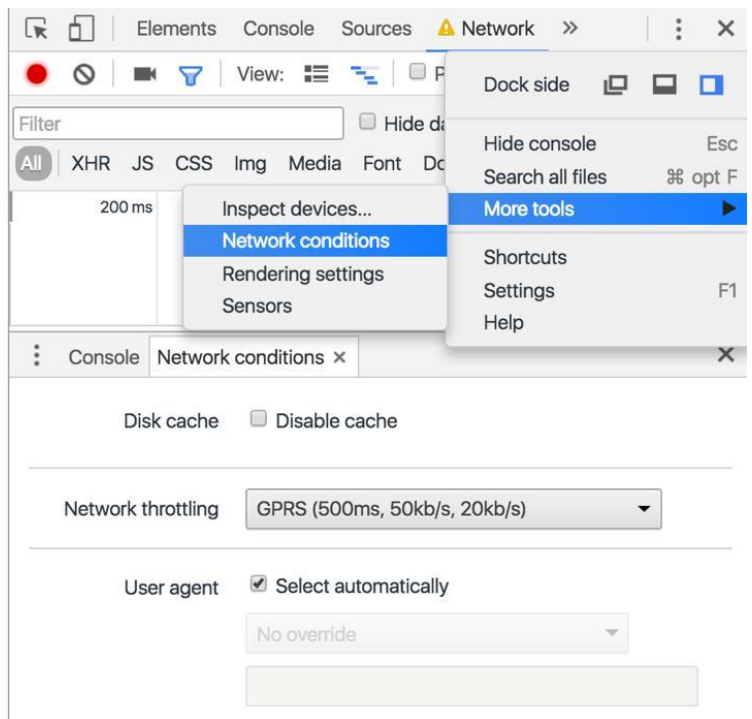
This screenshot is identical to the previous one, showing the 'Quotes' page on localhost:3000 with the same three quote cards and search bar.



Search...

The best preparation for tomorrow is doing your best today
— H. Jackson Brown
7

If opportunity doesn't knock, build a door
— Milton Berle



Chapter 8: Deploying to the Cloud

The screenshot displays the AWS Management Console interface. At the top, the browser address bar shows the URL: `https://us-west-2.console.aws.amazon.com/console/home?nc2=h_m_mc®ion=us-west-2#`. The console header includes the AWS logo, navigation menus for 'Services' and 'Edit', and user information for 'Samer Buna' in the 'Oregon' region. The main content area is titled 'Amazon Web Services' and is organized into several columns of service cards, each with an icon and a brief description. The categories include:

- Compute:** EC2 (Virtual Servers in the Cloud), EC2 Container Service (Run and Manage Docker Containers), Elastic Beanstalk (Run and Manage Web Apps), Lambda (Run Code in Response to Events).
- Storage & Content Delivery:** S3 (Scalable Storage in the Cloud), CloudFront (Global Content Delivery Network), Elastic File System (Fully Managed File System for EC2), Glacier (Archive Storage in the Cloud), Snowball (Large Scale Data Transport), Storage Gateway (Hybrid Storage Integration).
- Database:** RDS (Managed Relational Database Service), DynamoDB (Managed NoSQL Database), ElastiCache (In-Memory Cache), Redshift (Fast, Simple, Cost-Effective Data Warehousing), DMS (Managed Database Migration Service).
- Networking:** VPC (Isolated Cloud Resources), Direct Connect (Dedicated Network Connection to AWS), Route 53 (Scalable DNS and Domain Name Registration).
- Developer Tools:** CodeCommit (Store Code in Private Git Repositories), CodeDeploy (Automate Code Deployments), CodePipeline (Release Software using Continuous Delivery).
- Management Tools:** CloudWatch (Monitor Resources and Applications), CloudFormation (Create and Manage Resources with Templates), CloudTrail (Track User Activity and API Usage), Config (Track Resource Inventory and Changes), OpsWorks (Automate Operations with Chef), Service Catalog (Create and Use Standardized Products), Trusted Advisor (Optimize Performance and Security).
- Security & Identity:** Identity & Access Management (Manage User Access and Encryption Keys), Directory Service (Host and Manage Active Directory), Inspector (Analyze Application Security), WAF (Filter Malicious Web Traffic), Certificate Manager (Provision, Manage, and Deploy SSL/TLS Certificates).
- Analytics:** EMR (Managed Hadoop Framework), Data Pipeline (Orchestration for Data-Driven Workflows), Elasticsearch Service (Run and Scale Elasticsearch Clusters), Kinesis (Work with Real-Time Streaming Data), Machine Learning.
- Internet of Things:** AWS IoT (Connect Devices to the Cloud).
- Game Development:** GameLift (Deploy and Scale Session-based Multiplayer Games).
- Mobile Services:** Mobile Hub (Build, Test, and Monitor Mobile Apps), Cognito (User Identity and App Data Synchronization), Device Farm (Test Android, iOS, and Web Apps on Real Devices in the Cloud), Mobile Analytics (Collect, View and Export App Analytics), SNS (Push Notification Service).
- Application Services:** API Gateway (Build, Deploy and Manage APIs), AppStream (Low Latency Application Streaming), CloudSearch (Managed Search Service), Elastic Transcoder (Easy-to-Use Scalable Media Transcoding), SES (Email Sending and Receiving Service), SQS (Message Queue Service), SWF (Workflow Service for Coordinating Application Components).
- Enterprise Applications:** WorkSpaces (Desktops in the Cloud), WorkDocs (Secure Enterprise Storage and Sharing Service), WorkMail (Secure Email and Calendaring Service).

On the right side of the console, there is a 'Resource Groups' section with a 'Learn more' link. Below it, a 'Create a Group' button and a 'Tag Editor' button are visible. The 'Additional Resources' section includes links for 'Getting Started', 'AWS Console Mobile App', 'AWS Marketplace', and 'AWS re:Invent Announcements'. At the bottom of this section, the 'Service Health' dashboard is shown, indicating that 'All services operating normally' as of 'Jun 01 2016 20:01:01 GMT-0700'.

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Spot Requests
 - Reserved Instances
 - Scheduled Instances
 - Dedicated Hosts
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- LOAD BALANCING
 - Load Balancers
- AUTO SCALING
 - Launch Configurations
 - Auto Scaling Groups
- COMMANDS
 - Command History
 - Documents

Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Build and run distributed, fault-tolerant applications in the cloud with Amazon Simple Workflow Service.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

[Launch Instance](#)

Note: Your instances will launch in the US West (Oregon) region

Service Health

Service Status:

- US West (Oregon): This service is operating normally

Availability Zone Status:

- us-west-2a: Availability zone is operating normally
- us-west-2b: Availability zone is operating normally
- us-west-2c: Availability zone is operating normally

[Service Health Dashboard](#)

Scheduled Events

US West (Oregon):

No events

Account Attributes

Supported Platforms

VPC

Default VPC

vpc-7b08a310

[Resource ID length management](#)

Additional Information

- [Getting Started Guide](#)
- [Documentation](#)
- [All EC2 Resources](#)
- [Forums](#)
- [Pricing](#)
- [Contact Us](#)

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the [EC2 Launch Wizard](#). Or try these popular AMIs:

- Tableau Server (10 users)**
Provided by Tableau
Rating ★★★★★
Pay by the hour for Tableau software and AWS usage
[View all Business Intelligence](#)
- SAP HANA One 244GB**
Provided by SAP Inc (CAE)
Rating ★★★★★
Pay by the hour for SAP HANA One 244GB software and AWS usage
[View all Business Intelligence](#)

- 1. Choose AMI
- 2. Choose Instance Type
- 3. Configure Instance
- 4. Add Storage
- 5. Tag Instance
- 6. Configure Security Group
- 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.
 Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9abee4fb
 Free tier eligible
 Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-1
 Description: launch-wizard-1 created 2016-06-01T20:21:53.679-07:00

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

Instance Details [Edit instance details](#)

Storage [Edit storage](#)

[Cancel](#) [Previous](#) [Launch](#)

- 1. Choose AMI
- 2. Choose Instance Type
- 3. Configure Instance
- 4. Add Storage
- 5. Tag Instance
- 6. Configure Security Group
- 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name:
Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

EC2 Management Console x

https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LaunchInstanceWizard:

AWS Services Edit Samer Buna Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details Edit AMI

Ubuntu Server 14.04 LTS (HVM), SS...
Free tier eligible
Ubuntu Server 14.04 LTS (HVM), EBS Gen...
Root Device Type: ebs Virtualization type: hvm

Instance Type Edit instance type

Instance Type	ECUs	vCPU
t2.micro	Variable	1

Security Groups Edit security groups

Security group name	Description
launch-wizard-1	launch-wizard-1 cr

Type SSH HTTP Edit instance details

Instance Details Edit storage

Storage Edit tags

Tags Cancel Previous Launch

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name
graphql-project-server Download Key Pair

You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Cancel Launch Instances

Feedback English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

graphql-project-server.pem Show All

EC2 Management Console x

https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:sort=instanceld

AWS Services Edit Samer Buna Oregon Support

EC2 Dashboard Launch Instance Connect Actions

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances


Scheduled Instances

Dedicated Hosts

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-4fad6c94	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-191-153-32.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Personal apps | Heroku x

Heroku, Inc. [US] https://dashboard.heroku.com/apps

Dashboard Filter personal apps and pipelines +

samer@agilelabs.com

FAVORITES

★ Favorite any app to pin it here in the sidebar

Personal apps

Glimpse the future
Check out the new version of the Heroku Dashboard
Join the preview

You currently have no apps
To get started, click the button below and create a new app.
Create New App

heroku.com/home Blog Careers Privacy Policy Support

pacific-reaches-64379 - R x

Heroku, Inc. [US] https://dashboard.heroku.com/apps/pacific-reaches-64379/resources

Dashboard pacific-reaches-64379

samer@agilelabs.com

FAVORITES

★ Favorite any app to pin it here in the sidebar

Personal apps

Glimpse the future
Check out the new version of the Heroku Dashboard
Join the preview

< Apps pacific-reaches-64379

Resources Deploy Metrics Activity Access Settings

Dynos

This app has no process types yet
Add a Procfile to your app in order to define its process types
Find out more

Add-ons [FIND MORE ADD-ONS](#)

Quickly add add-ons from Elements

You haven't added any add-ons yet
Add-ons provide tools and services for developing, extending and operating your app

Heroku Add-ons pacific-reaches-64379 → Resources Activity Access Settings mongolab → Docs

Collections Users Stats Backups Tools

Collections + Add collection

[None at this time]

System Collections help





[None at this time]

Add new collection ✕

Collection name*

[show advanced options](#)

Cancel Create

COMPANY SUPPORT

About mLab mLab Status

Heroku Add-ons pacific-reaches-64379 → Resources Activity Access Settings mongolab → Docs

Create document

```
1 {
2   "text": "The best preparation for tomorrow is doing your best today",
3   "author": "H. Jackson Brown"
4 }
```

[Cancel and go back](#) [Create and go back](#) [Create and continue editing](#)

Heroku Add-ons pacific-reaches-64379 → Resources Activity Access Settings mongolab → Docs

Collection: quotes [↗](#)

Documents Indexes Stats Tools

Documents Delete all documents in collection + Add document

--- Start new search ---

All Documents

Display mode: list table (edit table view)

records / page 10 [1 - 3 of 3]

```

{
  "_id": {
    "$oid": "575084aadcbaf329529d6ad"
  },
  "text": "The best preparation for tomorrow is doing your best today",
  "author": "H. Jackson Brown"
}

```

```

{
  "_id": {
    "$oid": "575084cddcbaf329529d6b6"
  },
  "text": "If opportunity doesn't knock, build a door",
  "author": "Milton Berle"
}

```

```

{
  "_id": {
    "$oid": "575084e2dcba0f329529d6bd"
  },
  "text": "Try to be a rainbow in someone's cloud",
  "author": "Maya Angelou"
}

```

records / page 10 [1 - 3 of 3]

Documents (aka Objects)

From the "Documents" tab you can browse and search for objects in this collection. All standard query constructs are supported except for `map/reduce` queries. To use `map/reduce`, use the MongoDB shell (note that temporary result collections will be viewable in mLab).

You can also add, edit, and delete individual documents from here. Bulk collection updates are not yet supported in this UI (although they are supported in the shell).

https://pacific-reaches-64379.herokuapp.com

Search...

The best preparation for tomorrow is doing your best today
— H. Jackson Brown
1

If opportunity doesn't knock, build a door
— Milton Berle

Try to be a rainbow in someone's cloud
— Maya Angelou

Elements Console Sources Network Timeline Profiles

View: Preserve log Disable cache No throttling

Filter Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Stat..	Type	Initiator	Size	Time	Timeline - Start Time
pacific-reaches-64...	200	doc...	Other	610 B	104 ...	
bootstrap.min.css	200	style...	(index):5	25.1...	122 ...	
bundle.js	200	script	(index):11	473...	1.02 s	
graphql	200	fetch	bundle.js...	932...	113 ...	
glyphicons-halfling...	200	font	Other	(fro...	1 ms	
graphql	200	fetch	bundle.js...	325...	140 ...	

6 requests | 500 KB transferred | Finish: 15.02 s | DOMContentLoaded: 1.25 s | Load: 1.25 s

Console Network conditions

top Preserve log