



Building Innovative Apps with the QuickBooks API

Diana De Rose, Reg Ouellette, Peter Lavelle, Isha Shah

Session break-up

QuickBooks Online API

chata.ai

Intuit + GraphQL

AskQB

QuickBooks Online API



Today's speaker



Diana De Rose
Software Engineer
[@derosediana](#)

Agenda

Getting started with QuickBooks API

Tools & resources

Best practices

QuickBooks API resources

All QuickBooks API resources

| Customer | Vendor | Employee | Lists | Currency | Supporting |
|----------------|----------------|---------------|----------------|------------------|--------------|
| Estimate | Purchase Order | Time Activity | Account | Company Currency | Attachable |
| Invoice | Bill | Banking | Budget | Exchange Rate | Batch |
| Payment | Bill Payment | Deposit | Class | Tax | CDC |
| Sales Receipt | Purchase | Transfer | Department | Tax Agency | Company Info |
| Refund Receipt | Vendor Credit | Accounting | Item | Tax Code | Entitlements |
| Credit Memo | | Journal Entry | Payment Method | Tax Rate | Preferences |
| | | | Term | Tax Service | Reports |

Getting started

3 Easy steps

- Create an Intuit Developer Account
- Create an app
- Generate OAuth tokens

Secure | <https://developer.intuit.com>

intuit Developer

Search API Docs & Tools Community Help Sign In Sign Up

2M+ QuickBooks users are waiting for that one perfect app

From brainstorm to release and everything in between, we'll help you get there.

[Get Started](#)

Get ready for QuickBooks Connect Sydney 2018! [Register Now.](#)

Explore our APIs



QuickBooks Online

Integrate your app with QuickBooks Online to make accounting easier



QuickBooks Payments

Take payments and record transactions in QuickBooks Online with ease



QuickBooks Desktop

The original, desktop-only, QuickBooks API

Feedback

Developer tools

- OAuth playground
- Sandbox
- API explorer
- SDKs
- Sample code
- Postman


Sandbox

Sandbox URL – <https://sandbox-quickbooks.api.intuit.com>






Manage Sandboxes

You're using 5 out of 5 allocated companies.

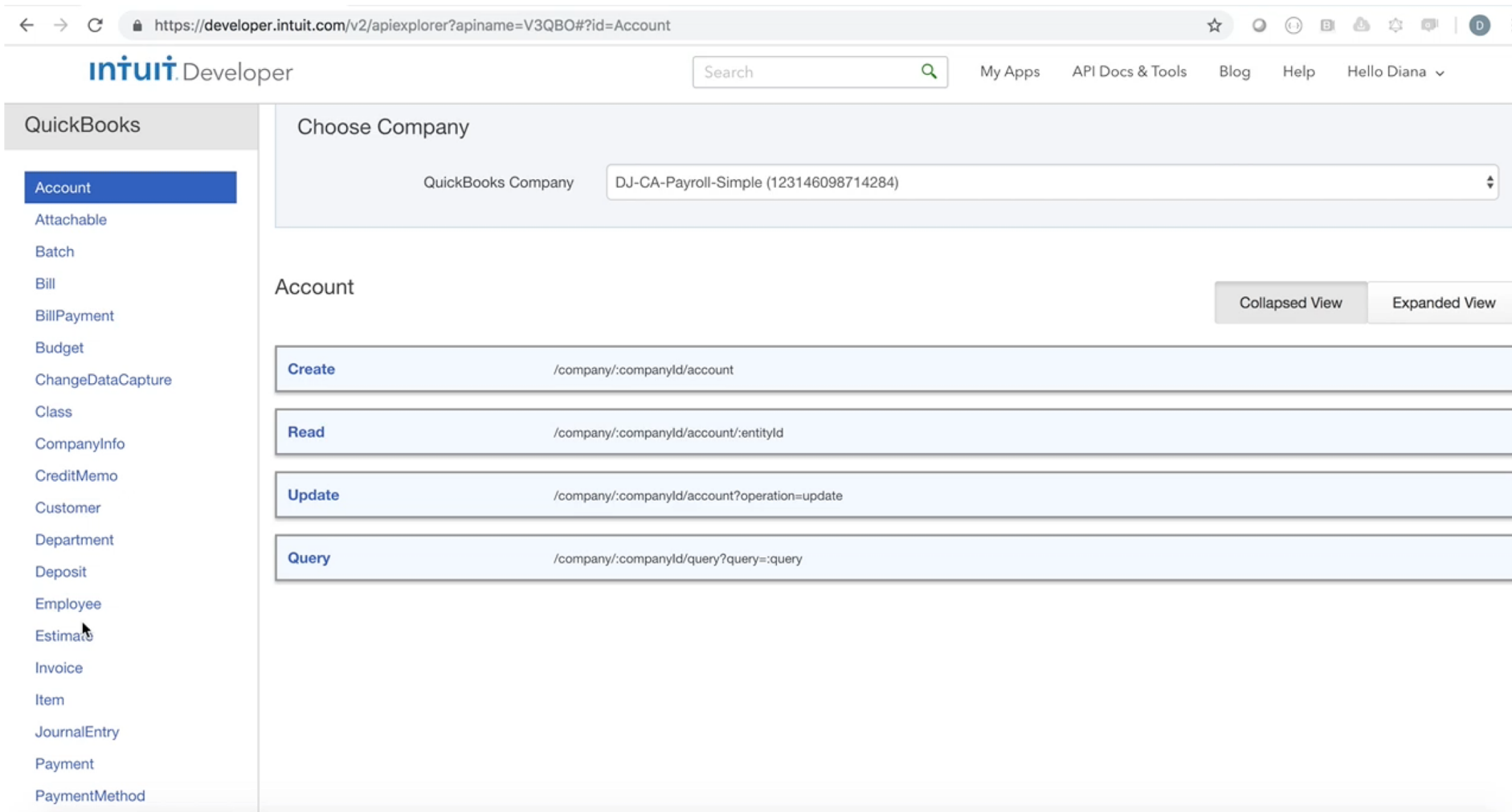
Test your app with more global QuickBooks companies

 United States

Add

| COMPANY NAME | INDUSTRY | ACTION |
|---|---------------------------|---------------------------------|
| <div><div>Sandbox Company_US_1 Company ID: 123145875112909 Payment: Enabled</div></div> | Construction Trades | Go to company ▼ |
| <div><div>Sandbox Company_AU_2 Company ID: 123145895037884 Payment: Not Supported</div></div> | Party planning services | Go to company ▼ |
| <div><div>Sandbox Company_CA_3 Company ID: 123145932194114 Payment: Not Supported</div></div> | Party planning services | Go to company ▼ |
| <div><div>Sandbox Company_UK_4 Company ID: 193514693654654 Payment: Not Supported</div></div> | Party planning services | Go to company ▼ |
| <div><div>Sandbox Company_FR_5 Company ID: 193514693712139 Payment: Not Supported</div></div> | Boulangeries commerciales | Go to company ▼ |

API Explorer



SDK

SDKs are open sourced – <https://github.com/Intuit>

Intuit SDK

Java
.NET
PHP

OAuth library

Java
.NET
PHP
Python
Node.js

Third-Party SDK

Node.js
Python
Ruby

Let the SDK do heavy-lifting for you, so you can focus on the cool features of your app

Samples

Samples are available at <https://github.com/IntuitDeveloper>

Languages

Java
.NET
PHP
Python
Node.js
Ruby
Go

CRUD

Java
.NET
PHP

Features

Concepts – Invoicing, Billing,
Inventory management etc.
Payments
Webhooks
OAuth2

Samples help you understand use cases and write code faster

Best practices

- Webhooks
- Batch
- Throttle limits
- Resiliency

Use SDKs to follow best practices

Why use Webhooks?

- Ensure data is in-sync
- Polling APIs is an inefficient way to get information
- Hollywood Principle - Don't call us, we will call you!

Why should you use Batch?

- Group several operations in a single HTTP request
- Reduces network overhead
- Optimize calls to the server and improve the scalability

Throttle limits

Adhere to throttle limits

- 10 concurrent requests per second per realmId
- 400 requests per min per realmId
- 40 batch requests/min per realmId, 10 payloads per batch request

Be Resilient, handle the unexpected

Downtime happens

- Be prepared, retry transactions instead of skipping

Manage errors gracefully

- Notify users of any errors and potential ways to fix them



CHATA.AI

Reg Ouellette

November 6, 2018

#QBConnect | WiFi: QBConnect Password not required

Today's speaker



Reg Ouellette

VP, Engineering and Integrations

[@Rego_Tweetn](https://twitter.com/Rego_Tweetn)

<https://www.linkedin.com/in/reg-ouellette/>

Agenda

What is chata.ai

Demo

High level architecture

chata.ai and v3 API

Future for chata.ai

Lessons learned

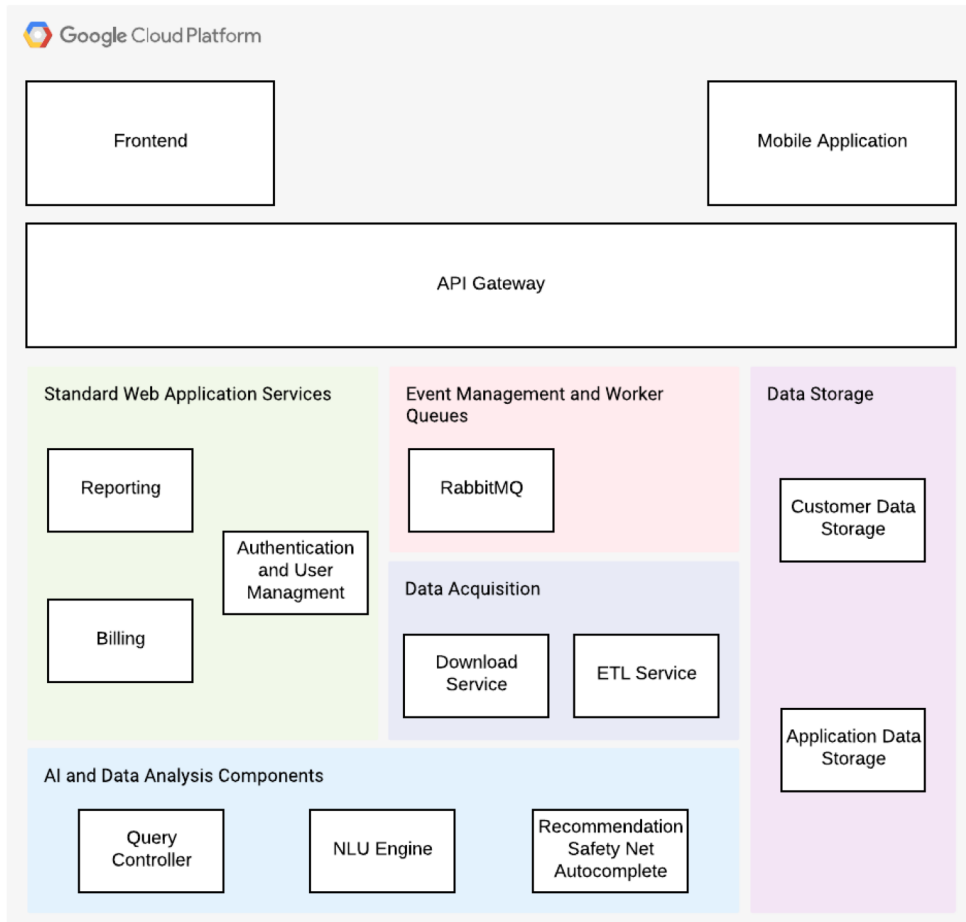


What is chata.ai

- chata.ai is a conversational analytics and reporting tool that does your data "dirty work".
- Natural Language Understanding (NLU) engine + Dynamic query builder
- Our mission is to give everyday people easy access to answers from their business data.

[Demo](#)

High level architecture



Notes

- Deployed in Google Cloud Platform.
- Microservices-based architecture.
- Kubernetes and Docker for orchestration of deployments.
- Customer data used by the application are physically segmented from each other.
- Data is encrypted at rest and in flight.

chata.ai and v3 API

V3 API categories

- Transactions resources
- Name list resources
- Supporting resources
- Report resources

Future for chata.ai

Become the "Hub" for all business data interaction



Lessons Learned

Leverage the SDKs

Prepare for scale early

- Batches for large downloads
- Prepare for throttling

Data wrangling

- Expect the unexpected

Resilience

Thanks!

@thechataHQ

Intuit + GraphQL



Today's speaker



Peter Lavelle
Software Engineer
@NerdBoots

Agenda

What is GraphQL?

Introduction to Queries and Mutations

Structure of a GraphQL Request

Building better apps with GraphQL

What is GraphQL?

Powerful features allow clients to define their own API

Query Language for APIs

GraphQL is a query language for APIs, and a runtime to fulfill those requests.

Ask for exactly what you need, and get it

Request specific fields from the API and get exactly what you requested, and nothing more.

Get many resources in one request

GraphQL queries allow you to retrieve data across many resources in a single request, and easily follow references between them.

"At its simplest, GraphQL is about asking for specific fields on objects." [graphql.org]

Queries and Mutations

Fields on objects can be requested from the server

```
{  
  company {  
    name  
  }  
}
```

```
{  
  "data": {  
    "company": {  
      "name": "Peter's Pallets"  
    }  
  }  
}
```

The shape of the query is reflected in the shape of the response, so clients know what to expect.

Queries and Mutations

Objects can also be represented by fields

```
{
  company {
    name
    transactions {
      dueDate
    }
  }
}
```

```
{
  "data": {
    "company": {
      "name": "Peter's Pallets",
      "transactions": [
        { "dueDate": "3/11/2016" },
        { "dueDate": "3/23/2016" }
      ]
    }
  }
}
```

Queries in GraphQL can traverse related objects, and access their fields.

Queries and Mutations

Arguments can be passed to Fields

```
{
  company {
    name
    transactions (type: invoice) {
      dueDate
    }
  }
}
```

```
{
  "data": {
    "company": {
      "name": "Peter's Pallets",
      "transactions": [
        { "dueDate": "12/1/2018" },
        { "dueDate": "12/4/2018" }
      ]
    }
  }
}
```

Every field and nested object in GraphQL can define its own set of arguments.

Queries and Mutations

Named operations can be beneficial to clients

```
query GetNameAndTxns {  
  company {  
    name  
    transactions (type: invoice) {  
      dueDate  
    }  
  }  
}
```

```
{  
  "data": {  
    "company": {  
      "name": "Peter's Pallets",  
      "transactions": [  
        { "dueDate": "12/1/2018" },  
        { "dueDate": "12/4/2018" }  
      ]  
    }  
  }  
}
```

Named operations are required to use multiple operations in a single request.

Queries and Mutations

Variables can pass dynamic data to arguments

```
query GetNameAndTxns ($type:String) {  
  company {  
    name  
    transactions (type: $type) {  
      dueDate  
    }  
  }  
}  
  
{  
  "type": "invoice"  
}
```

```
{  
  "data": {  
    "company": {  
      "name": "Peter's Pallets",  
      "transactions": [  
        { "dueDate": "12/1/2018" },  
        { "dueDate": "12/4/2018" }  
      ]  
    }  
  }  
}
```

Variables are highly beneficial in writing reusable, less complex code.

Queries and Mutations

Aliases allow clients to query the same fields with different arguments

```
{
  company {
    name
    invoices: transactions (type: invoice) {
      dueDate
    }
    bills: transactions (type: estimate) {
      dueDate
    }
  }
}
```

```
{
  "data": {
    "company": {
      "name": "Peter's Pallets",
      "invoices": [
        { "dueDate": "12/1/2018" }
      ],
      "bills": [
        { "dueDate": "11/19/2018" }
      ]
    }
  }
}
```

Aliases let clients rename fields to anything they want, defining their own API.

Queries and Mutations

Fragments are reusable collections of fields

```
{
  company {
    name
    invoices: transactions (type: invoice) {
      ... transactionFields
    }
    bills: transactions (type: bill) {
      ... transactionFields
    }
  }
}

fragment transactionFields on Transaction {
  dueDate
}
```

```
{
  "data": {
    "company": {
      "name": "Peter's Pallets",
      "invoices": [
        { "dueDate": "12/1/2018" }
      ],
      "bills": [
        { "dueDate": "11/19/2018" }
      ]
    }
  }
}
```

Fragments can be stored separately in your codebase, and used across GraphQL requests.

Queries and Mutations

Mutations allow modifications of server-side data

```
mutation CreateTransaction ($input: TxnInput!) {  
  createTransaction (input: $input) {  
    transaction {  
      dueDate  
    }  
  }  
}  
{  
  "input": {  
    "transaction": {  
      "dueDate": "11/30/2018",  
      "amount": 33.00  
    }  
  }  
}
```

```
{  
  "data": {  
    "createTransaction": {  
      "transaction": {  
        "dueDate": "11/30/2018"  
      }  
    }  
  }  
}
```

Multiple fields are supported in mutations, and they run in series, rather than parallel.

Structure of a GraphQL Request

HTTP Request is encoded when sent to the server

```
POST /graphql
Host: v4.api.intuit.com
```

```
{
  "query": "{\n  company {\n    name\n  }\n}\n",
  "variables": "",
  "operationName": ""
}
```

```
200 OK
```

```
{
  "data": {
    "company": {
      "name": "Peter's Pallets"
    }
  },
  "errors": {...}
}
```

GraphQL requests aren't rocket science. They are just an efficient way to request the same data.

Building better apps with GraphQL

Use the power of GraphQL to build faster and less complex apps



GraphQL benefits

As a recap:

- Eliminates over-fetching and under-fetching
- Allows clients to define the API that they need
- Fragments and variables make code reusable
- No breaking changes
- Used internally by Intuit

AskQB Demo



Today's speaker

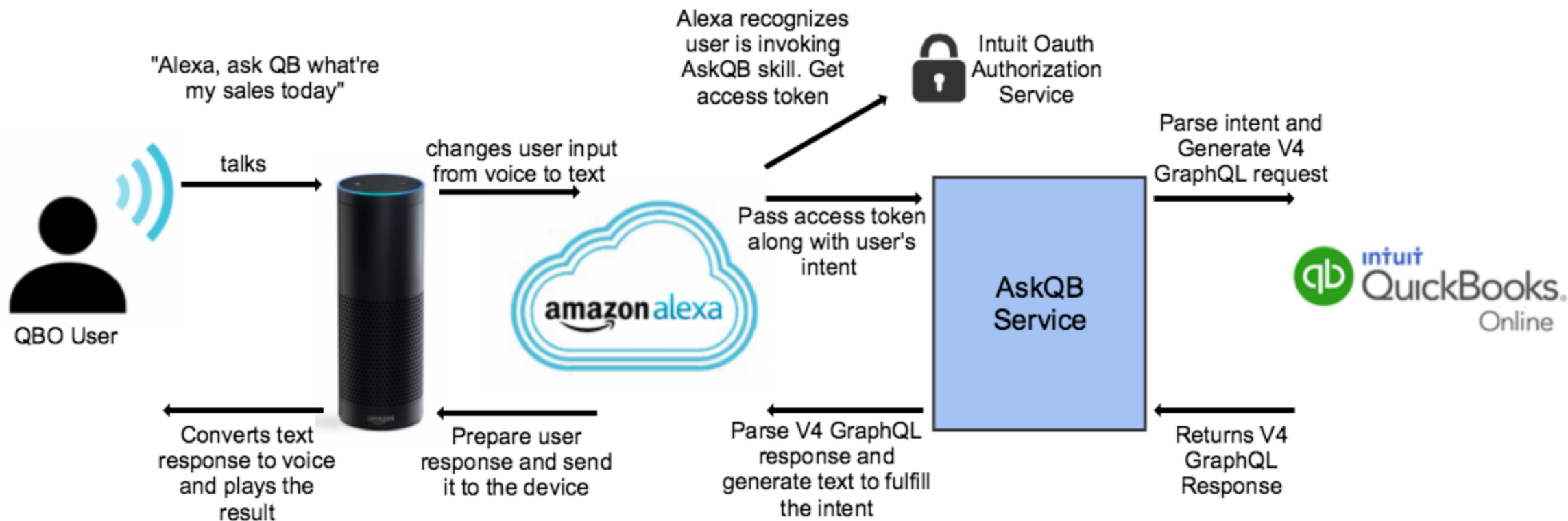


Isha Shah
Software Engineer

Accessing my Data with Voice – Any Device, anytime, anywhere



AskQB Skill diagram



Sample V4 GraphQL Request

Query all sales transactions for a given date

```
query{
  company{
    transactions(filterBy:"header.txnDate >= '2018-10-01' && header.txnDate < '2018-10-02' && type='SALE'"){
      edges{
        node{
          type
          header{
            txnDate
            txnStatus
            amount
          }
        }
      }
    }
    pageInfo{
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
  }
}
```

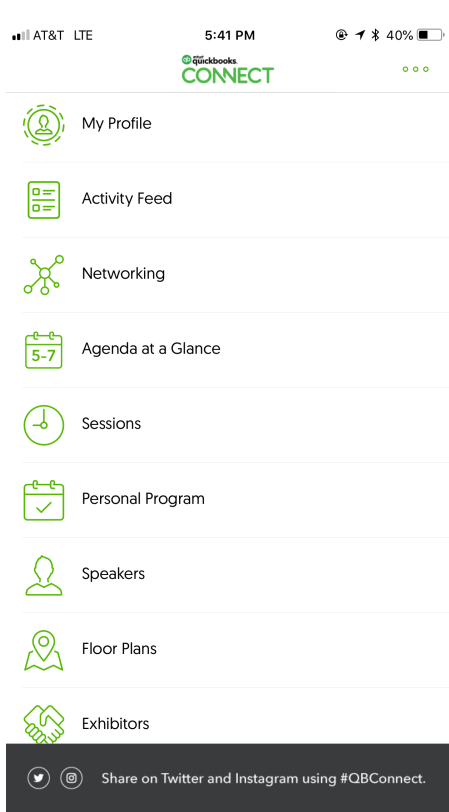
Sample V4 Response

```
{
  "data": {
    "company": {
      "transactions": {
        "edges": [
          {
            "node": {
              "header": {
                "amount": "12.00",
                "txnStatus": "PAID",
                "txnDate": "2018-10-01"
              },
              "type": "SALE"
            }
          }
        ],
        "pageInfo": {
          "hasNextPage": false,
          "hasPreviousPage": false,
          "endCursor": "c2ltcGxlLWN1cnNvcjA=",
          "startCursor": "c2ltcGxlLWN1cnNvcjA="
        }
      }
    }
  }
}
```

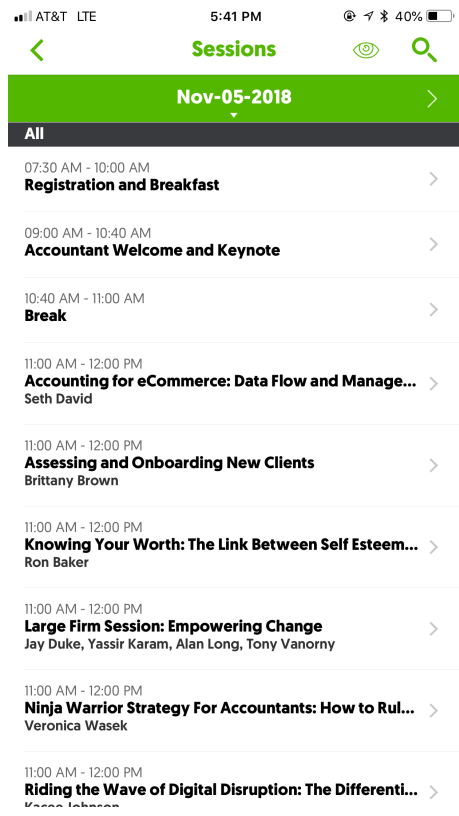
Questions?

Rate this session

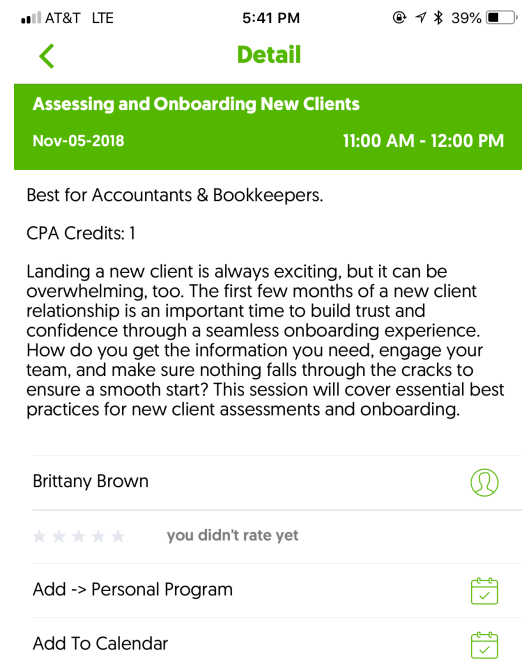
1. Select Sessions



2. Select Title



3. Add Rating



Material Download

Session slides can be found on the QuickBooks Connect agenda page

- 1) Find the session on the agenda
- 2) Select + for more information
- 3) Download PDF of slides

<https://quickbooksconnect.com/agenda/>

| | | |
|-------------------|---|---------------------|
| 11:00 am-12:00 pm | Breakout Sessions | |
| | Exhibits, Activations and Connections | + |
| | HR Essentials: 5 Steps to a Compliant Workplace | + |
| | Building Compassion: The Role We All Play in Creating Unbiased Businesses | + |
| | • session repeats | |
| | Freelance Finance: A Toolkit for the Self-Employed | • session repeats + |
| | The Fine Print: Legal Know-How for New Businesses | + |
| | Get Bossy: Develop Next-Generation Leadership Skills | + |
| | No Stone Unturned: New Funding Sources to Fuel Business Growth | + |
| | Marketing Your Business Part 1: Customer Acquisition | + |
| | Digital Savvy: Nurturing Your Online Brand | • session repeats + |



#QBConnect | WiFi: QBConnect Password not required