



Developing Web Apps with Forge Data Exchange and AWS Amplify

Jon Clark, Sr. Partner Solutions Architect, AWS
Enrico Chionna, BIM Technology Manager, AWS

September 2022



Who we are...



Enrico Chionna
BIM Technology Manager
AWS



Jon Clark
Sr. Partner Solutions Architect
AWS

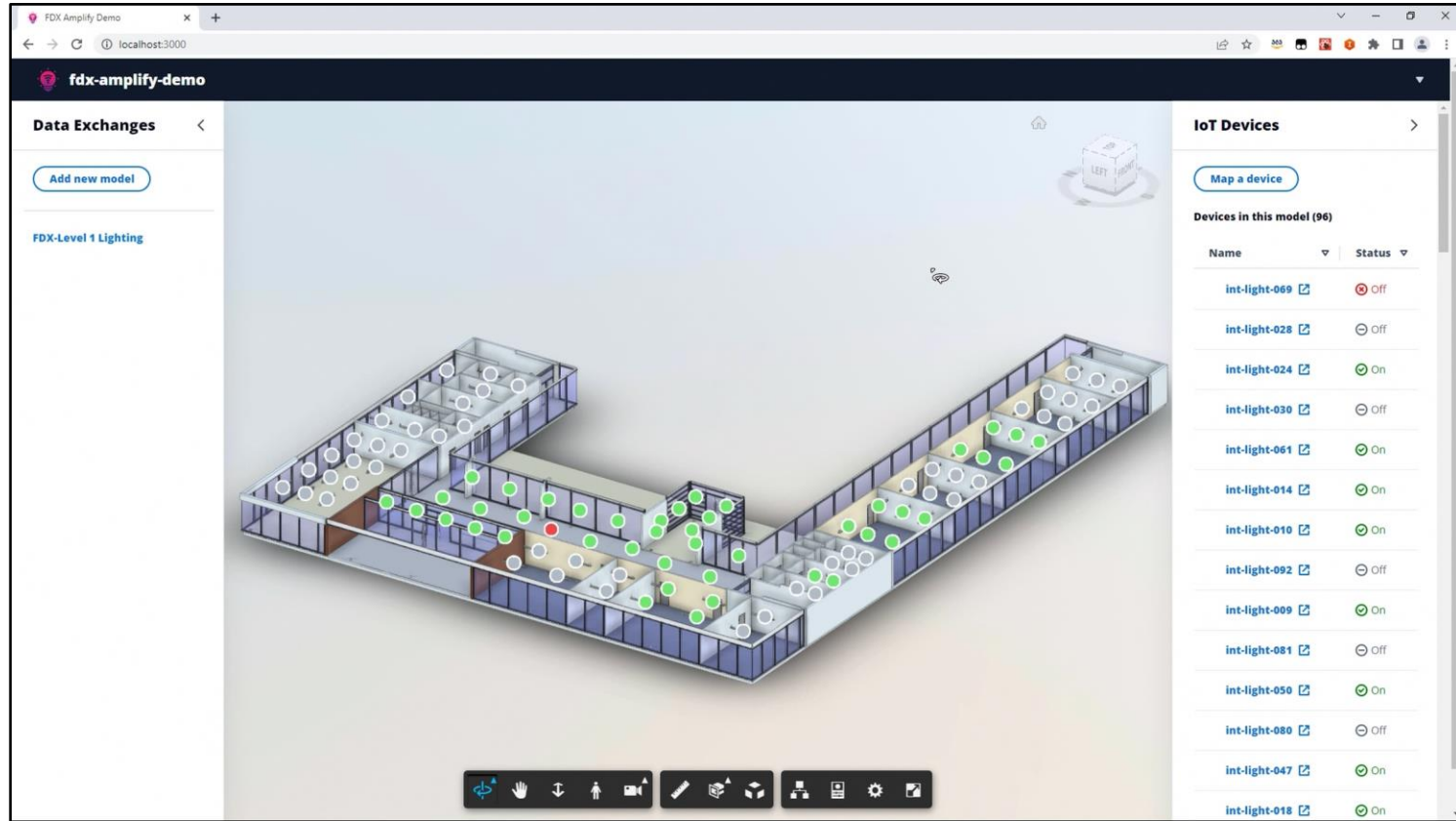
*“Developing Web Apps with **Forge Data Exchange** and **AWS Amplify**”*

Key learning objectives

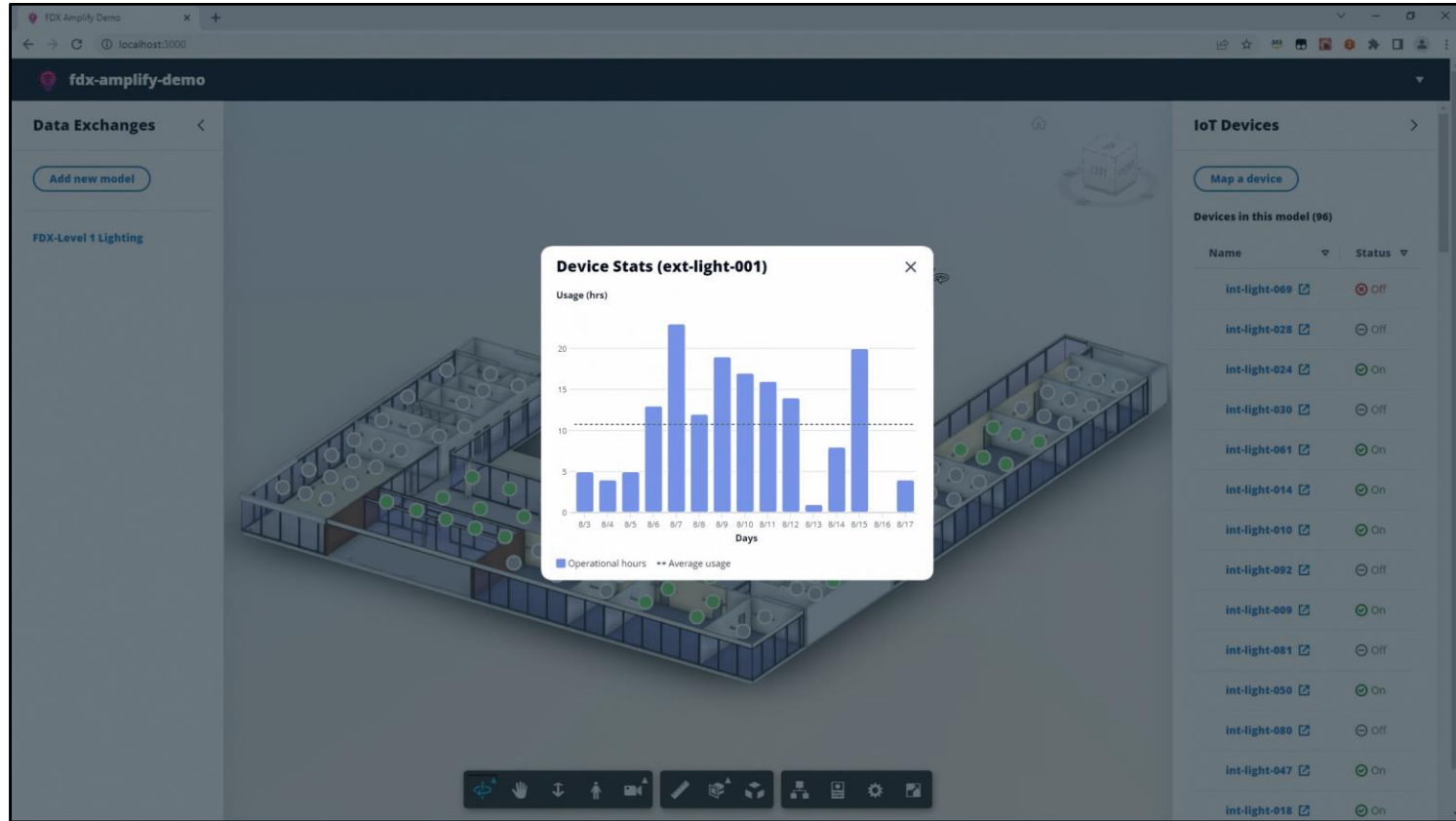
1. Understand the use of Autodesk Forge Data Exchange
2. Integrate disparate data into a single application
3. Build an application using AWS and Autodesk Forge
4. Identify new use cases for Autodesk Forge Data Exchange



What we'll be demoing



What we'll be demoing



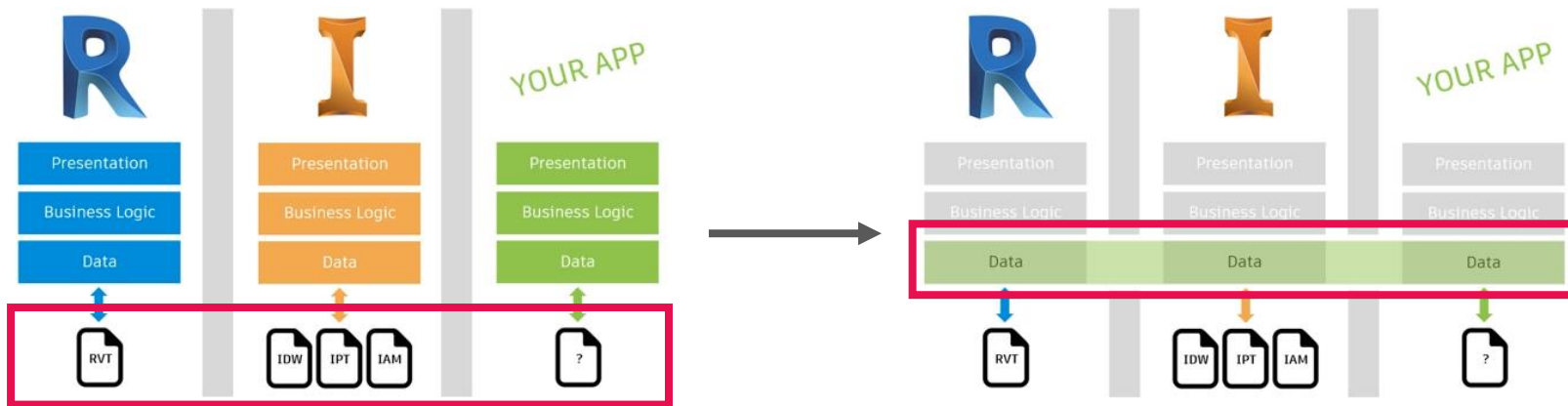
Data Exchange Overview

What is Autodesk Data Exchange?

How does Data Exchange work with Forge API?

Forge API + Data Layer

Forge is a platform of web service APIs that enable you to embed Autodesk components into your web or mobile applications.



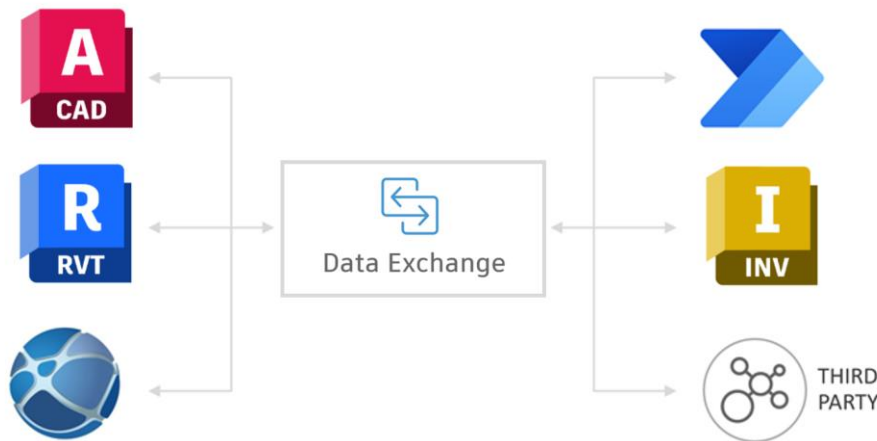
Monolithic binary files

Binary files with a **Data Layer**

Forge Data Exchange

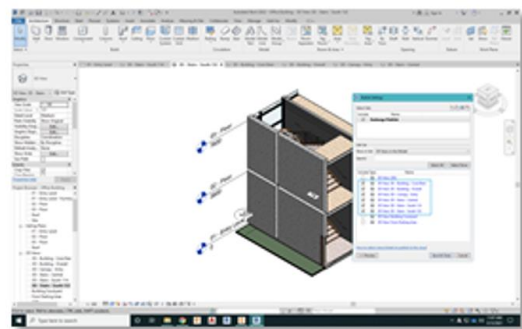
Share only the portions of your data that you need to share

- Works in Autodesk Docs, Revit 2023, and Inventor 2023
- Available in public beta on April 22, 2022
- Data Exchanges are built on the Forge Data platform



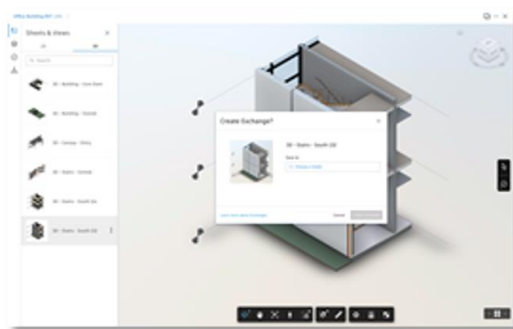
Forge Data Exchange

REVIT



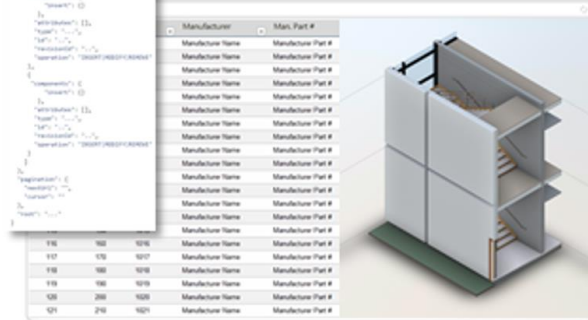
Revit user publishes a model that includes 3D views

AUTODESK DOCS



Autodesk Docs user selects a 3D view to create an exchange from

3RD PARTY APP



3rd party app uses Data Exchange Read APIs to load in property data from the exchange



AUTODESK
Construction Cloud



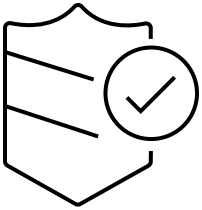
AWS Amplify

Build full-stack web and mobile apps.

What is AWS?

The most flexible and secure cloud computing environment.

Security



Security is priority #1

98 security standards
and compliance
certifications

Shared security model

Cost Optimized



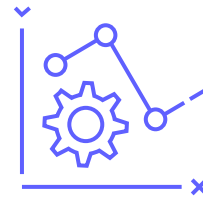
Pay only for what is
consumed

Agility



Innovate more
quickly and more
frequently

Elasticity



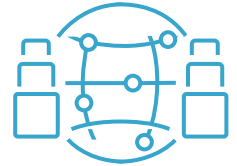
Automatically scale
up or down with
workload demands

Flexibility



200+ AWS services
supporting any cloud
workload

Go Global



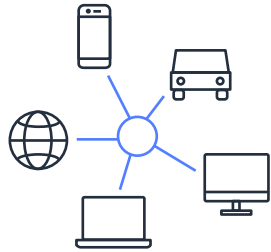
Deploy apps quickly
across 26 geos and
84 availability zones

AWS Amplify provides tools to build, ship, and scale your app

Build



Configure
application
backend



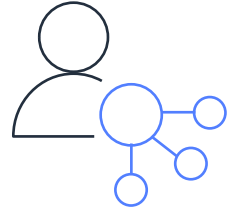
Build features and
connect frontends
to AWS Services

Ship



Deploy your app
frontend and
backend

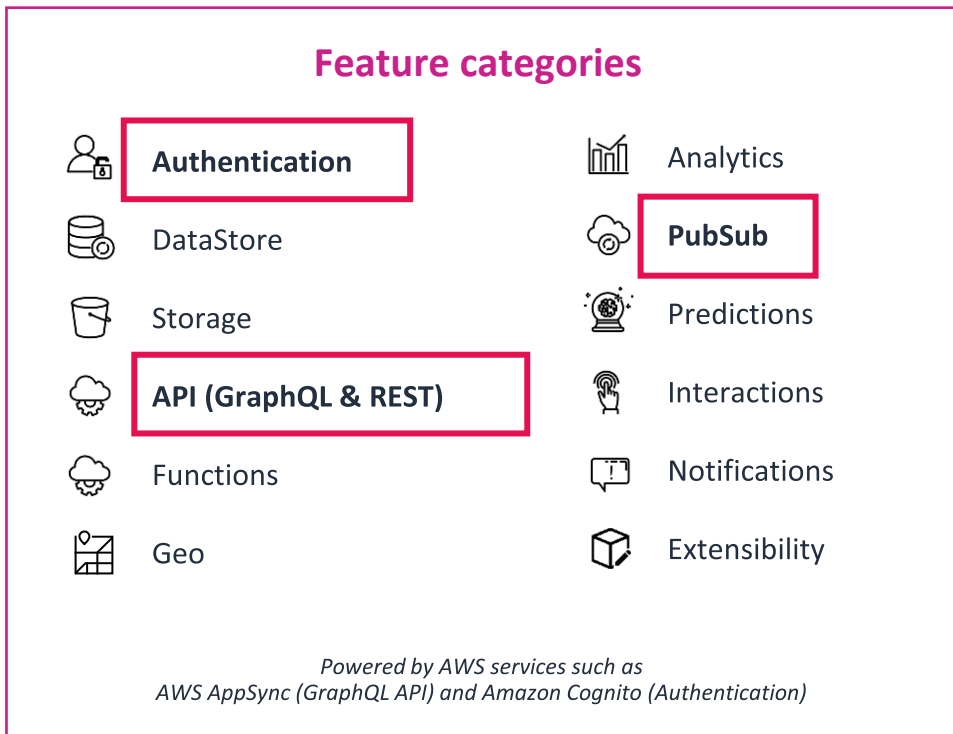
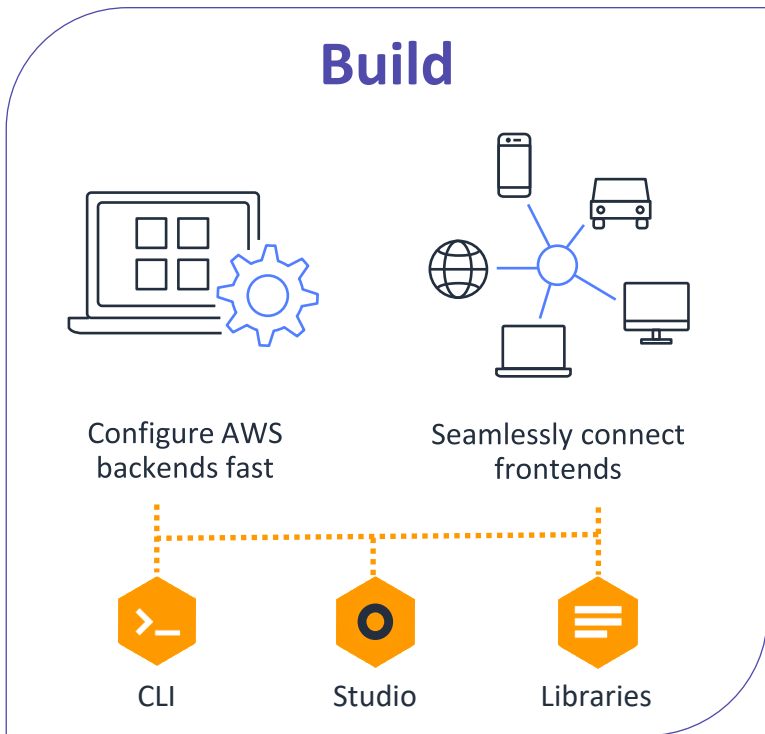
Scale



Customize and
integrate with CDK

BUILD with AWS Amplify

Backend feature categories



BUILD – Amplify Libraries

Frontend libraries for Web, iOS, Android, React Native, and Flutter



JavaScript



React



React Native



Angular



Ionic



Vue



Next.js



Android



iOS



Flutter

AWS Amplify

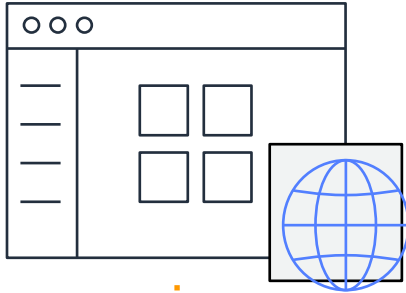
Backend development

BUILD – Environments

Toolchain to create, integrate, and manage the AWS cloud services for your app.

Amplify Studio

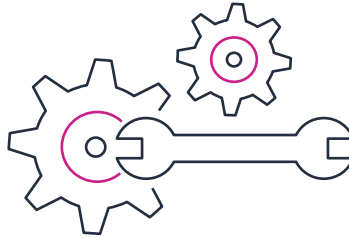
Visual development environment



Studio

Amplify CLI

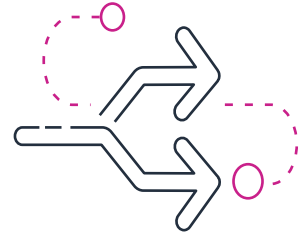
Full-stack CI/CD for developers



CLI

Amplify + CDK

Access to 175+ AWS resources



BUILD – Amplify Studio

Configure your backend app visually



Studio

- Model data with GraphQL API and DynamoDB tables
- Manage users, groups, file storage
- Configure authentication
- Extend with CLI toolchain to add functions

The screenshot shows the Amplify Studio interface for configuring data models. The left sidebar contains navigation options: Home, Manage (Content, Users), and Set up (Data, Auth, Function, Storage, REST API, Analytics, Predictions, Interactions, Notifications). The main area is titled 'Data' and includes a 'Save and Deploy' button. It displays three model configurations:

- Vote Model:** Fields include 'id' (ID type). A relationship 'candidates' is defined with 'Candidate' as the related model and 'm:n many Votes...' as the cardinality.
- Candidate Model:** Fields include 'id' (ID type), 'firstname' (String), 'lastname' (String), and 'electionID' (Relationship Source). A relationship 'votes' is defined with 'Vote' as the related model and 'm:n many Candi...' as the cardinality.
- Election Model:** Fields include 'id' (ID type), 'name' (String), and 'deadline' (AWSDate). A relationship 'candidates' is defined with 'Candidate' as the related model and '1:n one Election...' as the cardinality.

BUILD – Amplify CLI

Quickly create backends for your web and mobile applications



CLI

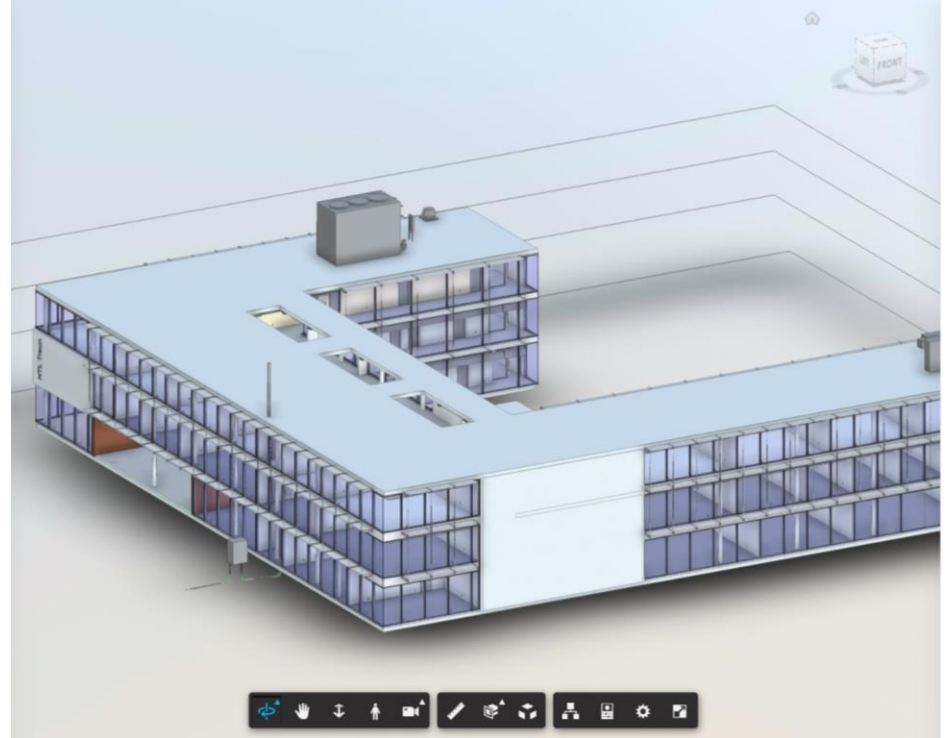
- Full-stack CI/CD for developers
- Model data with GraphQL API and DynamoDB tables
- Manage users, groups, file storage
- Configure authentication
- Transform schema definitions to GraphQL API and DynamoDB tables
- Codegen for Swift (iOS), Java (Android), and JavaScript

```
1  type Blog @model {
2    id: ID!
3    name: String!
4    posts: [Post] @connection(name: "BlogPosts")
5  }
6  type Post @model {
7    id: ID!
8    title: String!
9    blog: Blog @connection(name: "BlogPosts")
10   comments: [Comment] @connection(name: "PostComments")
11 }
12 type Comment @model {
13   id: ID!
14   content: String
15   post: Post @connection(name: "PostComments")
16 }
```

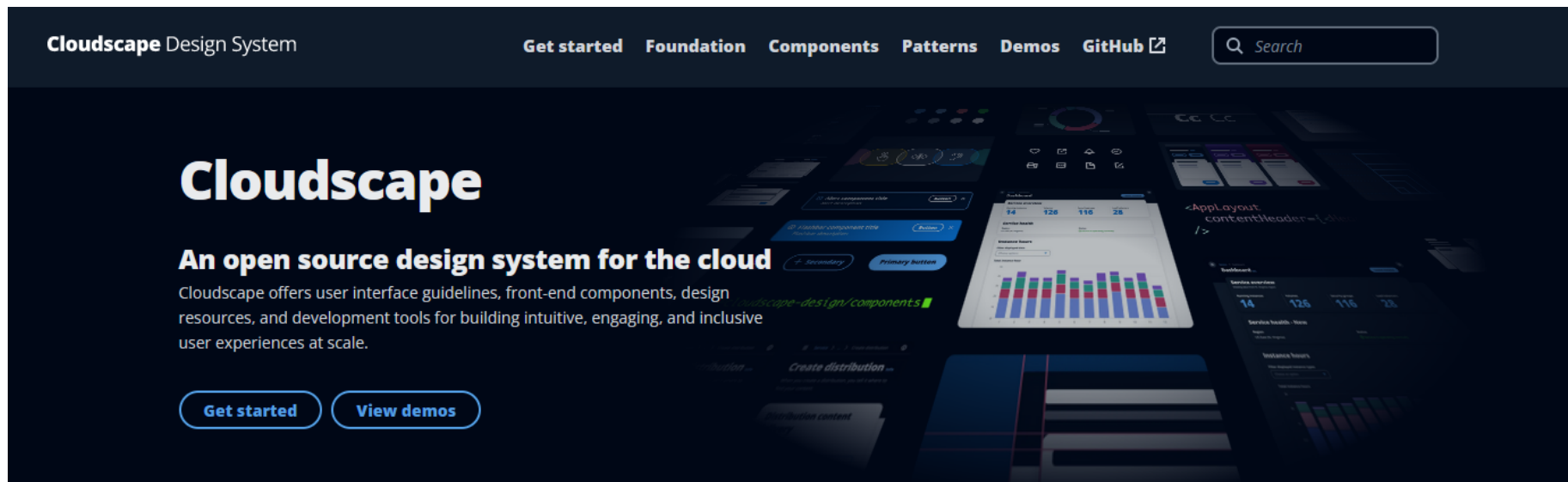
AWS Amplify UI Development

Autodesk Forge Viewer

- A WebGL-based, JavaScript library for 3D and 2D model rendering.
- Model data may come from AutoCAD, Fusion 360, Revit, etc.
- Compatible with all modern browsers



Cloudscape - <https://cloudscape.design>



Foundation

Learn about the visual foundation of our design system, such as [colors](#), [spacing](#), and [iconography](#).

Components

Discover our library of components. Built with React, they are tested, accessible, and responsive.

Patterns

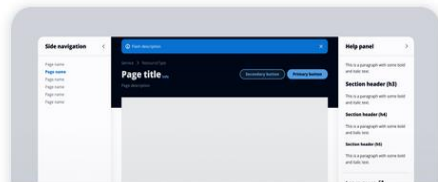
Explore our guidelines and best practices for building user experiences.

Demos

View Cloudscape in action by browsing through our demos.

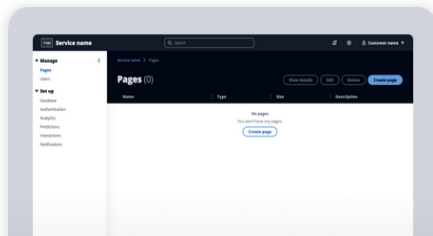


Cloudscape - <https://cloudscape.design>



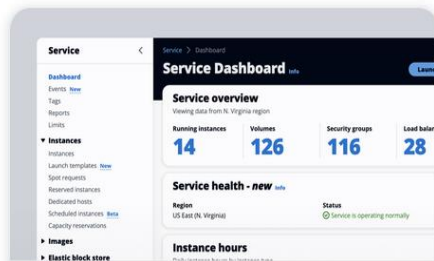
App layout

Provides the basic layout for all types of pages, including collapsible side navigation, tools panel, and split panel.



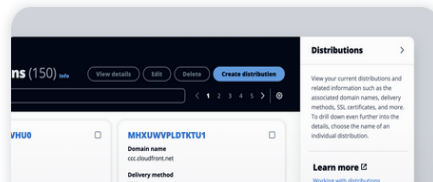
Top navigation

The top navigation, when paired with side navigation and breadcrumbs, forms a global, scalable navigation system



Side navigation

A list of navigational links that point to the pages within an application.



Help panel

The panel displays help content that relates to a concept, term, setting, option, or task within the main page content.

- Basic layout
- Collapsible panels

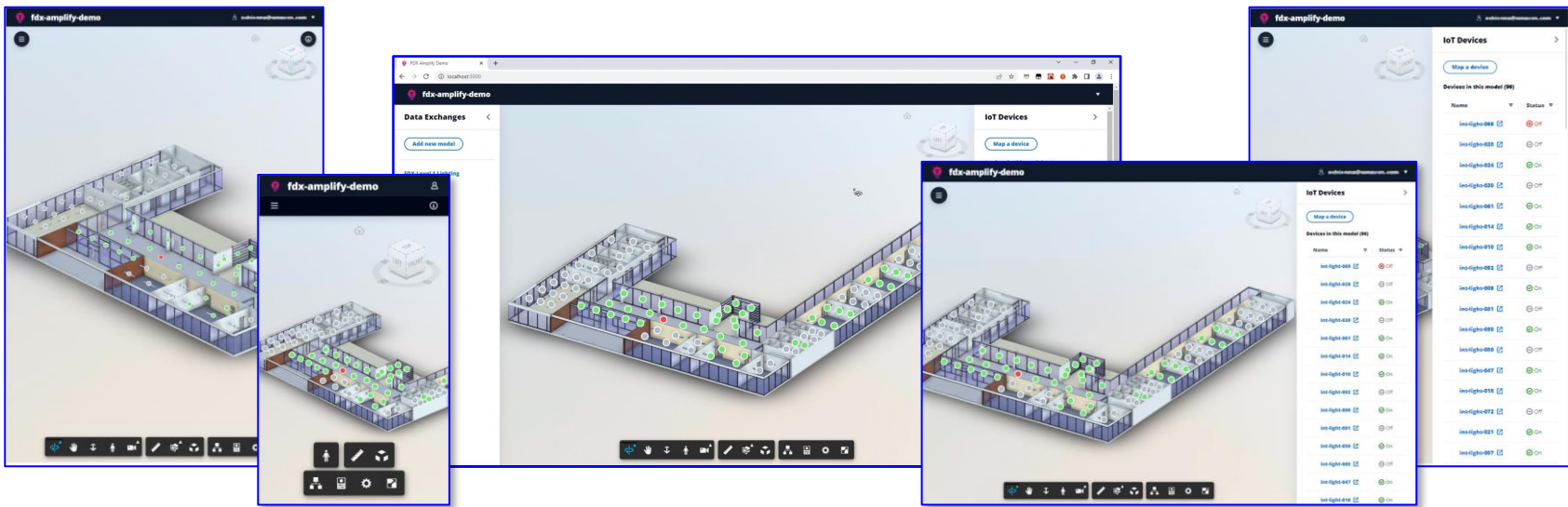
- Navigation
- Breadcrumbs

- Navigational links

- Contextual help content

Responsive

- Cloudscape React components are responsive
- Responsive web apps adapt to different screen sizes, resolutions, and orientations



Single-Page vs. Progressive vs. Mobile Apps

- **Single-Page Application (SPA)**
 - A website that loads content dynamically within a single page.
 - Runs within a single Document Object Model (DOM).
 - Requires network connection.
- **Progressive Web App (PWA)**
 - A web application designed to run as if it's a native mobile app.
 - Installable
 - Uses background workers that load cached data even without a network.
- **Native Mobile App**
 - Built to run on a specific mobile OS (iOS or Android).
 - Network connection not required.



Single-Page vs. Progressive vs. Mobile Apps

	Single Page App	Progressive Web App	Native Mobile App
Platform	Any browser, any device	Any browser, any device	Device OS
Language	HTML, CSS, JavaScript	HTML, CSS, JavaScript	Mobile-specific
Building and updating	Faster	Faster	Slower
Hosting/distributing	http or https hosting	https hosting (required)	App Store
Search engine	None	SEO	ASO
Installable	No	Yes	Yes
Offline-able	No	Yes	Yes

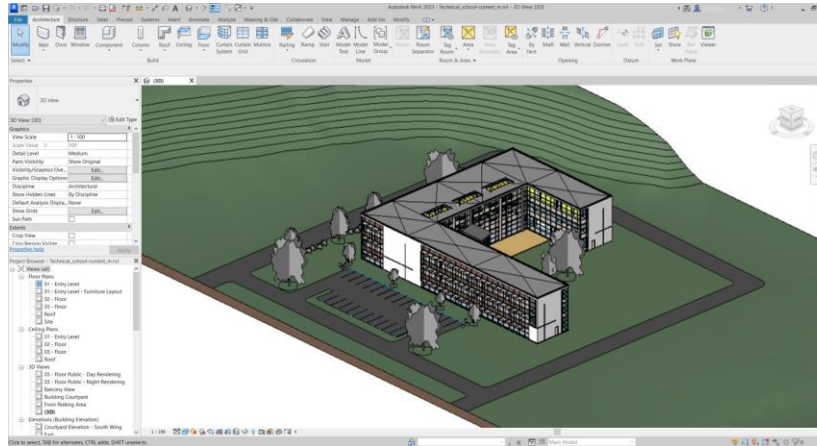
Use Case Spotlight

One example of a Data Exchange use case

Use case - Architect



Architect



AUTODESK
Construction Cloud

Use case – Building Manager

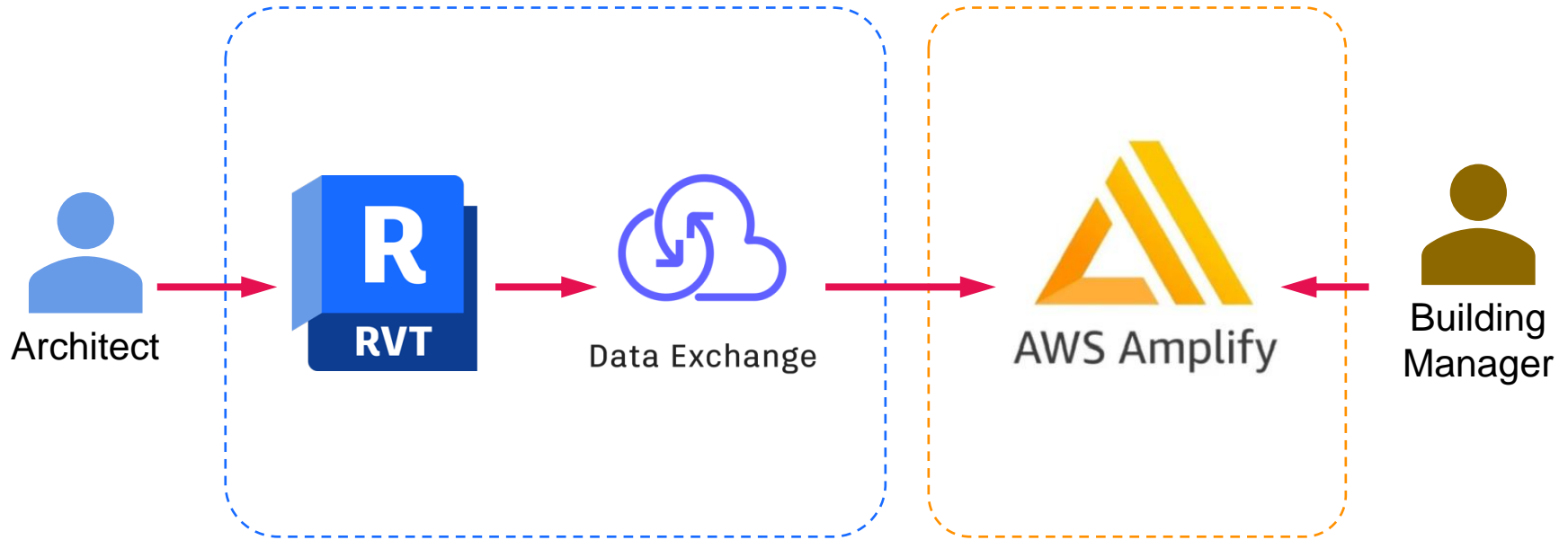


Building
Manager

Use case – Data Exchange



Data Exchange – Amplify workflow

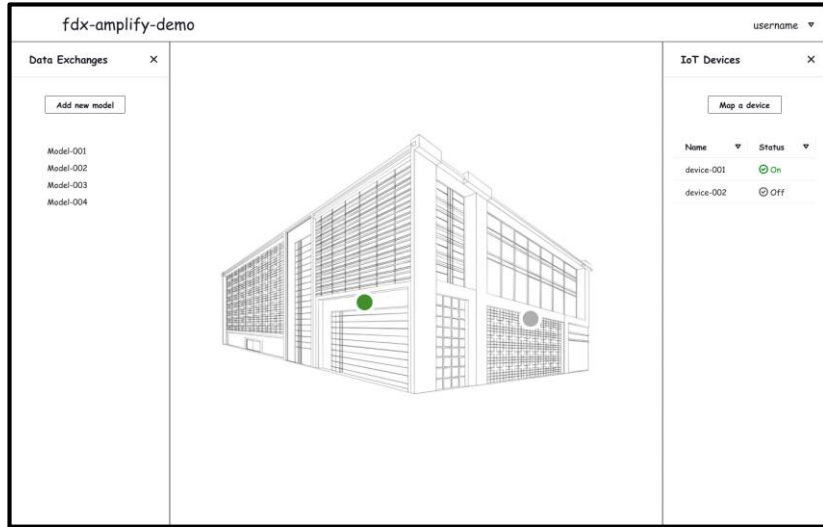


Share specific subsets of Revit data with third party applications

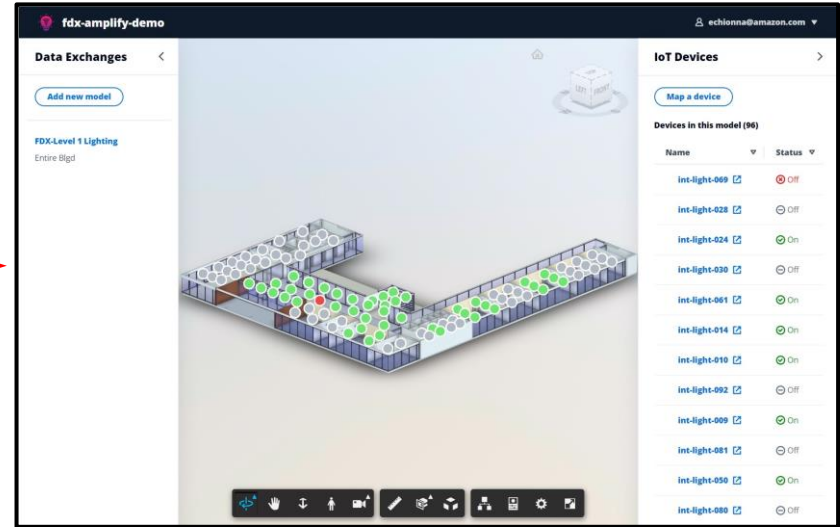
App Development Workflow

App UI Design

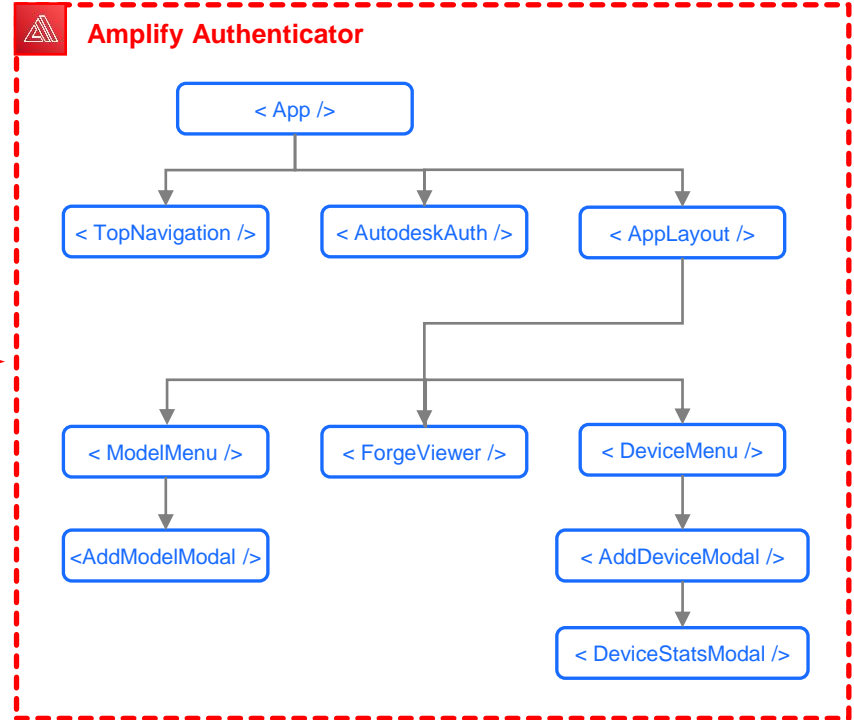
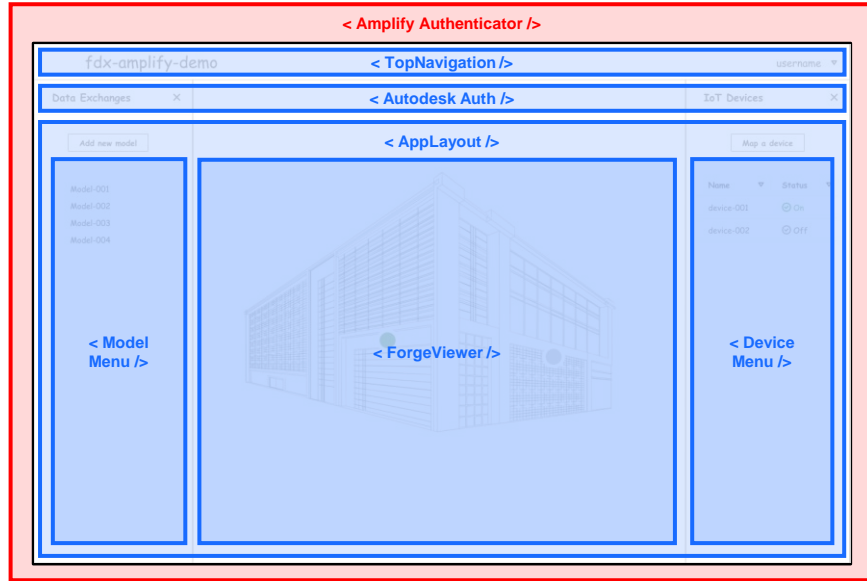
Wireframe



High-fidelity



React UI Components



 AWS Amplify UI  AWS Cloudscape

Forge API Used



Authentication

- Get 3-legged token with implicit grant



Viewer

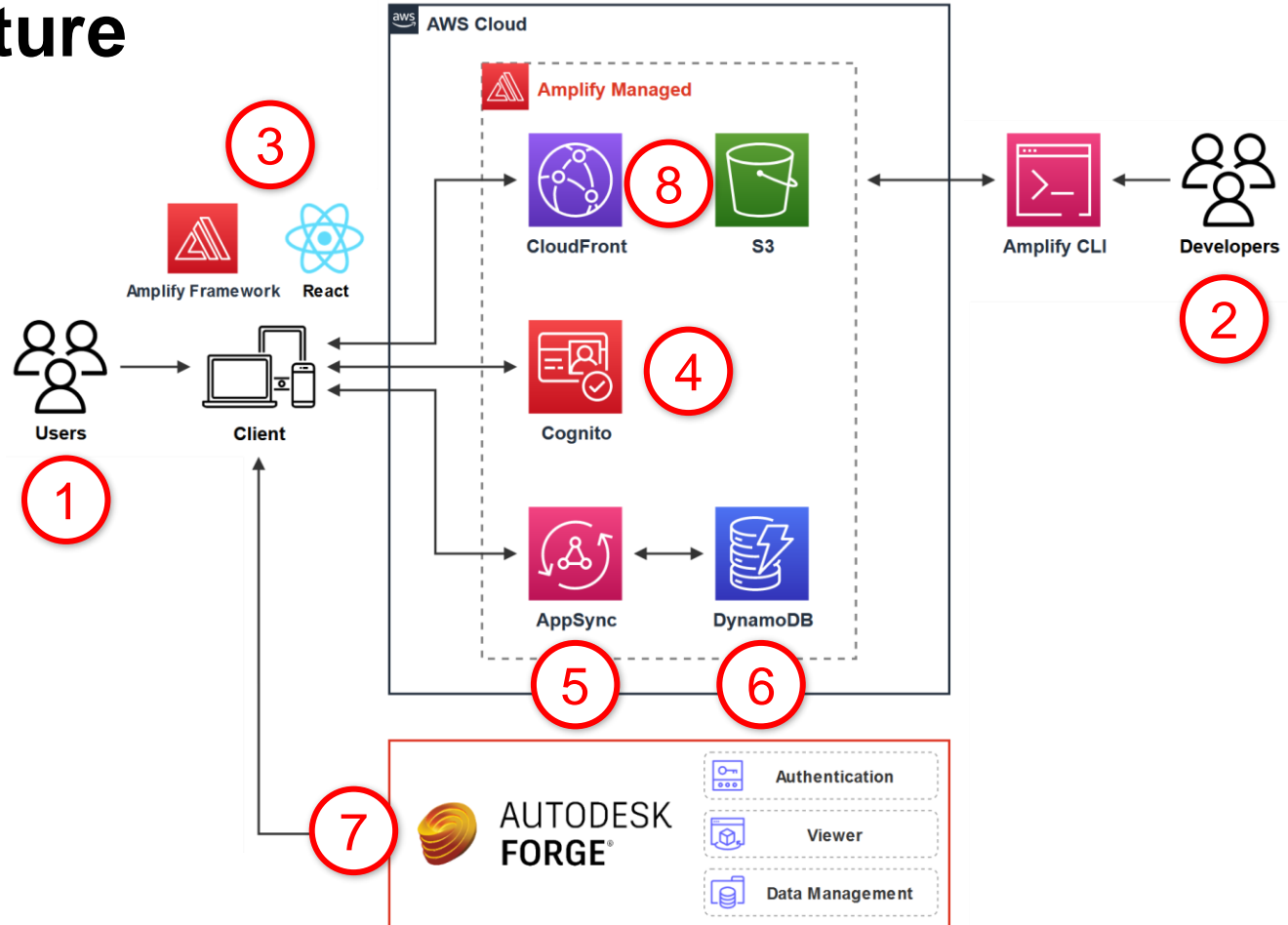
- Load and display models



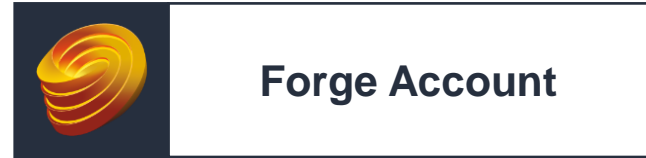
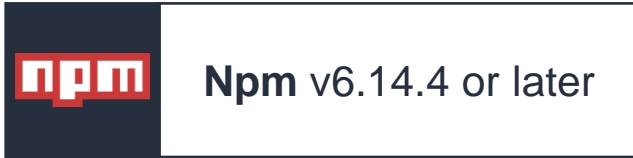
Data Management

- Get latest model version URN

Architecture



Prerequisites



Steps to Create an Amplify Project

- 1 Install Amplify CLI
- 2 Configure Amplify
- 3 Create a New React App
- 4 Initialize a New Amplify Project
- 5 Connect Amplify to the Frontend

Install & Configure Amplify CLI

1

```
npm install -g @aws-amplify/cli
```

2

```
amplify configure
```

```
...
```

Specify the AWS Region

? region: # Your preferred region

Specify the username of the new IAM user:

? user name: # User name for Amplify IAM user

Complete the user creation using the AWS console

```
...
```

Enter the access key of the newly created user:

? accessKeyId: # YOUR_ACCESS_KEY_ID

? secretAccessKey: # YOUR_SECRET_ACCESS_KEY

This would update/create the AWS Profile in your local machine

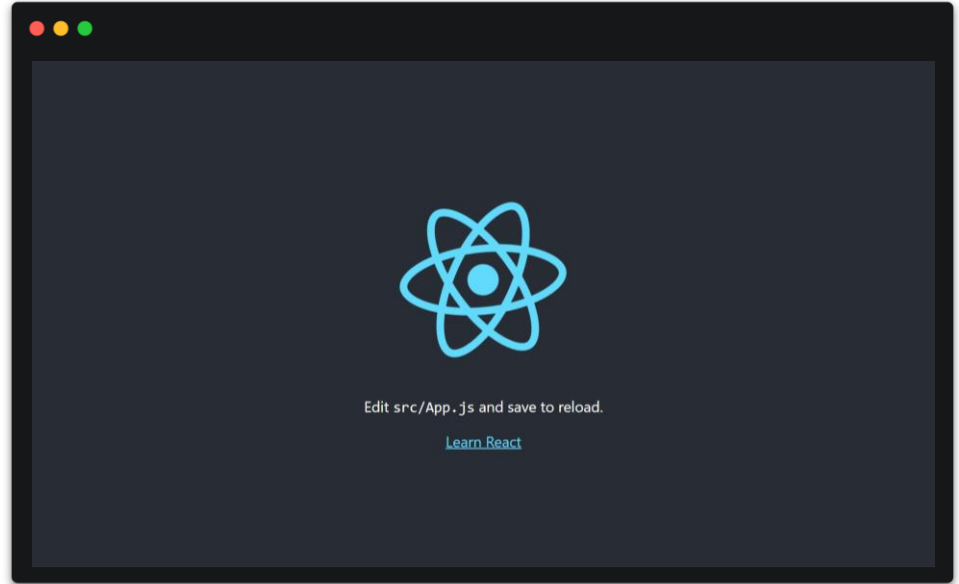
? Profile Name: # (default)

Successfully set up the new user.

Create a New React App

3

```
npx create-react-app fdx-amplify-demo  
cd fdx-amplify-demo  
npm start
```



Initialize a New Amplify Project

4

```
amplify init
```

Enter a name for the project (amplified) # Your project name

The following configuration will be applied:

Project information

```
| Name: amplified
| Environment: dev
| Default editor: Visual Studio Code
| App type: javascript
| Javascript framework: react
| Source Directory Path: src
| Distribution Directory Path: build
| Build Command: npm run-script build
| Start Command: npm run-script start
```

? Initialize the project with the above configuration? Yes

? Select the authentication method you want to use:

```
> AWS profile
   AWS access keys
```

? Please choose the profile you want to use:

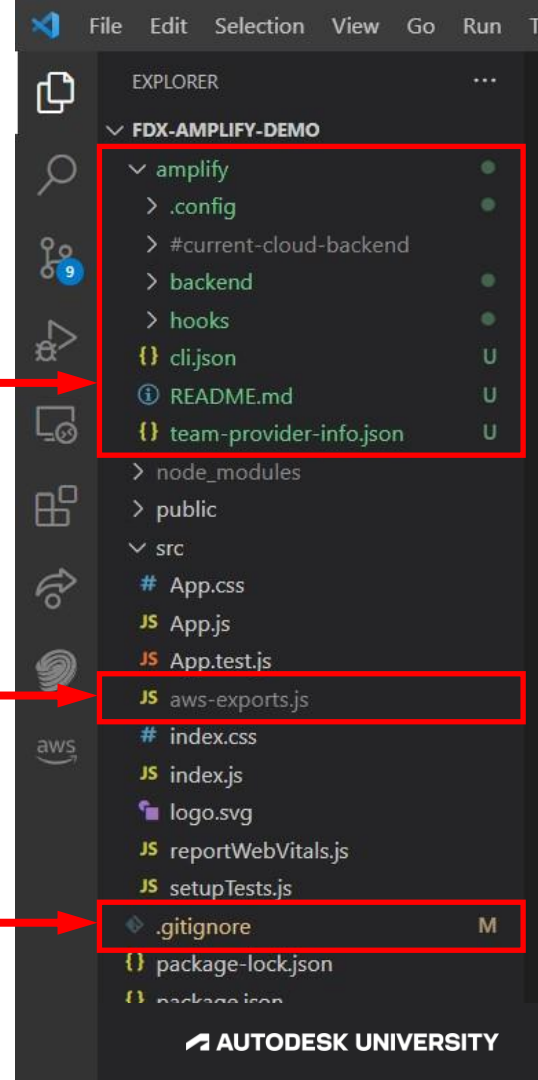
```
> YOUR_PROFILE # profile created earlier with `amplify configure`
   OTHER_PROFILE
```


Amplify Generated Files

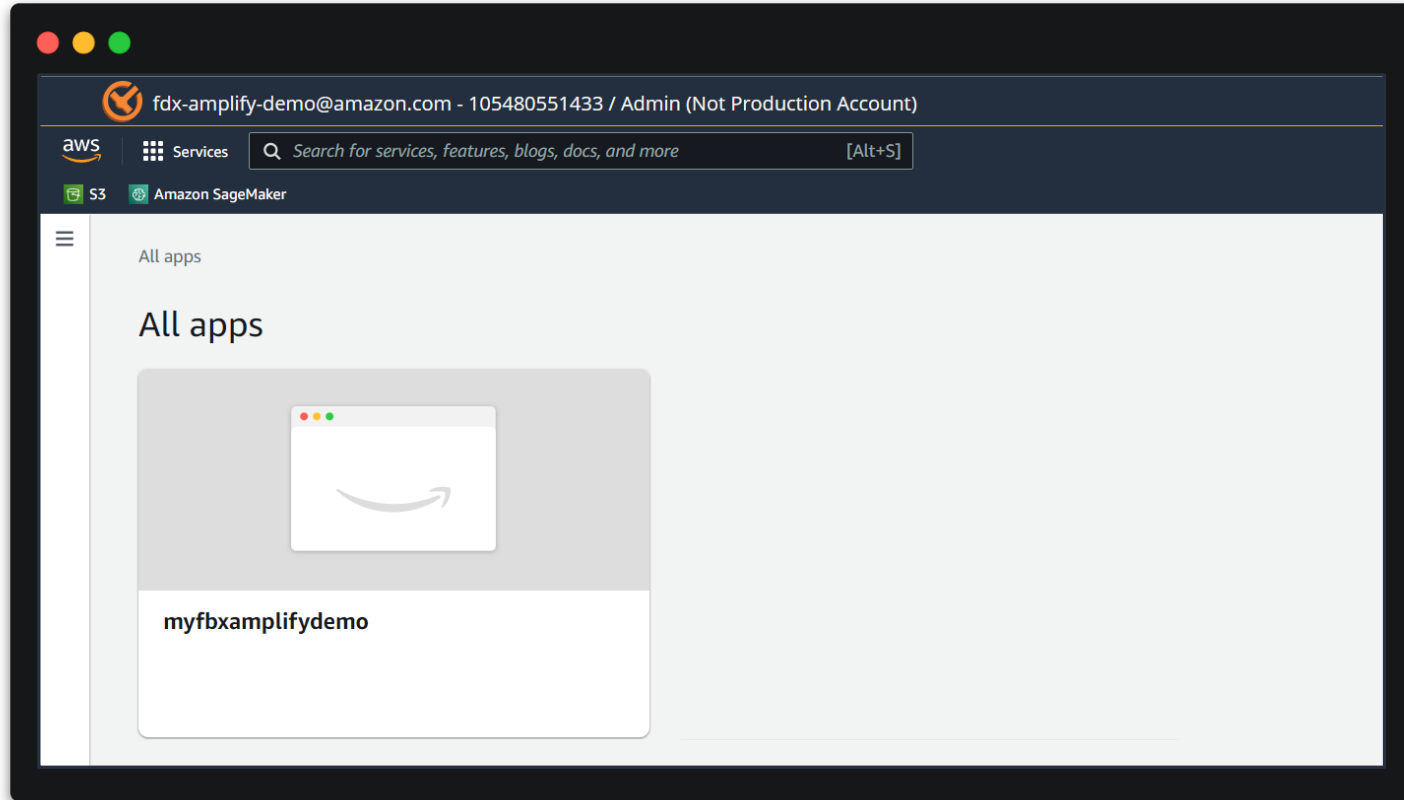
creates top level directory called **amplify** with backend definition

aws-export.js holds configuration for the backend services

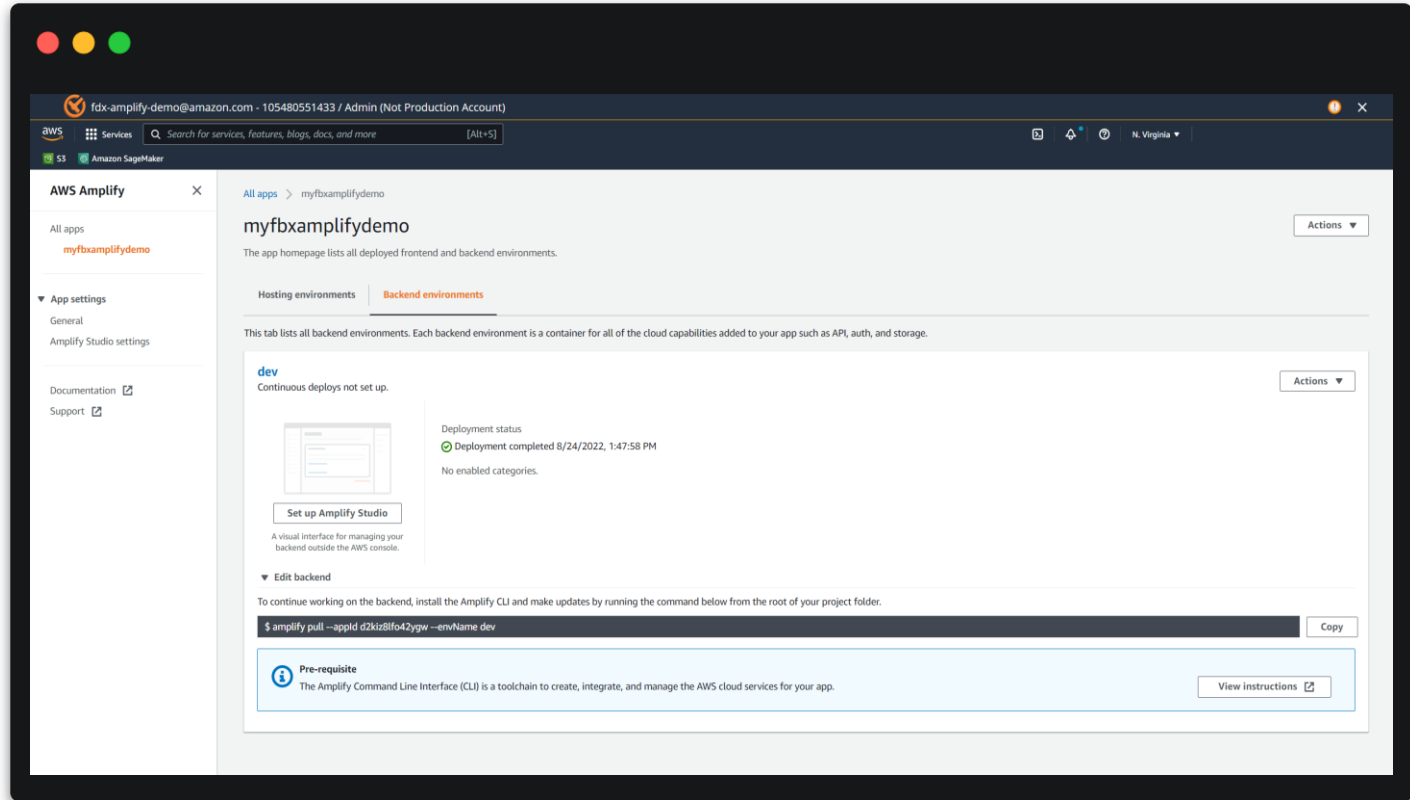
.gitignore adds some Amplify generated files to the ignore list



Amplify Generated Application in AWS Console



Amplify Generated Backend in AWS Console

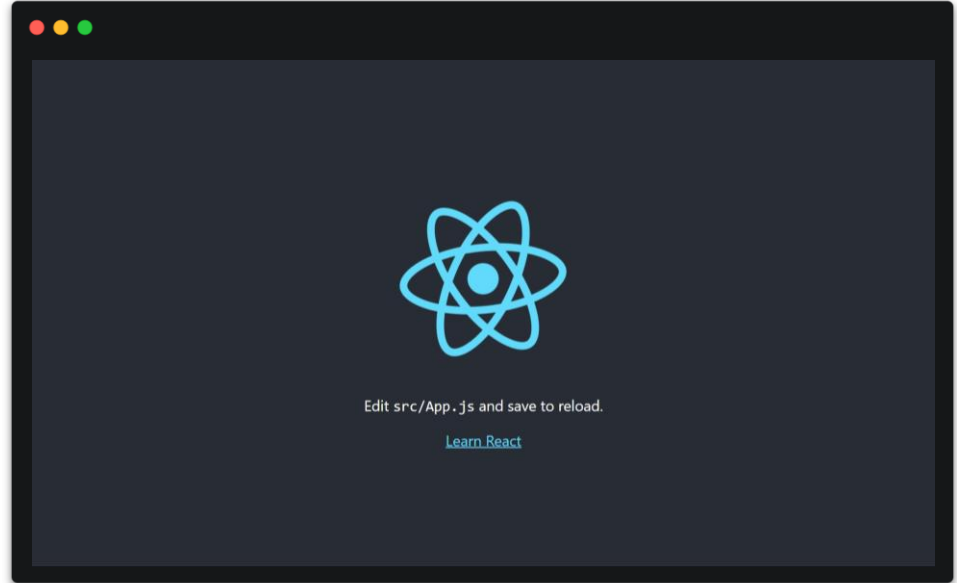


Connect Amplify to the Frontend

src/index.js

5

```
import { Amplify } from 'aws-amplify';  
import awsExports from './aws-exports';  
Amplify.configure(awsExports);
```



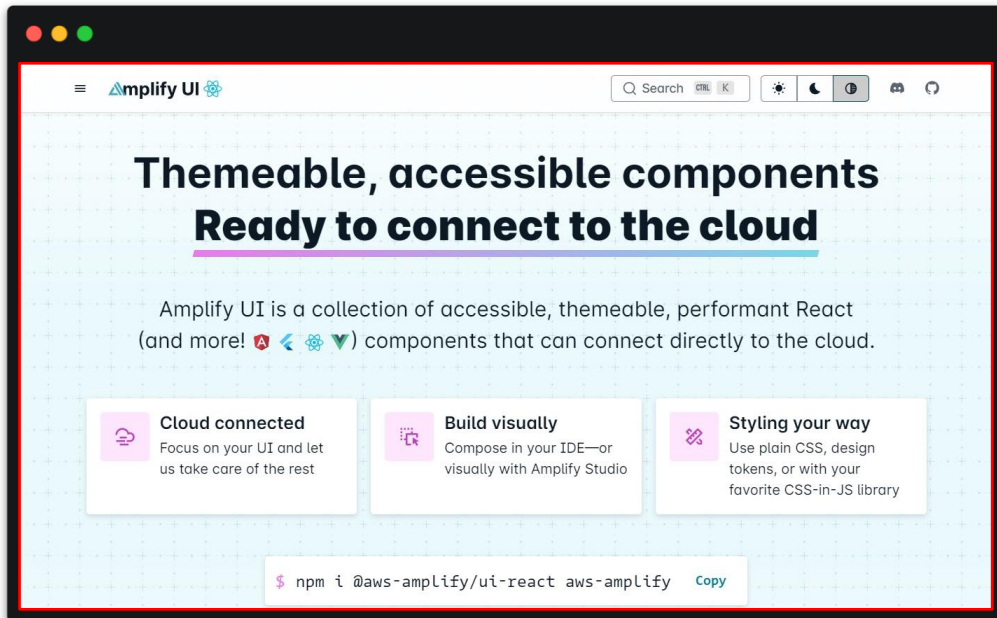
App Development

Adding Services: Authentication, API, Forge Viewer

Install Amplify UI Package

<https://ui.docs.amplify.aws/>

```
npm install aws-amplify @aws-amplify/ui-react
```



Install Cloudscape Packages

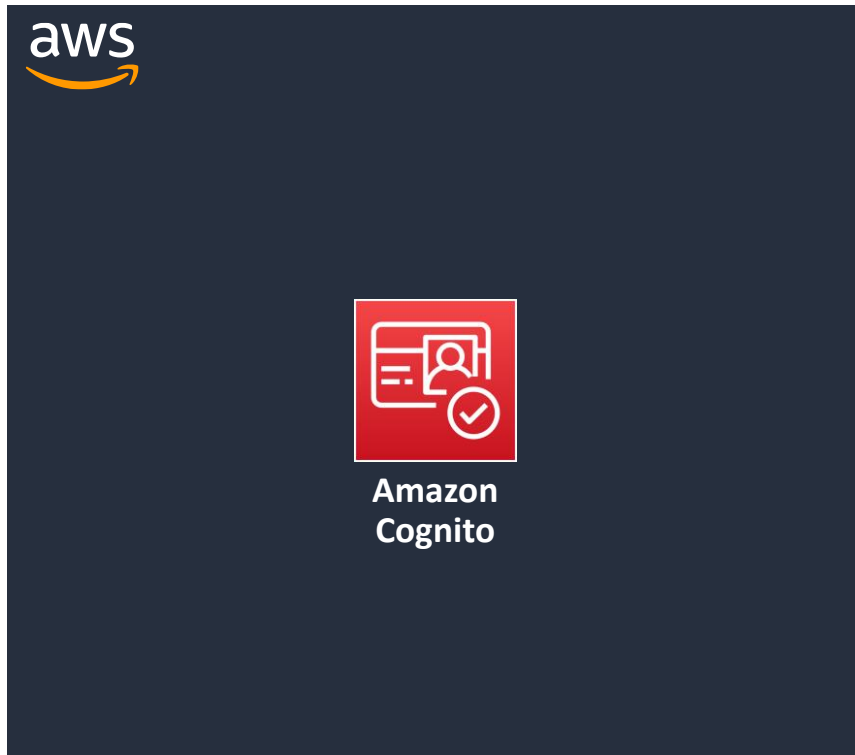
<https://cloudscape.design/>

```
npm install @cloudscape-design/global-styles  
npm install @cloudscape-design/components  
npm install @cloudscape-design/design-tokens
```



Add Authentication

- **Sign in**
 - Sign in with email and password
- **Sign up**
 - Sign up with email and password
 - Verify email with verification code
- **Recover password**
 - Reset forgotten password
 - Verify email with verification code
- **Custom styling**



Add Authentication

Sign In Create Account

Email

Password

Sign in

[Forgot your password?](#)

Create Authentication Service

```
amplify add auth
```

```
? Do you want to use the default authentication and security  
configuration? Default configuration  
? How do you want users to be able to sign in? Email  
? Do you want to configure advanced settings? No, I am done.
```

```
amplify push
```

```
amplify console
```

Create Authentication Service

The screenshot displays the AWS Amplify console interface. At the top, the navigation bar shows the AWS logo, 'Services', a search bar with 'sageMaker Canvas', and regional information for 'N. Virginia'. The left sidebar contains the 'AWS Amplify' header, a list of apps including 'myfbxamplifydemo', and a section for 'App settings' with links to 'General', 'Amplify Studio settings', 'Documentation', and 'Support'. The main content area is titled 'dev' and features tabs for 'Overview', 'Authentication' (which is selected), 'API', 'File storage', 'Analytics', and 'Functions'. Under the 'Authentication' tab, there are two panels: 'Users' and 'Federated identities'. The 'Users' panel includes a 'View in Cognito' link, a description of managing users and groups, and the app's name 'myfbxamplifydemodcdfa1f2_userpool_dcdfa1f2-dev'. The 'Federated identities' panel also has a 'View in Cognito' link, a description of federated identities, and the app's name 'myfbxamplifydemodcdfa1f2_identitypool_dcdfa1f2_dev'. Below these panels is an 'Edit backend' section with instructions to install the Amplify CLI and two terminal commands: '\$ amplify pull --appId d2kiz8lfo42ygw --envName dev' and '\$ amplify update auth', each with a 'Copy' button. A 'Pre-requisite' box at the bottom explains the Amplify CLI and includes a 'View instructions' link. The footer contains a 'Feedback' link, a language selection prompt, the year '© 2022, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Create Authentication Service

The screenshot shows the Amazon Cognito console interface. The top navigation bar includes the AWS logo, a 'Services' menu, a search bar, and a '[Alt+S]' shortcut. The left sidebar shows the 'Amazon Cognito' title and a list of options: 'User pools' (highlighted in orange) and 'Federated identities'. The main content area is titled 'myfbxamplifydemodcdfa1f2_userpool_dcdfa1f2-dev' with an 'Info' link and a 'Delete user pool' button. Below the title is a 'User pool overview' section containing a table with the following data:

User pool name	ARN	Created time
myfbxamplifydemodcdfa1f2_userpool_dcdfa1f2-dev	arn:aws:cognito-idp:us-east-1:105480551433:userpool/us-east-1_G29G1gJ0j	August 25, 2022 at 15:32 PDT
User pool ID	Estimated number of users	Last updated time
us-east-1_G29G1gJ0j	0	August 25, 2022 at 15:32 PDT

Below the overview is a 'Getting started' section with a right-pointing arrow. A horizontal tab bar follows, with 'Users', 'Groups', 'Sign-in experience' (highlighted in orange), 'Sign-up experience', 'Messaging', 'App integration', and 'User pool properties'. The 'Sign-in experience' tab is active, showing a 'Cognito user pool sign-in' section with an 'Info' link. Below this is a description: 'Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.' At the bottom of the sign-in options section, 'Email' is listed as an option. The footer of the console includes a 'Feedback' link, a language selection prompt, a 'Unified Settings' link, a copyright notice for 2022, and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Create Login UI

src/app.js (before)

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

src/app.js (after)

```
import logo from './logo.svg';
import './App.css';
import { withAuthenticator } from '@aws-amplify/ui-react';
import '@aws-amplify/ui-react/styles.css';

function App({ signOut, user }) {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

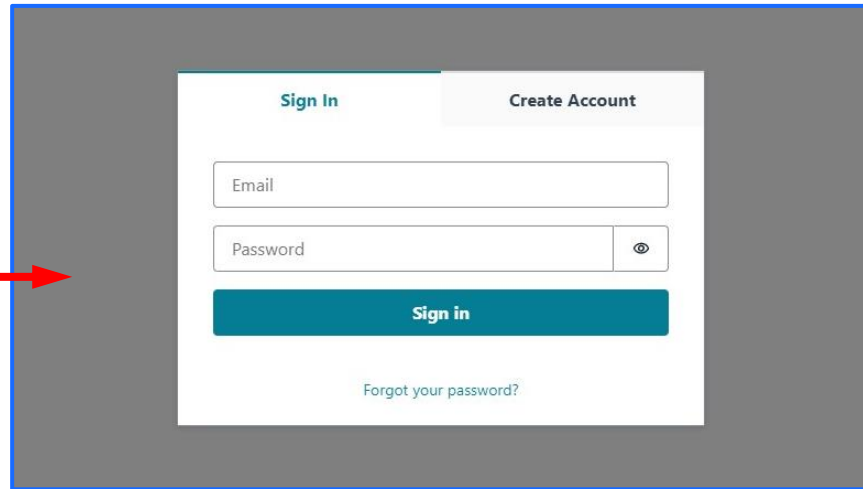
export default withAuthenticator(App);
```

Create Login UI

before



after



Login UI Custom Theme

<https://ui.docs.amplify.aws/react/theming>

src/index.js (before)

```
...
import { Amplify } from "aws-amplify";
import awsExports from "../aws-exports";
Amplify.configure(awsExports);

const root =
ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

src/index.js (after)

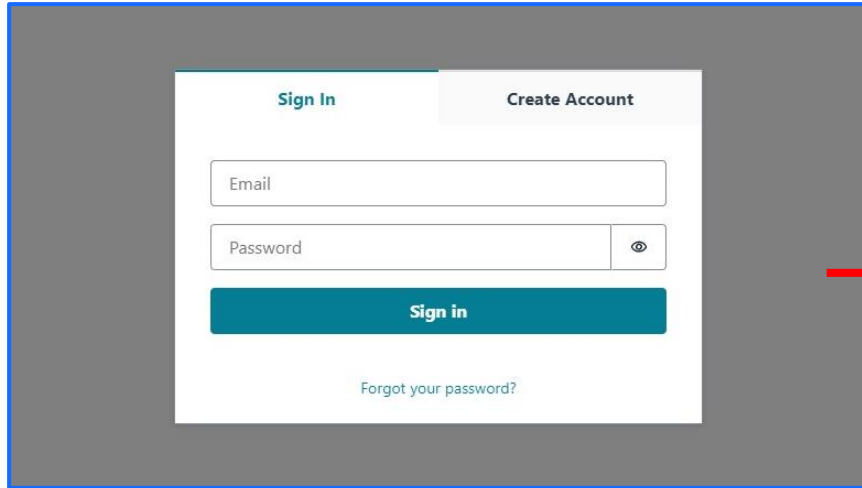
```
...
import { ThemeProvider } from "@aws-amplify/ui-react";
import * as awsui from "@cloudscape-design/design-tokens";
...

const theme = {
  name: "cloudscape-design",
  tokens: {
    colors: {
      brand: {
        primary: {
          80: { value: awsui.colorBackgroundButtonPrimaryDefault },
        },
      },
    },
  },
};

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <App />
    </ThemeProvider>
  </React.StrictMode>
);
```

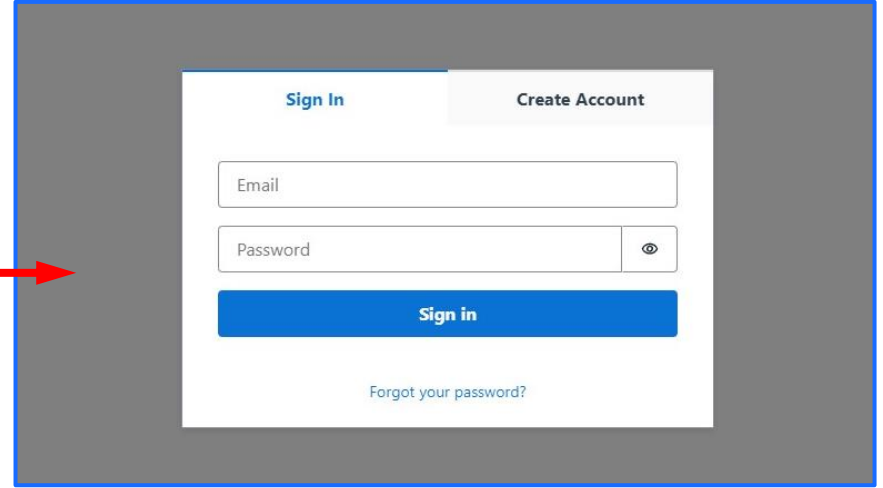
Login UI Custom Theme

before



The 'before' state shows a login form with a teal 'Sign In' button. The form includes tabs for 'Sign In' and 'Create Account', input fields for 'Email' and 'Password' (with a toggle icon), and a 'Forgot your password?' link.

after



The 'after' state shows the same login form but with a blue 'Sign In' button. A red arrow points from the 'before' state to the 'after' state, indicating the theme change.

Add GraphQL API Using Amplify

- **Queries**

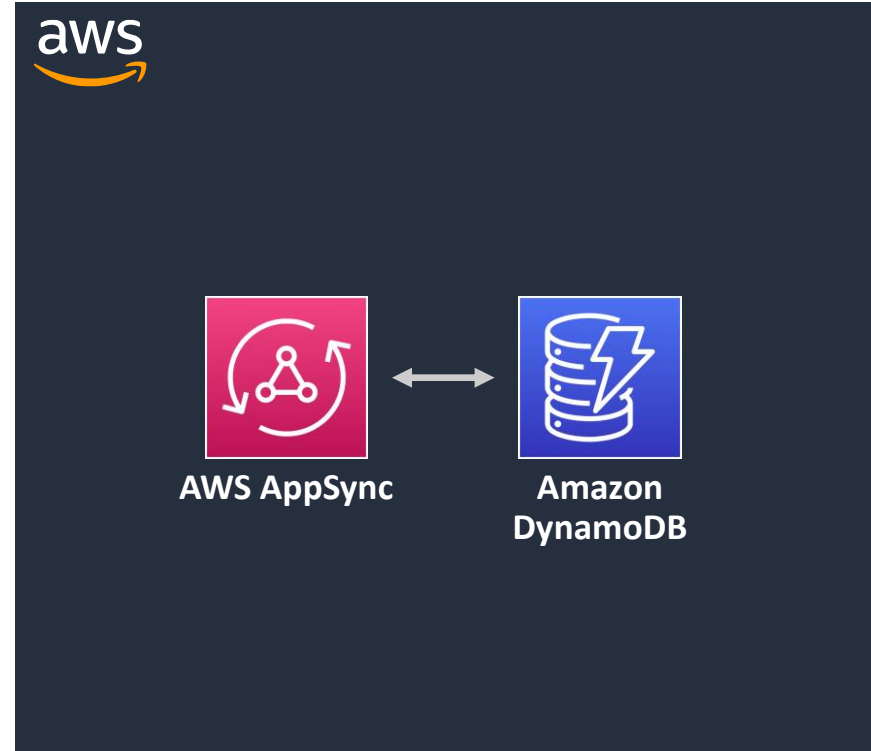
- Get list of models
- Get list of devices

- **Mutations**

- Create a new model
- Create a new device

- **Subscriptions**

- Get real-time notifications when a device is updated



Steps to Add an API

- 1 Create GraphQL API and Database
- 2 Edit GraphQL Schema
- 3 Deploy the API
- 4 Connect Frontend to API

Create GraphQL API and Database

1

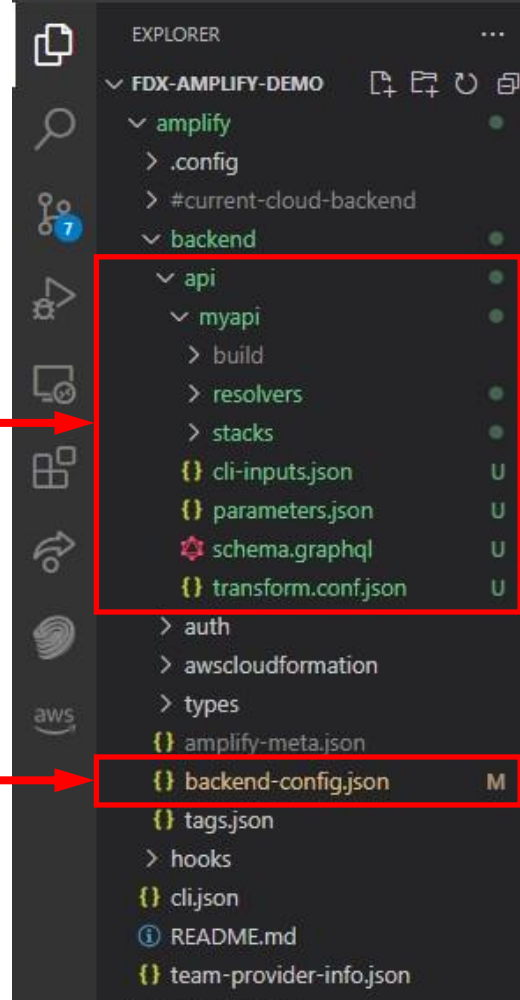
```
amplify add api
```

```
? Select from one of the below mentioned services:
# GraphQL
? Provide API name:
# myapi
? Choose the default authorization type for the API:
# API Key
? Enter a description for the API key:
# demo
? After how many days from now the API key should expire:
# 7 (or your preferred expiration)
? Do you want to configure advanced settings for the GraphQL
API:
# No
? Do you have an annotated GraphQL schema?
# No
? Choose a schema template:
# One-to-many relationship (e.g., "Blogs" with "Posts" and
"Comments")
? Do you want to edit the schema now?
# Yes
```

Amplify Generated Files

Created backend api folder with `schema.graphql`

Added new api settings to `backend-config.json`



Edit GraphQL Schema

schema.graphql (before)

```
input AMPLIFY { globalAuthRule: AuthRule = { allow: public } }

type Blog @model {
  id: ID!
  name: String!
  posts: [Post] @hasMany
}

type Post @model {
  id: ID!
  title: String!
  blog: Blog @belongsTo
  comments: [Comment] @hasMany
}

type Comment @model {
  id: ID!
  post: Post @belongsTo
  content: String!
}
```

schema.graphql (after)

2

```
type Model @model @auth(rules: [{ allow: owner }]) {
  id: ID!
  name: String!
  accProjectId: String!
  accEntityId: String!
  urn: String
  devices: [Device] @hasMany
}

type Device @model @auth(rules: [{ allow: owner }]) {
  id: ID!
  name: String!
  state: String @default(value: "On")
  dbId: Int!
  point: AWSJSON!
  model: Model @belongsTo
  modelDevicesId: ID!
}

type Subscription {
  onDeviceUpdateByModelId(modelDevicesId: ID!): Device
    @aws_subscribe(mutations: ["updateDevice"])
}
```

Deploy the API

3

```
amplify push
```

```
? Are you sure you want to continue? Y
```

```
# You will be walked through the following questions for GraphQL  
code generation
```

```
? Do you want to generate code for your newly created GraphQL  
API? Y
```

```
? Choose the code generation language target: javascript
```

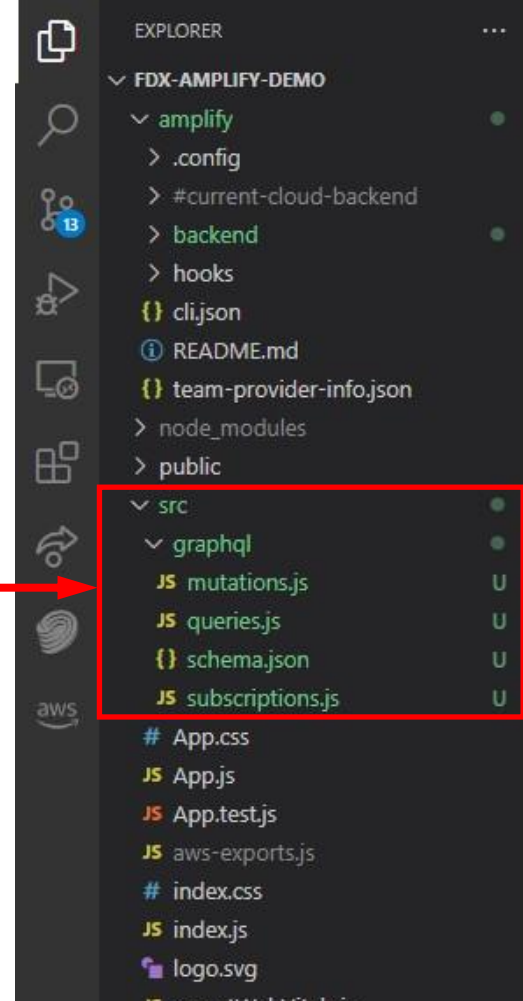
```
? Enter the file name pattern of graphql queries, mutations and  
subscriptions: src/graphql/**/*.js
```

```
? Do you want to generate/update all possible GraphQL operations  
- queries, mutations and subscriptions? Y
```

```
? Enter maximum statement depth [increase from default if your  
schema is deeply nested]: 2
```

Amplify Generated Files

Generated code for mutations, queries, and subscriptions



A Look Inside the Generated Code

Queries

Mutations

Subscriptions

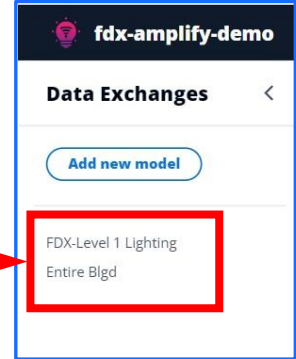
Connect Frontend to API - Queries

4

Get a list of models when the user signs in

```
import { API, graphqlOperation } from "aws-amplify";
import { listModels } from "../graphql/queries";

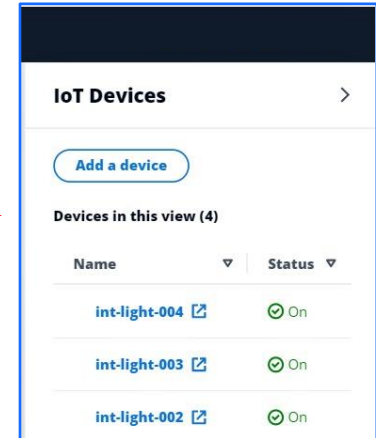
const modelsData = await API.graphql(graphqlOperation(listModels));
```



Get a list of devices by model ID after the user selects a model

```
import { API, graphqlOperation } from "aws-amplify";
import { listDevices } from "../graphql/queries";

const devicesData = await API.graphql(
  graphqlOperation(listDevices, {filter: { modelDevicesId: { eq: modelId } } })
);
```



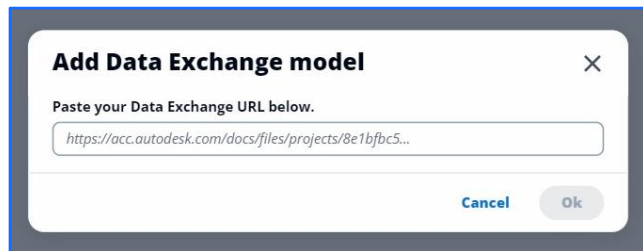
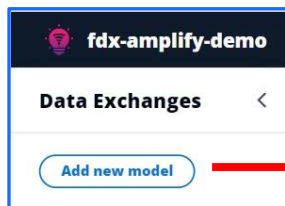
Connect Frontend to API - Mutations

4

Create new model with only 3 lines of code

```
import { API, graphqlOperation } from "aws-amplify";
import { createModel } from "../graphql/mutations";

const addModel = await API.graphql( graphqlOperation(createModel, { input: newModel }));
```

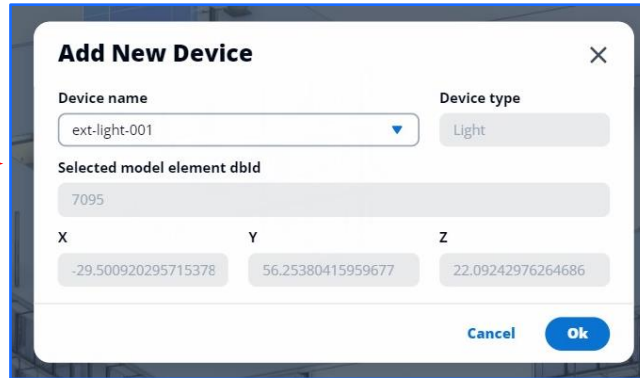
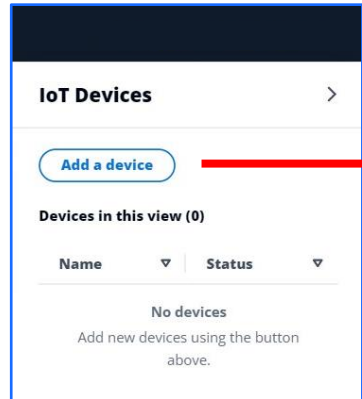


Connect Frontend to API - Mutations

4

Create new device with only 3 lines of code

```
import { API, graphqlOperation } from "aws-amplify";  
import { createDevice } from "../graphql/mutations";  
  
const addDevice = await API.graphql(graphqlOperation(createDevice, {input: device }));
```




Connect Frontend to API - Subscriptions

4

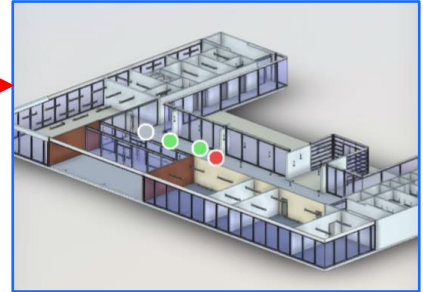
Subscribe to update of Devices by ModelID

```
import { API } from "aws-amplify";
import { onDeviceUpdateByModelId } from "../graphql/subscriptions";

const subscription = modelId => {
  subscriptionOnDeviceUpdate = API.graphql({
    query: onDeviceUpdateByModelId,
    variables: {
      modelDevicesId: modelId,
    },
  }).subscribe({
    next: ({ provider, value }) => console.log({ provider, value }),
    error: (error) => console.warn(error)
  });
};
```

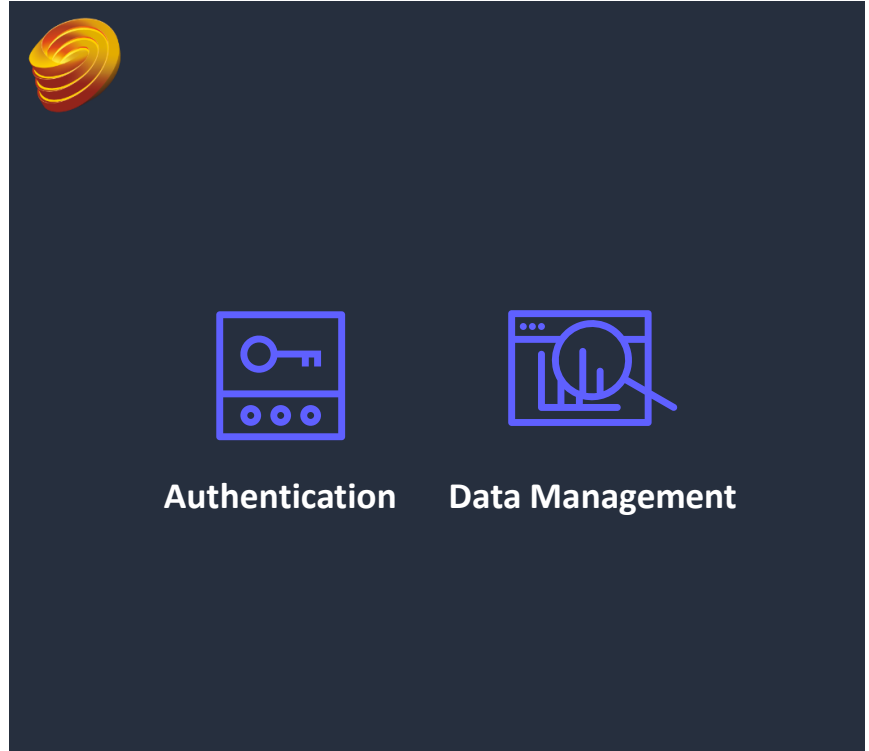


IoT Devices	
Map a device	
Devices in this model (4)	
Name	Status
int-light-004 ✎	Malfunctioning
int-light-003 ✎	On
int-light-002 ✎	On
int-light-001 ✎	Off



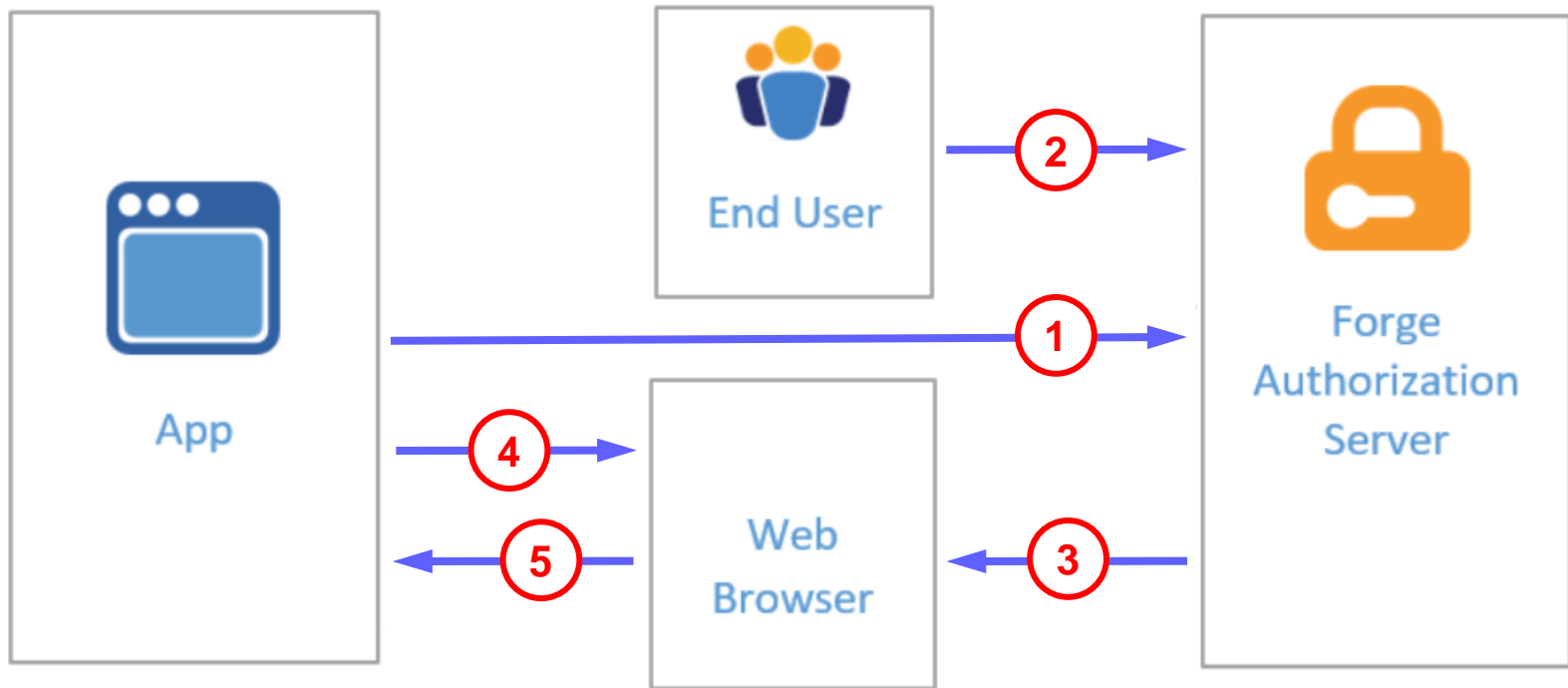
Add Forge API

- **Authentication API**
 - Get 3-legged token with implicit grant
- **Data Management API**
 - Get latest Data Exchange model version
 - Get Data Exchange model name



Get 3-Legged Token

<https://forge.autodesk.com/en/docs/oauth/v2/tutorials/get-3-legged-token-implicit/>



Get 3-Legged Token

<https://forge.autodesk.com/en/docs/oauth/v2/tutorials/get-3-legged-token-implicit/>

⊗ Autodesk auth missing

Log in with Autodesk is required before accessing models in Autodesk Construction Cloud.

Sign in with Autodesk

1. Redirect to the Authorization Web Flow

```
<a href="https://developer.api.autodesk.com/authentication/v1/authorize?
response_type=token&client_id=obQDn8P0GanGFQha4ngKKVWcxwyvFAGE&redirect_uri=http%3A%2F%2Fsampleapp.c
om%2Foauth2%2Fcallback&scope=data:read">Sign in with Autodesk</a>
```

2. Extract the Access Token from callback URL

http://sampleapp.com/oauth/callback?access_token=eyJhbGciOiJIUzI1NiIsImtpZCI6Imp3dF9zeW1tZXRYaWNfa2V5In0.eyJ1c2VyaWQiOiJlU0pHTTUSFjIUViLCJleHAiOiJlMDE0OTI5MjgsInNjb3BlIjpbXSwiY2xpZW50X2lkIjoibk9oYnJlPZw9HeHE4R2JBefVVR2NlCxpEwmVqQWxxSUSiLCJhdWQiOiJodHRwczovL2F1dG9kZXNrLmNvbS9hdWQvZGVhbmVtQ0MCI5Imp0aSI6Im12bmwyZ2tK0EU4Tkds2S2JEVko0S3BHaTRCYkZtRndyUmVrd2NjT3B3RU10TlVTdnZrNnlnNlWSGo3d29WwjmifQ.Niy8dwBQVuhcaCTClZgttJleuKIoQtN8yoT1ZJWgNg&token_type=Bearer&expires_in=86399

Get Data Exchange Info

forge.autodesk.com/en/docs/data/v2/reference/http/projects-project_id-items-item_id-tip-GET/

ACC URL:

<https://acc.autodesk.com/docs/files/projects/8e1bfbc5-7807-4190-90ba-429cb434cd9c?folderUrn=urn%3Aadsk.wipprod%3Afs.folder%3Aco.cFILI3uETZWniC1FJUibvg&entityId=urn%3Aadsk.wipprod%3Adm.lineage%3AZE5IZ5jYTuevINQMmJ5igw&viewModel=detail&moduleId=folders>

Get the latest ("tip") version of a given item (Data Exchange)

```
const axios = require("axios").default;

const getItemTipVersion = async (accessToken, projectId, itemId) => {
  const url = `https://developer.api.autodesk.com/data/v1/projects/${projectId}/items/${encodeURIComponent(itemId)}/tip`;
  const res = await axios.get(url, {
    headers: {
      "Content-Type": "application/json",
      Authorization: "Bearer " + accessToken,
    },
  });
  return res?.data?.data;
};
```


Get Data Exchange Info – Sample Response

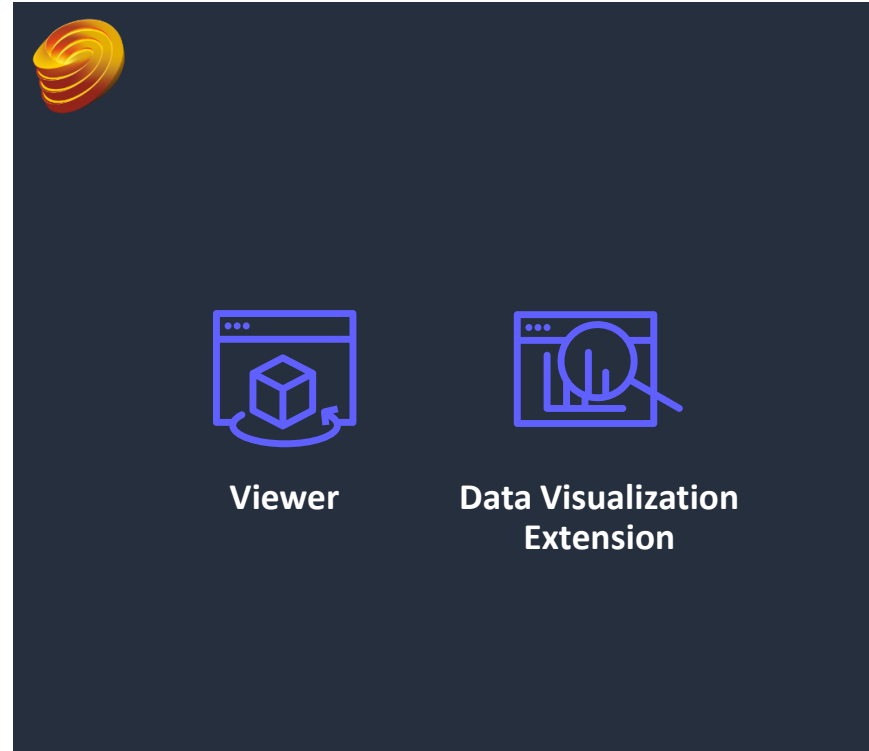
```
const sampleResponse_GetItemTipVersion = {
  data: {
    type: "versions",
    id: "urn:adsk.wiprod:fs.file:vf.FXwwyfYbSCWx10l7cpgk9w?version=1",
    attributes: {
      name: "FDX-Level 1 Lighting",
      displayName: "FDX-Level 1 Lighting",
      createTime: "2022-08-06T04:16:07.0000000Z",
      createUser: "Enrico Chionna",
      ...
    },
    ...
  },
  relationships: {
    ...
    derivatives: {
      data: {
        type: "derivatives",
        id: "dXJu0mFkc2sud2lwcHJvZDpmcy5maWxlOnZmLkZYd3d5Zl1iU0NXeDFPbDdjcGdrOXc_dmVyc2lvbj0x",
        ...
      },
      ...
    },
    ...
  },
  ...
};
```

Data Exchange Name

Data Exchange URN

Add Forge Viewer & Sprites

- **Viewer**
 - Load models in web viewer
- **Data Visualization Extension**
 - Add sprites representing IoT devices



Add Forge Viewer & Sprites

https://forge.autodesk.com/en/docs/dataviz/v1/developers_guide/introduction/

Viewer

The screenshot shows the Autodesk Forge Viewer documentation page. The header includes the Autodesk logo and navigation links: Platform Vision, Solutions, Getting Started, Documentation, Community, Support, Pric, and a SIGN IN button. The main content area is titled 'Using React Components' under the 'Developer's Guide' section. It explains that all React components used by the Reference Application are available via the 'forge-dataviz-iot-react-components' NPM package. A code block shows the command to install the package: 'npm install forge-dataviz-iot-react-components'. Below this, it shows how to import components and styles into a React application. An example code snippet is provided at the bottom, showing the import of the 'Viewer' component and the 'ChronosTimeSlider' component, and the definition of a 'SampleApp' function. A 'Feedback' button is visible in the bottom right corner.

AUTODESK Platform Vision Solutions Getting Started Documentation Community Support Pric Q SIGN IN

Data Visualization Version 1

Documentation / Data Visualization / Developer's Guide

Using React Components

All React components used by the Reference Application are available via the [forge-dataviz-iot-react-components](#) NPM package.

Installing and Using the Components

To install the package:

```
npm install forge-dataviz-iot-react-components
```

To import a component:

```
import { X } from "forge-dataviz-iot-react-components"; // To import component "X"
import "forge-dataviz-iot-react-components/dist/main.bundle.css"; // To import styles
```

Example:

```
// To import the Viewer component
import { Viewer } from "forge-dataviz-iot-react-components";
import { ChronosTimeSlider } from "forge-dataviz-iot-react-components";

function SampleApp(props) {

  return (
```

Developer's Guide

- Introduction
 - The Data Visualization Extension
- QuickStart
- Examples
- Advanced Topics
- NPM Packages
 - Using React Components
 - Using IoT Data Modules
 - Modifying NPM Packages

API Reference

Feedback

Data Visualization Extension (Sprites)

The screenshot shows the Autodesk Forge Data Visualization Extension documentation page. The header includes the Autodesk logo and navigation links: Platform Vision, Solutions, Getting Started, Documentation, Community, Support, Pric, and a SIGN IN button. The main content area is titled 'Quick Source-Code Examples' under the 'Developer's Guide' section. It provides a quick overview of the extension and links to the 'QuickStart' guide. A code block shows the command to install the package: 'npm install forge-dataviz-iot-react-components'. Below this, it shows how to import components and styles into a React application. An example code snippet is provided at the bottom, showing the import of the 'Viewer' component and the 'ChronosTimeSlider' component, and the definition of a 'SampleApp' function. A 'Feedback' button is visible in the bottom right corner.

AUTODESK Platform Vision Solutions Getting Started Documentation Community Support Pric Q SIGN IN

Data Visualization Version 1

Documentation / Data Visualization / Developer's Guide

Quick Source-Code Examples

This section of the Data Visualization QuickStart Guide provides a few quick HTML examples that you can examine and run to learn more about using the Data Visualization extension.

Before running these examples, we recommend first installing and launching the Reference Application, as described in [Running the Reference Application](#). This will make sure that you have the necessary pre-requisites for running the Data Visualization extension, and that you can view and interact with the results.

Adding sprite viewables to a scene

Here is a complete example HTML file that loads a Navisworks model and adds two sprites into the scene. In addition, it adds a smooth camera transition for effect.

```
import { X } from "forge-dataviz-iot-react-components"; // To import component "X"
import "forge-dataviz-iot-react-components/dist/main.bundle.css"; // To import styles
```

Developer's Guide

- Introduction
 - The Data Visualization Extension
- QuickStart
- Examples
- Advanced Topics
- NPM Packages

API Reference

Feedback

Costs - Autodesk



Revit

Architect: PAID

Vendor: FREE



ACC / Docs

Architect: PAID

Vendor: FREE



Forge

Architect: FREE

Vendor: FREE

Costs – AWS

<https://aws.amazon.com/free/>

<div>SECURITY, IDENTITY, & COMPLIANCE</div> <div>Free Tier</div> <div>ALWAYS FREE</div> <div>Amazon Cognito</div> <div>50,000</div> <div>MAUs each month</div> <div>Simple and Secure User Sign-Up, Sign-In, and Access Control.</div> <div>The Your User Pool feature has a Free Tier of 50,000 MAUs each month**</div> <div>10 GB of cloud sync storage. Expires 12 months after sign-up.</div> <div>1,000,000 sync operations per month. Expires 12 months after sign-up.</div>	<div>FRONT-END WEB & MOBILE</div> <div>Free Tier</div> <div>12 MONTHS FREE</div> <div>AWS AppSync</div> <div>250 k</div> <div>query or data modifications per month</div> <div>Develop, secure and run GraphQL APIs at any scale.</div> <div>The Free Tier offers the following monthly usage levels at no charge for 12 months:</div> <div>250,000 query or data modification operations</div> <div>250,000 real-time updates</div> <div>600,000 connection-minutes</div> <div>The AWS AppSync Free Tier automatically expires after 12 months</div>	<div>DATABASE</div> <div>Free Tier</div> <div>ALWAYS FREE</div> <div>Amazon DynamoDB</div> <div>25 GB</div> <div>of storage</div> <div>Fast and flexible NoSQL database with seamless scalability.</div> <div>25 GB of Storage</div> <div>25 provisioned Write Capacity Units (WCU)</div> <div>25 provisioned Read Capacity Units (RCU)</div> <div>Enough to handle up to 200M requests per month.</div>	<div>FRONT-END WEB & MOBILE</div> <div>Free Tier</div> <div>12 MONTHS FREE</div> <div>AWS Amplify Hosting</div> <div>15 GB</div> <div>served per month</div> <div>Fully managed CI/CD and hosting service for fast, secure, and reliable static websites and server-side rendered web apps.</div> <div>Build & Deploy - 1,000 build minutes per month</div> <div>Hosting - 5 GB stored per month & 15 GB served per month</div>
--	---	--	---

Resources and References

- **Autodesk Forge:**
 - <https://forge.autodesk.com>
- **Autodesk Data Exchange:**
 - <https://forge.autodesk.com/blog/data-exchange-released-forge-data-exchange-apis-now-available-public-beta>
 - https://forge.autodesk.com/en/docs/fdx/v1/developers_guide/fd_overview
 - https://forge.autodesk.com/en/docs/data/v2/reference/http/projects-project_id-items-item_id-tip-GET
- **Autodesk Authentication:**
 - <https://forge.autodesk.com/en/docs/oauth/v2/tutorials/get-3-legged-token-implicit>
- **Autodesk Forge Viewer:**
 - https://forge.autodesk.com/en/docs/viewer/v7/developers_guide/overview/
 - https://forge.autodesk.com/en/docs/dataviz/v1/developers_guide/introduction
- **Amplify Framework Documentation:**
 - <https://ui.docs.amplify.aws>
- **Amplify UI:**
 - <https://ui.docs.amplify.aws>
- **Figma:**
 - <https://www.figma.com>
- **React:**
 - <https://reactjs.org>
- **GraphQL:**
 - <https://graphql.org>
 - <https://aws.amazon.com/graphql>
 - <https://aws.amazon.com/graphql/graphql-dynamodb-data-modeling>
- **Cloudscape:**
 - <https://cloudscape.design>
- **AWS Free Tier:**
 - <https://aws.amazon.com/free>



Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2022 Autodesk. All rights reserved.

Autodesk Confidential & Proprietary Information - Please do not post, copy or distribute without authorization.