SiQ API Changes

2021 March

Below details the SIQ API Changes:

1 Mutations

1.1 UpdateCompanySetting

```
mutation UpdateCompanySettingsMutation($input: UpdateCompanySettingsInput!) {
  updateCompanySettings(input: $input) {
    viewPeerFutureBookings
    viewPeerCurrentBookings
    showOtherBookers
  }
}
######### Variables ##########
{
  "input": {
    "showOtherBookers": true,
    "viewPeerFutureBookings": true,
    "viewPeerCurrentBookings": true
  }
}
```

1.2 AddCapacityToLocations

```
mutation AddCapacityToLocationsMutation($input: AddCapacityToLocationsInput!) {
    addCapacityToLocations(input: $input) {
        locations {
      id
      locationCapacity{
        capacity
      }
    }
  }
}


######### Variables #########
{
  "input":{
    "capacity": 40,
    "locationIds":
["TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk"]
  }
}


######### Response #########
{
  "data": {
    "addCapacityToLocations": {
      "locations": [
        {
          "id":
"TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk",
          "locationCapacity": {
            "capacity": 40
          }
        }
      ]
    }
  }
}
```

1.3 UpdateLocationsCapacity

```
mutation UpdateLocationsCapacityMutation($input: UpdateLocationsCapacityInput!)
{
    updateLocationsCapacity(input: $input) {
        locations {
      id
      locationCapacity{
        capacity
      }
    }
  }
}


######### Variables #########
{
  "input":{
    "capacity": 40,
    "locationIds":
["TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk"]
  }
}


######### Response #########
{
  "data": {
    "updateLocationsCapacity": {
      "locations": [
        {
          "id":
"TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk",
          "locationCapacity": {
            "capacity": 80
          }
        }
      ]
    }
  }
}
```

1.4 DeleteLocationsCapacity

```
mutation DeleteLocationsCapacityMutation($input: DeleteLocationsCapacityInput!)
{
      deleteLocationsCapacity(input: $input) {
            locations {
        id
        locationCapacity{
          capacity
        }
      }
    }
}


######### Variables ##########
{
  "input":{
    "locationIds":
["TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk"]
  }
}


######### Response ##########
{
  "data": {
    "deleteLocationsCapacity": {
      "locations": [
        {
          "id":
"TG9jYXRpb24tTG9jYXRpb24uZjJmOTI5ZTQtOTk5MS00OTc2LTliOTYtNGUxMDljOWNmMTlk",
          "locationCapacity": null
        }
      ]
    }
  }
}
```

2 Queries

2.1 CompanySettings Graph

CompanySettings graph now has 3 new boolean settings related to bookings. They can be updated via the normal updateCompanySettingsMutation (example above). The default values for the new fields:`viewPeerFutureBookings`, `viewPeekCurrentBookings`, `showOtherBookers` are false

```
query {
  viewer {
    company {
      settings {
        viewPeerFutureBookings
        viewPeerCurrentBookings
        showOtherBookers
      }
    }
  }
}
```

2.2 Space Graph

The space graph now accepts a toDate for filtering bookings. The argument is optional and accepts an iso8601 string date time format.

```
query spaces(id: $ID!, $fromDate: String, $toDate: String) {
  floor:node(id: $id) {
    ... on Floor {
      spaces {
        bookings(fromDate: $fromDate, toDate: $toDate) {
          id
        }
      }
    }
  }
}
######### Variables #########
{
  "id": "SOME_FLOOR_ID",
  "fromDate": "2021-02-18T19:05:00+00:00",
  "toDate": "2021-02-19T23:59:59+00:00"
}
```

2.3 Company Graph

The company graph has 2 new fields 'locations' and 'locationCapacities".

```
{
  viewer {
    company {
        locations {
          id
          city
          state
          country
        }
        capacityLocations {
        capacity
        locations {
          id
          city
          state
          country
        }
      }
    }
  }
}
```

2.4 ProjectionImport related endpoints

2.4.1 ProjectionImport

The `projectionImport` graph has a new field `skippedCount`. Skipped rows are distinct from failedCount because a projectionImport doesn't require for all growth/absolute values to be entered for projections. Values that aren't entered will default to previously entered values and continue with execution.

2.4.2 ProjectionImportFailed

The `projectionImportFailed` graph has been completely revamped to include relevant information about the import file and what causes a failed row from being imported. This includes a new graph type called `ProjectionImportError` with fields: absoluteAmount, buildingCode, date, departmentName, floorCode, numberAmount, and rowNum. These fields will map to the applicable import file and give more details about the buildingProjectionImport or floorProjectionImport.

```
{
  viewer {
    company {
      projection_imports {
        edges {
          node {
            skippedCount
            failed {
              errors { absoluteAmount building department floor numberAmount }
              object {
                absoluteAmount
                buildingCode
                date
                departmentName
                floorCode
                numberAmount
                rowNum
              }
            }
          }
        }
      }
    }
  }
}
```