TnID v2 API proposal specs

This document presents a draft for TnID V2 API (graphQL)

It’s just a draft so some adjustments may be made during the development, but it should not change significantly. Some functions might be renamed, and some props might be added or removed as necessary.

The company graphql api is accessible at /company.

API call specs - company to company

Search Companies

- It will return a list of companies matching the query and in case queried props are visible to the querier, it will be possible to query by:
  - Legal name (with name field)
  - Brand name (with name field)
  - Profile name (with name field)
  - TaxID
  - Telephone number
  - Email
  - Webpage (social media or other)
- Each result will return a list of companies
- It is possible to pass a combination of queryable props

```
query (
      $name: String
      $taxId: String
      $email: String
      $telephoneNumber: String
      $webpage: String
      $limit: Int
  ) {
      companies (
      name: $name
      taxId: $taxId
      email: $email
      telephoneNumber: $telephoneNumber
      webpage: $webpage
      limit: $limit
      ) {
      id
      legalName
      brandName
      profileName
      taxId
      }
  }
```

Create B2B Connection request

- By providing the invited company ID and the connection type, a connection request will be sent to the invited company

```
mutation (
     $invitedCompanyId: ID!
     $connectionType: B2bConnectionType!
 ) {
     createB2bConnectionRequest (
     invitedCompanyId: $invitedCompanyId
     connectionType: $connectionType
     ) {
     id
     status
     type
     insertedAt
     respondedAt
     updatedAt
     company {
     id
     }
     user {
     id
     }
     invitedCompany {
     id
     }
     }
 }
```

Invite Company

- As much as possible data should be included in the invitation model so the invitee can only verify and confirm entered data (of course, they can change if necessary)
- As part of the invitation, at least one person needs to be included in the invitation so that after they register they can take over the management of the newly created company
- Connection request needs to be confirmed/accepted by the newly created company legal representative

- Most properties are flagged private so the newly created company data is not exposed

```
mutation (
      $company: CompanyInput!
      $representatives: [InviteUserInput!]!
      $connectionType: B2bConnectionType!
  ) {
      createB2bInvite (
      company: $company
      representatives: $representatives
      connectionType: $connectionType
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedCompany {
      id,
      legalName,
      brandName,
      taxId
      }
      }
  }
```

List Organization Types

```
query
      {
      organizationTypes
      {
      id
      displayName
      }
  }

```

List Vertical Types

```
query
      {
      verticalTypes
      {
      id
      displayName
      description
      }
  }
```

List B2B Connections

- Call returns list of connected companies and the relationship type(s) (client, vendor, partner, other)

```
query (
      $includedType: B2bConnectionType
      $excludedType: B2bConnectionType
      $limit: Int
  ) {
      b2bConnections (
      includedType: $includedType
      excludedType: $excludedType
      limit: $limit
      ) {
      id
      type
      createdAt
      updatedAt
      startedAt
      company {
      id
      }
      connectedCompany {
      id
      }
      }
  }

```

List Pending B2B Connection Requests

- Returns a list of pending connection requests this company sent to other companies
- Includes company data, invitation status, and proposed connection type (vendor, client, partner..)

```
query (
      $invitedCompanyId: ID
      $includedType: B2bConnectionType
      $excludedType: B2bConnectionType
      $limit: Int
  ) {
      pendingB2bConnectionRequests (
      invitedCompanyId: $invitedCompanyId
      includedType: $includedType
      excludedType: $excludedType
      limit: $limit
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedCompany {
```

```
      id
      }
      }
  }
```

Revoke B2B Connection Request

- If a company wants to revoke a connection request it can choose from the list of pending requests and revoke it

```
mutation (
      $requestId: ID!
  ) {
      revokeB2bConnectionRequest (
      requestId: $requestId
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedCompany {
      id
      }
      }
  }
```

List Received B2B Connection Requests

- The company can list pending connection requests they need to respond

```
query (
      $invitingCompanyId: ID
      $includedType: B2bConnectionType
      $excludedType: B2bConnectionType
      $limit: Int
  ) {
      receivedB2bConnectionRequests (
      invitingCompanyId: $invitingCompanyId
      includedType: $includedType
      excludedType: $excludedType
      limit: $limit
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedCompany {
      id
      }
      }
  }
```

Respond to B2B Connection Request

- If they receive a connection request this is how they respond to it

```
mutation (
      $requestId: ID!
      $response: ConnectionRequestResponse!
  ) {
      respondB2bConnectionRequest (
      requestId: $requestId
      response: $response
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedCompany {
      id
      }
      }
  }
```

API call specs - company to people

List B2C Connections

- Returns list of b2c connections with information about the connection, the user and the company

```
query (
      $includedType: B2cConnectionType
      $excludedType: B2cConnectionType
      $limit: Int
  ) {
      b2cConnections (
      includedType: $includedType
      excludedType: $excludedType
      limit: $limit
      ) {
      id
      type
      insertedAt
      updatedAt
      startedAt
      company {
      id
      }
      connectedUser {
      id
      }
      }
  }
```

List Pending B2C Connection Requests

- Returns a list of B2C connection requests that the user did not respond to with the proposed relationship type(owner, admin, customer, partner….)

```
query (
      $invitedUserId: ID
      $includedType: B2cConnectionType
      $excludedType: B2cConnectionType
      $limit: Int
  ) {
      pendingB2cConnectionRequests (
      invitedUserId: $invitedUserId
      includedType: $includedType
      excludedType: $excludedType
      limit: $limit
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedUser {
      id
      }
      }
  }
```

List B2C Subscriptions

- Returns a list of B2C subscriptions for the company including the subscription channel, and all the relevant details

```
query (
      $userId: ID
      $includedTopic: B2cSubscriptionTopic
      $excludedTopic: B2cSubscriptionTopic
      $expirationDateBefore: Date
      $expirationDateAfter: Date
      $emailChannelEnabled: Boolean
      $smsChannelEnabled: Boolean
      $voiceChannelEnabled: Boolean
      $limit: Int
  ) {
      b2cSubscriptions (
      userId: $userId
      includedTopic: $includedTopic
      excludedTopic: $excludedTopic
      expirationDateBefore: $expirationDateBefore
      expirationDateAfter: $expirationDateAfter
      emailChannelEnabled: $emailChannelEnabled
      smsChannelEnabled: $smsChannelEnabled
      voiceChannelEnabled: $voiceChannelEnabled
      limit: $limit
      ) {
```

```
    id
    subscriptionTopics
    shareInformationWithTheCompany
    stopAllCommunications
    expirationDate
    notificationStartTime
    notificationEndTime
    timezone
    insertedAt
    updatedAt
    company {
    id
    }
    user {
    id
    }
    email {
    email
    }
    smsTelephoneNumber {
    number
    }
    voiceTelephoneNumber {
    number
    }
    }
}
```

List Pending B2C Subscription Requests

- Returns list of pending B2C subscription requests

```
query (
    $includedType: B2cConnectionType
    $excludedType: B2cConnectionType
    $limit: Int
) {
    pendingB2cConnectionRequests (
    ) {
    id
    type
    status
    insertedAt
    updatedAt
    respondedAt
    company {
    id
    }
    invitedUser {
    id
    }
    user {
    id
    }
    }
}
```

Search User (People)

- Search for users who have existing profiles
- Name field queries full name or username fields

```
query (
      $name: String
      $email: String
      $telephoneNumber: String
      $limit: Int
  ) {
      users (
      name: $name
      email: $email
      telephoneNumber: $telephoneNumber
      limit: $limit
      ) {
      id
      firstName
      lastName
      middleName
```

```
    username
    }
}
```

Invite User (People)

- Provide as much information as possible and create shallow profiles
- Send the invites to those people and connection requests

```
  mutation (
    $user: InviteUserInput!
    $connectionType: B2cConnectionType!
) {
    createB2cInvite (
    user: $user
    connectionType: $connectionType
    ) {
    id
    status
    type
    insertedAt
    respondedAt
    updatedAt
    company {
    id
    }
    user {
    id
    }
    invitedUser {
    id
    firstName
    lastName
    }
    }
}
```

Send B2C Connection Request

- If they find people using search they can send them a connection request

```
mutation (
      $invitedUserId: ID!
      $connectionType: B2cConnectionType!
  ) {
      createB2cConnectionRequest (
      invitedUserId: $invitedUserId
      connectionType: $connectionType
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedUser {
      id
      }
      }
  }
```

Send B2C Subscription Request

- When user found company can send them a subscription request

```
mutation (
      $invitedUserId: ID!
      $requestEmailSubscription: Boolean
      $requestSmsSubscription: Boolean
      $requestVoiceSubscription: Boolean
  ) {
      createB2cSubscriptionRequest (
      invitedUserId: $invitedUserId
      requestEmailSubscription: $requestEmailSubscription
      requestSmsSubscription: $requestSmsSubscription
      requestVoiceSubscription: $requestVoiceSubscription
      ) {
      id
      status
      email_requested
      sms_requested
      voice_requested
      insertedAt
      updatedAt
      respondedAt
      company {
      id
      }
      user {
      id
      }
      invitedUser {
      id
      }
      }
  }
```

Revoke Pending B2C Connection Request

- Find the request on the list and revoke it

```
mutation (
      $requestId: ID!
  ) {
      revokeB2cConnectionRequest (
      requestId: $requestId
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      company {
      id
      }
      user {
      id
      }
      invitedUser {
      id
      }
      }
  }
```

Revoke Pending B2C Subscription Request

- Find the request on the list and revoke it

```
mutation (
      $requestId: ID!
  ) {
      revokeB2cSubscriptionRequest (
      requestId: $requestId
      ) {
      id
      status
      email_requested
      sms_requested
      voice_requested
      insertedAt
      updatedAt
      respondedAt
      company {
```

```
     id
     }
     user {
     id
     }
     invitedUser {
     id
     }
     }
  }
```

List pending C2B Connection Requests

- List pending requests the company needs to respond to

```
query (
     $userId: ID
     $includedType: B2cConnectionType
     $excludedType: B2cConnectionType
     $limit: Int
  ) {
     pendingC2bConnectionRequests (
     userId: $userId
     includedType: $includedType
     excludedType: $excludedType
     limit: $limit
     ) {
     id
     status
     type
     insertedAt
     respondedAt
     updatedAt
     user {
     id
     }
     invitedCompany {
     id
     }
     respondedByUser {
      id
      }
      }
  }
```

Respond C2B Connection Request

- Respond to the connection request received from people

```
mutation (
      $requestId: ID!
      $response: ConnectionRequestResponse!
  ) {
      respondC2bConnectionRequest (
      requestId: $requestId
      response: $response
      ) {
      id
      status
      type
      insertedAt
      respondedAt
      updatedAt
      user {
      id
      }
      invitedCompany {
      id
      }
      respondedByUser {
      id
      }
      }
  }
```

Update company profile

- Update company profile fields

```
mutation (
     $legalName: String
     $brandName: String
     $profileName: String
     $taxId: String
     $yearFounded: Int
     $aboutUs: String
     $metadata: Json
 ) {
     updateCompany (
     legalName: $legalName
     brandName: $brandName
     profileName: $profileName
     taxId: $taxId
     yearFounded: $yearFounded
     aboutUs: $aboutUs
     metadata: $metadata
     ) {
     id
     legalName
     brandName
     profileName
     taxId
     yearFounded
     aboutUs
     metadata
     }
 }
```

API call specs - user

The user graphql api is accessible at /user.

Create Spam Report

- Create Spam report

```
mutation (
    $fromNumber: String!
    $toNumber: String!
    $channelType: SpamReportChannelType!
    $timestamp: NaiveDateTime!
    $issueType: SpamReportIssueType
    $userNote: String
    $messageContent: String
    $metadata: Json
) {
    createSpamReport (
    fromNumber: $fromNumber
    toNumber: $toNumber
    channelType: $channelType
    timestamp: $timestamp
    issueType: $issueType
    userNote: $userNote
    messageContent: $messageContent
    metadata: $metadata
    ) {
    id
    fromNumber
    toNumber
    userNote
    messageContent
    channelType
    issueType
    status
    createdAt
    updatedAt
    timestamp
    metadata
    user {
    id
    }
    }
}
```

List Spam Reports

- List Spam Reports with optional filters

```
query (
    $channelType: SpamReportChannelType
    $issueType: SpamReportIssueType
    $includedStatus: SpamReportStatus
    $excludedStatus: SpamReportStatus
    $limit: Int
) {
    spamReports (
    channelType: $channelType
    issueType: $issueType
    includedStatus: $includedStatus
    excludedStatus: $excludedStatus
    limit: $limit
    ) {
    id
    fromNumber
    toNumber
    userNote
    messageContent
    channelType
    issueType
    status
    createdAt
    updatedAt
    timestamp
    metadata
    user {
    id
    }
    }
}
```

Update user profile

- Update user profile fields

```
mutation (
     $username: String
     $firstName: String
     $lastName: String
     $middleName: String
     $birthdate: Date
     $aboutMe: String
     $metadata: Json
 ) {
     updateUser (
     username: $username
     firstName: $firstName
     lastName: $lastName
     middleName: $middleName
     birthdate: $birthdate
     aboutMe: $aboutMe
     metadata: $metadata
     ) {
     id
     username
     firstName
     lastName
     middleName
     birthdate
     aboutMe
     metadata
     }
 }
```

API call specs - authentication

Company auth

- Create a Client Secret in TnID for the company
- Note the client_id and client_secret values
- Make a request to the /auth/token endpoint of the api based on the example body
- Use the access_token from the response to interact with the graphql api at /company

```
{
   "client_id": CLIENT_ID_VALUE,
   "client_secret": CLIENT_SECRET_VALUE
}
```

User auth

- Make a request to /auth/create_user_otp with the telephone_number of an existing user or a not existing user and optional information
- Upon success an SMS will be sent to the provided number with an otp_code

```
{
   "telephone_number": "3024343434333",
   "first_name": "John",
   "last_name": "Doe",
   "email": "email@address.com"
}
```

- Make a request with the telephone_number and the received otp_code to /auth/token
- Use the access_token from the response to interact with the graphql api at /user

```
{
   "telephone_number": "3024343434333",
   "otp_code": "698125"
}
```