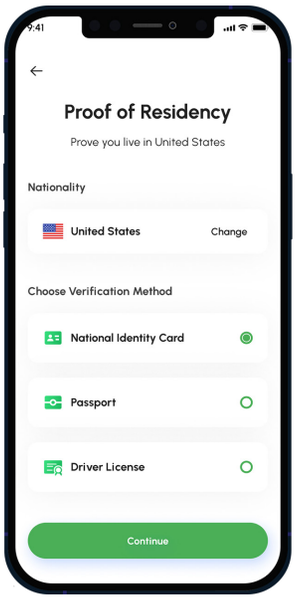
IDVERIFACT
DATA MODEL CONSISTENCY

IDVerifact delivers **data model consistency** across their identity verification APIs for various business use cases such as ***document identity verification, telco based address verification, passive fraud verification based on email, phone or IP address.***

LETS SEE
HOW IT WORKS

Verifying a typical government issued photo ID such as driver's license, IDVerifact would always return the attributes in a consistent hierarchy and attribute below. Below is a sample example for the same.




```
identityCheckResponse
driversLicense
      idData
            idDocumentType
            idRegion
            idCountry
            idNumber
            idBirthDate
            idExpiryDate
      personalData
            firstName
            lastName
            gender
            address
            address1
            address2
            city
            country
```

The data in these attributes can be mapped from any provider, even if they use a different name for them or place them at a different location in the API payload. For e.g.

Provider A might have is as follows:

```
responseCustomerData
      extractedIdData
            idType
            idState
            idCountryCode
            idDateOfBirth
            idDateOfExpiry
```

Provider B might have is as follows:

```
identity_data
      document_id
            id_category
            id_province
            id_country
            id_DOB
            id_Expiry
```

```
=  Unified and consistent error handling resulting in unified error codes
```

IDverifact 80 Richmond St. W, Suite 1401, Toronto, Ontario, M5H 2A3 | info@idverifact.com

IDVERIFACT
REMOVES COMPLEXITY

Customers that are implementing identity verification with multiple providers must understand and integrate with varied API signatures, error codes and message. This results in mapping complexity and managing different API signatures and Error codes. IDVerifact takes away that complexity and delivers a unified consistent harmonized response structure making it simple.

IDVerifact will always have a control data section in the API response that includes the following attributes

```
statusCode - represents the IDVerifact status code for the composite API
statusMessage - represents IDVerifact status message for the composite API
responseDateTime - timestamp when API response was generated
uniqueId - Unique ID of the call requested from the consumer
```

The error section will be present in case any error occurs which will include the following attributes.

```
errorType - representing the type of error which could be Auth, Entitlement,
Validation or Provider errors
errorCode - representing the IDVerifact Error code translated from various providers
description - representing the error description of the error code
downstreamData
      downstreamAdapter - representing the provider adapter that caused the error
      downstreamStatus - Provider error description
```

IDVerifact will always have a consistent response Data which will include the field error type and error code when the data submitted is incorrect from a provider's perspective. Different providers will mostly have different error response signatures and different error codes. Below are some examples.

Provider A might return the error as follows:

```
status_code - 400
status_message - Request could not be processed because of bad data
error_message - Invalid value for the field telephone_number
```

Provider B might return the error as follows:

```
responseCode - 2 (without any error description)

responseCode - 3
responseMessage - Field \"requestId\" is required.
```

Provider C might return the error as follows without any error code:

```
message - The request has incomplete content. List of pos-
sible non-empty elements expected: 'firstName'/'lastName' OR
'scriptNameFullName'.
```

IDVerifact 80 Richmond St. W, Suite 1401, Toronto, Ontario, M5H 2A3 | info@idverifact.com