NORGES IDRETTSFORBUND

API Documentation
External Membership Services

Document Version 1.9

11 December 2020

Table of Content

1 Change History

No	Changes	Date
1	• Remove membership payment and active membership payment entity messages from section Entities Posted to Queue • Remove payment entity JSON schema for membership payment and active membership payment in section Entities JSON schema	06-Feb-2020
2	• In POST Membership, SSN would not be mandatory. • In POST Membership, NationalityID would be used to decide is membership request for Norwegian or Foreign member. • List of Membership Status IDs updated with descriptions. • ToOrgtype JSON added	11-Mar-2020
3	• Added description of how to generate developerUserID • Updated simplified flow for the membership • Updated person merge with all messages related to person merge	22-May-2020
4	• Updated person merge message sequence as per new implementation	23-Jun-2020
5	• Specify MembershipRequest-guidelines for 3.party • Remove SSN in MembershipRequest	14-Oct-2020
6	• Added sections for Corporate Sports APIs	11-Dec-2020

2 Introduction

This document covers the API definitions developed by Norges idrettsforbund og olympiske og paralympiske komité, herewith referred to as NIF for short. NIF API products consists of a set of RESTful services.

This document is written for developers, programmers and the solution providers of third-party membership solutions (herewith referred to as Integration Partners), who seek to sync data with NIF. NIFs requirements must be met to achieve a contract to provide service through NIF APIs to NIF organizations, prior to engaging in a delivery contract with given clubs.

This document also includes how to connect to the APIs and how to read from NIFs enterprise service bus (ESB) hosted on Microsoft Azure cloud.

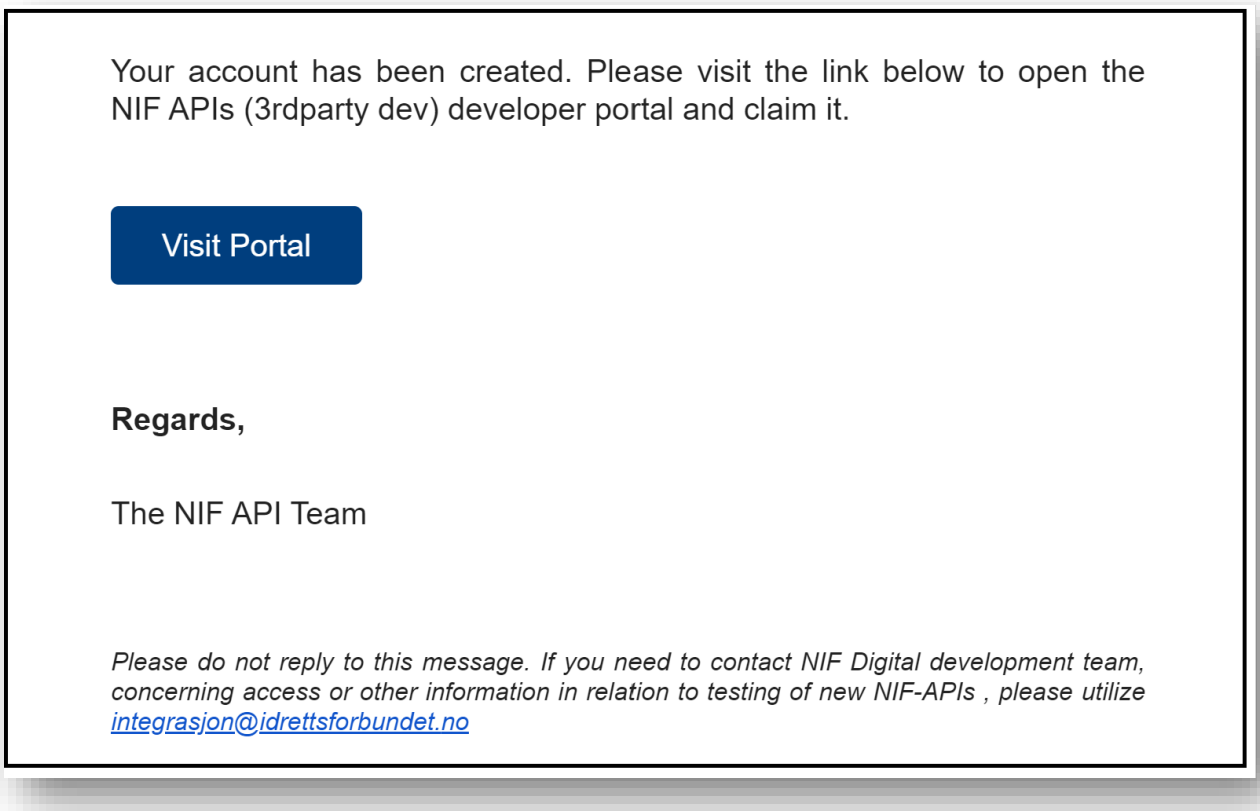
3 Preparations

For any Integration Partner to connect to and consume NIF APIs, the request must go through the following stages:

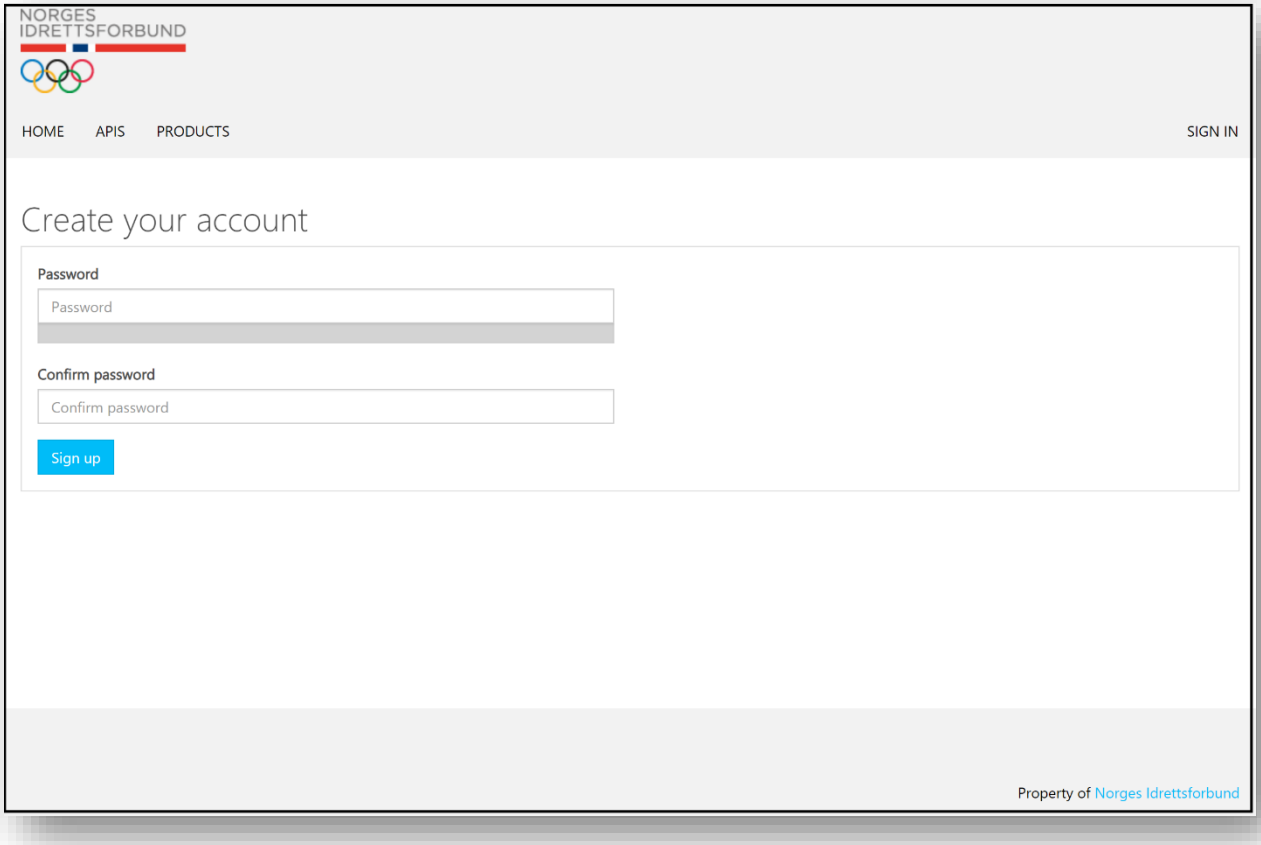
- Identification
- Authentication
- Authorization

3.1 Identification

An Integration Partner must be registered with NIF to be correctly identified. When registered, the main contact person at the Integration Partner receives an invitation email to the NIF API developer portal.

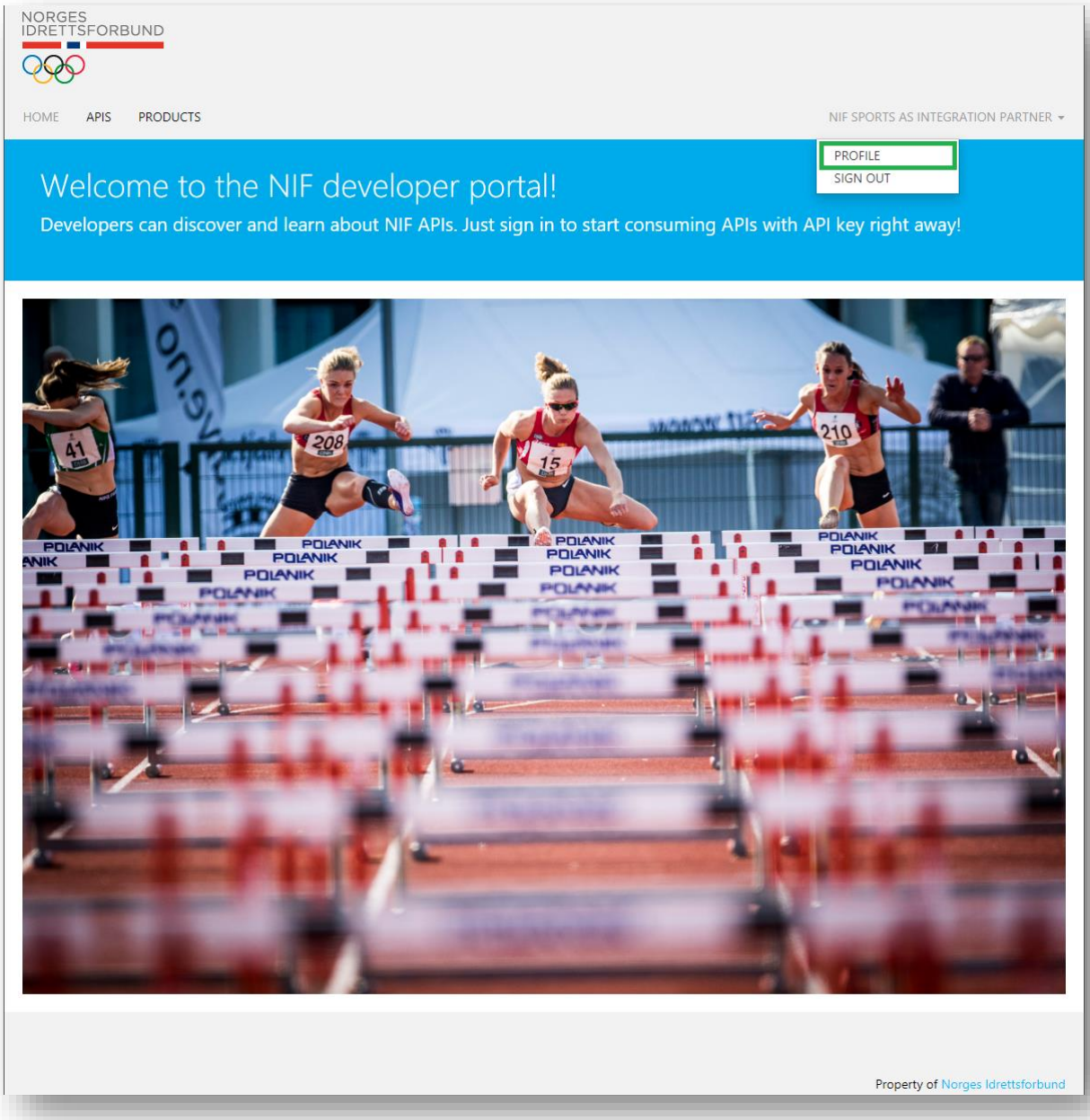


On clicking the link in the email, the Integration Partner can setup their user and login credentials on the NIF API developer portal. These credentials must not be shared with any non-registered entity.

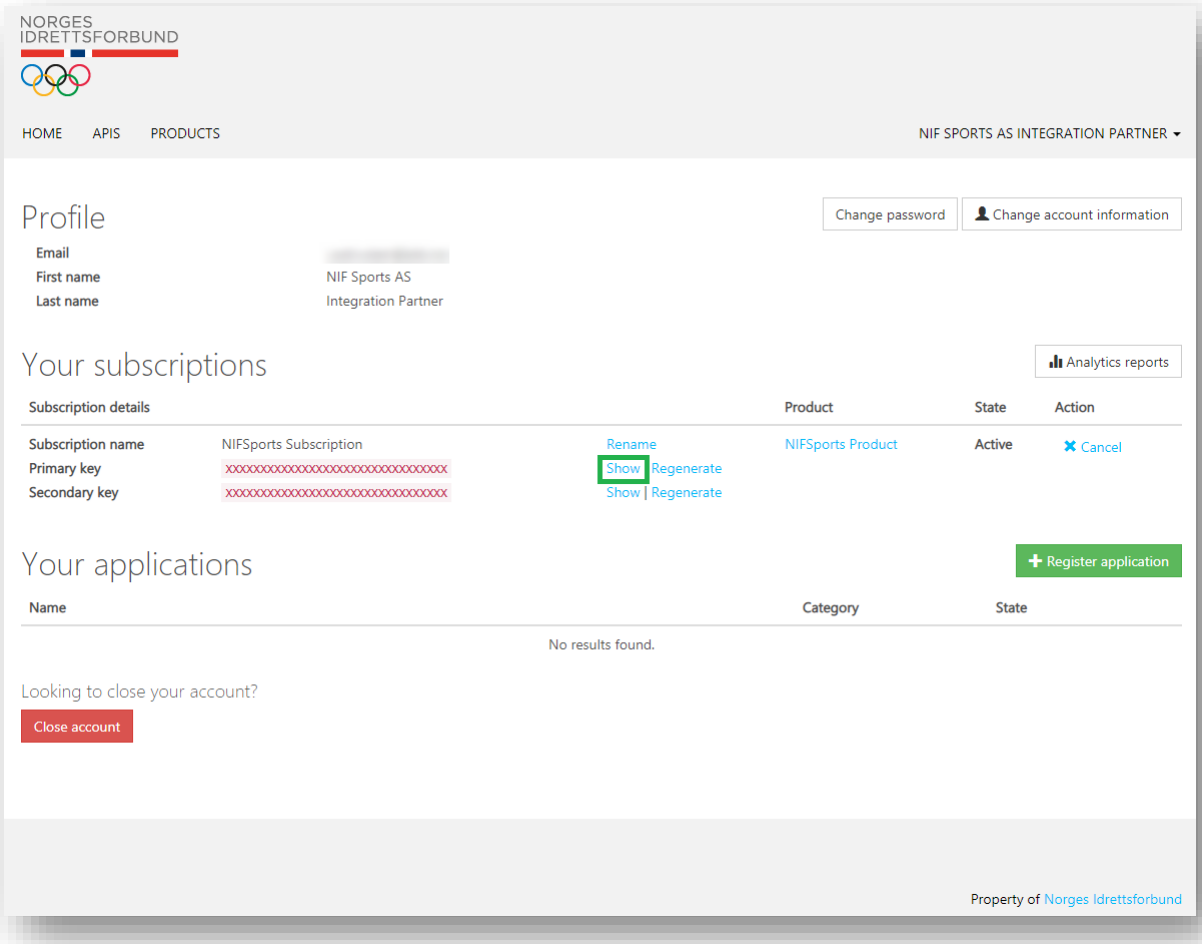

NORGES
IDRETTSFORBUND
HOME
APIS
PRODUCTS
SIGN IN
Create your account
Password
Password
Confirm password
Confirm password
Sign up
Property of Norges Idrettsforbund

3.2 Authentication

On logging into the portal, one may view a list of NIF API products in the NIF API developer portal but will only be able to successfully request the APIs which pass the NIF authentication using this API key.

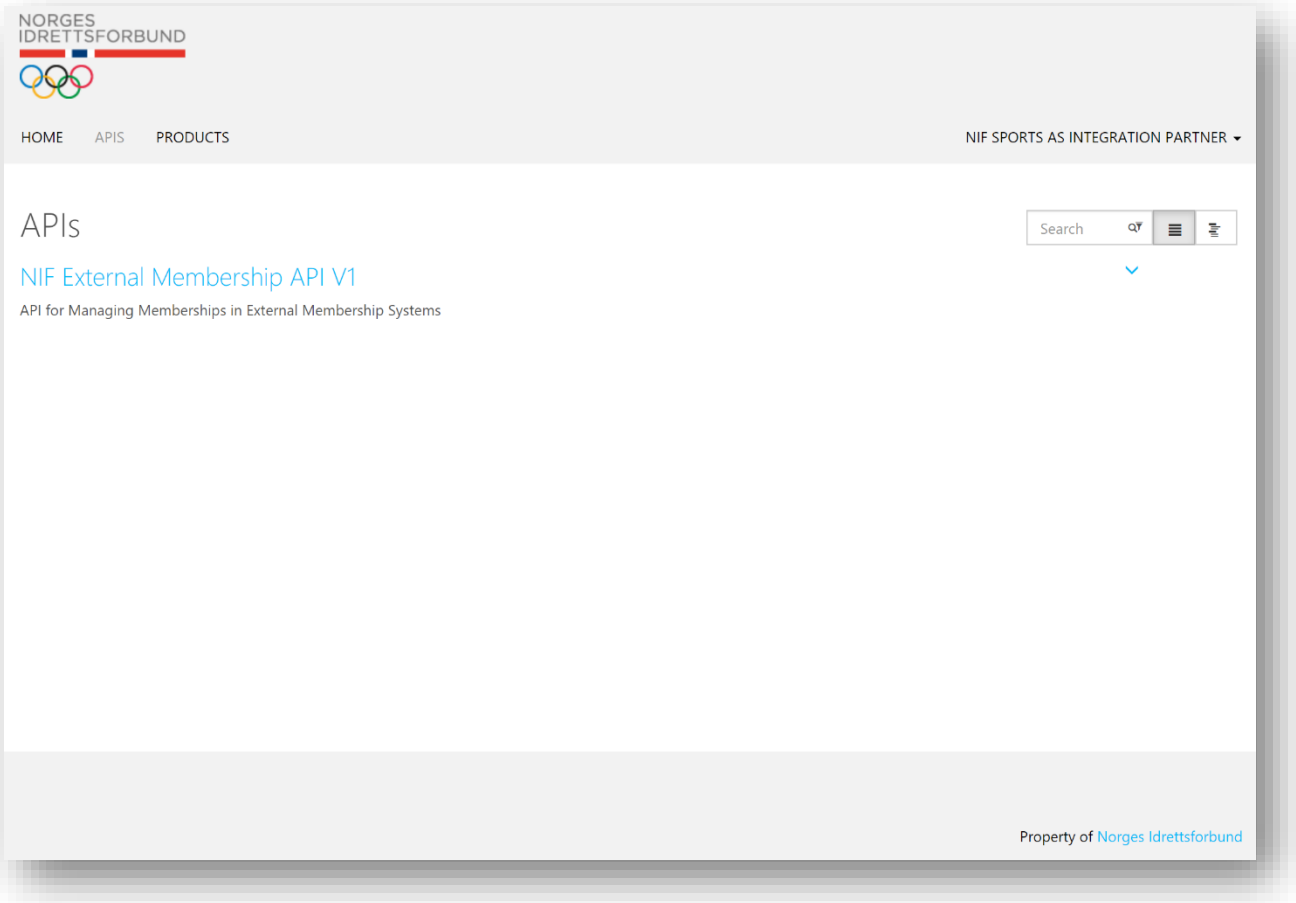


The Integration Partner can find its API Primary key which will be used for its authentication during API requests under API products.


NORGES IDRETTSFORBUND
HOME
APIS
PRODUCTS
NIF SPORTS AS INTEGRATION PARTNER
Profile
Change password
Change account information
Email
First name
NIF Sports AS
Last name
Integration Partner
Your subscriptions
Analytics reports
Subscription details
Product
State
Action
Subscription name
NIFSports Subscription
Rename
NIFSports Product
Active
Cancel
Primary key
Show
Regenerate
Secondary key
Show | Regenerate
Your applications
Register application
Name
Category
State
No results found.
Looking to close your account?
Close account
Property of Norges Idrettsforbund

3.3 Authorization

The Integration Partner needs access token to be authorized in order to request for and manage club data. Club data is highly sensitive and secured, hence, only Integration Partner authorized by the clubs through legal contracts can access respective club data. NIF maintains a registry of clubs and their authorized Integration Partner to generate a unique access token for the match.



This token must be fetched by the Integration Partner by invoking the GET Token API (Gets the bearer token to access APIs for specific club) in the NIF API developer portal. To get token of club Integration Partner needs to provide clubID and version. It is not required to enter developerUserID.

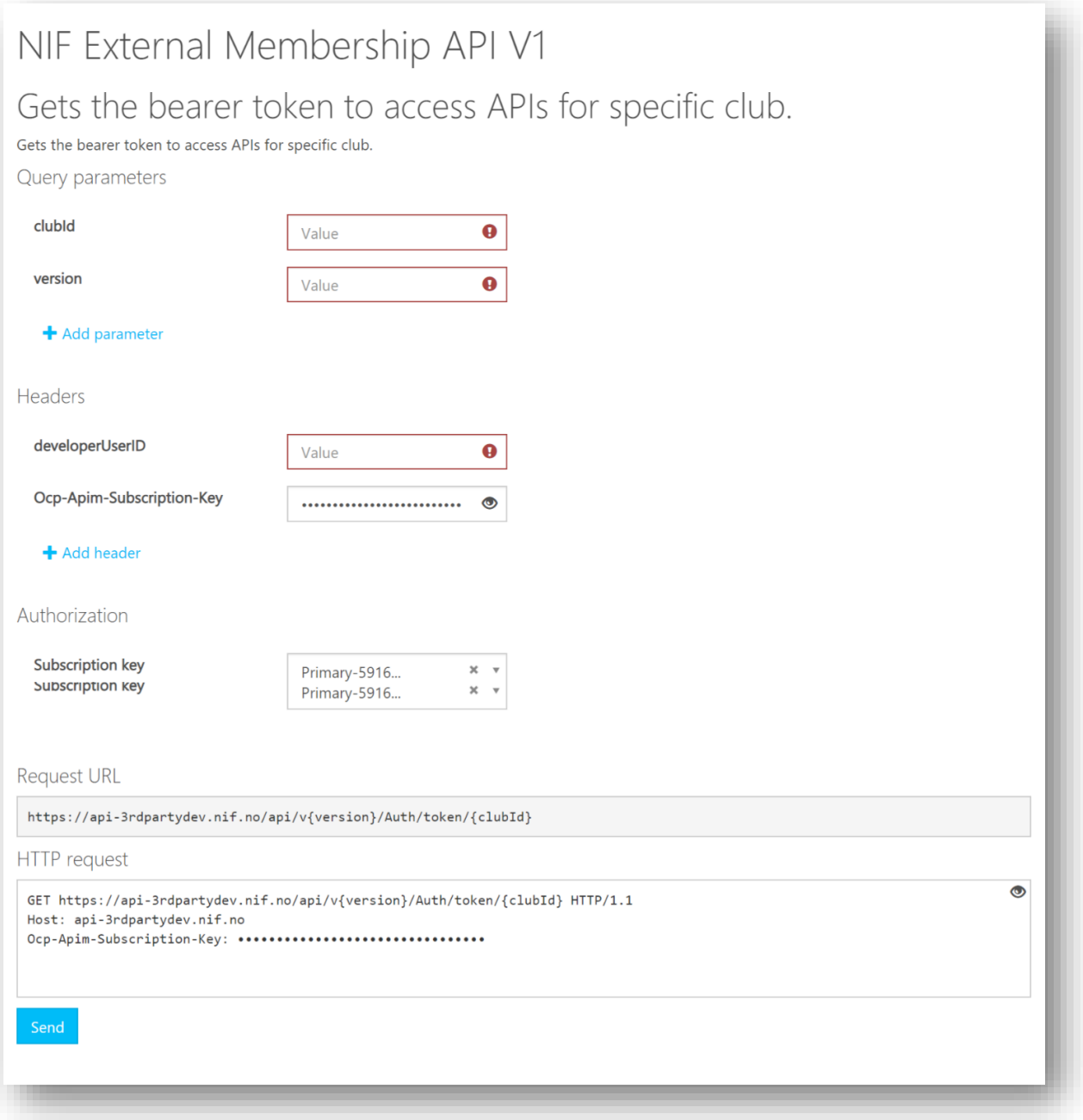
This token must be fetched by the Integration Partner by invoking the GET Token API (Gets the bearer token to access APIs for specific club) in the NIF API developer portal. To get token of club Integration Partner needs to provide clubID and version. It is required to enter developerUserID.

The developerUserId will be created as below:

If the developer email id is donald.duck@idrettsforbundet.no then the developerUserID for this developer will be donald-duck-idrettsforbundet-no

The GET Token API is restricted to the portal and cannot be accessed and/or invoked from outside (e.g. directly through code or web API tools).

NOTE: Along with the API key the access token is highly confidential and must not be shared with unauthorized entities in any way. Both the API key and access token are mandatory parameters in request header of all APIs.



The GET Token API is restricted to the portal and cannot be accessed and/or invoked from outside (e.g. directly through code or web API tools).

NOTE: *Along with the API key the access token is highly confidential and must not be shared with unauthorized entities in any way. Both the API key and access token are mandatory parameters in request header of all APIs*

Enterprise Service Bus Queue

In addition to the NIF API developer portal, the Integration Partner will also receive connection string to NIF's ESB queue in a separate email. Please read section 3.2 for more details on using the ESB queues to listen and consume messages.

4 Communication

There will be two modes of communication between NIF and the Integration Partners:

- NIF API (REST API)
- Microsoft Azure Service Bus Queue

4.1 NIF API (REST API)

The REST API uses HTTP requests to GET, PUT, POST and DELETE data.

Using this communication, the Integration Partner can fetch initial lookup data and provide data changes to NIF through HTTPS communication. Required parameters to be set as header in NIF API requests are:

1. Ocp-Apim-Subscription-Key

This key defines access rights granted to the Integration Partner or NIF APIs. This key is a mandatory parameter in API request header to invoke NIF APIs to identify and authenticate the Integration Partner. Integration Partners can get their subscription key from the API management portal after logging in.

2. Authorization

To authorize an API request the Integration Partner must provide the authorization token in API request header. This authorization token is unique to a club for an integration partner. These keys will define data access granted to the Integration Partner for accessing and managing data on behalf of the club. Kindly refer to sections 2.2 and 2.3 to understand how to get authorization tokens.

4.2 Service Bus Queue

Integration partner will receive data updates from NIF via a dedicated Azure Service Bus Queue.

NIF will publish relevant changes to integration partner scope to the Azure Service Bus Queue. A message with changed object and delta changes is provided. The Integration partner subscribes to the Azure Service Bus Queue to consume posted messages.

Note: *Each Integration partner will have a dedicated instance of Service Bus Queue. All messages concerning clubs who has an engagement with the given integration partner will be published on the Integration Partner's dedicated queue. The messages will be in FIFO (In technical terms It is a Session Enabled Queue).*

Once a message is read from the service bus Queue, it will be removed from the queue. NIF does not keep a history of messages sent to the integration partner over queue. The integration partner must maintain the message history if they need to refer to it later, after reading it from the queue.

Message in the Queue will follow the below mentioned schema:

Message Properties

Property	Type	Value
ObjectId	int	Id of the entity
EntityName	string	Name of the entity.
ChangeType	Change type enumeration	1 - Added 2 - Modified 3 - Deleted 9 - Informational

Note:

1. *Messages with change types 'Added' or 'Modified' have delta object schema and data.*
2. *Informational messages will always have a full object schema and data.*

Message Body

The message body contains the JSON message defined in Queue Message Structure.

Service Bus Queue Guidelines

- Service Bus Queue session is enabled, so the listener must be configured with the option for session.
- Due to the nature of sessions, all messages will be received in FIFO manner.
- The listener can be configured in different modes.
    - *ReceiveAndDelete*: In this mode the message is removed from queue once received.
    - *Peek*: In this mode the messages is not removed from the queue until the receiver mark that Message as "Complete".
- Once the message is removed from the queue it cannot be recovered or replayed.
- The informational messages (ChangeType = 9) will be received before the main change.

5 Enumerations

4.1 Membership Status

Name	Value	Comment
Active	2	Membership is active
Cancelled	4	If membership is cancelled by user or enddate has reached

6 Queue Message Structure

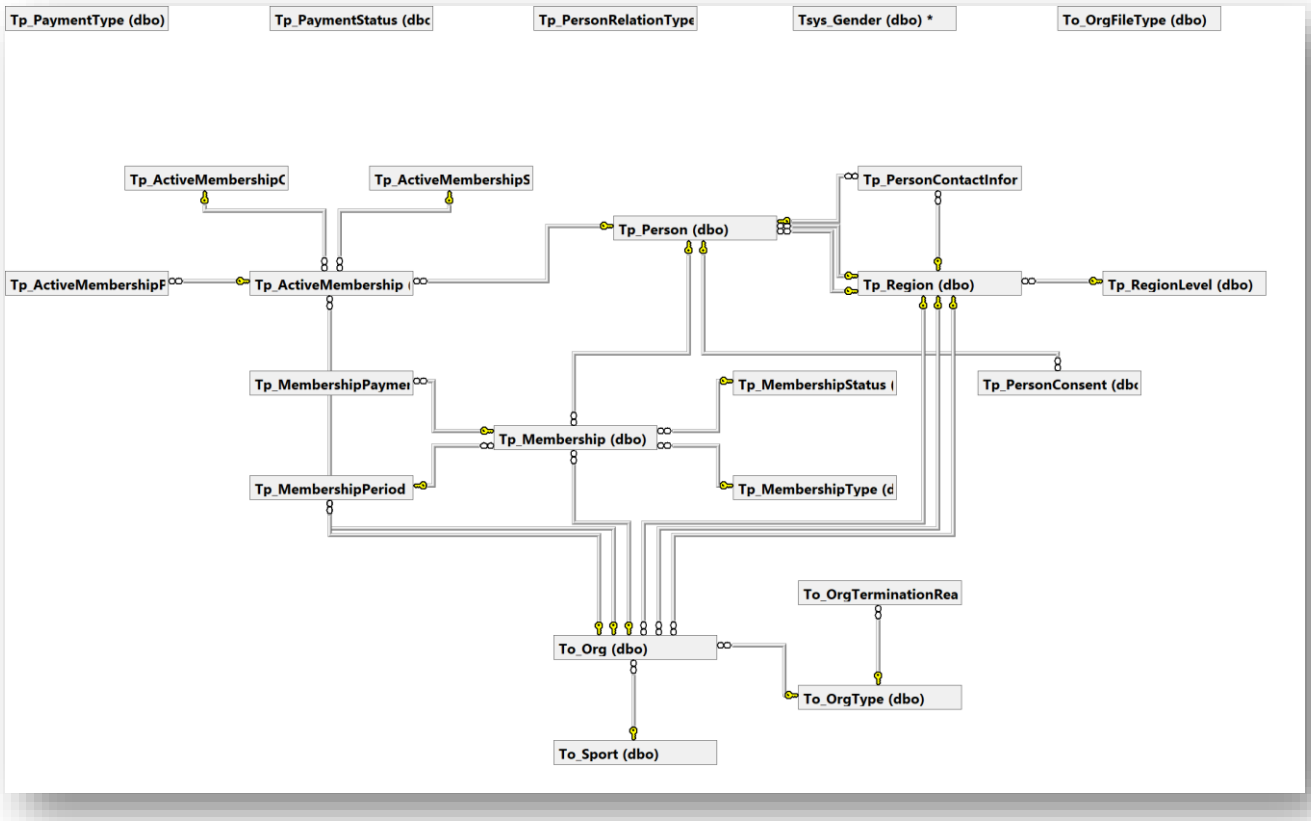
6.1 Entities posted to Queue

No	Entities	Publish Message to Queue
1	TpMembership	When a person is created and then membership is assigned via POST Membership API When a club chooses to switch its Integration Partner, NIF allows an overlapping period between going live on new solution and shutting down integration on old solution. During this period if any membership is registered on either of the solutions, then the other Integration Partner will receive this message in their queue.
2	TpActiveMembership	Similar to the aforementioned scenario, if active membership is registered in either of the Integration Partner solutions during the overlapping period, the other solution will receive this message in their queue.
3	TpPerson	When a person is created through POST Membership API Additionally, similar to the aforementioned scenario, if person is registered in either of the Integration Partner solutions during the overlapping period, the other solution will receive this message in their queue.
4	TpPersonContactInformation	When person contact information is updated.
5	ToOrg	When an organization is created or updated.
6	ToOrgTerminationReason	When an organization is terminated
7	ToOrgType	When there are any changes in org type master/lookup table.
8	ToSport	When any sport is created or updated.
9	TpMembershipPeriod	When club membership Period is updated
10	Tp_PaymentType	When payment is type is added or updated
11	Tp_PaymentStatus	When payment status is updated

6.2 Message Body Schema

No.	Entities JSON Schema
1	TpMembership
2	TpActiveMembership
3	TpPerson
4	TpPersonContactInformation
5	ToOrg
6	ToOrgTerminationReason
7	ToOrgType
8	ToSport
9	TpMembershipPeriod
10	TpPaymentStatus
11	TpPaymentType

6.3 Entities Relationships



The API endpoints allow integration partners to perform operations on a club's behalf. The only prerequisites to use an API endpoint is a valid subscription key and an authorization key. The APIs utilize JSON data format for responses (and in some cases, for requests).

Successful request example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
}
```

Failed request example

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
   "message": "Invalid personId",
   "error code": "1002"
}
```

6.4 General Response Codes

6.4.1 HTTP Success Codes

Code	Text	Description
200	The request has succeeded	The meaning of success depends on the HTTP method: GET: The resource has been fetched and is transmitted in the message body. PUT or POST: The resource describing the result of the action is transmitted in the message body.

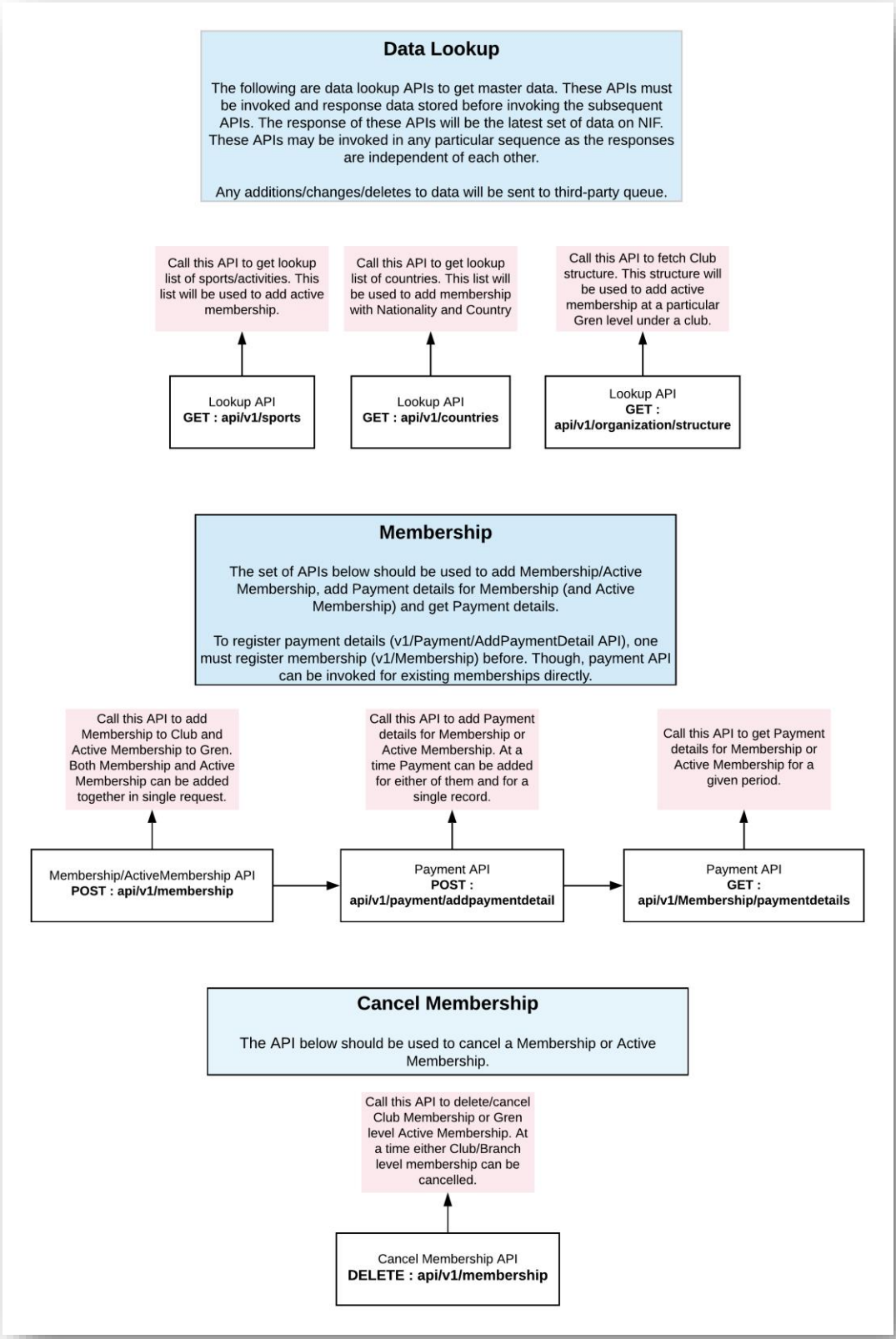
6.4.2 HTTP Error Codes

Code	Text	Description
400	Bad Request	This response means the server did not understand the request due to invalid syntax.
401	Unauthorized	Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.
404	Not Found	The server cannot find the requested resource. In an API, this can also mean that the endpoint is valid but the resource itself does not exist.
500	Internal Server Error	The server has encountered a situation it does not know how to handle.
502	Bad Gateway	This error response means the server, while working as a gateway to get the response needed to handle the request, received an invalid response.

6.4.3 Commonly used Data types

Type	Description	Format
Integer	Whole positive or negative numbers	
Date	Date	yyyy-mm-dd
DateTime	Date and time information	yyyy-mm-dd hh:mm:ss
Decimal	Floating point values	000.00
String	Free text	
Boolean	True/false values	true / false

7 Using APIs


Data Lookup
The following are data lookup APIs to get master data. These APIs must be invoked and response data stored before invoking the subsequent APIs. The response of these APIs will be the latest set of data on NIF. These APIs may be invoked in any particular sequence as the responses are independent of each other.
Any additions/changes/deletes to data will be sent to third-party queue.
Call this API to get lookup list of sports/activities. This list will be used to add active membership.
Call this API to get lookup list of countries. This list will be used to add membership with Nationality and Country
Call this API to fetch Club structure. This structure will be used to add active membership at a particular Gren level under a club.
Lookup API GET : api/v1/sports
Lookup API GET : api/v1/countries
Lookup API GET : api/v1/organization/structure
Membership
The set of APIs below should be used to add Membership/Active Membership, add Payment details for Membership (and Active Membership) and get Payment details.
To register payment details (v1/Payment/AddPaymentDetail API), one must register membership (v1/Membership) before. Though, payment API can be invoked for existing memberships directly.
Call this API to add Membership to Club and Active Membership to Gren. Both Membership and Active Membership can be added together in single request.
Call this API to add Payment details for Membership or Active Membership. At a time Payment can be added for either of them and for a single record.
Call this API to get Payment details for Membership or Active Membership for a given period.
Membership/ActiveMembership API POST : api/v1/membership
Payment API POST : api/v1/payment/addpaymentdetail
Payment API GET : api/v1/Membership/paymentdetails
Cancel Membership
The API below should be used to cancel a Membership or Active Membership.
Call this API to delete/cancel Club Membership or Gren level Active Membership. At a time either Club/Branch level membership can be cancelled.
Cancel Membership API DELETE : api/v1/membership

7.1 GET Token

Overview

Method	API Endpoint	Description
GET	api/v{version}/auth/token/{clubId}	Allows Integration Partner to get authorization token to be used for subsequent API requests. **This API must be called from the Developer portal of Azure API Management**. The representative must login with username and password.

Resource Information

Response Format	JSON
Requires Authentication	No

Response Codes

	Code	Type	Text	Description
1	200	Success		
2	1018	Error	Invalid club Id	
3	401	Error	Unauthorized	

Response Example

[
“token” :
“eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6IkhpdGVzaFBhcnQyIiwiQ2x1YklkIjoiMTk0MzAiLCJDbHViRGV0YWlsIjoiMTk0MzB-NzF-LTQzIiwibmJmIjoxNTY2OTc0OTA0LCJleHAiOjE1NjY5Nzg1MDQsImlhdCI6MTU2Njk3NDkwNCwiaXNzIjoic2VsZiIsImF1ZCI6ImZ1bi10cGkifQ._YXZwmdSMhmxldBew48Ii1BbZAXgGwXAfUPuEbV40c8”
]

7.2 GET Sports

Overview

Method	API Endpoint	Description
GET	/api/v{version}/sports	Allows Integration Partners to download all NIF sports.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested.	
2	Authorization	Yes	This will be the token defining access for an on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. Each Integration Partner is provided a separate and unique key.	

Response Codes

	Code	Type	Text	Description
1	200	Success		

Response Example


```
[
  {
    "sportId": "32",
        "sportName": "Ski",
        "sportDescription": "demo sport description",
    "sports": [
      {
                "sportId": 132,
        "sportName": "Alpint",
        "sportDescription": "demo sport description"
      },
      {
        "sportId": 130,
        "sportName": "Langrenn",
        "sportDescription": "demo sport description"
      }
    ]
  },
  {
    "sportId": "17",
    "sportName": "Friidrett",
    "sportDescription": "demo sport description",
    "sports": [
      {
        "sportId": 220,
        "sportName": "Friidrett på bane",
        "sportDescription": "demo sport description"
      },
      {
        "sportId": 434,
        "sportName": "Løp utenfor bane",
        "sportDescription": "demo sport description"
      }
    ]
  }
]
```

7.3 GET Countries

Overview

Method	API Endpoint	Description
GET	/api/v{version}/countries	Fetches all countries available from NIF.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested	
2	Authorization	Yes	This is the token to define access for an on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. Each Integration Partner is provided a separate and unique key.	

Response Codes

	Code	Type	Text	Description
1	200	Success		

Response Example


```
[
  {
    "countryId": 1500159,
    "countryName": "Peru",
    "isoAlpha2": "PE",
    "isoAlpha3": "PER"
  },
  {
    "countryId": 1500125,
    "countryName": "Monaco",
    "isoAlpha2": "MC",
    "isoAlpha3": "MON"
  },
  {
    "countryId": 1500121,
    "countryName": "Luxembourg",
    "isoAlpha2": "LU",
    "isoAlpha3": "LUX"
  }
]
```

7.4 GET Organization Structure

7.4.1 External Membership API

Overview

Method	API Endpoint	Description
GET	/api/v{version}/organization/structure	Gets the organization structure (branches) of club along with its details.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested.	
2	Authorization	Yes	This will be the token defining access for the on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. Each Integration Partner is provided a separate and unique key.	

Response Codes

	Code	Type	Text	Description
1	200	Success		Clubs, groups and branch(es)

Response Example


```
{
  "clubId": "1111111",
  "clubName": "Heming Idrettslaget",
  "describingName": "Heming Idrettslaget",
  "email": "kontoret@heming.no",
  "city": "Oslo",
  "county": "Norway",
  "longitude": "72.5714° E",
  "latitude": "23.0225° N",
  "groups": [
   {
    "groupId": "221333",
    "groupName": "Heming, IL - Alpint",
    "describingName": "Heming, IL - Alpint",
    "sportId": "32",
    "sportName": "Ski",
    "branches": [
     {
      "describingName": "Heming, IL - Alpint",
      "sportId ": "132",
      "sportName": "Alpint",
      "branchId": "221333",
      "branchName": "Heming, IL - Alpint"
     },
     {
      "branchId": "211424",
      "branchName": "Heming, IL - Hopp",
      "describingName": "Heming, IL - Hopp",
      "sportId": "129",
      "sportName": "Hopp"
     }
    ]
   }
  ]
}
```

7.4.2 Company Sports API

Overview

Method	API Endpoint	Description
GET	/api/v{version}/organization/structure	Gets the organization structure (branches) of Corporate Sports Club's along with its details.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested. Pass Corporate Sport club's ID here to get the org structure for Corporate Sport club.	
2	Authorization	Yes	This will be the token defining access for the on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. Each Integration Partner is provided a separate and unique key.	

Response Codes

	Code	Type	Text	Description
1	200	Success		Clubs, groups and branch(es)

Response Example


```
{
   "clubId": 16235,
   "clubName": "PetrOl B.I.L.",
   "describingName": "PetrOl B.I.L.",
   "email": "franziska.blystad@npd.no",
   "city": "STAVANGER",
   "county": "Norge",
   "longitude": null,
   "latitude": null,
   "groups": [
      {
         "groupId": 766441,
         "groupName": "PetrOl B.I.L.",
         "describingName": "PetrOl B.I.L. - Curling",
         "sportId": 204,
         "sportName": "Curling",
         "branches": []
      }
]
}
```

7.5 POST Membership

7.5.1 External Membership API

Overview

Method	API Endpoint	Description
POST	/api/v{version}/membership	It allows adding a membership for Norwegian or Foreign persons alike using POST Membership. Membership represents the connection of a person with a club and/or its branch for a specific sport. If a person´s nationality is Norwegian, membership request would be considered as Norwegian person request, otherwise request would be considered as a foreign person request. The API adds branch level membership data. It will also add club level membership. If clubId is passed in the request the API adds club level membership. If a branchId is passed, the request adds branch level membership. **If the Person exists in the third party membership solution, the PersonId must be sent in the AddMembership-request.** When the membership request contains PersonId then all the information is fetched from NIF by using PersonId. If PersonId is not passed in the request, then the person is searched for, based on first name, last name, date of birth, gender, postal code and nationality. If no result in this search, a new person will attempt to be created. If the data is imprecise or wrong, and the person most likely already exists, a failure in user verification will arise when end user tries to validate with SSN. If the PersonId is sent and the response is “invalid person id”, the third party is not synchronized correctly. Work around is to send exact person information, according to entry in The National Population Register. Club level membership requests are subject to a two-phase verification active for 24 hours. The verification link is sent to the user via email and phone SMS. This must be passed for request to be fulfilled. After user verification an OTP code is sent by SMS and must be entered to the two-phase temporary site. Once completed request is passed on.

		If a person is already validated in NIFs database, no OTP is required, and the person will not be able to change any personal information in the verification form. All personal information is kept updated by the person itself in Min idrett, by logging in with Idrettens ID. As soon as two-phase verification is completed successfully, person club and branch level membership are added. This API will respond with `traceId` along with other needed parameters. This `traceId` will be used to trace the request response with messages received in enterprise service bus queue. After two-phase verification, person and membership records in queue has `traceId`. In case the person is already validated there is no `traceId` in the message in the queue or the response. If Membership at Club or Branch level system will respond with existing MembershipId/ActiveMembershipId. **Entities to get affected by this API:** Tp_Person Tp_Membership Tp_ActiveMembership

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested.	
2	Authorization	Yes	This is the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	IsClubLevel	Optional	True in case of Club level membership request, false in case of branch level membership.	true/false

2	StartDate	Optional	StartDate for membership is required if ClubId is not provided, the system will ignore this field when ClubId is provided. **Possible Value** Must be of today's date year. For example, if today's date is 19-Sep-2019. Then start date must be in between 1-Jan-**2019** to 31-Dec-2019.	"2019-06-07"
3	Person	Mandatory		
4	PersonId	Optional	If PersonId is known	1234567
5	FirstName	Optional Max Length 50	Required if PersonId is not provided, system will ignore it if PersionId is provided.	"Ola"
6	LastName	Optional Max Length 50	Required if PersonId is not provided. The system will ignore if PersionId is provided.	"Carter"
7	BirthDate	Optional	Required if PersonId is not provided. The system will ignore if PersionId is provided	"1980-07-05"
8	PostCode	Optional Max Length 50	Required if PersonId is not provided The system will ignore if PersionId is provided	"879534"
9	Email	Required Min Length 8 Max Length 100	Either of email or mobile is required The system will ignore if PersionId is provided	"helge.carter@nic.com"
10	MobilePhone	Required Min Length 8 Max Length 12	Either of Email or Mobile is required System will ignore if PersionId is provided	"8556974521"
11	CountryCode	Optional Min Length 2 Max Length 6	CountryCode is required in the case MobilePhone is provided, and if PersonId is not provided. System will ignore if PersionId is provided.	+47 , +91

12	GenderId	Optional	Required if PersonId is not provided. System will ignore if PersionId is provided. **Possible Value** 1 - Mann 2 - Kvinne 5 - Ukjent	2
14	CountryId	Optional (when PersonId is supplied/Membership is being added for a Norwegian citizen)	Required if PersonId is not provided, system will ignore if PersionId is provided.	1500152
15	NationalityId	Required	Nationality of person is required. (Pass the country id from get counties API)	1500152
16	AddressLIne1	Optional Max Length 50	First line of address	"Blålyngveien 898209 Fauske"
17	AddressLIne2	Optional Max Length 50	Second line of address	"Blålyngveien 898609 Oslo"
18	City	Optional Max Length 50	City of residence	"Oslo"
19	PrivatePhone	Optional Min Length 8 Max Length 18	Private phone number	"78965872"
20	WorkPhone	Optional Min Length 8 Max Length 18	Work phone number	"78965872"
21	IsSecretAddress	Optional	If the person has configured to hide their address from viewership on the UI, this flag will be set to true. When set to true, it is the responsibility of the Integration	true/false

			Partner to hide this information from viewership on their UI.	
22	IsSecretPrivatePhone	Optional	Similar to aforementioned scenario, if the person's private phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
23	IsSecretWorkPhone	Optional	Similar to aforementioned scenario, if the person's work phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
24	IsSecretMobilePhone	Optional	Similar to aforementioned scenario, if the person's mobile phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
25	IsSecretEmail	Optional	Similar to aforementioned scenario, if the person's email is marked as secret, the Integration Partner must also hide it from their UI.	true/false
26	Sports	Optional	Branch level Membership	[]
27	BranchId	Mandatory		1238
28	SportId	Mandatory	(Pass the sport id from get sports API)	
29	StartDate	Mandatory	StartDate for branch level Membership **Possible Value** 1) Must be of today's date year. For example, if today's date is 19-Sep-2019. Then start date must be in between 1-Jan-2019 to 31-Dec-2019 2) Must be greater than Club level membership start date. 3) Must be of Club level membership's start date year. For example, if Club level membership's start date is 19-Sep-2019. Then start date must be in between 1-Jan-2019 to 31-Dec-2019.	
30	EndDate	Optional	**Possible Value** 1) Must be greater than today's date.	

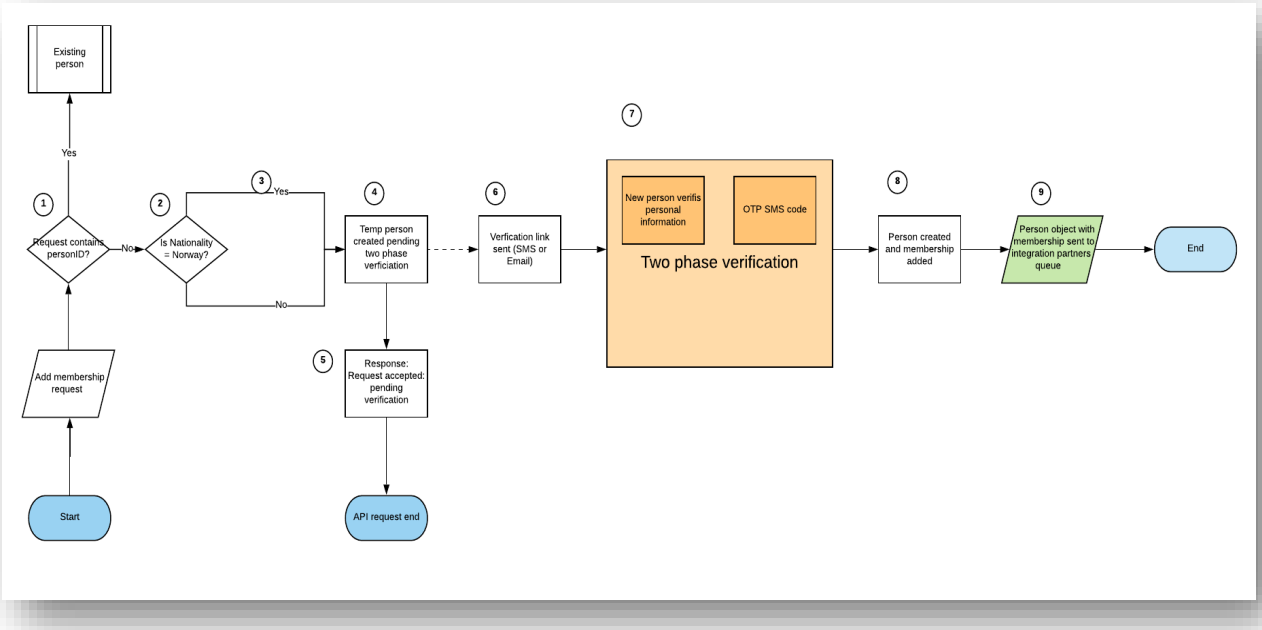
NORGES IDRETTSFORBUND

Two-phase verification link and OTP matrix

	Cases	Send two-phase link	OTP SMS Code	Note
1	Add Club level membership with new person (Norwegian)	Yes	Yes	
2	Add Club level membership with new person (Non-Norwegian)	Yes	No	
3	Add Club level membership with validated existing person	Yes	No	
4	Add Club level membership with unvalidated existing person (Norwegian)	Yes	Yes	
5	Add Club level membership with unvalidated existing person (Non-Norwegian)	Yes	No	
6	Add Club level and Branch level membership with new person (Norwegian)	Yes	Yes	
7	Add Club level and Branch level membership with new person (Non-Norwegian)	Yes	No	
8	Add Club level and Branch level membership with validated existing person	Yes	No	
9	Add Club level and Branch level membership with unvalidated existing person (Norwegian)	Yes	Yes	
10	Add Club level and Branch level membership with unvalidated existing person (Non-Norwegian)	Yes	No	
11	Add Branch level membership unvalidated existing person	No	No	
12	Add Branch level membership with validated existing person	No	No	

13	Add Branch level membership without club level membership with new Person	No	No	Branch level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
14	Add Branch level membership without club level membership for validated existing person	No	No	Branch level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
15	Add Branch level membership without club level membership for unvalidated existing person (Norwegian)	No	No	Branch level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
16	Add Branch level membership without club level membership for unvalidated existing person (Non-Norwegian)	No	No	Branch level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.

Simplified add member flow



Response codes

	Code	Text	Description
1	200		
2	1001	Either PersonId or a combination of (FirstName, LastName, BirthDate, PostCode, NationalityId) must be supplied.	
3	1002	Invalid Person Id.	
4	1006	Invalid Branch Id.	
5	1007	Start date cannot be past/future year date.	
6	1010	Either Email or MobilePhone and CountryCode must be supplied.	
7	1013	Branch level membership information missing.	
8	1018	Invalid club Id.	
9	1034	Person not found.	
10	1036	Start date is required.	
11	1038	Branch start date must be grater then membership start date.	
12	1039	Invalid request parameters.	

13	1042	Group level membership for Club is not allowed.	
14	3001	Membership not found for specific period from date.	

Request Example


```
{
  "startDate": "2019-08-27",
  "isClubLevel": true,
  "person": {
    "personId": 8756934,
    "firstName": "Helge",
    "lastName": "Carter",
    "birthDate": "1980-07-05",
    "postCode": "0010",
    "email": "helge.carter@nic.com",
    "mobilePhone": "85569745",
    "genderId": 2,
    "ssn": "",
    "countryId": 1500152,
    "nationalityId": 1500152,
    "addressLine1": "Blålyngveien 898209 Fauske",
    "addressLine2": "Blålyngveien 898609 Oslo",
    "city": "Oslo",
    "privatePhone": "78553987",
    "workPhone": "78553977",
    "isSecretAddress": true,
    "isSecretWorkPhone": true,
    "isSecretMobilePhone": false,
    "isSecretEmail": true,
    "countryCode": "+47",
    "isSecretPrivatePhone": true
  },
  "sports": [
    {
      "branchId": 827527,
      "sportId": 220,
      "startDate": "2019-08-27",
      "endDate": "2019-09-26"
    },
    {
      "branchId": 819760,
      "sportId": 207,
      "startDate": "2019-08-27",
      "endDate": "2019-09-26"
    }
  ]
}
```

Response Example


```
{
  "personId": 8756934,
  "membershipId": 0,
  "clubId": 19428,
  "clubName": "Gjermundsen Auto B.I.L.",
  "startDate": "2019-08-27",
  "traceId": "7AC07206-471E-4015-9ED7-E6DB55ED7201",
  "sports”: [
    {
      "branchId": 819760,
      "sportId": 207,
      "branchName": "Berg IL",
      "activeMembershipId": 0
    },
    {
      "branchId": 827527,
      "sportId": 220,
      "branchName": "Berg IL",
      "activeMembershipId": 0
    }

  ],
  "responseMessage": "Membership request is accepted at Club Level. It is pending
for two phase verification."
}
```

7.5.2 Corporate Sports API

Overview

Method	API Endpoint	Description
POST	/api/v{version}/membership	It allows adding a membership for Norwegian or Foreign persons alike using POST Membership. Membership represents the connection of a person with a club and/or its branch for a specific sport. If a person´s nationality is Norwegian, membership request would be considered as Norwegian person request, otherwise request would be considered as a foreign person request. The API adds group level membership data. It will also add club level membership. If clubId is passed in the request the API adds club level membership for corporate Sports club. If a groupId is passed, the request adds group level membership. If branchId is passed in the request for Corporate Sports club then the response will show an error. **If the Person exists in the Corporate Sports membership solution, the PersonId must be sent in the AddMembership-request.** When the membership request contains PersonId then all the information is fetched from NIF by using PersonId. If PersonId is not passed in the request, then the person is searched for, based on first name, last name, date of birth, gender, postal code and nationality. If no result in this search, a new person will attempt to be created. If the data is imprecise or wrong, and the person most likely already exists, a failure in user verification will arise when end user tries to validate with SSN. If the PersonId is sent and the response is “invalid person id”, the third party is not synchronized correctly. Work around is to send exact person information, according to entry in The National Population Register. Club level membership requests are subject to a two-phase verification active for 24 hours. The verification link is sent to the user via email and phone SMS. This must be passed for request to be fulfilled. After user verification an OTP code is sent by SMS and must be entered to the two-phase temporary site. Once completed request is passed on.

		If a person is already validated in NIFs database, no OTP is required, and the person will not be able to change any personal information in the verification form. All personal information is kept updated by the person itself in Min idrett, by logging in with Idrettens ID. As soon as two-phase verification is completed successfully, person club and group level membership are added. This API will respond with `traceId` along with other needed parameters. This `traceId` will be used to trace the request response with messages received in enterprise service bus queue. After two-phase verification, person and membership records in queue has `traceId`. In case the person is already validated there is no `traceId` in the message in the queue or the response. If Membership at Club or group level system will respond with existing MembershipId/ActiveMembershipId. **Entities to get affected by this API:** Tp_Person Tp_Membership Tp_ActiveMembership

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Corporate Sports Club's Id for which API is being requested. Pass Corporate Sports club's club ID here to post membership request for Corporate Sports Club or its group.	
2	Authorization	Yes	This is the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example

1	IsClubLevel	Optional	True in case of Club level membership request, false in case of Group level membership.	true/false
2	StartDate	Optional	StartDate for membership is required if ClubId is not provided, the system will ignore this field when ClubId is provided. **Possible Value** Must be of today's date year. For example, if today's date is 19-Sep-2019. Then start date must be in between 1-Jan-**2019** to 31-Dec-2019.	"2019-06-07"
3	Person	Mandatory		
4	PersonId	Optional	If PersonId is known	1234567
5	FirstName	Optional Max Length 50	Required if PersonId is not provided, system will ignore it if PersionId is provided.	"Ola"
6	LastName	Optional Max Length 50	Required if PersonId is not provided. The system will ignore if PersionId is provided.	"Carter"
7	BirthDate	Optional	Required if PersonId is not provided. The system will ignore if PersionId is provided	"1980-07-05"
8	PostCode	Optional Max Length 50	Required if PersonId is not provided The system will ignore if PersionId is provided	"879534"
9	Email	Required Min Length 8 Max Length 100	Either of email or mobile is required The system will ignore if PersionId is provided	"helge.carter@nic.com"
10	MobilePhone	Required Min Length 8 Max Length 12	Either of Email or Mobile is required System will ignore if PersionId is provided	"8556974521"
11	CountryCode	Optional Min Length 2 Max Length 6	CountryCode is required in the case MobilePhone is provided, and if PersonId is not provided.	+47 , +91

			System will ignore if PersionId is provided.	
12	GenderId	Optional	Required if PersonId is not provided. System will ignore if PersionId is provided. **Possible Value** 1 - Mann 2 - Kvinne 5 - Ukjent	2
14	CountryId	Optional (when PersonId is supplied/Membership is being added for a Norwegian citizen)	Required if PersonId is not provided, system will ignore if PersionId is provided.	1500152
15	NationalityId	Required	Nationality of person is required. (Pass the country id from get counties API)	1500152
16	AddressLIne1	Optional Max Length 50	First line of address	"Blålyngveien 898209 Fauske"
17	AddressLIne2	Optional Max Length 50	Second line of address	"Blålyngveien 898609 Oslo"
18	City	Optional Max Length 50	City of residence	"Oslo"
19	PrivatePhone	Optional Min Length 8 Max Length 18	Private phone number	"78965872"
20	WorkPhone	Optional Min Length 8 Max Length 18	Work phone number	"78965872"

21	IsSecretAddress	Optional	If the person has configured to hide their address from viewership on the UI, this flag will be set to true. When set to true, it is the responsibility of the Integration Partner to hide this information from viewership on their UI.	true/false
22	IsSecretPrivatePhone	Optional	Similar to aforementioned scenario, if the person's private phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
23	IsSecretWorkPhone	Optional	Similar to aforementioned scenario, if the person's work phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
24	IsSecretMobilePhone	Optional	Similar to aforementioned scenario, if the person's mobile phone is marked as secret, the Integration Partner must also hide it from their UI.	true/false
25	IsSecretEmail	Optional	Similar to aforementioned scenario, if the person's email is marked as secret, the Integration Partner must also hide it from their UI.	true/false
26	Sports	Optional	Branch level Membership	[]
27	BranchId	Mandatory		1238
28	SportId	Mandatory	(Pass the sport id from get sports API)	
29	StartDate	Mandatory	StartDate for branch level Membership **Possible Value** 1) Must be of today's date year. For example, if today's date is 19-Sep-2019. Then start date must be in between 1-Jan-2019 to 31-Dec-2019 2) Must be greater than Club level membership start date. 3) Must be of Club level membership's start date year. For example, if Club level membership's start date is 19-Sep-2019. Then start date must be in between 1-Jan-2019 to 31-Dec-2019.	
30	EndDate	Optional	**Possible Value** 1) Must be greater than today's date.	

NORGES IDRETTSFORBUND

Two-phase verification link and OTP matrix

	Cases	Send two-phase link	OTP SMS Code	Note
1	Add Club level membership with new person (Norwegian)	Yes	Yes	
2	Add Club level membership with new person (Non-Norwegian)	Yes	No	
3	Add Club level membership with validated existing person	Yes	No	
4	Add Club level membership with unvalidated existing person (Norwegian)	Yes	Yes	
5	Add Club level membership with unvalidated existing person (Non-Norwegian)	Yes	No	
6	Add Club level and Group level membership with new person (Norwegian)	Yes	Yes	
7	Add Club level and Group level membership with new person (Non-Norwegian)	Yes	No	
8	Add Club level and Group level membership with validated existing person	Yes	No	
9	Add Club level and Group level membership with unvalidated existing person (Norwegian)	Yes	Yes	
10	Add Club level and Group level membership with unvalidated existing person (Non-Norwegian)	Yes	No	
11	Add Group level membership unvalidated existing person	No	No	
12	Add Group level membership with validated existing person	No	No	

13	Add Group level membership without club level membership with new Person	No	No	Group level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
14	Add Group level membership without club level membership for validated existing person	No	No	Group level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
15	Add Group level membership without club level membership for unvalidated existing person (Norwegian)	No	No	Group level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.
16	Add Group level membership without club level membership for unvalidated existing person (Non-Norwegian)	No	No	Group level membership cannot be added without club level membership. Hence two-phase verification would not be initiated.

Response codes

	Code	Text	Description
1	200		
2	1001	Either PersonId or a combination of (FirstName, LastName, BirthDate, PostCode, NationalityId) must be supplied.	
3	1002	Invalid Person Id.	
4	1006	Invalid Branch Id.	
5	1007	Start date cannot be past/future year date.	
6	1010	Either Email or MobilePhone and CountryCode must be supplied.	
7	1018	Invalid club Id.	

8	1034	Person not found.	
9	1036	Start date is required.	
10	1039	Invalid request parameters.	
11	1041	Branch level membership for Corporate Club is not allowed.	
12	1042	Group level membership for Club is not allowed.	
13	1043	Group level membership information missing.	
14	1044	Invalid Corporate Club Group Id.	
15	1045	Group start date must be greater than membership start date.	
16	3001	Membership not found for specific period from date.	

Request Example


```
{
  "isClubLevel": true,
  "startDate": "2020-01-01T00:00:00",
  "person": {
    "firstName": "ola1",
    "lastName": "trump1",
    "birthDate": "1993-12-13",
    "postCode": "0010",
    "email": "ola.t1@mailinator.com",
    "mobilePhone": "",
    "countryCode": "",
    "genderId": 1,
    "ssn": "",
    "countryId": 1500152,
    "nationalityId":1500093,
    "addressLine1": "testaddress",
    "addressLine2": "testaddress2",
    "city": "Rena"
    "privatePhone": "78553987",
    "workPhone": "78553977",
    "isSecretAddress": true,
    "isSecretWorkPhone": true,
    "isSecretMobilePhone": false,
    "isSecretEmail": true,
    "countryCode": "+47",
    "isSecretPrivatePhone": true
  },
   "groups": [
    {
      "groupId": 810529,
      "sportId": 130,
      "startDate": "2020-01-01T00:00:00"
    }
  ]
}
```

Response Example


```
{
    "personId": null,
    "updatedPersonId": null,
    "membershipId": null,
    "clubId": 16235,
    "clubName": "PetrOl B.I.L.",
    "startDate": "2020-01-01T00:00:00",
    "traceId": "d72a124d-6c8c-4afc-9c25-f840c8c4d4eb",
    "sports": [],
    "groups": [],
    "person": null,
    "responseMessage": "Membership request is accepted at Club Level. It is
pending for two phase verification."
}
```

7.6 POST Payment detail

7.6.1 External Membership API

Overview

Method	API Endpoint	Description
POST	/api/v{version}/membership/ Addpaymentdetail	Add payment details for a club level membership and/or branch level membership. This API will also change membership to paid. **Entities to get affected by this API** Tp_MembershipPayment Tp_ActiveMembershipPayment Note: Integration Partner should neglect “Is Paid” Parameter.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested	
2	Authorization	Yes	This is the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management’s subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	Amount	Mandatory	Membership charge	
2	Comment	Optional Max Length 200		
3	ExternalId	Optional Max Length 25	Transaction reference number.	

4	IsFamilyMember	Optional	If the payment is being posted for a family member	
5	IsPaid	Mandatory	If payment is already paid.	
6	OrgId	Mandatory	Payment type 1 ClubId, Payment type 2 BranchId	
7	PaidDate	Mandatory	Date of payment **Possible Value** 1) Must be greater than today's date.	
8	PaidToOrgId	Mandatory	Organization receiving the payment (club)	
9	PaymentCategory	Mandatory Max Length 127	Eg. Online, Invoice	
10	PaymentTypeId	Mandatory	0 - Paid for Membership, 1 - License, 2 -Training fee, 3 - Registration fee 4 - Other	
11	PeriodFrom	Mandatory	Membership from date **Possible Value** 1) If PaymentType is membership then Year of PeriodFrom and PeriodTo must be the same. 2) If PaymentType is training fee then PeriodFrom must be greater than membership from date.	
12	PeriodTo	Mandatory	Membership to date **Possible Value** 1) Must be greater than PeriodFrom date.	
13	PersonId	Mandatory	Person for which payment is being done.	

Response Codes

	Code	Text	Description
1	200	Payment details added	

2	3000	Period from and period to are not having same year.	
3	3001	Membership not found for specific period from date.	
4	3006	Invalid period from date and period to date.	
5	3007	Invalid paid date.	
6	3008	Invalid org id.	
7	3009	Invalid payment type.	
8	3010	Invalid Period	
9	1006	Invalid branchId.	

Request Example


```
{
  "amount": "450",
  "comment": "This was paid for club membership",
  "externalId": "3248",
  "isFamilyMember": "true",
  "isPaid": "false",
  "orgId": "7896",
  "paidDate": "2015-05-08",
  "paidToOrgId": "4568",
  "paymentCategory": "Club membership",
  "paymentTypeId": 0,
  "periodFrom": "2015-05-08",
  "periodTo": "2015-06-08",
  "personId": "8675682"
}
```


Response Example


```
{
  "message": "string""Payment details added for membership of {personId} with {orgId} for period {periodFrom} to {periodTo}",
  "updatedPersonId": 0,
  "paymentTypeId": 0,
  "paymentId": 0,
  "paymentCategory": "string"
}
```


7.6.2 Corporate Sports API

Overview

Method	API Endpoint	Description
POST	/api/v{version}/membership/ Addpaymentdetail	Add payment details for a Corporate Sports level membership and/or Group level membership. This API will also change membership to paid. **Entities to get affected by this API** Tp_MembershipPayment Tp_ActiveMembershipPayment Note: Integration Partner should neglect “Is Paid” Parameter.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested. Pass Corporate Sport club's ID here to post payment details for the membership of corporate sports club or its group.	
2	Authorization	Yes	This is the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	Amount	Mandatory	Membership charge	
2	Comment	Optional Max Length 200		
3	ExternalId	Optional Max Length 25	Transaction reference number.	
4	IsFamilyMember	Optional	If the payment is being posted for a family member	
5	IsPaid	Mandatory	If payment is already paid.	
6	OrgId	Mandatory	Payment type 0 ClubId, Payment type 2 GroupId	
7	PaidDate	Mandatory	Date of payment **Possible Value** 1) Must be greater than today's date.	
8	PaidToOrgId	Mandatory	Organization receiving the payment (club)	
9	PaymentCategory	Mandatory Max Length 127	Eg. Online, Invoice	
10	PaymentTypeId	Mandatory	0 - Paid for Membership, 2 -Group Membership fee,	
11	PeriodFrom	Mandatory	Membership from date	

			Possible Value 1) If PaymentType is membership then Year of PeriodFrom and PeriodTo must be the same. 2) If PaymentType is Group Membership fee then PeriodFrom must be greater than membership from date.	
12	PeriodTo	Mandatory	Membership to date **Possible Value** 1) Must be greater than PeriodFrom date.	
13	PersonId	Mandatory	Person for which payment is being done.	

Response Codes

	Code	Text	Description
1	200	Payment details added	
2	3000	Period from and period to are not having same year.	
3	3001	Membership not found for specific period from date.	
4	3006	Invalid period from date and period to date.	
5	3007	Invalid paid date.	
6	3008	Invalid org id.	
7	3009	Invalid payment type.	
8	3010	Invalid Period	
9	1006	Invalid branchId.	

Request Example


```
{
  "amount": 777.0,
  "comment": "test payment",
  "externalId": "llqgJLRvGNU2uhYrpMQA",
  "isFamilyMember": false,
  "isPaid": true,
  "orgId": 16235,
  "paidDate": "2020-12-02",
  "paidToOrgId": 16235,
  "paymentCategory": "test payment Category",
  "paymentTypeId": 0,
  "periodFrom": "2020-07-01",
  "periodTo": "2020-12-31",
  "personId": 9283002
}
```


Response Example


```
{
   "message": "Payment details added for membership of 9283002 with 16235 for period 2020-07-01 to 2020-12-31.",
   "updatedPersonId": 9283002,
   "paymentTypeId": 0,
   "paymentId": 1206920,
   "paymentCategory": "test payment Category"
}
```

7.7 GET Membership Payment Details

7.7.1 External Membership API

Overview

Method	API Endpoint	Description
GET	/api/v{version}/membership/paymentdetails	Returns list of payment information made for club level membership.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested.	
2	Authorization	Yes	This will be the token defining access to on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be a separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	PersonId	Yes	Person Id for which the request is made	
2	FromDate	Yes	From date range **Possible Value** 1) From date should not be of year less than Today's date year. For example if today's date is 19-Sep-2019, then From date must be greater than 1-Jan-2018	
3	ToDate	Yes	End date range **Possible Value** 1) Must be greater than ToDate.	

Response Codes

	Code	Type	Text	Description
1	200	Success		
2	1022	Error	PersonId is required.	
3	1023	Error	From Date is required.	
4	1024	Error	To Date is required.	
5	1025	Error	From date should not be of year less than Today's date year	
6	1026	Error	To date should not be greater than From date.	

Response Example

```
[
  {
    "paymentTypeId": 2,
    "paidDate": "2017-05-22",
    "periodFrom": "2017-05-21"
  },
  {
    "paymentTypeId": 1,
    "paidDate": "2019-12-31",
    "periodFrom": "2019-01-01"
  }
]
```

7.7.2 Corporate Sports API

Overview

Method	API Endpoint	Description
GET	/api/v{version}/membership/paymentdetails	Returns list of payment information made for Corporate Sports club level membership.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested. Pass Corporate Sport club's ID here to get payment details for the membership of corporate sports club.	
2	Authorization	Yes	This will be the token defining access to on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be a separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	PersonId	Yes	Person Id for whom the request is made	
2	FromDate	Yes	From date range **Possible Value** 1) From date should not be of year less than Today's date year. For example if today's date is 19-Sep-2019, then From date must be greater than 1-Jan-2018	
3	ToDate	Yes	End date range **Possible Value** 1) Must be greater than ToDate.	

Response Codes

	Code	Type	Text	Description
1	200	Success		
2	1022	Error	PersonId is required.	
3	1023	Error	From Date is required.	
4	1024	Error	To Date is required.	

5	1025	Error	From date should not be of year less than Today's date year	
6	1026	Error	To date should not be greater than From date.	

Response Example


```
{
    "paymentDetails": [
        {
            "paymentTypeId": 0,
            "paidDate": "2020-12-02",
            "periodFrom": "2020-07-01"
        },
        {
            "paymentTypeId": 0,
            "paidDate": "2020-12-02",
            "periodFrom": "2020-07-01"
        }
    ],
    "updatedPersonId": 9283002
}
```

7.8 Delete Membership

7.8.1 External Membership API

Overview

Method	API Endpoint	Description
DELETE	/api/v{version}/membership	Cancels an existing membership at club level as well as from branch level (sports). If branch Id is passed in the request then the API cancels branch level membership, otherwise it cancels the Club level membership. Today's date will be passed as end date for canceled memberships.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested	
2	Authorization	Yes	This will be the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	PersonId	Mandatory	Person Id	5347891
2	BranchId	Optional	If Branch Id is passed in the request then the API cancels the branch level membership, otherwise it cancels the Club level membership.	1238
3	`IsClubLevel`	Optional	If true, the system deletes club level membership, and Branch Id should not be passed.	true

Response Codes

	Code	Text	Description
1	200		
2	1002	Invalid PersonId	
3	1006	Invalid BranchId	
4	1019	Please provide Either Club Id or Branch Id to identify Club level or branch level membership delete request.	
5	1021	No club level membership found for delete.	
6	1020	No branch level membership found for delete.	
7	1042	Group level membership for club is not allowed.	

Request Example

```
{
  "personId": 5347891,
  "endDate": "2017-06-20",
  “branchId”: 1238
}
```

Response Example

"Club level membership cancellation request accepted for {personId} with {clubId}."
Or
"Branch level membership cancelled for {personId} with {branchId}."

7.8.2 Corporate Sports API

Overview

Method	API Endpoint	Description
DELETE	/api/v{version}/membership	Cancels an existing membership at Corporate Sports club level as well as from Group level. If Group Id is passed in the request then the API cancels Group level membership, otherwise it cancels the Corporate Sports Club level membership. Today's date will be passed as end date for canceled memberships.

Resource Information

Response Format	JSON
Requires Authentication	Yes

Request Headers

	Name	Required	Description	Example
1	ClubId	Yes	Club Id for which API is being requested. Pass Corporate Sport club's ID here to delete membership for Corporate sports club or group of corporate club.	
2	Authorization	Yes	This will be the token to define access for on-boarded club.	
3	Ocp-Apim-Subscription-Key	Yes	API management's subscription key. There will be separate key for each Integrator.	

Request Parameters

	Name	Required	Description	Example
1	PersonId	Mandatory	Person Id	5347891
2	GroupId	Optional	If Group Id is passed in the request then the API cancels the Group level membership, otherwise it cancels the Corporate Sports Club level membership.	1238
3	IsClubLevel	Optional	If true, the system deletes club level membership, and Group Id should not be passed.	true

Response Codes

	Code	Text	Description
1	200		
2	1002	Invalid PersonId	
3	1019	Please provide Either Club Id or Group Id to identify Club level or Group level membership delete request.	
4	1021	No Corporate Sports club level membership found for delete.	
5	1041	Branch level membership for Corporate Club is not allowed.	
6	1044	Invalid Corporate Club Group Id.	
7	1047	No group level membership found for delete.	

Request Example

```
{
  "personId": 5347891,
  "endDate": "2017-06-20",
  “groupId”: 1238
}
```

Response Example

```
{
    "message": "Club level membership cancellation request accepted for 9289406 with 16235.",
    "updatedPersonId": 9289406
}
```

8 Person merge

Duplicate records may be introduced to the system. Duplicate person records may be merged and processed. When two or more person records are merged, a new person record is created with a new PersonId. This new PersonId is used as the foreign key to the membership, payment and other person related entities.

The Integration Partner are informed of person merge function through the Queue.
There will be an information message (changeType = 9) about the affected record before it's changed or created.
Integration Partner should expect the following sequence of messages in case of person merge:

1. Insert message for New Person with Merged person IDs
2. Old person 1 message with change type 2
3. Old person 2 message with change type 2
4. Information message about New organization/club Membership
    a. Information message about New Person with Merged person IDs
    b. Add new Membership message with Merged Membership IDs
5. Information message about old organization/club membership
    a. Information message about old person 1
    b. Soft delete old membership (Old person 1 Club membership(s))
6. Information message about old organization/club membership
    a. Information message about old person 2
    b. Soft delete old membership (Old person 2 Club membership(s))
7. Information message about New organization/Gren for new person
    a. Information message about New Person with Merged person IDs
    b. Add new ActiveMembership message with Merged ActiveMembership IDs
8. Information message about old organization/Gren membership
    a. Information message about old person 1
    b. Soft delete old Activemembership (Old person 1 Gren membership(s))
9. Information message about old organization/Gren membership
    a. Information message about old person 2
    b. Soft delete old Activemembership (Old person 2 Gren membership(s))
10. Update Sports number for new person from old person 1
11. Update Sports number for new person from old person 2
12. Delete old gren Membership person 1
13. Delete old club Membership person 1
14. Delete old gren Membership person 2
15. Delete old club Membership person 2
16. Delete old person 1
17. Delete old person 2

It is the responsibility of the Integration Partner to read person merge messages and process them on their system to remove duplicate person records and maintain consistency with NIF data.

Example of messages integration partner should expect when duplicate person records are merged.

1. Insert message for New Person with Merged person IDs


```
    "body": {
    "PersonId": 8339484,
    "SportNo": "M010620TRO03",
    "GenderId": 1,
    "BirthDate": "2020-06-01T00:00:00",
    "IsDead": false,
    "DeathDate": null,
    "IsDeleted": false,
    "IsValidated": false,
    "LastName": "Adolfsen",
    "FirstName": "Trond",
    "AccountId": null,
    "CitizenshipRegionId": 1500152,
    "ProfilePictureId": null,
    "AddressLine1": "Oslo",
    "AddressLine2": null,
    "PostCode": "0010",
    "City": "OSLO",
    "PostCodeId": 14892,
    "Country": "Norge",
    "CountryId": 1500152,
    "Email": "Trond.Adolfsen@mailinator.com",
    "Email2": null,
    "Email3": null,
    "PrivatePhone": null,
    "WorkPhone": null,
    "MobilePhone": null,
    "RegistrationPhone": null,
    "RegistrationPhoneCountryCode": null,
    "IsSecretAddress": false,
    "IsSecretPrivatePhone": false,
    "IsSecretWorkPhone": false,
    "IsSecretMobilePhone": false,
    "IsSecretEmail": false,
    "IsSecretEmail2": null,
    "IsSecretEmail3": null,
    "OrgIdOwner": 19426,
    "MergedPersonIds": "8339482,8339483"
  }
```


2. Information message about New organization/club Membership


```
"body": {
  "OrgId": 19426,
  "OrgNo": "KL01010001",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb",
  "OrgTypeId": 5,
```

```
  "AccountId": 5284797,
  "IsActive": true,
  "SportId": 16,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": "883927842",
  "MunicipalityId": 500001,
  "FoundedDate": "2005-05-31T00:00:00",
  "AddressLine1": "Idd sportsklubb",
  "AddressLine2": "PB 7 Risum",
  "PostCode": "1761",
  "City": "HALDEN",
  "PostCodeId": 215339,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
  "Homepage": "www.iddsk.no",
  "Telephone1": null,
  "Telephone2": "09876542",
  "MobilePhone": "87654321",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": "5.347138599999994",
  "Latitude": "60.4837661",
  "OrgIdOwner": 19426,
  "ParentOrgId": null,
  "ParentOrgTypeId": null
}
```


3. Information message about New Person with Merged person IDs


```
"body": {
  "PersonId": 8339484,
  "SportNo": "M010620TRO03",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
```

```
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": null,
  "IsSecretEmail3": null,
  "OrgIdOwner": 19426,
  "MergedPersonIds": "8339482,8339483"
}
```


4. Add new Membership message with Merged Membership IDs


```
    "body": {
  "MembershipId": 24704636,
  "PersonId": 8339484,
  "OrgId": 19426,
  "FromDate": "2020-06-12T00:00:00",
  "IsDeleted": 0,
  "IsPassive": 0,
  "MembershipStatusId": 2,
  "OrgIdOwner": 19426,
  "MergedMembershipIds": "39921546,39921554"
}
```


5. Information message about old organization/club membership


```
"body": {
  "OrgId": 19426,
  "OrgNo": "KL01010001",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb",
  "OrgTypeId": 5,
  "AccountId": 5284797,
  "IsActive": true,
  "SportId": 16,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": "883927842",
```

```
  "MunicipalityId": 500001,
  "FoundedDate": "2005-05-31T00:00:00",
  "AddressLine1": "Idd sportsklubb",
  "AddressLine2": "PB 7 Risum",
  "PostCode": "1761",
  "City": "HALDEN",
  "PostCodeId": 215339,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
  "Homepage": "www.iddsk.no",
  "Telephone1": null,
  "Telephone2": "09876542",
  "MobilePhone": "87654321",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": "5.347138599999994",
  "Latitude": "60.4837661",
  "OrgIdOwner": 19426,
  "ParentOrgId": null,
  "ParentOrgTypeId": null
}
```


6. Information message about old person 1


```
"body": {
  "PersonId": 8339482,
  "SportNo": "M010620TRO01",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
```

```
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": false,
  "IsSecretEmail3": false,
  "OrgIdOwner": 19426,
  "MergedPersonIds": null
}
```


7. Soft delete old membership (Old person 1 Club membership(s))


```
"body": {
  "MembershipId": 39921546,
  "PersonId": 8339482,
  "OrgId": 19426,
  "IsDeleted": 1,
  "IsPassive": 1,
  "MembershipStatusId": 4
}
```


8. Information message about old organization/club membership


```
"body": {
  "OrgId": 19426,
  "OrgNo": "KL01010001",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb",
  "OrgTypeId": 5,
  "AccountId": 5284797,
  "IsActive": true,
  "SportId": 16,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": "883927842",
  "MunicipalityId": 500001,
  "FoundedDate": "2005-05-31T00:00:00",
  "AddressLine1": "Idd sportsklubb",
  "AddressLine2": "PB 7 Risum",
  "PostCode": "1761",
  "City": "HALDEN",
  "PostCodeId": 215339,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
```

```
  "Homepage": "www.iddsk.no",
  "Telephone1": null,
  "Telephone2": "09876542",
  "MobilePhone": "87654321",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": "5.347138599999994",
  "Latitude": "60.4837661",
  "OrgIdOwner": 19426,
  "ParentOrgId": null,
  "ParentOrgTypeId": null
}
```


9. Information message about old person 2


```
 {
    "body": {
  "PersonId": 8339483,
  "SportNo": "M010620TRO02",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": false,
  "IsSecretEmail3": false,
```

```
  "OrgIdOwner": 19426,
  "MergedPersonIds": null
}
```


10. Soft delete old membership (Old person 2 Club membership(s))


```
  "body": {
  "MembershipId": 39921554,
  "PersonId": 8339483,
  "OrgId": 19426,
  "IsDeleted": 1,
  "IsPassive": 1,
  "MembershipStatusId": 4
}
```


11. Information message about New organization/Gren for new person


```
    "body": {
  "OrgId": 819185,
  "OrgNo": "GN01010001381",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb - Orientering",
  "OrgTypeId": 14,
  "AccountId": 5669501,
  "IsActive": true,
  "SportId": 153,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": null,
  "MunicipalityId": 500001,
  "FoundedDate": null,
  "AddressLine1": "Nyborg",
  "AddressLine2": "Aspedammen",
  "PostCode": "1766",
  "City": "HALDEN",
  "PostCodeId": 10946,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
  "Homepage": "www.iddsk.no",
  "Telephone1": "",
  "Telephone2": null,
  "MobilePhone": "98856568",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": null,
```

```
  "Latitude": null,
  "OrgIdOwner": 819185,
  "ParentOrgId": 61874,
  "ParentOrgTypeId": 6
}
```


12. Information message about New Person with Merged person IDs


```
  "body": {
  "PersonId": 8339484,
  "SportNo": "M010620TRO03",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": null,
  "IsSecretEmail3": null,
  "OrgIdOwner": 19426,
  "MergedPersonIds": "8339482,8339483"
}
```


13. Add new ActiveMembership message with Merged ActiveMembership IDs


```
"body": {
  "ActiveMembershipId": 24704642,
  "PersonId": 8339484,
  "OrgId": 819185,
  "SportId": 153,
  "FromDate": "2020-06-12T00:00:00",
  "IsDeleted": 0,
```

```
  "IsPassive": 0,
  "ActiveMembershipStatusId": 2,
  "OrgIdOwner": 819185,
  "MergedActiveMembershipIds": "39921552,39921559"
}
```


14. Information message about old organization/Gren membership


```
"body": {
  "OrgId": 819185,
  "OrgNo": "GN01010001381",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb - Orientering",
  "OrgTypeId": 14,
  "AccountId": 5669501,
  "IsActive": true,
  "SportId": 153,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": null,
  "MunicipalityId": 500001,
  "FoundedDate": null,
  "AddressLine1": "Nyborg",
  "AddressLine2": "Aspedammen",
  "PostCode": "1766",
  "City": "HALDEN",
  "PostCodeId": 10946,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
  "Homepage": "www.iddsk.no",
  "Telephone1": "",
  "Telephone2": null,
  "MobilePhone": "98856568",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": null,
  "Latitude": null,
  "OrgIdOwner": 819185,
  "ParentOrgId": 61874,
  "ParentOrgTypeId": 6
}
```


15. Information message about old person 1

```
    "body": {
  "PersonId": 8339482,
  "SportNo": "M010620TRO01",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": false,
  "IsSecretEmail3": false,
  "OrgIdOwner": 19426,
  "MergedPersonIds": null
}
```


16. Soft delete old Activemembership (Old person 1 Gren membership(s))


```
    "body": {
  "ActiveMembershipId": 39921552,
  "PersonId": 8339482,
  "OrgId": 819185,
  "SportId": 153,
  "IsDeleted": 1,
  "IsPassive": 1,
  "ActiveMembershipStatusId": 4
}
```


17. Information message about old organization/Gren membership

```
    "body": {
  "OrgId": 819185,
  "OrgNo": "GN01010001381",
  "OrgName": "Idd Sportsklubb",
  "Abbreviation": "Idd Sportsklubb",
  "DescribingName": "Idd Sportsklubb - Orientering",
  "OrgTypeId": 14,
  "AccountId": 5669501,
  "IsActive": true,
  "SportId": 153,
  "TerminationDate": null,
  "TerminationReasonId": null,
  "Comment": null,
  "OrganisationNumber": null,
  "MunicipalityId": 500001,
  "FoundedDate": null,
  "AddressLine1": "Nyborg",
  "AddressLine2": "Aspedammen",
  "PostCode": "1766",
  "City": "HALDEN",
  "PostCodeId": 10946,
  "Country": "Norge",
  "CountryId": 1500152,
  "VisitAddressLine1": "Aspedammenveien 46",
  "VisitAddressLine2": "1766",
  "VisitPostCode": "HALDEN",
  "VisitCity": null,
  "VisitPostCodeId": 10946,
  "Email": "info@iddsk.no",
  "Homepage": "www.iddsk.no",
  "Telephone1": "",
  "Telephone2": null,
  "MobilePhone": "98856568",
  "Fax": null,
  "IsSecretAddress": false,
  "IsSecretTelephone1": false,
  "IsSecretTelephone2": false,
  "IsSecretMobilePhone": false,
  "IsSecretFax": false,
  "IsSecretEmail": false,
  "Longitude": null,
  "Latitude": null,
  "OrgIdOwner": 819185,
  "ParentOrgId": 61874,
  "ParentOrgTypeId": 6
}
```


18. Information message about old person 2


```
  {
    "body": {
  "PersonId": 8339483,
  "SportNo": "M010620TRO02",
  "GenderId": 1,
  "BirthDate": "2020-06-01T00:00:00",
  "IsDead": false,
```

```
  "DeathDate": null,
  "IsDeleted": false,
  "IsValidated": false,
  "LastName": "Adolfsen",
  "FirstName": "Trond",
  "AccountId": null,
  "CitizenshipRegionId": 1500152,
  "ProfilePictureId": null,
  "AddressLine1": "Oslo",
  "AddressLine2": null,
  "PostCode": "0010",
  "City": "OSLO",
  "PostCodeId": 14892,
  "Country": "Norge",
  "CountryId": 1500152,
  "Email": "Trond.Adolfsen@mailinator.com",
  "Email2": null,
  "Email3": null,
  "PrivatePhone": null,
  "WorkPhone": null,
  "MobilePhone": null,
  "RegistrationPhone": null,
  "RegistrationPhoneCountryCode": null,
  "IsSecretAddress": false,
  "IsSecretPrivatePhone": false,
  "IsSecretWorkPhone": false,
  "IsSecretMobilePhone": false,
  "IsSecretEmail": false,
  "IsSecretEmail2": false,
  "IsSecretEmail3": false,
  "OrgIdOwner": 19426,
  "MergedPersonIds": null
}
```


19. Soft delete old Activemembership (Old person 2 Gren membership(s))


```
"body": {
  "ActiveMembershipId": 39921559,
  "PersonId": 8339483,
  "OrgId": 819185,
  "SportId": 153,
  "IsDeleted": 1,
  "IsPassive": 1,
  "ActiveMembershipStatusId": 4
}
```


20. Update Sports number for new person from old person 1


```
"body": {
"PersonID": 8339484
  "SportsNo": "M13129DAV01"
}
```


21. Update Sports number for new person from old person 2

```
"body": {
"PersonID": 8339484
  "SportsNo": "M13129DAV01"
}
```

22. Delete old gren Membership person 1

```
"body": {
  "ActiveMembershipId": 39921552
}
```

23. Delete old club Membership person 1

```
    "body": {
  "MembershipId": 39921546
}
```

24. Delete old gren Membership person 2

```
    "body": {
  "ActiveMembershipId": 39921559
}
```

25. Delete old club Membership person 2

```
  {
    "body": {
  "MembershipId": 39921554
}
```

26. Delete old person 1

```
    "body": {
  "PersonId": 8339483
}
```

27. Delete old person 2

```
    "body": {
  "PersonId": 8339482
}
```

9 Entities JSON Schema

9.1 TpMembership


```
{
  "Tp_Membership": {
    "properties": {
      "MembershipId": {
        "type": "integer"
      },
      "PersonId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "FromDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "ToDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "IsDeleted": {
        "type": [
          "integer",
          "null"
        ]
      },
      "IsPassive": {
        "type": [
          "integer",
          "null"
        ]
      },
      "MembershipNumber": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OriginalMembershipDate": {
        "type": [
          "string",
          "null"
        ],
```

```
        "format": "date-time"
      },
      "MembershipStatusId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "MembershipTypeId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "IsPaid": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "MembershipPeriodId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```

9.2 TpActiveMembership


```
{
  "Tp_ActiveMembership": {
    "properties": {
      "ActiveMembershipId": {
        "type": "integer"
      },
      "PersonId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "SportId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "FromDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "ToDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "IsDeleted": {
        "type": [
          "integer",
          "null"
        ]
      },
      "IsPassive": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OriginalMembershipDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "ActiveMembershipStatusId": {
```

```
        "type": [
          "integer",
          "null"
        ]
      },
      "ActiveMembershipCategoryId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "ActiveMembershipPeriodId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```

9.3 ToOrgType


```
{
  "To_OrgType": {
    "properties": {
      "OrgTypeId": {
        "type": "integer"
      },
      "OrgTypeNo": {
        "type": [
          "string",
          "null"
        ]
      },
      "OrgTypeName": {
        "type": [
          "string",
          "null"
        ]
      },
      "IsActive": {
        "type": [
          "boolean ",
          "null"
        ]
      },
      "ActivityLevelId": {
        "type": [
          "int"
        ]
      },
      "ReadPublic": {
        "type": [
          "integer",
          "null"
        ]
      },
      "ReadLoggedIn": {
        "type": [
          "integer",
          "null"
        ]
      },
      "ReadDetailOwner": {
        "type": [
          "integer",
          "null"
        ]
      },
      "ReadOrgLine": {
        "type": [
          "int",
          "null"
        ]
      },
      "ReadTypeOwner": {
        "type": [
          "integer",
          "null"
        ]
```

```
      },
      "ReadPersonal": {
        "type": [
          "integer",
          "null"
        ]
      },
      "UpdatePublic": {
        "type": [
          "int ",
          "null"
        ]
      },
      "UpdateLoggedIn": {
        "type": [
          "integer",
          "null"
        ]
      },
      "UpdateDetailOwner": {
        "type": [
"integer",
“null”
]
      },
     "UpdateOrgLine": {
        "type": [
          "integer",
          "null"
        ]
      },
     "UpdateTypeOwner": {
        "type": [
          "integer",
          "null"
        ]
      },
     "UpdatePeronal": {
        "type": [
          "integer",
          "null"
        ]
      },
     "OrgIdOwner": {
        "type": [
          "integer",
          "null"
        ]
      }
    }
  }
}
```

9.4 TpPerson


```
"Tp_Person": {
  "properties": {
    "PersonId": {
      "type": "integer"
    },
    "SportNo": {
      "type": [
        "string",
        "null"
      ]
    },
    "GenderId": {
      "type": "integer"
    },
    "BirthDate": {
      "type": [
        "string",
        "null"
      ],
      "format": "date-time"
    },
    "IsDead": {
      "type": "boolean"
    },
    "DeathDate": {
      "type": [
        "string",
        "null"
      ],
      "format": "date-time"
    },
    "IsDeleted": {
      "type": "boolean"
    },
    "IsValidated": {
      "type": "boolean"
    },
    "LastName": {
      "type": [
        "string",
        "null"
      ]
    },
    "FirstName": {
      "type": [
        "string",
        "null"

      ]
    },
    "AccountId": {
      "type": [
        "integer",
        "null"
      ]
    },
    "CitizenshipRegionId": {
      "type": [
```

```
      "integer",
      "null"
    ]
  },
    "ProfilePictureId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "AddressLine1": {
    "type": [
      "string",
      "null"
    ]
  },
  "AddressLine2": {
    "type": [
      "string",
      "null"
    ]
  },
  "PostCode": {
    "type": [
      "string",
      "null"
    ]
  },
  "City": {
    "type": [
      "string",
      "null"
    ]
  },
  "PostCodeId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "Country": {
    "type": [
      "string",
      "null"
    ]
  },
  "CountryId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "Email": {
    "type": [
      "string",
      "null"
    ]
  },
  "Email2": {
```

```
      "type": [
        "string",
        "null"
      ]
    },
    "Email3": {
      "type": [
        "string",
        "null"
      ]
    },
    "PrivatePhone": {
      "type": [
        "string",
        "null"
      ]
    },
    "WorkPhone": {
      "type": [
        "string",
        "null"
      ]
    },
    "MobilePhone": {
      "type": [
        "string",
        "null"
      ]
    },
    "IsSecretAddress": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "IsSecretPrivatePhone": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "IsSecretWorkPhone": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "IsSecretMobilePhone": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "IsSecretEmail": {
      "type": [
        "boolean",
        "null"
      ]
    },
```

```
    "IsSecretEmail2": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "IsSecretEmail3": {
      "type": [
        "boolean",
        "null"
      ]
    },
    "OrgIdOwner": {
      "type": "integer"
    }
  }
 }
}
```

9.5 TpPersonContactInformation


```
{
  "Tp_PersonContactInformation": {
    "properties": {
      "PersonContactInformationId": {
        "type": "integer"
      },
      "PersonId": {
        "type": "integer"
      },
      "AddressLine1": {
        "type": [
          "string",
          "null"
        ]
      },
      "AddressLine2": {
        "type": [
          "string",
          "null"
        ]
      },
      "PostCode": {
        "type": [
          "string",
          "null"
        ]
      },
      "City": {
        "type": [
          "string",
          "null"
        ]
      },
      "PostCodeId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "Country": {
        "type": [
          "string",
          "null"
        ]
      },
      "CountryId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "Email": {
        "type": [
          "string",
          "null"
        ]
      },
      "PrivatePhone": {
```

```
        "type": [
          "string",
          "null"
        ]
      },
      "WorkPhone": {
        "type": [
          "string",
          "null"
        ]
      },
      "MobilePhone": {
        "type": [
          "string",
          "null"
        ]
      },
      "IsSecretAddress": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "IsSecretPrivatePhone": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "IsSecretWorkPhone": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "IsSecretMobilePhone": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "IsSecretEmailAddress": {
        "type": [
          "boolean",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": [
          "integer",
          "null"
        ]
      }
    }
  }
}
```

9.6 ToOrg


```
{
  "ToOrg": {
    "properties": {
      "OrgId": {
        "type": "integer"
      },
      "OrgNo": {
        "type": [
          "string",
          "null"
        ]
      },
      "OrgName": {
        "type": [
          "string",
          "null"
        ]
      },
      "Abbreviation": {
        "type": [
          "string",
          "null"
        ]
      },
      "DescribingName": {
        "type": [
```

```
      "string",
      "null"
    ]
  },
  "OrgTypeId": {
    "type": "integer"
  },
  "AccountId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "IsActive": {
    "type": "boolean"
  },
  "SportId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "TerminationDate": {
    "type": [
      "string",
      "null"
```

```
    ],
    "format": "date-time"
  },
  "TerminationReasonId": {
    "type": [
      "integer",
      "null"
    ]
  },
  "Comment": {
    "type": [
      "string",
      "null"
    ]
  },
  "OrganisationNumber": {
    "type": [
      "string",
      "null"
    ]
  },
  "MunicipalityId": {
    "type": [
      "integer",
      "null"
    ]
```

```
  },
  "FoundedDate": {
    "type": [
      "string",
      "null"
    ],
    "format": "date-time"
  },
  "AddressLine1": {
    "type": [
      "string",
      "null"
    ]
  },
  "AddressLine2": {
    "type": [
      "string",
      "null"
    ]
  },
  "PostCode": {
    "type": [
      "string",
      "null"
    ]
  },
```

```
"City": {
  "type": [
    "string",
    "null"
  ]
},
"PostCodeId": {
  "type": [
    "integer",
    "null"
  ]
},
"Country": {
  "type": [
    "string",
    "null"
  ]
},
"CountryId": {
  "type": [
    "integer",
    "null"
  ]
},
"VisitAddressLine1": {
  "type": [
```

```
      "string",
      "null"
    ]
  },
  "VisitAddressLine2": {
    "type": [
      "string",
      "null"
    ]
  },
  "VisitPostCode": {
    "type": [
      "string",
      "null"
    ]
  },
  "VisitCity": {
    "type": [
      "string",
      "null"
    ]
  },
  "VisitPostCodeId": {
    "type": [
      "integer",
      "null"
```

```
    ]
  },
  "Email": {
    "type": [
      "string",
      "null"
    ]
  },
  "Homepage": {
    "type": [
      "string",
      "null"
    ]
  },
  "Telephone1": {
    "type": [
      "string",
      "null"
    ]
  },
  "Telephone2": {
    "type": [
      "string",
      "null"
    ]
  },
```

```
"MobilePhone": {
  "type": [
    "string",
    "null"
  ]
},
"Fax": {
  "type": [
    "string",
    "null"
  ]
},
"IsSecretAddress": {
  "type": [
    "boolean",
    "null"
  ]
},
"IsSecretTelephone1": {
  "type": [
    "boolean",
    "null"
  ]
},
"IsSecretTelephone2": {
  "type": [
```

```
      "boolean",
      "null"
    ]
  },
  "IsSecretMobilePhone": {
    "type": [
      "boolean",
      "null"
    ]
  },
  "IsSecretFax": {
    "type": [
      "boolean",
      "null"
    ]
  },
  "IsSecretEmail": {
    "type": [
      "boolean",
      "null"
    ]
  },
  "Longitude": {
    "type": [
      "string",
      "null"
```

```
      ]
    },
    "Latitude": {
      "type": [
        "string",
        "null"
      ]
    },
    "OrgIdOwner": {
      "type": "integer"
    }
  }
 }
}
```

9.7 ToOrgTerminationReason


```
{
  "To_OrgTerminationReason": {
    "properties": {
      "TerminationReasonId": {
        "type": "integer"
      },
      "TerminationReasonName": {
        "type": [
          "string",
          "null"
        ]
      },
      "Description": {
        "type": [
          "string",
          "null"
        ]
      },
      "RestrictedToOrgTypeId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "IsActive": {
        "type": "boolean"
      },
      "IsTemporary": {
        "type": "boolean"
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
```

```
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```

9.8 ToSport


```
{
  "To_Sport": {
    "properties": {
      "SportId": {
        "type": "integer"
      },
      "ParentSportId": {
        "type": "integer"
      },
      "SportCode": {
        "type": [
          "string",
          "null"
        ]
      },
      "SportName": {
        "type": [
          "string",
          "null"
        ]
      },
      "Description": {
        "type": [
          "string",
          "null"
        ]
      },
      "IsActive": {
        "type": "boolean"
      },
      "FromDate": {
        "type": "string",
        "format": "date-time"
      },
      "ToDate": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "IsValidForReporting": {
        "type": "boolean"
      },
      "IsCalendarSeason": {
        "type": "boolean"
      },
      "IsIndividualSport": {
        "type": "boolean"
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
```

```
      "type": "integer"
    },
    "LCFunctionId": {
      "type": [
        "integer",
        "null"
      ]
    },
    "LCDateTime": {
      "type": [
        "string",
        "null"
      ],
      "format": "date-time"
    },
    "LCFunctionalityId": {
      "type": [
        "integer",
        "null"
      ]
    },
    "OrgIdOwner": {
      "type": "integer"
    }
  }
 }
}
```

9.9 TpActiveMembershipStatus


```
{
  "TpActiveMembershipStatus": {
    "properties": {
      "ActiveMembershipStatusId": {
        "type": "integer"
      },
      "ActiveMembershipStatusName": {
        "type": [
          "string",
          "null"
        ]
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```

9.11 TpMembershipPeriod


```
{
  "Tp_MembershipPeriod": {
    "properties": {
      "MembershipPeriodId": {
        "type": "integer"
      },
      "MembershipPeriodName": {
        "type": [
          "string",
          "null"
        ]
      },
      "FromDate": {
        "type": "string",
        "format": "date-time"
      },
      "ToDate": {
        "type": "string",
        "format": "date-time"
      },
      "Available": {
        "type": "boolean"
      },
      "OrgId": {
        "type": "integer"
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      },
```

```
      "To_Org": {
        "$ref": "#/definitions/To_Org"
      }
    }
  }
}
```

9.12 TpMembershipStatus


```
{
  "TpMembershipStatus": {
    "properties": {
      "MembershipStatusId": {
        "type": "integer"
      },
      "MembershipStatusName": {
        "type": [
          "string",
          "null"
        ]
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```

9.13 TpPaymentStatus


```
{
  "Tp_PaymentStatus": {
    "properties": {
      "PaymentStatusId": {
        "type": "integer"
      },
      "PaymentStatusName": {
        "type": [
          "string",
          "null"
        ]
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
```


9.13 TpPaymentType


```
{
  "Tp_PaymentType": {
```

```
    "properties": {
      "PaymentTypeId": {
        "type": "integer"
      },
      "PaymentTypeName": {
        "type": [
          "string",
          "null"
        ]
      },
      "FAFunctionId": {
        "type": "integer"
      },
      "FADatetime": {
        "type": "string",
        "format": "date-time"
      },
      "FAFunctionalityId": {
        "type": "integer"
      },
      "LCFunctionId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "LCDatetime": {
        "type": [
          "string",
          "null"
        ],
        "format": "date-time"
      },
      "LCFunctionalityId": {
        "type": [
          "integer",
          "null"
        ]
      },
      "OrgIdOwner": {
        "type": "integer"
      }
    }
  }
}
```


Disclaimer

This version of the API documentation is intended to enlighten Integration Partners negotiating a contract with NIF and cannot be subject to third party distribution. Sole purpose is to describe API Product access and methods Referred data in this document are referred as examples only.