

SCHACHDE DSB-Mitglieder-API

ANGEBOT

Einführung

NU hat dem DSB die Ablösung des MiVIS und DeVIS Systems als zwei getrennte, aber miteinander verbundene Systeme angeboten. Dabei soll einerseits das nuLiga-System beim DSB und in allen Landesverbänden für die Mitgliederverwaltung eingesetzt werden (Ablösung des bestehenden MiVIS Systems). Dazu sind nur Teilmodule von nuLiga erforderlich und der DSB hat im April 2023 diese Teilmodule als SaaS-Lizenz "nuLigaLight" lizenziert.

Andererseits soll der gesamte Funktionsumfang von nuLiga für die Ligaverwaltung in einzelnen Landesverbänden zum Einsatz kommen. Neben der Mitgliederverwaltung können dann diese Verbände die integrierten Funktionen von nuLiga für die Planung, Organisation und Durchführung des Ergebnisdienstes für die Mannschaftswettkämpfe nutzen.

Neben der Einführung von nuLiga/nuLigaLight soll ebenfalls ein neues System "DWZ-Service" entwickelt werden, das die Berechnung der DWZ durchführt. Es ist geplant, dass dieses System automatisch Mitglieder- und Ergebnisdaten aus nuLiga/nuLigaLight bezieht. Ergebnisse zu Turnieren und anderen (externen) Ligaverwaltungssystemen sollen wie bisher in Form von Turnierdatei in den DWZ-Service importiert werden und dort, zusammen mit den Ergebnissen aus nuLiga für die Berechnung der DWZ herangezogen werden. Der DWZ Service soll außerdem für die Umsetzung diverser Schnittstellen herangezogen werden.

Die Einführung von nuLiga/nuLigaLight sowie des neuen DWZ-Services war für die Saison 2023/24 geplant.

Der DSB hat nun die Beauftragung der kostenintensiven Entwicklung des DWZ-Services verschoben. Der Auftrag zur Umsetzung der Mitgliederverwaltung ist jedoch im April 2023 erfolgt. Das erlaubt es auch, dass die Landesverbände, die nuLiga für den Ergebnisdienst einsetzen wollen, wie geplant mit der Saison 2023/24 einsteigen können.

Es stellt sich nun die Frage, wie ein Übergangsszenario umgesetzt werden kann, bei dem nuLigaLight als neue Mitgliederverwaltung für einen begrenzten Zeitraum mit dem alten MiVIS und DeWIS System zusammenarbeitet.

nuLigaLight-Standalone Konzept

Die Grundgedanken des Konzeptes sind:

- An die Stelle einer engen Kopplung von nuLigaLight und dem geplanten DWZ-Service tritt eine offen gehaltene Schnittstelle, die auch eine lose Kopplung mit dem alten DeWIS System erlaubt. Auf dieser Schnittstelle kann dann auch zukünftig der neue DWZ-Service aufsetzen. Arbeitstitel "DSB-Mitglieder-API"
- Die interaktiven Funktionen des bestehenden MiVIS Systems werden deaktiviert, sodass dieses System nur noch als "Datenlieferant" für das DeWIS System und andere, heute angebundene Systeme wie zB die DSB homepage, die Generierung von "Hintergrunddateien" für Turnierprogramme oder verschiedene Ergebnisdienste dient.
- Die Entwickler des MiVIS/DeWIS Systems, bzw. Experten, die die Struktur der Datenbank kennen, realisieren das Gegenstück zum DSB-Mitglieder-API, sodass die Datenänderungen aus nuLigaLight dorthin synchronisiert werden.
- Drittsysteme, die bisher ihre Daten aus "Hintergrunddateien", über SOAP oder andere Wege von MiWIS/DeWIS bezogen haben, können später auf das DSB-Mitglieder-API migrieren, um updates aus der Mitgliederverwaltung zu konsumieren.

DSB-Mitglieder-API

REST-Push

Das API, das eine lose Kopplung mit der Mitgliederverwaltung erlaubt, wird als **push service** unter Anwendung der **REST-Methode** umgesetzt. nuLigaLight ist dabei der **producer** und beliefert einen oder mehrere **consumer** mit den relevanten Mitgliederdaten. Das bedeutet:

- nuLigaLight als **producer** führt https-requests aus und übergibt dabei in den http-headers, URL-Parametern und im request-body Daten für die Identifikation, die Steuerung und zu den Inhalten der Datenobjekte.
- Die Gegenseiten (**consumer**) des APIs stellen jeweils einen http-endpoint zur Verfügung, der die Syntax dieser requests versteht bzw. umsetzt. Die Basis-URL dieses endpoints und die notwendige Authentifizierungsinformation wird für jeden consumer in nuLigaLight konfiguriert.
- Das DSB-Mitglieder-API wird mittels einer Kafka message queue an nuLigaLight angebunden.
- Je nach Anforderung wird entweder der gesamte, aktuelle DB-Inhalt über diese Infrastruktur transferiert (zB nächtlicher Abgleich aller Verbands-, Vereins- und Mitgliederdaten), oder bei jeder Datenänderung adhoc übertragen (zB Neuanlage eines Spielers).

Endpoints

- Für jeden Endpoint wird eine Basis-URL konfiguriert, zB für das bestehende MiVIS System: <https://mivis.dsb.de/api>
- Zur Absicherung sollte ein bearer-token zur Anwendung kommen
- Am endpoint werden folgende HTTP-methods verwendet:
 - PUT für den update bzw insert (upsert) von Objekten
 - DELETE für das Löschen von Objekten
- Die Identifikation erfolgt jeweils über eine UUID die Teil der URL sind:
 - Verbände, upsert per PUT: `.../api/federations/{fed-uuid}`
- Der consumer antwortet auf jeden request mit einem response in dem insbesondere über den http status code der Erfolg der Operation bekannt gegeben wird. Im Fall eines Fehlers oder Zugriffsproblems enthält der response body eine Fehlerbeschreibung.
- Die Dateninhalte werden bei den PUT Operationen als request body im JSON-Format übertragen (data transfer objects, DTOs). ZB das DSBFederationDTO:

```
{
  "uuid" : "d3a11424-d2bd-11ed-afaf-0242ac120002",
  "federation-nr" : 8,
  "nickname" : "SVW",
  "region_uuid" : "e29c2c2a-d2bd-11ed-afaf-0242ac120002",
  "name" : "Schachverband Württemberg e.V."
}
```

DSB-Mitglieder-API Basisausführung

Resource	endpoint (request URL path)	request body
Regionen	PUT <code>.../regions/{reg-uuid}</code>	DSBRegionDTO
Verbände	PUT <code>.../federations/{fed-uuid}</code>	DSBFederationDTO
Vereine	PUT <code>.../federations/{fed-uuid}/clubs/{club-uuid}</code>	DSBClubDTO
	DELETE <code>.../federations/{fed-uuid}/clubs/{club-uuid}</code>	
Adressen	PUT <code>.../federations/{fed-uuid}/addresses/{addr-uuid}</code>	DSBAdresseDTO
	DELETE <code>.../federations/{fed-uuid}/addresses/{addr-uuid}</code>	
Personen	PUT <code>.../federations/{fed-uuid}/persons/{pers-uuid}</code>	DSBPersonDTO
	DELETE <code>.../federations/{fed-uuid}/persons/{pers-uuid}</code>	
Vereinsmitglieder, Spieler	PUT <code>.../federations/{fed-uuid}/club/{club-uuid}/member/{clubmem-uuid}</code>	DSBClubMemberDTO
	DELETE <code>.../federations/{fed-uuid}/club/{club-uuid}/member/{clubmem-uuid}</code>	
Vereinsfunktionen	PUT <code>.../clubRoles/{role-uuid}</code>	DSBClubRoleDTO
Vereinsfunktionäre	PUT <code>.../federations/{fed-uuid}/club/{club-uuid}/officials/{role-uuid}</code>	DSBClubOfficialDTO
	DELETE <code>.../federations/{fed-uuid}/club/{club-uuid}/officials/{role-uuid}</code>	

DSB-Mitglieder-API Erweiterungen

Zweck des DSB-Mitglieder-APIs ist die **Synchronisation** bzw **Replikation** mit externen Datenbanken. Die oben ausgeführte Basisversion setzt eine solches Replikations-API für die Daten der Mitgliederverwaltung um. nuLiga und nuVerband verwalten darüber hinaus weitergehende Informationen, die in einem on-demand-Verfahren über nuPortalRS zur Verfügung gestellt werden. Dort ist der Transaktionsablauf umgekehrt: die consumer stellen über einen https-requests eine Anfrage zu einem konkreten Objekt bzw einer konkreten Resource und nuPortalRS antwortet im Response mit einem entsprechenden DTO.

Wenn zu weitergehenden Datenstrukturen ebenfalls ein Replikationsbedarf besteht, dann kann das push-API entsprechend erweitert werden. Datenstrukturen aus nuVerband wäre zB:

- Verbandsfunktionen und Funktionäre (Verband, Bezirk, Kreis)
- Ehrungstypen und Geehrte
- Lizenztypen und Lizenzinhaber

MiVIS/DeWIS API (Notizen des MS-Teams call mit dem MiVIS-Team des SVW vom 14.4.2023)

Entitiesets

- Vereine
 - (1:m) Sportstaetten
 - (1:1) Adresse
 - Vereinsnummer <V><K>000
 - (1:1) Post/Korrespondenzadresse
 - (1:0) Rechnungsadresse
- Adressen
- Regionen
- Verbände
- Personen
 - PKZ: permanente, eindeutige Personennummer
 - (1:1) Adresse
- Vereinsmitglieder
 - Mitgliedernummer <V><K>000-0000
 - Spielrecht: aktiv | passiv | kein Spielrecht
 - von
 - bis
- Funktionäre
 - von
 - bis
- Vereins-Kreis-/Bezirks-/Verbands-Funktionen
- Ehrungstypen
- Geehrte
 - am
- Lizenztypen
- Lizenzinhaber
 - von
 - bis