Implementing the ICAR ADE standards

Webinar
ICAR ADE WG

April 22nd 2020, 14.00 PM CEST, ICAR ‘virtual room’

April 23rd 2020, 21.00 PM CEST, ICAR ‘virtual room’

Version 1.0
24-4-2020
Content

14.00 Welcome & Introduction to the subject (Andrew Cooke, NZ)

14.10 ICAR-ADE defined messages (Erwin Speybroeck, BE)

14.20 Prototype of ADE API based on standard messages (Anton Hokkonen, Fi)

14.30 How to start using ADE messages in your data exchange (Arjan Lamers, NL)

14.40 Questions

15.00 eom
Welcome and introduction

Why:
• Common standards and guidelines make data interchange easier and more effective

How:
• Modern, standards-based web technologies
• Open source, collaborative approach
Audience and Scope

Audience:
• Business decision makers
• Product owners and business analysts
• Software and data developers

Scope:
• Individual animals (at least initially)
• Wider than milking bovines – all livestock covered by ICAR
• First the “data” standards – then the exchange
Content – in version 1

Version: ADE-1

- Animal
- Generic device
- Registration
- Arrivals
- Departures
- Deaths
- Embryos
- Semen straws
- Heats
- Mating
- Insemination
- Pregnancy check
- Abortion
- Parturition
- Lactation
- Test Day Result
- Milking Visit
- Drying Off
- Weight
Content – in future

Next versions

- Condition scores
- Conformation scores
- Breeding values
- Health diagnoses
- Health treatments (medicines, procedures)
- Groups
Content – in future

Version: ADE-1

Detailed version numbering \(<major>.<minor>.<patch>\)

- Major version number – breaking changes only – infrequent
- Minor version number – new additions – regular cadence
- Patch – extremely minor corrections
How to get started

https://github.com/adewg/ICAR
How to get started

https://github.com/adewg/ICAR

- Arrays or lists of resources
- List values included in resources
- Example resources
- The main things to GET or update
- Other data types included in resources
- API methods to GET or update resources and collections
ICAR ADE defined messages

ADE-1-milk  Milk messages approved by the working group, waiting for a test

ADE-1-other  Other messages approved by the working group, waiting for a test

ADE-1-registration  Registration messages approved by the working group, waiting for a test

ADE-1-reproduction  Reproduction messages approved by the working group, waiting for a test
# ICAR ADE defined messages

**ADE-1-registration**  Registration messages approved by the working group, waiting for a test

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td><code>/locations/{location-scheme}/{location-id}/births</code></td>
<td>Get the births for a certain location</td>
</tr>
<tr>
<td>GET</td>
<td><code>/locations/{location-scheme}/{location-id}/deaths</code></td>
<td>Get the deaths for a certain location</td>
</tr>
<tr>
<td>GET</td>
<td><code>/locations/{location-scheme}/{location-id}/arrivals</code></td>
<td>Get the arrivals for a certain location</td>
</tr>
<tr>
<td>GET</td>
<td><code>/locations/{location-scheme}/{location-id}/departures</code></td>
<td>Get the departures for a certain location</td>
</tr>
<tr>
<td>GET</td>
<td><code>/locations/{location-scheme}/{location-id}/animals</code></td>
<td>Get the animals for a certain location</td>
</tr>
</tbody>
</table>
# ICAR ADE defined messages

**ADE-1-reproduction** Reproduction messages approved by the working group, waiting for a test

| GET | /locations/{location-scheme}/{location-id}/pregnancy-checks | Get the pregnancy checks for a certain location |
| GET | /locations/{location-scheme}/{location-id}/heats | Get the heats for a certain location |
| GET | /locations/{location-scheme}/{location-id}/inseminations | Get the inseminations for a certain location |
| GET | /locations/{location-scheme}/{location-id}/drying-offs | Get the drying off for a certain location |
| GET | /locations/{location-scheme}/{location-id}/abortions | Get the abortion events for a certain location |
| GET | /locations/{location-scheme}/{location-id}/parturitions | Get the parturition events for a certain location |
| GET | /locations/{location-scheme}/{location-id}/mating-recommendations | Get the mating recommendations for a certain location |
ICAR ADE defined messages

ADE-1-milk  Milk messages approved by the working group, waiting for a test

GET /locations/{location-scheme}/{location-id}/milking-visits  Get the data for milking visits

GET /locations/{location-scheme}/{location-id}/test-day-results  Get the data for test day results

GET /locations/{location-scheme}/{location-id}/lactations  Get the data for lactations
ICAR ADE defined messages

ADE-1-other  Other messages approved by the working group, waiting for a test

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/locations/{location-scheme}/{location-id}/weights</td>
<td>Get the weight data of animals</td>
</tr>
<tr>
<td>GET</td>
<td>/locations/{location-scheme}/{location-id}/devices</td>
<td>Get the device data for a location</td>
</tr>
</tbody>
</table>
ICAR ADE defined messages

I Need other messages

- Take part in the ADE working group
- Bring in an issue with your request on https://github.com/adewwg/ICAR/issues
ICAR ADE defined messages - principles

- Example
  
  [Link to example](http://editor.swagger.io/?url=https://raw.githubusercontent.com/adewg/ICAR/master/url-schemes/exampleUrlScheme.json)
Identification with *icarIdentifierType*

Generalised way of identifying:

- Animals
- Locations
- Trait labels
- Many other things

Two parts:

- scheme  e.g. “nl.ubn” or “eu.animalId” or “org.gs1.gln”
- id      A unique identifier within that scheme

https://github.com/adewg/ICAR/blob/master/location-scheme.md
Observations – based on icarEventCoreResource

Record observations or actions for individual animals

**Required**
- **id** (a unique id of the event, at least at location level)
- **animal** (identifier)
- **location** (identifier)
- **eventDateTime** (ISO date time, UTC format)

**Not required**
- **traitlabel** (If event = formal trait, identifies the recording system and trait.)
- **responsible** (person object : event is carried out or authorised by a person)
- **contemporaryGroup** (contemporary group code that affects statistical analysis)
Prototype ADE API/client based on ADE

- ICAR ADE standard code generation for server/client side
- several generic code generators available (codegen, openapi-generator, etc.)

- Web Service (.NET Web API)
  - endpoints
  - models and enums
    http://ade-api.azurewebsites.net/

- Client SPA (Angular 8)
  - client models and enumeration types
  - data services
  - Data presentation
    http://ade-client.azurewebsites.net/
## Tools – openapi-generator

https://github.com/OpenAPITools/openapi-generator

| Languages/Frameworks |  |
|-----------------------|--| |
| **API clients**       | ActionScript, Ada, Apex, Bash, C, C# (.net 2.0, 3.5 or later, .NET Standard 1.3 - 2.0, .NET Core 2.0), C++ (cpp-resdk, Qt5, Tizen), Clojure, Dart (1.x, 2.x), Elixir, Elm, Eiffel, Erlang, Go, Groovy, Haskell (http-client, Servant), Java (Jersey1.x, Jersey2.x, OkHttp, Retrofit1.x, Retrofit2.x, Feign, RestTemplate, RESTEasy, Vertx, Google API Client Library for Java, Rest-assured, Spring 5 Web Client, MicroProfile Rest Client), k8s, Kotlin, Lua, Nim, Node.js/JavaScript (ES5, ES6, AngularJS with Google Closure compiler annotations, Flow types, Apollo GraphQL DataStore), Objective-C, OCaml, Perl, PHP, PowerShell, Python, R, Ruby, Rust (rust, rust-server), Scala (akka, http4s, scalaz, sttp, swagger-async-httpclient), Swift (2.x, 3.x, 4.x, 5.x), Typescript (AngularJS, Angular (2.x - 8.x), Aurelia, Axios, Fetch, Inversify, Jquery, Node, Rxjs) |
| **Server stubs**       | Ada, C# (ASP.NET Core, NancyFx), C++ (Pistache, Restbed, Qt5 QHTTPEngine), Erlang, F# (Giraffe), Go (net/http, Gin), Haskell (Servant), Java (MSF4J, Spring, Undertow, JAX-RS: CDI, CXF, Inflector, Jersey, RestEasy, Play Framework, PKMST, Vertx), Kotlin (Spring Boot, Ktor, Vertx), PHP (Laravel, Lumen, Slim, Silex, Symfony, Zend Expressive), Python (Flask), NodeJS, Ruby (Sinatra, Rails5), Rust (rust-server), Scala (Akka, Finch, Legorn, Play, Scaletta) |
| **API documentation generators** | HTML, Confluence Wiki, Asciidoc |
| **Configuration files** | Apache2 |
| **Others**             | GraphQL, JMeter, MySQL Schema, Protocol Buffer |
.NET Core API – stub service

- openapi-generator for "aspnetcore" used for API stub generation
- contains models, enums and default API endpoint definitions
- includes swagger docs and descriptions
How to start using ADE messages

- Demo based on JoinData
  - See https://integration.join-data.net/api/docs#datahub-api

- What will we demo?
  - Proposed URL scheme
  - Proposed Authentication (OAuth2)
  - Example data for a dummy Dutch farm
Questions
Thank you