



Miljøministeriet
Miljøstyrelsen

Grænseflade til indberetning af prøveresultater til Miljøstyrelsens Autorisationssystem for brug af Bekæmpelsesmidler (MAB)

Dato

23-02-2021

Version

Status

2.06

Gældende fra 14. september 2020

Ansvarlig

Stener Bonde

Ændringshistorik

Version	Kapitel/afsnit	Beskrivelse	Dato/initialer
1.0			7/8-2019 stbon
1.01	2	Miljøstyrelsen er dataejer.	18/9-2019 stbon
1.02	3 / 5	Testsystem: https://mab-pp.mst.dk/swagger/docs/v1	20/9-2019 stbon
2.01	1 4.2	Prøve id for Miljøstyrelsens fag/uddannelser Beskrivelse af hvordan man opdaterer et resultat	7/2-2020 stbon
2.02	3	Der må ikke indberettes tilmelding til et fremtidigt kursus.	19/2-2020 stbon
2.03	1	Opfølgingskursus kode 49491 tilføjet tabellen	1/5-2020 stbon
2.04	1	Fagkode 17086 tilføjet i tabellen	14/9-2020 stbon
2.05	1	Fagkoder 17086, M0006, M0010 og M0011 tilføjet i tabellen.	30/10-2020 stbon
2.06	1	Fagkode 49491 rettet (stod som 49419)	23/01-2021 stbon
2.06	0	Nyt logo på forsiden	23/01-2021 stbon

Indhold

1	Indledning	4
2	Adgang til kursus indberetnings funktionalitet	5
3	Webservice OpenAPI specification.....	5
4	Webservice operations beskrivelser	5
4.1	/Api/CourseRegistration/RegisterSignup	6
4.2	/Api/CourseRegistration/RegisterCompletion	6
4.3	/Api/Test/Ping	7
4.4	/Api/Test/PingAuth.....	7
5	Test med SwaggerUI.....	7
6	Kode eksempel	9
7	Request/response eksempler	12
7.1	Endpoint: /Api/CourseRegistration/RegisterCompletion.....	12
7.1.1	Med valideringsfejl	12
7.1.2	Uden valideringsfejl	12

1 Indledning

Dette dokument beskriver vilkår for at indberette oplysninger til Miljøstyrelsens Autorisationssystem for brug af Bekæmpelsesmidler (herefter kaldet MAB), som følge af nedlukning af EasyA.

MAB skal vide hvilke kurser en borger har bestået når borgeren ønsker en given autorisation. En autorisation udstedes på baggrund af et bestået kursus og MAB har hidtil forespurgt EasyA om et kursus er bestået. Når EUD- og AMU-institutionerne fra 1. juli 2020 ikke længere kan indberette til EasyA, skal oplysningerne vedr. kurser, indberettes til MAB.

Kurser (skolefag), som MAB udsteder autorisation på baggrund af, ses i denne oversigt.

Tabel 1. Fag og autorisationer

Fagidentifikation	Fag	Autorisation
03212	Sprøjteteknik (del 1 & 4)	Ingen
03233	Planteværn - Delprøve 2 og 3	S1 - Sprøjtetecertifikat
04342	Plantesundhedslære (Anlægsgartner)	S1 - Sprøjtetecertifikat
04670	Plantesundhed, sprøjtning (Greenkeeper)	S1 - Sprøjtetecertifikat
06588	Sprøjtetecertifikat (skovskolen)	S1 - Sprøjtetecertifikat
06992	Plantedyrkning (del 2 & 3)	S1 - Sprøjtetecertifikat
11963	Plantebeskyttelse, (delprøve 2 og 3)	S1 - Sprøjtetecertifikat
11965	Jordbrugsmaskinfører, (delprøve 2 og 3)	S1 - Sprøjtetecertifikat
11998	Plantebeskyttelse (gartneruddannelse)	S1 - Sprøjtetecertifikat
14247	Plantebeskyttelse, (delprøve 1 og 4)	Ingen
14249	Plantebeskyttelse 2 (delprøve 2+3)	S1 - Sprøjtetecertifikat
17086	Landmand, planter. Plantedyrkning.	S1 - Sprøjtetecertifikat
17087	Jordbrugsmaskinfører. Maskiner.	Ingen
40837	Sprøjtetecertifikat, opfølgning	S1 - Sprøjtetecertifikat
42250	Sprøjtetecertifikat, Skov	S1 - Sprøjtetecertifikat
42389	Plantebeskyttelse i gartneri, sprøjtetecertifikat (anlægsgartner)	S1 - Sprøjtetecertifikat
42390	Plantebeskyttelse i landbrug, Sprøjtetecertifikat	S1 - Sprøjtetecertifikat
47586	Opfølgningskursus, sprøjtetecertifikat gartneri	S1 - Sprøjtetecertifikat
47757	Forhandlercertifikat	F1 – Salg til private
48076	Opfølgningskursus, sprøjtetecertifikat skovbrug	S1 - Sprøjtetecertifikat
48142	Opfølgningskursus, sprøjtetecertifikat anlægsgartner	S1 - Sprøjtetecertifikat
48170	Hånd- og rygsprøjtetecertifikat	S2 - Hånd- og rygsprøjtetecertifikat
48175	Opfølgningskursus, sprøjtetecertifikat landbrug	S1 - Sprøjtetecertifikat
49491	Opfølgningskursus, hånd- og rygsprøjtetecertifikat	S2 – Hånd- og rygsprøjtetecertifikat
M0001	Almen uddannelse, grunduddannelse	Ingen
M0002	Landbrug, grunduddannelse	S1 - Sprøjtetecertifikat
M0003	Anlægsgartner, grunduddannelse	S1 - Sprøjtetecertifikat
M0004	Gartneri, grunduddannelse	S1 - Sprøjtetecertifikat
M0005	Skovbrug, grunduddannelse	S1 - Sprøjtetecertifikat
M0006	Hånd- og rygcertifikat, grunduddannelse	S2 - Hånd- og rygsprøjtetecertifikat
M0010	Bejdsecertifikat, grunduddannelse	S3 - Bejdsecertifikat
M0011	Forhandlercertifikat, grunduddannelse	F1 – Salg til private

Indberetningerne sker ved at skolens studieadministrative system kalder en webservice i MAB. Det er denne webservice, der beskrives i dokument.

Adgang til kursus indberetnings funktionalitet

For at få adgang til at indberette kursus informationer skal skolen oprettes i MAB af Miljøstyrelsen, som tildeler **ClientId** og **Secret**, der skal bruges når servicen kaldes.

Det er det studieadministrative systems opgave, såfremt det håndterer flere skoler, at kalde MAB med det korrekte **ClientId** og levere korrekt information.

Hvis man gerne vil i gang inden man har credentials kan man hente OpenAPI spec, generere en klient med AutoRest og kalde **/Api/Test/Ping** for at se at der er hul igennem. Herefter er det et spørgsmål om at koble credentials på for at kunne kalde de andre operationer.

Henvendelser til Miljøstyrelsen om at blive oprettet sendes i mail til pesticides@mst.dk

Ift. GDPR er Miljøstyrelsen dataejer af de informationer, der indberettes.

2 Webservice OpenAPI specification

WebAPI'et på MAB tillader skolerne at indberette oplysninger om kurser til MAB. Der kan indberettes 2 former for aktivitet.

1. Tilmelding til et fremtidigt kursus (**bruges ikke**)
2. Afslutning af kursus med bestået/ikke bestået status (påkrævet)

Ad 1) Det kan indberettes at eleven er tilmeldt kurset en gang i fremtiden. Denne mulighed benyttes ikke pga. forhold omkring GDPR.

Ad 2) Der skal indberettes resultatet af kursus/prøve.

API'et anvender OAuth2 ClientCredentials flow.

WebAPI'et er dokumenteret via OpenAPI så client kode kan genereres via f.eks. AutoRest (v 2.0.4283 eller nyere) (<https://github.com/Azure/autorest>)

Test: <https://mab-pp.mst.dk/swagger/docs/v1>

Prod: <https://mab.mst.dk/swagger/docs/v1>

3 Webservice operations beskrivelser

Der findes andre operationer end kursus indberetning på WebAPI'et, men ift. indberetning af kursus deltagelse/beståelse er det kun operationer under **/Api/CourseRegistration** der er interessante.

Til implementering/test er der også nogle test operationer så man kan teste at der er hul igennem både med og uden credentials. Disse test operationer findes under **/Api/Test**.

3.1 /Api/CourseRegistration/RegisterSignup (bruges ikke)

Bruges til at indberette en fremtidig kursustilmelding.

Request objekt:

```
{
  "CivilRegistrationIdentifier": "string", // Cpr nummer på borgeren
  "Signups": [ // Et array af kursus tilmeldinger
    {
      "CourseCode": "string", // UVM fagkode, eks. "32010"
      "StartDate": "2019-06-13T08:05:00.414Z", // Hvornår starter kurset
      "EndDate": "2019-06-13T08:05:00.414Z" // Hvornår slutter kurset
    }
  ]
}
```

Response object:

```
{
  "Errors": [ // Et array af fejl objekter
    {
      "Property": "string", // Hvilken property på request objektet er årsag til fejl, eks StartDate
      "Message": "string" // Hvad er der galt med property værdien
    }
  ],
  "Success": true // Angiver om request blev behandlet uden problemer (hvis true er Errors array tomt,
                  // hvis false findes der mindst et objekt i Errors)
}
```

3.2 /Api/CourseRegistration/RegisterCompletion

Bruges til at indberette et afsluttet kursus med information om kurset er bestået/ikke bestået.

Request objekt:

```
{
  "CivilRegistrationIdentifier": "string", // Cpr nummer på borgeren
  "Completions": [ // Et array af kursus afslutninger
    {
      "CompletionDate": "2019-06-13T08:05:00.434Z", // Dag for eksamen eller sidste kursus dag
      "Passed": true, // Er kurset bestået/ikke bestået
      "CourseCode": "string", // UVM fagkode, eks. "03212"
      "StartDate": "2019-06-13T08:05:00.434Z", // Hvornår startede kurset
      "EndDate": "2019-06-13T08:05:00.434Z" // Hvornår sluttede kurset
    }
  ]
}
```

Bemærk at CourseCode skal være på 5 tegn. Foranstil med 0 (nul) hvis kurset har 4 eller færre cifre. Der kan KUN anvendes de kursuskoder der er vist i tabel 1. Andre kursuskoder vil blive ignoreret af systemet og der kommer pt. ikke en fejlbesked.

Hvis man ikke kender start- og slut-dato for kurset, udfyldes disse felter med samme dato som for "CompletionDate".

En indberetning kan ikke slettes, men den kan opdateres.

Opdateringer af indberettet kursus data sker ved at vi sammenligner CourseCode og EndDate for et CPR-nummer med tidligere registreringer. Hvis disse felters værdier matcher med en tidligere registrering opdateres StartDate, CompletionDate og Passed. Hvis der ikke findes en tidligere registrering, oprettes det som en ny registrering.

Response object:

```
{
  "Errors": [ // Et array af fejlobjekter
    {
      "Property": "string", // Hvilken property på request objectet er årsag til fejl, eks StartDate
      "Message": "string" // Hvad er der galt med property værdien
    }
  ],
  "Success": true // Angiver om request blev behandlet uden problemer (hvis true er Errors array tomt,
                  // hvis false findes der mindst et objekt i Errors)
}
```

3.3 /Api/Test/Ping

Bruges til at teste at man kan kalde WepAPI'et.

Der er ingen parametre og response er en streng, f.eks.

"Pong @ 2019-06-13T08:52:40.8356099+00:00"

3.4 /Api/Test/PingAuth

Bruges til at teste at man kan kalde WebAPI'et med credentials.

Der er ingen parametre og response er en streng, f.eks.

"Pong auth @ 2019-06-13T08:52:40.8356099+00:00"

4 Test med SwaggerUI

SwaggerUI er slået til så man kan browse API'et og tilmed kalde det direkte fra browseren (med credentials).

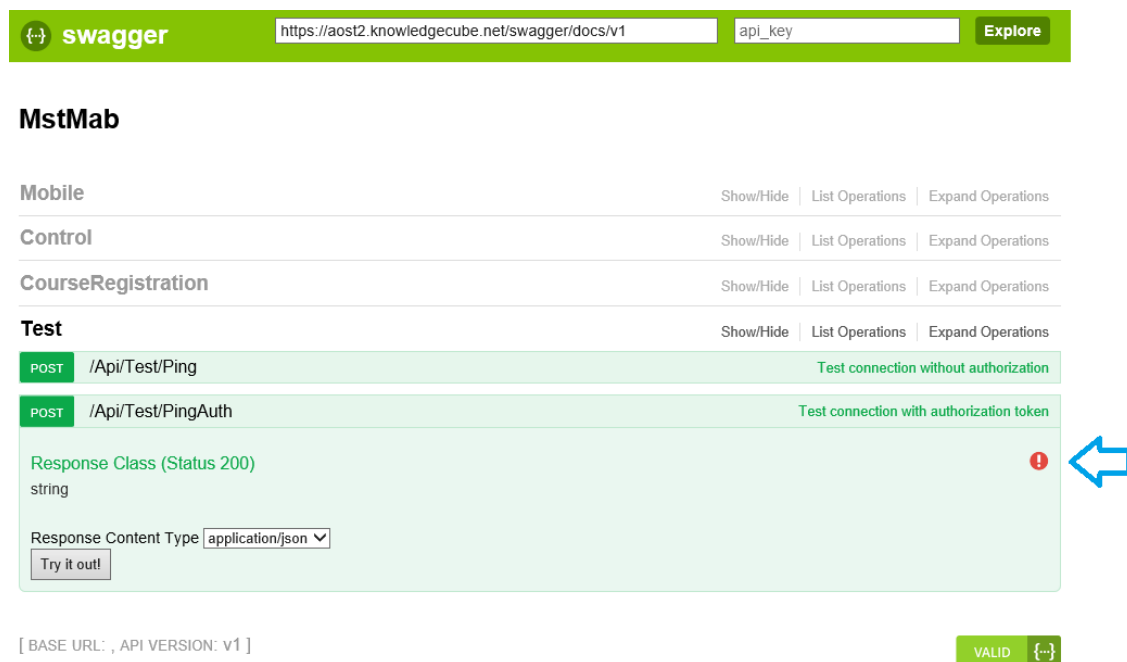
Note: Hvis man tester via SwaggerUI skal man bruge aost2.knowledgecube.net og ikke mabt2.knowledgecube.net, det skyldes at Identity serveren p.t. er registreret på aost2 URL, og den kan p.t. ikke håndtere requests fra SwaggerUI fra mabt2 URL.

Test: <https://mab-pp.mst.dk/swagger/>

Prod: <https://mab.mst.dk/swagger>

Det anbefales at starte med at teste credentials via SwaggerUI og /Api/Test/PingAuth inden man begynder kode implementering, så man er sikker på at man har gyldige credentials.

Hvis man vil teste credentials i SwaggerUI skal der trykkes på det lille røde advarselsikon for at indtaste credentials. Se pilen på figuren.



The image shows the SwaggerUI interface for the 'MstMab' API. The top bar includes the Swagger logo, the URL 'https://aost2.knowledgecube.net/swagger/docs/v1', an 'api_key' input field, and an 'Explore' button. The main content area lists several API endpoints: 'Mobile', 'Control', 'CourseRegistration', and 'Test'. The 'Test' tab is selected, showing two endpoints: 'POST /Api/Test/Ping' and 'POST /Api/Test/PingAuth'. The 'POST /Api/Test/PingAuth' endpoint is expanded, showing a 'Response Class (Status 200)' of type 'string' and a 'Response Content Type' of 'application/json'. A red warning icon is visible in the response area, with a blue arrow pointing to it. Below the response area, there is a 'Try it out!' button. At the bottom of the interface, there is a status bar showing '[BASE URL: , API VERSION: v1]' and a 'VALID' button with a Swagger logo.

Herefter vælges Basic auth og ClientId og Secret indtastes, CitizenEducationRegistration scope vælges og der trykkes Authorize.

Available authorizations

OAuth2.0

OAuth2 ClientCredentials flow

Token URL: <https://aost2.knowledgecube.net/identity/connect/token>

flow: application

Setup client authentication.

Type: **Basic auth** ▼

ClientId:

Secret:

API requires the following scopes. Select which ones you want to grant to Swagger UI.

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. [Learn how to use](#)

☒ CitizenEducationRegistration

Write access to course registration and completion

Authorize

Cancel

Det røde ikon er nu blå hvilket betyder at man er autoriseret logget på.



Man kan herefter kalde f.eks. **PingAuth** eller **CourseRegistration** operationer med credentials automatisk tilknyttet requests.

5 Kode eksempel

Et kode eksempel for en .Net Core console applikation.

Installer NuGet pakkerne:

- IdentityModel
- Microsoft.Client.Runtime

Hent OpenAPI specification for API'et

- <https://mabt2.knowledgecube.net/swagger/docs/v1>

Installer AutoRest (<https://github.com/Azure/autorest>)

- npm install -g autorest

Lav API client klasse med AutoRest

OpenAPI specifikationen er her gemt som mab.json i sin egen folder i projektet da AutoRest laver en del filer.

- autorest --input-file=mab.json --csharp --output-folder=mab_api --namespace=MabApi

```
using System;
using System.Net.Http;
using System.Threading.Tasks;
using IdentityModel.Client;
using MabApi.Models;

namespace MabApiSample
{
    class Program
    {
        private const string _baseUrl = "https://mabt2.knowledgecube.net/";
        private const string _issuer = _baseUrl + "identity/connect/token";
        private const string _clientId = "<spørg mst>";
        private const string _secret = "<spørg mst>";

        /// <summary>Scope required for a school using API to submit citizen course registration + completion</summary>
        private const string ScopeCitizenEducationRegistration = "CitizenEducationRegistration";

        static void Main(string[] args)
        {
            var t1 = CourseRegistrationCompletionTest();
            var t2 = PingTest();
            Task.WaitAll(t1, t2);
        }
    }
}
```

```

        Console.WriteLine("Done. Hit any key to exit");
        Console.ReadKey();
    }

    /// <summary>Test that we can use autorest client to post data to MAB API</summary>
    public static async Task PingTest()
    {
        var client = new MabApi.MstMab(new Uri(_baseUrl));

        var pingReq = await client.Test.PingWithHttpMessagesAsync();
        Console.WriteLine(pingReq.Response.Content.ReadAsStringAsync().Result);

        var token = await GetClientToken();
        // will fail with unauthorized status code
        //var pingAuthReq = client.Test.PingAuthWithHttpMessagesAsync().Result;
        client.HttpClient.DefaultRequestHeaders.Authorization = new System.Net.Http
.Headers.AuthenticationHeaderValue("bearer", token.AccessToken);
        var pingAuthReq = await client.Test.PingAuthWithHttpMessagesAsync();
        Console.WriteLine(pingAuthReq.Response.Content.ReadAsStringAsync().Result);
    }

    /// <summary>Test that we can call course completion endpoint on API</summary>
    public static async Task CourseRegistrationCompletionTest()
    {
        var client = new MabApi.MstMab(new Uri(_baseUrl));
        var token = await GetClientToken();
        client.HttpClient.DefaultRequestHeaders.Authorization = new System.Net.Http
.Headers.AuthenticationHeaderValue("bearer", token.AccessToken);
        var request = new RegisterCompletionParameters
        {
            CivilRegistrationIdentifier = "1231231231",
            Completions = new[]
            {
                new CourseCompletion
                {
                    CourseCode = "12345",
                    StartDate = DateTime.Now,
                    EndDate = DateTime.Now,
                    CompletionDate = DateTime.Now.AddDays(1), // will result in val
                    idation error '{"Errors":[{"Property":"CompletionDate","Message":"Fremtidig dato er ikk
                    e tilladt"}],"Success":false}'
                    Passed = false,
                }
            }
        };
        var response = await client.CourseRegistration.RegisterCompletionWithHttpMe
        ssagesAsync(request);
        Console.WriteLine(response.Response.Content.ReadAsStringAsync().Result);
    }

    private static async Task<TokenResponse> GetClientToken(string issuer = _issuer
    , string clientId = _clientId, string secret = _secret, string scope = ScopeCitizenEduc
    ationRegistration)
    {
        var client = new HttpClient();
    }

```

```
        var response = await client.RequestClientCredentialsTokenAsync(new ClientCr  
edentialsTokenRequest  
    {  
        Address = issuer,  
        ClientId = clientId,  
        ClientSecret = secret,  
        Scope = scope,  
    });  
  
        return response;  
    }  
}
```

6 Request/response eksempler

6.1 Endpoint: /Api/CourseRegistration/RegisterCompletion

6.1.1 Med valideringsfejl

Request:

```
{
  "CivilRegistrationIdentifier": "1231231231",
  "Completions": [
    {
      "CompletionDate": "2019-06-14T16:00:08.7606701+02:00",
      "Passed": false,
      "CourseCode": "12345",
      "StartDate": "2019-06-13T16:00:08.7605799+02:00",
      "EndDate": "2019-06-13T16:00:08.7606284+02:00"
    }
  ]
}
```

Response:

```
{
  "Errors": [
    {
      "Property": "CompletionDate",
      "Message": "Fremtidig dato er ikke tilladt"
    }
  ],
  "Success": false
}
```

6.1.2 Uden valideringsfejl

Request:

```
{
  "CivilRegistrationIdentifier": "1231231231",
  "Completions": [
    {
      "CompletionDate": "2019-06-12T16:04:30.9787041+02:00",
      "Passed": false,
      "CourseCode": "12345",
      "StartDate": "2019-06-13T16:04:30.9785814+02:00",
      "EndDate": "2019-06-13T16:04:30.9786357+02:00"
    }
  ]
}
```

Response

```
{
  "Errors": [],
  "Success": true
}
```