

Migration Dimona REST webservice v1 → v2

Contents

Introduction	2
Différences entre v1 et v2.....	2
POST /declarations.....	2
Explication.....	2
Différences dans le body.....	2
GET /declarations/{declarationId}/{dimonaNbr}.....	11
Explication.....	11
Différences.....	11
Nouveautés dans la v2	12

Introduction

Une nouvelle version du webservice REST Dimona est déployé. Afin d'utiliser cette nouvelle version, quelques mises à jours seront nécessaires.

Ce document est document est destiné aux utilisateurs actuels du WS REST Dimona v1. Il détaille les différentes adaptations nécessaires pour migrer vers la v2.

Différences entre v1 et v2

POST /declarations

Explication

Permet de créer une nouvelle déclaration Dimona (In, Out, Update ou Cancel) mais également de créer des enregistrements journaliers (In, Update ou Cancel).

Différences dans le body

address

Changements au niveau « address » (peu importe l'endroit où se situe cet objet) :

- postBox ➔ boxNumber
- zipCode ➔ postCode
- city ➔ renommé municipality qui est composé de name et code¹

Paramètres identiques au niveau « address » :

- street
- houseNumber
- country

v1	v2
<pre>"address": { "street": "Rue de la gare", "houseNumber": "1", "postBox": "A", "zipCode": "1000", "city": "Bruxelles", "country": "999" }</pre>	<pre>"address": { "street": "Rue de la gare", "houseNumber": "1", "boxNumber": "A", "postCode": "1000", "municipality": { "name": "Bruxelles", "code": 99999 }, "country": 999 }</pre>

¹ code n'est pas utilisé pour la création de Dimona. Le champ « name » est suffisant.

employer

Changements au niveau « employer » :

- nssRegistrationNumber → employerId
- provisionalNssRegistrationNumber → employerId
- companyId → enterpriseNumber

v1	v2
<pre>"employer": { "nssRegistrationNumber": "5999999999", "provisionalNssRegistrationNumber": "5999999989", "companyId": "1827180768" }</pre>	<pre>"employer": { "employerId": 5999999999, "enterpriseNumber": "1827180768" }</pre>

worker

Changements au niveau “worker”:

- lastName ➔ familyName
- firstName ➔ givenName
- firstName middleName ➔ givenNames
- birthPlace ➔ placeOfBirth
- birthPlaceCountry ➔ countryOfBirth
- address (voir [Address](#))

Paramètres identiques au niveau « worker » :

- ssin
- initial
- birthDate
- nationality
- gender

v1	v2
<pre>"worker": { "ssin": "93379704628", "lastName": "Doe", "firstName": "John", "middleName": "Bob", "initial": "G", "birthDate": "2000-12-31", "birthPlace": "UCCLE", "birthPlaceCountry": "999", "nationality": "999", "gender": "0", "address": { "street": "Rue de la gare", "houseNumber": "1", "postBox": "A", "zipCode": "1000", "city": "Bruxelles", "country": "999" } }</pre>	<pre>"worker": { "ssin": "93379704628", "familyName": "Doe", "givenName": "John", "givenNames": "John Bob", "initial": "G", "birthDate": "2000-12-31", "placeOfBirth": "UCCLE", "countryOfBirth": 999, "nationality": 999, "gender": 0, "address": { "street": "Rue de la gare", "houseNumber": "1", "boxNumber": "A", "postCode": "1000", "municipality": { "name": "Bruxelles", "code": 99999 } }, "country": 999 }</pre>

declarationStatus

Changements au niveau « declarationStatus » :

- dimonaDeclarationId ➔ declarationId
- dimonaPeriodId ➔ renommé period qui contient id
- anomaliesCollection (avec errorId, labelFr, labelNI) ➔ renommé anomalies composé de errorId, label (composé de nl, fr et de), type et value.

- informationsCollection (composé de informationType et informationDetail) ➔ informationsCollection (composé de type et value)

Paramètres identiques au niveau « declarationStatus» :

- result

v1	v2
<pre> "declarationStatus": { "dimonaDeclarationId": "string", "result": "string", "dimonaPeriodId": "123456789123", "anomaliesCollection": [{ "errorId": "string", "labelFr": "string", "labelNl": "string" }], "informationsCollection": [{ "informationType": "string", "informationDetail": "string" }] } </pre>	<pre> "declarationStatus": { "declarationId": 678519516, "result": "string", "period": { "id": 123456789123 }, "anomalies": [{ "errorId": "string", "label": { "nl": "string", "fr": "string", "de": "string" }, "type": "string", "value": "string" }], "informationsCollection": [{ "type": "string", "value": "string" }] } </pre>

dimonaIn

Changements au niveau « dimonaIn » :

- [identique] references
 - declarationReference → declaration
 - employerReference → employer
 - workerReferencees → worker
- startingDate → startDate
- startingHour → startHour
- endingDate → endDate
- endingHour → endHour
- dimonaFeatures → features . Il est composé de :
 - [identique] jointCommissionNumber
 - [identique] workerType
 - [identique] subEntityNumber
 - usingEmployer → userEnterprise qui est composé de :
 - [identique] jointCommissionNumber
 - usingEmployerName → name
 - usingCompanyId → enterpriseNumber
 - [nouveau] employerId
 - Address → voir [Address](#)
- studentPlaceOfWork (composé de denomination et address) → studentPlaceOfWork (composé de name et address (voir [Address](#)))

Paramètres identiques au niveau « dimonaIn » :

- plannedDaysNumber
- plannedHoursNumber
- constructionControlCards (composé de firstMonthC32ANumber et nextMonthC32ANumber)
- smallStatutesInformation (composé de employmentNature, activityWithRisk, workerStatus)

v1	v2
<pre>"dimonaIn": { "references": { "declarationReference": "string", "employerReference": "string", "workerReference": "string" }, "startingDate": "2000-12-31", "startingHour": "15:45", "endingDate": "2000-12-31", "endingHour": "16:45", "plannedDaysNumber": 111, "plannedHoursNumber": 160, "dimonaFeatures": { "jointCommissionNumber": "124", "workerType": "STU", "subEntityNumber": "1234567891", "usingEmployer": { "jointCommissionNumber": "124",</pre>	<pre>"dimonaIn": { "references": { "declaration": "string", "employer": "string", "worker": "string" }, "startDate": "2000-12-31", "startHour": "1545", "endDate": "2000-12-31", "endHour": "1645", "plannedDaysNumber": 111, "plannedHoursNumber": 160, "features": { "jointCommissionNumber": "124", "workerType": "STU", "subEntityNumber": "1234567891", "userEnterprise": { "jointCommissionNumber": "124",</pre>

<pre> "usingEmployerName": "string", "usingCompanyId": "1075672849", "address": { "street": "Rue des nuages", "houseNumber": "10", "postBox": "B", "zipCode": "1000", "city": "Bruxelles", "country": "Belgique" } }, "constructionControlCards": { "firstMonthC32ANumber": "123456789123", "nextMonthC32ANumber": "123456789124" }, "studentPlaceOfWork": { "denomination": "studentPlace", "address": { "street": "Rue des étudiants", "houseNumber": "15", "postBox": "C", "zipCode": "1000", "city": "Bruxelles", "country": "Belgique" } }, "smallStatutesInformation": { "employmentNature": "worker", "activityWithRisk": 401, "workerStatus": "F1" } } </pre>	<pre> "name": "string", "enterpriseNumber": "1075672849", "employerId": 5999999999, "address": { "street": "Rue des nuages", "houseNumber": "10", "boxNumber": "B", "postCode": "1000", "municipality": { "name": "Bruxelles", "code": 99999 } }, "country": 999 }, } }, "constructionControlCards": { "firstMonthC32ANumber": "123456789123", "nextMonthC32ANumber": "123456789124" }, "studentPlaceOfWork": { "name": "studentPlace", "address": { "street": "Rue des étudiants", "houseNumber": "15", "boxNumber": "C", "postCode": "1000", "municipality": { "name": "Bruxelles", "code": 99999 } }, "country": 999 }, } }, "smallStatutesInformation": { "employmentNature": "worker", "activityWithRisk": 401, "workerStatus": "F1" } } } </pre>
---	--

dimonaOut

Changements au niveau « dimonaOut » :

- [identique] references
 - declarationReference → declaration
 - employerReference → employer
 - workerReferencees → worker
- endingDate → endDate
- dimonaNumber → periodId

v1	v2
<pre>"dimonaOut": { "references": { "declarationReference": "string", "employerReference": "string", "workerReference": "string" }, "dimonaNumber": "123456789123", "endingDate": "2000-12-31" }</pre>	<pre>"dimonaOut": { "references": { "declaration": "string", "employer": "string", "worker": "string" }, "periodId": 123456789123, "endDate": "2000-12-31" }</pre>

dimonaUpdate

Changements au niveau « dimonaUpdate » :

- [identique] references
 - declarationReference → declaration
 - employerReference → employer
 - workerReferencees → worker
- startingDate → startDate
- startingHour → startHour
- endingDate → endDate
- endingHour → endHour
- dimonaNumber → periodId

Paramètres identiques au niveau « dimonaUpdate » :

- plannedDaysNumber
- plannedHoursNumber

v1	v2
<pre>"dimonaUpdate": { "references": { "declarationReference": "string", "employerReference": "string", "workerReference": "string" }, "dimonaNumber": "123456789123", "startingDate": "2000-12-31", "startingHour": "15:45", "endingDate": "2000-12-31", "endingHour": "16:45", "plannedDaysNumber": "111", "plannedHoursNumber": "160", }</pre>	<pre>"dimonaUpdate": { "references": { "declaration": "string", "employer": "string", "worker": "string" }, "periodId": 123456789123, "startDate": "2000-12-31", "startHour": "1545", "endDate": "2000-12-31", "endHour": "1645", "plannedDaysNumber": 111, "plannedHoursNumber": 160 }</pre>

dimonaCancel

Changements au niveau « dimonaCancel » :

- [identique] references
 - declarationReference → declaration
 - employerReference → employer
 - workerReferencees → worker
- dimonaNumber → periodId

v1	v2
<pre>"dimonaCancel": { "references": { "declarationReference": "string", "employerReference": "string", "workerReference": "string" }, "dimonaNumber": "123456789123" }</pre>	<pre>"dimonaCancel": { "references": { "declaration": "string", "employer": "string", "worker": "string" }, "periodId": 123456789123 }</pre>

GET /declarations/{declarationId}/{dimonaNbr}

Explication

Permet de retrouver le statut d'une déclaration sur base du numéro de Dimona

Différences

Le parameter dimonaNbr (v1) a été modifié en declarationId (v2).

Nouveautés dans la v2

Dans cette nouvelle version du webservice, il est également possible de

- **Rechercher une (des) déclaration(s) Dimona en se basant sur un ou plusieurs paramètres** grâce au POST /declarations/search. Pour toute recherche, il faudra compléter au moins l'une de ces informations : employerId ou entrepriseNumber ou entrepriseNumber of a userEnterprise ou NISS.
- **Recherche une période en se basant sur le numéro de période** grâce au GET /periods/{periodId}.
- **Rechercher une liste de(s) période(s) Dimona en se basant sur un ou plusieurs paramètres** grâce au POST /periods/search. Pour toute recherche, il faudra compléter au moins l'une de ces informations : employerId ou entrepriseNumber ou entrepriseNumber of a userEnterprise ou NISS.
- **Rechercher une liste de(s) relations(s) Dimona en se basant sur un ou plusieurs paramètres** grâce au POST /relations/search. Pour toute recherche, il faudra compléter au moins l'une de ces informations : employerId ou entrepriseNumber ou entrepriseNumber of a userEnterprise ou NISS.
- **Créer des enregistrements journaliers** grâce au POST /declarations. En effet, la création des enregistrements journaliers est réalisée de la même manière que la création des Dimona In/Out/Update/Cancel. Il est maintenant possible de créer des DailyRegistration In, Update et Cancel.
- **Rechercher un (des) enregistrement(s) journalier(s)** en se basant sur un ou plusieurs paramètres grâce au GET /dailyRegistrations. Pour toute recherche, il faudra compléter au minimum le numéro de période de la Dimona.
- **Rechercher un enregistrement journalier** en se basant sur le numéro d'enregistrement journalier grâce au GET /dailyRegistrations/{dailyRegistrationId}.

Toutes les informations concernant ces nouveautés se trouvent dans la documentation ([openapi dimona v2](#)).