

# CoreAPI - Starterspakket

## Basis functionele documentatie

### Algemene weetjes

#### Deprecated endpoints en attributen

Omdat er regelmatig nieuwe functionaliteiten worden voorzien is het niet mogelijk om alle bestaande endpoints en attributen eendeloos te blijven ondersteunen. Om aan te geven welke endpoints en attributen op termijn zullen verdwijnen wordt er gewerkt met de aanduiding deprecated.

Via Swagger zijn de deprecated endpoints duidelijk aangeduid doordat ze doorstreept zijn. Voorbeeld.

#### Occupation

POST	<del>/integration/occupation</del>
DELETE	<del>/integration/occupation/{occupationId}</del>
GET	<del>/integration/occupation/{occupationId}</del>
PATCH	<del>/integration/occupation/{occupationId}</del>

Deprecated attributen vind je in Swagger bij de response- en request modellen. Voorbeeld.

```
company (integer, optional): [Deprecated: ] [Deprecated: the company id linked to the branch will be used],  
contact (Array[Integration.Communication.PersonCommunication], optional): [Deprecated: ] [Deprecated: Use PersonCommunication endpoints instead],
```

Meer informatie vind je via deze link ([NL](#) en [FR](#))

Een overzicht van de deprecated endpoints/attributen en de eventuele alternatieven vind je via [deze pagina](#)

#### Datamodellen van de response en request

Via Swagger kan je meer informatie vinden over de request en response data-modellen. Per attribuut staat er een beknopte functionele beschrijving en referenties naar andere endpoints indien relevant.

Klik op "Model" bij het response model, bv :

GET /integration/client/{clientId}/communication

Retrieves all existing communications from a client

## Implementation Notes

Returns the information of all communication options of the client with the given ID.

## Response Class (Status 200)

The found communications

**Model** Example Value

```
Integration.Communication.ClientCommunication {
  id (integer, optional): An unique never changing identifier. An unique never changing identifier,
  isdefault (boolean, optional): True if the communication is default. Default communications are used with serveral processes, ie. print documents, mailings, ....
  True if the communication is default. Default communications are used with serveral processes, ie. print documents, mailings, ....
  type (string): A type to identify the communication, for example e-mail, mobile, .... A type to identify the communication, for example e-mail, mobile, .... Issue a
  GET request to /integration/codes with kind 13 for all possible codes,
  value (string): The e-mail address, mobile number, fax number, .... according the communcation type. The e-mail address, mobile number, fax number, ....
  according the communcation type
}
```

Of bij het request model, bv :

**Model** Example Value

```
Integration.Client.ClientCommunicationToAdd
{
  isdefault (boolean, optional): True if the
  communication is default. Default
  communications are used with serveral
  processes, ie. print documents, mailings, ....
  type (string): A type to identify the
  communication, for example e-mail, mobile, ....
  Issue a GET request to /integration/codes with
  kind 13 for all possible codes,
  value (string): The e-mail address, mobile
  number, fax number, .... according the
  communcation type
}
```

Over sommige attributen (we zijn ze nog verder aan het aanvullen) kan je *hoveren* zodat nog extra informatie wordt getoond.

vb. de max. lengte

```
instead],
externalid (string, optional): External id that can be used to store a third
party id,
financialrating (string, optional): The financial rating of the client.. Issue
a GET request to /integration/codes with kind 19 for all possible codes,
homepage (string, optional): The homepage (url) of the client.,
info (string, optional): Information in free text
```

## Authenticatie met een token

Als de API wordt geïnstalleerd zal ook een authenticatietoken worden doorgegeven. Dit token moet samen met de string “WB “ bij elke call naar de API worden doorgegeven. Bijvoorbeeld:

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: WB  
ABC123IKBENEENTOKEN'  
'http://<url>/webservice/integration/codes?kind=1000&language=nl'
```

In principe vervalt een API token niet, al zou er op eenvoudige aanvraag een nieuw token kunnen worden gegenereerd indien nodig.

## Welk zijn de verplichte gegevens in het request model?

Via Swagger kan je per attribuut te weten komen of het gegeven al dan niet verplicht is. Attributen met aanduiding “optional” zijn niet verplicht. Attributen zonder aanduiding zijn wel verplicht.

```
Integration.Client.Contact.ClientContactToAdd {  
  active (boolean): Active clientcontact can be used ie. for employment  
    contracts. Non-active clientcontact cannot be used anymore,  
  birthdate (string, optional): Date of birth,  
  communications  
    (Array[Integration.Client.Contact.ClientContactCommunicationToAdd],  
    optional): A list of communications of the client,  
  externalid (string, optional): External id that can be used to store a third  
    party id,  
  firstname (string): First name,
```

## Codes en omschrijvingen

Gegevens in PratoFlex worden vaak gemodelleerd met generieke codes. D.w.z. dat het uitzendbedrijf voor bepaalde lijsten zelf nieuwe codes kan aanmaken, wijzigen en verwijderen.

De response modellen van de API geven enkel de waarde van de code terug. Als de omschrijving nodig is kan dat worden opgevraagd via het endpoint **GET /integrations/codes**.

Als je bijvoorbeeld de communicatiekanalen van een klant wilt ophalen via **GET /integration/client/{clientId}/communication**, dan merk je in Swagger dat dit type gegeven aan codesoort 13 gekoppeld is:

GET /integration/client/{clientId}/communication

Retrieves all existing communications from a client

## Implementation Notes

Returns the information of all communication options of the client with the given ID.

## Response Class (Status 200)

The found communications

Model | Example Value

## Integration.Communication.ClientCommunication {

**id** (integer, optional): An unique never changing identifier. An unique never changing identifier,**isdefault** (boolean, optional): True if the communication is default. Default communications are used with serveral processes, ie. print documents, mailings, .... True if the communication is default. Default communications are used with serveral processes, ie. print documents, mailings, ....**type** (string): A type to identify the communication, for example e-mail, mobile, .... A type to identify the communication, for example e-mail, mobile, .... Issue a GET request to /integration/codes with kind 13 for all possible codes.**value** (string): The e-mail address, mobile number, fax number, .... according the communication type. The e-mail address, mobile number, fax number, .... according the communication type

}

Je kan dan via de volgende GET de codes ophalen die bij codesoort 13 horen:

`https://<url>/webservice/integration/codes?kind=13&language=nl`  
 (language=fr voor de Franse omschrijvingen)

Als respons krijg je dan een oplijsting van alle opties die bij het type 'communicatiekanalen' van de klant horen.

```
[
  {
    "description": "Telefoon",
    "descriptionshort": null,
    "externalcode": null,
    "id": "1",
    "kind": "13",
    "language": "nl",
    "validfrom": null,
    "validuntil": null
  },
  {
    "description": "Fax",
    "descriptionshort": null,
    "externalcode": null,
    "id": "2",
    "kind": "13",
    "language": "nl",
    "validfrom": null,
    "validuntil": null
  },
  enzovoort
]
```

## Werking van een PATCH

Bij alle PATCH api-calls die gebeuren, gaan we er vanuit dat **enkel** de velden die aangepast moeten worden, meegestuurd worden. Alle velden die niet meegestuurd worden, behouden dus hun originele waarde.

# HTTP Responses

We hanteren de algemene HTTP responses volgens bv.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

De meest gebruikte zijn :

- 200 OK
- 201 Created
- 202 Accepted
- 204 No Content : De DELETE endpoints geven dit terug
- 400 Bad Request : Er is een (functionele) foutmelding teruggeven met de response. De foutmelding zou in de meeste gevallen moeten verklaren wat er moet gecorrigeerd worden. Vaak zijn foutmeldingen te interpreteren door een gebruiker die de gegevens aanpast.
- 401 Unauthorized : Is het token correct? zie ook [Authenticatie met een token](#)
- 404 Not Found : Zijn de id's correct?
- 500 : Er is een systeemfout gebeurd (vb. API is down omwille van een upgrade maar ook onverwachte foutmeldingen). **Deze requesten kan je via een retry mechanisme opnieuw uitvoeren, vb. 5x. Indien deze foutmeldingen niet opgelost geraken is het noodzakelijk dat dat melden via [customercare@prato.be](mailto:customercare@prato.be)**

## Callback URL's

Sommige calls kunnen synchroon geen sluitend antwoord geven over het resultaat van een operatie. Bijvoorbeeld als er een contract verwijderd wordt, en daar is een Dimonnummer aan gekoppeld, dan moeten we eerst het Dimonnummer annuleren bij de RSZ. Dit verloopt via een apart proces en daar kunnen we niet synchroon op wachten.

Daarom moet er een callback URL doorgegeven worden, die wij gebruiken om het resultaat van de actie uiteindelijk aan mee te delen. Als je hiervan gebruik wilt maken dan moet er éénmalig een Base URL door Prato geregistreerd worden. Als de callback URL met OAuth2.0 wordt beveiligd, dan moet ook de nodige gegevens hiervoor aan Prato doorgegeven worden (client ID, client secret, OAuth Service URL, scope, ...).

Indien je wenst gebruikt te maken van callback URL's contacteer dan Prato via [customercare@prato.be](mailto:customercare@prato.be) met vermelding van de base URL en de OAuth2.0 gegevens.

Voorbeeld : Verwijder een contract met een Dimona-nummer : het contract kan pas verwijderd worden als Dimona geannuleerd is.

Wanneer de integrerende partij een call gaat uitvoeren om een contract te verwijderen, moet deze dan ook 2 extra query parameters doorgeven. De eerste is de relatieve URL. Dit wordt geplakt aan de baseurl uit de voorgenoemde parameter.

De andere is de versie van de callback logica die gebruikt wordt. Dit mechanisme laat ons toe om improvements en veranderingen in te bouwen in het callback mechanisme, en dat integrerende partijen op eigen tempo kunnen overgaan naar de nieuwe manier van werken.

```
curl -X DELETE --header 'Accept: application/json' --header 'Authorization: WB ABC123IKBENEENTOKEN' 'http://<url>/webservice/integration/employmentContract/7757693?url=relative%2Fcallback%2Furl&version=V1'
```



# Use cases

## Basisflow



Uitzendkracht



Klant

<b>Creatie</b>	<p>Naam, taal Domicilie adres</p> <p>Naam, taal, BTW nummer Domicilie adres</p>
<b>Klaarmaken voor contract</b>	<p>Extra persoonsinformatie Fiscale gezinssituatie Verzendwijze documenten Statuut</p> <p>Werkregime (Vakbondsopties) Markeren als arbeidsklaar</p>
<b>Contract aanmaken</b>	<p>Statuut en werkregime Jobomschrijving en reden Q/S, ADV en stelsel Rooster Loon en coëfficient</p>
<b>Prestaties doorgeven</b>	<p>Contract Datum Uren en prestatiecode (prestaties) Aantal en premiecode (premies)</p>

## Persoon aanmaken

Een persoon aanmaken kan via **POST /integration/person**. Om een persoon aan te maken zijn slechts een aantal velden verplicht (zie Swagger), maar je kan natuurlijk al veel meer informatie meegeven bij het aanmaken. Ook moet een persoon verplicht een domicilie adres hebben. Met de volgende body kan je een persoon aanmaken:

```
{
  "branch": 1,
  "firstname": "Voorbeeldpersoon1",
  "name": "Prato",
  "language": 1,
  "addresses": [
    {
      "adresstype": 1,
      "city": "Hasselt",
      "country": 150,
      "street": "Kempische Steenweg",
      "houzenumber": "293/35",
      "validfrom": "2022-01-01T00:00:00.000Z",
      "zipcode": "3500"
    }
  ]
}
```

Opmerking : Het adrestype “domicilie” kan in bepaalde omgevingen (<5%) verschillend zijn van de waarde 1, dat kan voorkomen doordat voor de klant afwijkende codes zijn gedefinieerd. De correcte code voor de omgeving kan je opvragen via het endpoint GET /integration/codes met parameter *kind=86*

## Klant aanmaken

Een klant aanmaken kan via **POST /integration/client**. Om een klant aan te maken zijn slechts een aantal velden verplicht (zie Swagger), maar je kan natuurlijk al veel meer informatie meegeven bij het aanmaken. Ook moet een klant verplicht een adres hebben van type 100 (een domicilie adres). Met de volgende body kan je een klant aanmaken:

```
{
  "addresses": [
    {
      "city": "Hasselt",
      "country": 150,
      "houzenumber": "26",
      "street": "Gouverneur Verwilghensingel",
      "type": 100,
      "zipcode": "3500"
    }
  ],
  "branch": 1,
  "language": "1",
  "name": "Voorbeeldklant",
  "taxnumber": "BE0454896841"
}
```

Opmerking : in tegenstelling tot de adrestypes van een persoon is het domicilieadres voor een klant altijd de waarde 100.

## Persoon aanvullen zodat contracten kunnen worden aangemaakt

Om een contract voor een persoon te kunnen aanmaken, is extra contractgerelateerde informatie van een persoon vereist.

### Geboortedatum, nationaliteit en INSZ

Alvorens er een statuut kan aangemaakt worden, is er ook nog wat extra standaard persoonsinformatie nodig (geboortedatum, nationaliteit, INSZ).

De informatie kon ook via de POST worden meegegeven, indien ze later wordt aangevuld kan het via de **PATCH /integration/person**.

```
{
  "inss": "95072500528",
  "nationality": "150",
  "birthdate": "1995-07-25"
}
```

### Fiscale info

Er moet informatie meegegeven worden over de fiscale gezinssituatie van de persoon. Deze info kan meegegeven worden via **POST /integration/person/{personid}/taxationhistory**. Hier kan je gegevens meegeven zoals burgerlijke staat, aantal kinderen ten laste, enzovoort. Via de volgende body geven we door dat onze testpersoon ongehuwd is en we geven 1 kind ten laste mee:

```
{
  "maritalstate": "1",
  "validfrom": "2022-07-20",
  "childrenatexpensenumber": 1
}
```

### Verzendwijze documenten

De verzendwijze van documenten voor een persoon moet meegegeven worden. Deze info kan meegegeven worden via **PATCH /integration/person**. Via de volgende body stellen we een e-mail adres in als verzendwijze voor onze voorbeeldpersoon:

```
{
  "deliverymethoddocuments": {
    "deliverymethodtype": "1",
    "mailto": "testpersoon@prato.be"
  }
}
```

Opmerking : Het kan zijn dat de omgeving van de klant geconfigureerd is dat er geen e-mail en Unified Post mogelijk is. Zeker voor testomgevingen kan dat het geval zijn. Zo'n omgevingen verwachten geen actie rond verzendwijze, maar we raden toch aan om deze call altijd te voorzien omdat de productieomgevingen vaak wel geconfigureerd zijn met e-mail en Unified Post.

### Statuut

Een persoon heeft ook een statuut nodig, voor hij een contract kan krijgen. Deze info kan meegegeven worden via **PATCH /integration/{personid}/statute**. Hiermee geef je een categorie mee (arbeider, bediende, student, flex), en een betaalperiode (wekelijks, maandelijks). Via het taxtypeinternal, configureer je hoe het BV percentage moet worden vastgelegd. Met de volgende body geven we onze voorbeeld persoon een arbeidsstatuut.



```
{
  "branchid": 1,
  "section": "115",
  "taxtypeinternal": 1
}
```

Opmerking : Let op, de code voor dit statuut kan verschillen van omgeving tot omgeving. De waardes voor de klant kunnen opgevraagd via GET /integration/codes met *kind=96*

## Een prospect/klant aanvullen zodat contracten kunnen worden aangemaakt

Om een contract te kunnen aanmaken, is bepaalde informatie van een klant vereist.

### Werkregime

In het werkregime van een klant geef je gegevens mee zoals werknemerscategorie, paritair comité, aantal betaalde/onbetaalde ADV, aantal gewerkte uren bij fulltime, enzovoort. Dit kan ingesteld worden via **POST** of **PATCH** /integration/client/{clientid}/workregime. Via de volgende body configureren we het werkregime voor een Arbeider, paritair comité 125, een GUR (=gemiddeld uurrooster) van 38, RUR (reëel uurrooster) van 40 met betaalde ADV (arbeidsduurvermindering)

```
{
  "averagetotalworkhours": 38,
  "category": "1",
  "description": "Arbeider 38/40 betaalde ADV",
  "jointcommittee": "125",
  "realtotalworkhours": 40,
  "validfrom": "2022-07-20",
  "workingsystem": "5",
  "worktimereductionfulltime": "paid",
  "worktimereductionparttime": "paid"
}
```

### Klant markeren als arbeidsklaar

Voor je aan een klant contracten kan toekennen, moet deze gemarkeerd worden als arbeidsklaar. Dit kan via **PUT** /integration/client/{clientid}/setreadyforemploymentcontracts. Er is geen body nodig voor deze PUT.

## Contract aanmaken

Een contract aanmaken kan via **POST** /integration/employmentcontract. Hier moet verplicht meegegeven worden over welke persoon en welk statuut het gaat, alsook over welk klanten werkregime. Verder moet je hier ook het rooster meegeven en informatie zoals het loon. In de volgende body, maken we een contract aan voor de voorbeeldpersoon bij de voorbeeldklant.

{{clientId}} = de id van de klant die in de vorige stappen werd aangemaakt

{{employeeid}} = de id van het statuut (**niet** de persoon) die in de vorige stappen werd aangemaakt

{{workregimeld}} = de id van het werkregime dat in voor de klant in de vorige stappen werd aangemaakt

```
{
  "branch": 1,
  "begindate": "2022-07-25",

```

```

"enddate": "2022-07-31",
"clientid": {{clientid}},
"statuteid": {{employeeid}},
"jobdescriptionid": "409",
"hourlywage": 12.39,
"coefficient": 1.91,
"selectiontempworker": "1",
"remunerationmethod": "0",
"type": "0",
"turnoverbranch": "1",
"reasonofemployment": "2",
"meansoftransport": "2",
"clientworkregimeid": {{workregimeid}},
"workingsystem": 5,
"q": 40.00,
"s": 40.00,
"weeklyhoursfulltime": 38.00,
"weeklyhoursparttime": 0.00,
"worktimereductionpaid": 2.00,
"worktimereductionunpaid": 0.00,
"workschedule": {
  "monday": [
    {
      "timeworked": "8"
    }
  ],
  "tuesday": [
    {
      "timeworked": "8"
    }
  ],
  "wednesday": [
    {
      "timeworked": "8"
    }
  ],
  "thursday": [
    {
      "timeworked": "8"
    }
  ],
  "friday": [
    {
      "timeworked": "8"
    }
  ]
}
}

```

## Prestaties en premies ingeven

Voor contracten kan je prestaties en premies ingeven.

Prestaties zijn het effectief aantal uren dat er per dag gewerkt/gepresteerd is. Deze aantallen kunnen afwijken van de contracturen, zowel minder als meer.

Premies zijn extra verloningen (vb. 1€ extra per gewerkte dag, 10€ ploegenpremie, 15€ premie nachtarbeid, ...).

## Prestaties ingeven

Je kan via **POST /integration/employmentcontract/{employmentcontractid}/performances** prestaties doorgeven voor een contract. De prestatiecode, datum en aantal uren zijn verplicht. Optioneel kan je ook kostenplaats, post en afdeling meegeven. Via het onderstaande voorbeeld worden voor 3 dagen prestaties meegegeven:

De mogelijke waarden voor de *code* kan je vinden via GET /integration/codes met *kind=2000*

```
[
  {
    "code": "AD",
    "date": "2022-07-25",
    "hours": 8.0
  },
  {
    "code": "AD",
    "date": "2022-07-26",
    "hours": 8.0
  },
  {
    "code": "AD",
    "date": "2022-07-27",
    "hours": 8.0
  }
]
```

## Premies ingeven

Je kan via **POST /integration/employmentcontract/{employmentcontractid}/premiums** premies ingeven voor een contract. De premiecode, aantal premies en datum zijn verplicht. Optioneel kan je ook bedrag, kostenplaats, post en afdeling meegeven. Via het onderstaande voorbeeld worden voor 3 dagen premies meegegeven:

De mogelijke waarden voor de *code* kan je vinden via GET /integration/codes met *kind=1000*

```
[
  {
    "code": "201",
    "date": "2022-07-25",
    "number": 1,
    "amount": 10
  },
  {
    "code": "201",
    "date": "2022-07-26",
    "number": 1,
    "amount": 10
  },
  {
    "code": "201",
    "date": "2022-07-27",
    "number": 1,
    "amount": 10
  }
]
```

# Queues

Er is een mogelijkheid om een wijzigingen die in PratoFlex gebeuren aan personen, klanten en contracten via een queue te verwerken.

vb. Een ander systeem is master van de gegevens naam & adres voor een persoon, maar niet voor de fiscale gegevens (vb. aantal kinderen ten laste). Het andere systeem is wel geïnteresseerd in de fiscale gegevens.

- Het andere systeem zal via de API de naam en adres van een persoon up2date houden
- Als een consulent via PratoFlex het aantal kinderen ten laste wijzigt zal PratoFlex de id van de persoon publiceren naar een queue
- Het andere systeem kan de id van de queue halen en de nieuwe gegevens via de API opvragen
- De nieuwe fiscale gegevens kunnen dan ook in het andere systeem worden bewaard

Als je hier meer informatie over wenst contacteer dan de Customer Success Manager van de klant of [customercare@prato.be](mailto:customercare@prato.be)