1 Authentication

- HTTPS
- Anyone wishing to address Link2Prisma must have a certificate provided by Sodeco.
  This certificate contains a username that must also exist in Prisma. Based on this username, the employer's social-accounting secretariat itself can determine which rights are linked to this certificate.

2 Methodology

- Link2Prisma is an asynchronous API.
- Each request enters a queue which can be found on the web database.
- A unique identifier is sent as a response.
- The requests Result and WorkerExists are two exceptions to that. The API will process these directly.
- A locally run programme constantly reads and processes the queue.
- When the programme processes a record from the queue, the status of the record will be modified and, if possible, a response will be added.
- The request Result allows the client to check the status of the record and retrieve a response at any time through a unique identifier.

3 Remarks

- Dates are always communicated in yyyymmdd format.
- Link2Prisma is deployed by social secretaries, which means that each social secretariat has a separate URI.
- Test URI Sodeco: https://link2prisma.sodeco.info/link2prisma.svc
- Before receiving performances through Link2Prisma, they first have to be processed using a programme called INLEZEN EXTERNE PRESTATIES (reading external performances). Select the option LINK2PRISMA. Then select your employer(s) and run the programme. In order to do this, the employer's workflow must indicate under timekeeping that it uses Link2Prisma!

4 Employee data

4 1 Requesting employee data

Employee data are always requested per employer.

HTTP–METHOD:	GET
ADD HEADER:	Employer (= external number of the employer)
BODY:	*(not applicable)*

POTENTIAL ENDPOINTS :

/worker
Requesting all employees.
Personal data of the employee: previous contracts, address history, family status, contact information (phone number, email address, …), history of tax withholding, salary data.

This is limited to 100 employees.
To retrieve more than 100 employees, employees must be separated into groups. The labels ‘limit’ and ‘offset’ can be used for this.

Limit

- Indicates the number of employees
- Can be any number up to 100

Offset

- Indicates the starting position
- Does not equal the number of the employee, but their position in the list

EXAMPLE: GET worker?limit=100&offset=200

/worker/{StartDate}/{EndDate}
Retrieve the employees who are employed within the specified period (startdate en enddate)
>> see /worker

/worker/{workerID}
Retrieve one employee
>> see /worker

/worker/{workerID}/contract
Retrieve the contract history of an employee

/worker/{workerID}/contract/{RefDate}
Retrieve one contract from an employee
The contract being queried is the one that applies to the RefDate.

/worker/{workerID}/address
Retrieve the addresse history of an employee

/worker/{workerID}/address/{RefDate}
Retrieve one address of an employee
The address being queried is the one that applies to the RefDate.

/worker/{workerID}/communication
Retrieve all contact details of an employee
This includes phone, e-mail, cell phone from both work and private.
These cannot be retrieved separately, only per employee.

/worker/{workerID}/familystatus
Retrieve the history of an employee's familystatus

/worker/{workerID}/familystatus/{RefDate}
Retrieve one family status of an employee.
The family state being queried is the one that applies to the RefDate.

/worker/{workerID}/tax
Retrieve the history of an employee's (deviating) withholding tax calculation settings

/worker/{workerID}/tax/{RefDate}
Retrieve one (deviating) withholding tax calculation setting from an employee
The (deviating) withholding tax calculation setting being queried is the one that applies to the RefDate.

/worker/{workerID}/replacement/{RefDate}
Retrieve the replacement details for an employee.
The replacement detail being queried are the ones that apply to the RefDate.

/newWorkerNumber
Retrieve the next employee number

/modifications/{StartDate}/{EndDate}
Retrieval of all changes within the specified period (startdate and enddate)
Changes made by the requester itself are filtered out of the list.

/firstdayofunemployment/{firstDayOfUnemplID}
Retrieving the first day of unemployment
As FirstDayOfUnemplID you have to pass in the sequence number.

/worker/{workerID}/companycar/{carID}
Retrieving a company car belonging to an employee
Here carID is the startingdate of the company car.

/companycar{carID}
Retrieving a company car that is not linked to an employee
Here carID is the startdate of the company car.

/worker/{workerID}/companycar/
Retrieving the history of company cars belonging to an employee

/companycar
Retrieve the history of all company vehicles of an employer

/worker/{workerID}/divergentSalaryScale
Retrieval of an employee's deviating pay scales

/worker/{workerID}/absencenote
Retrieval of an employee’s sick notes.

/worker/{workerID}/absencenote/{noteID}
Retrieving a specific sick note
Here, the noteID is the date the sick note was created.

/worker/{workerID}/dimona
Retrieval of all dimonas of an employee

/worker/{workerID}/dimona/{Startdate}
Retrieving one dimona of an employee where the startdate is the start date of the Dimona

/absences/{startdate}
Retrieve the (approved) absences of all employees as of the specified startdate
The absences are grouped by employee.

/worker/{workerID}/absences/{startdate}
Retrieve an employee's (approved) absences from the specified startdate

/worker/{workerID}/divergentPayment
Retrieving an employee's deviating payments

/worker/{workerID}/costcentre
Retrieving all cost centers of an employee

/worker/{workerID}/costcentre/{startdate}
Retrieving the cost centers of an employee where startdate is the start date of the cost center

/worker/{workerID}/leavecounters/{year}
Retrieve the leave counters of an employee.
Year is the year in which the leave counter is active.
If it is not the current year, you get the end balance of that year.

leavecounters/{year}/{counterID}
Retrieve one type of leave counter for all employees of an employer.
CounterID is the number of a leave counter.
Year is the year in which the leave counter is active.
If it is not the current year, you get the end balance of that year.

/department

Requesting the departments of an employer.

/costcentre

Requesting the costcentres of an employer.

/parcom

Requesting the joint committees of an employer

/schedule

Requesting an employer’s timetables.

/nssocat

Requesting the social security categories of an employer.

/profcat

Requesting the occupational categories of an employer.
The occupational categories are listed per joint committee.
An employer may belong to several joint committees. In that case, you receive a list of professional categories for each joint committee.

RESPONSE	If the request was processed properly, a 202 "accepted" is returned. This means that the message has been accepted and is ready to be processed by our local program. The body of the response always contains a unique identifier. With this identifier you can follow up the request and retrieve the result.

4 2 Adding employee data

HTTP–METHOD:	POST
ADD HEADER	Employer (= employer’s external number)
BODY:	JSON

POTENTIAL ENDPOINTS:

/worker
Adding an employee
This includes all the employee's details, which are at least: Personal details, Contract, Address, Family status

/worker/{workerID}/contract
Adding a new contract

/worker/{workerID}/address
Adding a new address

/worker/{workerID}/communication
Adding a new contact detail

/worker/{workerID}/familystatus
Adding a new family status

/worker/{workerID}/tax
Adding a new (deviating) withholding tax calculation setting

/worker/{workerID}/replacement
Adding a new replacement

/absences
Adding absences
Several employees with their absences can be sent in one request.
If an absence already exists (same date and code) it will be changed.

/prestations

Adding performances

Several employees with their performances can be sent in one request.

If there are already performances for a certain employee/day, these will be overwritten by the new performances. In case of corrections, the entire day must be sent again.

/prestations/completed

After adding the performances they still have to be processed in Prisma via the program TRANSFERT OF EXTERNAL PERFORMANCES. Through this call this can be called and executed automatically. If this does not happen the social secretariat will have to do it manually.

/worker /{workerID}/ dimona

Request to send a dimona

When adding a new employee or changing a contract we do not automatically send a Dimona. If it is desired that we do the Dimona it must be done separately via this endpoint.

In order to do a Dimona IN, the employee does not yet need to exist in Prisma. However, one should already know what his future number will be so that the dimona can later be linked to the employee.

/firstdayofunemployment

Adding a first day of unemployment

When adding a first unemployment day, you don't have to provide a serial number. This is determined by us and can be requested via the unique identifier.

The ticket number is the number we receive back from the RSZ (national social security office) and therefore does not need to be included.

/worker/{workerID}/companycar

Adding a company car that belongs to an employee

/companycar

Adding a company that doesn’t belong to an employee

When a company car isn’t linked to an employee, catRSZ is mandatory

/worker/{workerID}/salarycomposition

Adding salary composition details to the employee

/worker/{workerID}/absencenote

Adding a sick note for an employee

/worker/{workerID}/divergentPayment
Adding a deviating payment to an employee

/schedule
Adding an employer’s timetable
Via the code of the timetable, it can later be linked to an employee.

/worker/{workerID}/costcentre
Adding one of more cost centers with a start date.
The sum of the percentages must always be 100

RESPONSE:	If the request was processed properly, a 202 "accepted" is returned. This means that the message has been accepted and is ready to be processed by our local program. The body of the response always contains a unique identifier. With this identifier you can follow up the request and retrieve the result.

4 3 Changing employee data

HTTP–METHOD:	PUT
ADD HEADER	Employer (= employer's external number)
BODY:	JSON (When modifying, only the mandatory fields and the modified fields must be forwarded. The fields that are not transmitted will not be modified.)

POTENTIAL ENDPOINTS:

/worker
Modifying an employee
All the employee's details must be sent, which are at least: Personal details, Contract, Address, Family status

/worker/{workerID}
Modifying an employee
This concerns the personal data of the employee
Contracts, adresses, etc.. must be modified seperately.

/worker/{workerID}/contract/{contractID}
Modifying a contract.
Contractid is the (old) startdate of the contract.

/worker/{workerID}/address/{addressID}
Modifying an addres
AddressID is the (old) startdate of the address.

/worker/{workerID}/familystatus/{familystatusID}
Modifying a family status.
In this context, \`familystatusID\` represents the (old) start date of the family status.

/worker/{workerID}/tax/{taxID}
Changing a special calculation of corporate withholding tax.
Here, \`taxID\` corresponds to the (old) start date of the special corporate withholding tax calculation.

/worker/{workerID}/replacement/{replDateID}

Altering a replacement.
In this case, `replDateID` represents the (old) start date of the replacement.

/firstdayofunemployment

Changing the first day of unemployment.
No ID needs to be provided via this endpoint. Instead, the body should contain either the sequence number or the ticket number.

/worker/{workerID}/companycar/{carID}

Modifying an employee's company car.
In this context, `carID` represents the starting date of the company car.

/companycar{carID}

Adding a company car not associated with an employee.
Here, `carID` corresponds to the starting date of the company car.

/worker/{workerID}/salarycomposition/{salarycode}/{salarydate}

Changing salary details.
The `salarycode` refers to the salary code you wish to modify, and `salarydate` is the start date of the salary details.

/worker/{workerID}/absencenote/{noteID}

Modifying a sick note for an employee.
In this case, `noteID` represents the date of issuance of the sick note.

/worker/{workerID}/divergentpayment/{startdate}/{typeID}

Changing a divergent payment, where `startdate` is the start date of the divergent payment, and `typeID` is the type of divergent payment.

/worker/{workerID}/costcentre/{startdate}

Modifying a cost center.
If multiple cost centers exist with the same start date, all of them must be re-submitted when making changes. The sum of the percentages for these cost centers should always equal 100%.

RESPONSE:

If the request was processed properly, a 202 "accepted" is returned.
This means that the message has been accepted and is ready to be processed by our local program.
The body of the response always contains a unique identifier.
With this identifier you can follow up the request and retrieve the result.

NOTICE
Contact details cannot be modified directly.
If you wish to make changes, they must first be deleted and then re-added.

4 4 Deletion of Employee Data

HTTP-METHOD:	DELETE
ADD HEADER:	Employer (= external number of the employer)
BODY:	*(not applicable except when deleting contact information)*

POTENTIAL ENDPOINTS:

/worker/{workerID}/contract/{contractID}
Deleting a contract.
Here, `contractID` is the start date of the contract.

/worker/{workerID}/address/{addressID}
Deleting an address.
Here, `addressID` is the start date of the address.

/worker/{workerID}/communication
Deleting contact information.
The body consists of the complete contact information to be deleted.

/worker/{workerID}/familystatus/{familystatusID}
Deleting a family status.
Here, `familystatusID` is the start date of the family status.

/worker/{workerID}/tax/{taxID}
Deleting a special calculation of corporate withholding tax.
Here, `taxID` is the start date of the special corporate withholding tax calculation.

/worker/{workerID}/replacement/{replDateID}
Deleting a replacement.
Here, `replDateID` is the start date of the replacement.

/absences
Deleting absences.
Multiple employees with their absences to be deleted can be sent in one request.

/prestations

Deleting performances.
Multiple employees with their performances to be deleted can be sent in one request.

/firstdayofunemployment

Deleting the first day of unemployment.
No ID needs to be provided via the endpoint, but the body should include either the sequence number or the ticket number.

/worker/{workerID}/companycar/{carID}

Deleting an employee's company car.
Here, `carID` is the starting date of the company car.

/companycar{carID}

Deleting a company car not associated with an employee.
Here, `carID` corresponds to the starting date of the company car.

/worker/{workerID}/salarycomposition/{salarycode}/{salarydate}

Deleting salary details.
Here, the `salarycode` is the salary code you wish to delete, and `salarydate` is the start date of the salary details.

/worker/{workerID}/absencenote/{noteID}

Deleting a sick note for an employee.
Here, `noteID` is the date of issuance of the sick note.

/worker/{workerID}/divergentpayment/{startdate}/{typeID}

Deleting a divergent payment, where `startdate` is the start date of the divergent payment, and `typeID` is the type of divergent payment.

/worker/{workerID}/costcentre/{startdate}

Deleting a cost center.
All cost centers on the `startdate` will be deleted.

RESPONSE: If the request was processed properly, a 202 "accepted" is returned.
This means that the message has been accepted and is ready to be processed by our local program.
The body of the response always contains a unique identifier.
With this identifier you can follow up the request and retrieve the result.

NOTICE
An employee cannot be deleted entirely. An employee must always have at least one contract and one address associated with them.

4 5 Requesting the Result

HTTP–METHOD:	GET
ADD HEADER:	*(none)*
BODY:	*(not applicable)*

ENDPOINT:

/Result/{GUID}
In this context, {GUID} represents the unique identifier that you received as a response during the original request.

RESPONSE:

If the request has been processed successfully, a 200 "OK" response will be returned, and the body will contain a JSON object.

The JSON object includes the following information:

- Unique identifier of the original request.
- Action of the original request (get, insert, update, delete).
- Type of the original request (employee, contract, address, etc.).
- Employer of the original request.
- Status of the original request: The status indicates whether the original request has been processed and if any errors occurred.
- Response: If the original request is a GET request and has been processed correctly, the response contains the requested data in JSON format. If the original request adds, modifies, or deletes data, the response remains empty except for performance-related requests. If there are performances that could not be processed, the response will contain an overview of those.

4 6 Checking if an Employee already exists

HTTP–METHOD:	GET
ADD HEADER:	Employer (= external number of the employer)
BODY:	*(not applicable)*

ENDPOINT:

/workerExistst/{rrnr}
In this context, {rrnr} represents the national registration number (rijksregisternummer) of the individual for whom the employee number is being requested.

RESPONSE:

If the request has been processed succesfully, a 200 (OK) response will be returned, and the body will contain a JSON-object.

The JSON-object is structured as follows:

- Unique identifier
- WorkerExists (True/False)
- WorkerNumber

If WorkerExists is False, the WorkerNumber will be 0.

5 DOCUMENTS

5 1 Requesting Documents

HTTP–METHOD:	GET
ADD HEADER:	Employer (external number of the employer)
BODY:	*(not applicable)*
ENDPOINTS:	

/document/listing/{StartDate}/{EndDate}
Requesting a list of available documents within the specified period.
The list includes documents for both the employer and the employees of that employer. Each document has a unique number, and this number can be used to retrieve the document itself.
The schema provides information about the types of documents that can be requested.

/document/listing/withdrawals/{StartDate}
Requesting a list of withdrawn documents from a specified date.
Withdrawn documents are documents that the social secretariat has recalled for some reason.
This request is NOT specific to an employer; the response includes the employer's number along with the document number.

/document/{documentID}
Retrieving a document using the document number.
The document number can be obtained through the listing.
The result consists of a base64 converted byte-array.

RESPONSE:	If the request was processed properly, a 202 "accepted" is returned. This means that the message has been accepted and is ready to be processed by our local program. The body of the response always contains a unique identifier. With this identifier you can follow up the request and retrieve the result.

5 2 Uploading documents to Prisma

HTTP–METHOD:	POST
ADD HEADER:	Employer (external number of the employeur)
BODY:	document.JSON
ENDPOINT:	/document
EXPLANATION OF THE DATA:	

Filename

The name of the file, including the file extension.

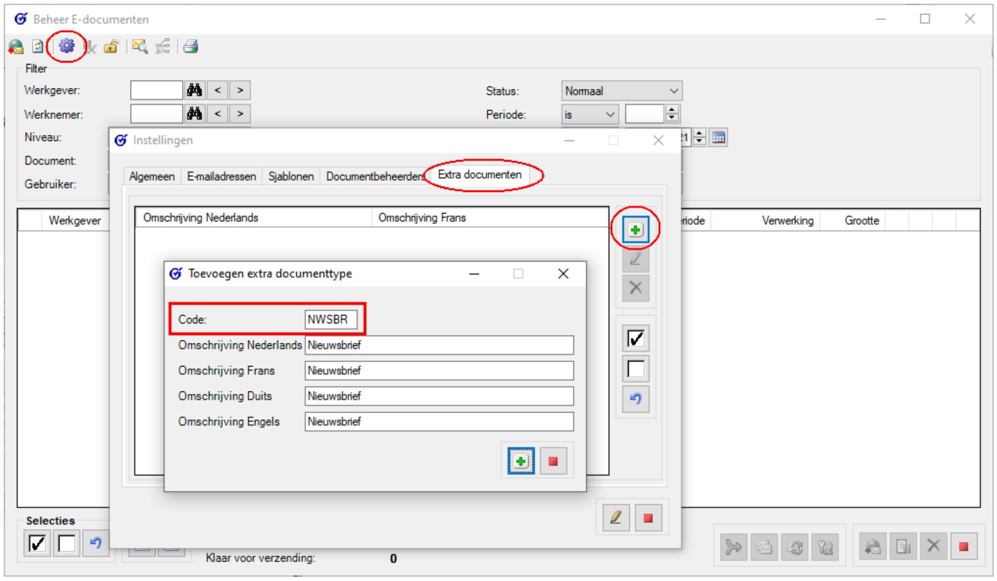
PublicationDate

The date associated with the document in Prisma (processing date).

DocumentType

The document types can be defined by the social secretariat itself using the SETTINGS button ⚙ in the programme BEHEER E-DOCUMENTEN *(Specifiek > Beheer e-documenten > Beheer e-documenten)*.
Document types are managed there on the tab EXTRA DOCUMENTEN. When uploading the document, the CODE set here should be provided.

EXAMPLE



Status: Normaal

Periode: is

Code: NWSBR

Omschrijving Nederlands: Nieuwsbrief

Omschrijving Frans: Nieuwsbrief

Omschrijving Duits: Nieuwsbrief

Omschrijving Engels: Nieuwsbrief

Klaar voor verzending: 0

WorkerID

The employee's identification number.
If left blank, the document is linked to the employer.

Month

The month to which the document is related.

Year

The year to which the document is related.

Language

The language of the document.

Document

Base64 representation of the byte stream of the document.

RESPONSE:	If the request was processed properly, a 202 "accepted" is returned. This means that the message has been accepted and is ready to be processed by our local program. The body of the response always contains a unique identifier. With this identifier you can follow up the request and retrieve the result.