# Next building block cd-customer-platform

## Create country database

See separate SQL dump, import as new table in database

Merge country-array.php with language file(s): we will need a country-name language file per language.

## Create first database supported grid “Companies”

First, import the provided company.sql and distribution\_partner.sql as a small feed

The companies grid will list all companies for a given user.

The data of the grid will come primarily from a API call to the application.

Important fundamental things we need to build here (which will be used for other request also is):

* Routing, we need to build a sensible routing system for the API calls.
  + Laravel supports routing configs for API apart from the main routes, we would like to use that option
  + We use RESTful API requests, for the companies request this means the endpoint will be something like /companies

With query parameters like these:

* + - offset: the paginator offset
    - sort: sorting key(s), can be multiple, separated by “,” chacter like: “sort=name:desc,city:asc” or a workable alternative notation (as advised by Laravel)
    - pagesize: the pagesize (default: 30)
    - term: generic search term, will query the backend for name, zipcode, address and city
    - ..must be easily expandable for more!
* Authentication: the API will be protected by a username:password authentication header (like a token). Here we will look at Laravel best practices first (from their docs).
  + Important for the scope of companies (and other requests) is the role a user has and to which distribution\_partner\_id is connected.

A distribution\_partner is a company (record) which itself is a client of CreditDevice B.V. this company has its own clients, but these clients will be registered in our database as they sell our products. We define 2 types of distribution partners:

1. Root: this is CreditDevice B.V. itself, meaning this distribution partner is the main connection to all companies. If a user is connected to a root distribution\_partner, this means all companies can be requested by the API.
2. Partner: this is a client of CreditDevice B.V., meaning this distribution partner has company records (referring in company table distribution\_partner\_id to the company record of the partner), only companies can be requested with distribution\_partner\_id of the partner

(in simple SQL, this part will always be added: WHERE company.distribution\_partner\_id = <PARTNER\_ID>

## Query

The result-set will be a combination of the following tables:

Company table

Connected to address, where address.is\_primary = 1 and address.company\_id = company.id

Connected to address, where address.country\_code = country.iso2\_code

The authentication will lead to the additional automatic filtering (company.distribution\_partner\_id = <PARTNER\_ID>) if required.

The API will give a JSON result back, for now, we will use the databasefields also as keys for the json resultset. The API will fetch all (active) addresses for a company and present the result in a structured way like the enclosed example (company.json) (may be subject to little changes or syntax mistakes). This example json is for one record, the api will return a json collection of many results of course.

The resultset will be like such a format (but Laravel may have standardized way of doing this, I do not know):

{

“collection”:[

….all company records here

],

“meta”:{

“items”:<total>

}

}

It should be possible to do a curl command with the correct endpoint and credentials to get the response we want. Like: curl –user <username>:<password> <endpoint>