# Companies grid

With the basic companies API in place, we can now use this functionality to create the companies grid. When creating this, we will most likely add new data/filter options to the API to make it work.

## Api changes

### Language parameter

We will want to pass a language parameter to the API, with this, specific fields can be translated to the appropriate languagecode. For instance: we want the country description also in our json response. This value has to be translated from the country translation table which we now have. The language parameter defines which language will be used. Two possible approaches for this:

1. Add the languagecode as an extra header to the API call (preferred);
2. Add the languagecode as an extra query parameter to the API call (less preferred).

### Companies

See the companies.json file for an updated view of the data in the companies API response. We now want all this data in the resultset.

### Address

We want to add, delete and get all addresses with a given company id. We will require the functions for that. Please use companies API functions as main guideline for the layout of the response in json. Use the database-fields as json response fields as much as possible. (every record in the json will be something like {“company\_id”:”1”, “is\_primary”:”1”, “street”:”Rudolf tappenbeckweg” etc….}

### Contact

We want to add, delete and get all contacts for a given company id. We will require the functions for that. Please use companies API functions as main guideline for the layout of the response in json. Use the database-fields as json response fields as much as possible.

### Segmentation

We want to add, delete and get all segments for a given company id. We will require the functions for that. Please use companies API functions as main guideline for the layout of the response in json. Use the database-fields as json response fields as much as possible.

Please run segmentation.sql to update the database. We now added distribution\_partner\_id to this table. This means: every distribution\_partner\_id has its own segmentation records.

### Contacttype

Use the contact\_type sql to change the contact\_type table to add the first contacttypes we will need.

## Grid display

The grid will display, by default, all companies visible for the current user. The grid, for now, will have the following columns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Name | City | Registration number | Segmentation | Owner |

The ideal flow will be that, using AJAX, the grid is filled. The paginator, filter box should also be made available and filter/paginate the results by using AJAX calls.

Registration number: comes from company table

Number: comes from company table (the client number)

City: comes from address table, join with address with flag “is\_primary” = 1. There can be only one address with primary for every company id.

Segmentation value comes from segmentation table (description field).

Owner comes from company\_contact, join with contact and join with contact\_type, contact\_type code needs to be “company\_owner”. There can be only one contact\_type with code “company” owner for every company\_id.