# 

|  |
| --- |
| DNB Eiendom |
| Meglerfront |
| Documentation |

Contents

[1 Background 6](#_Toc405799122)

[2 Meglerfront access 6](#_Toc405799123)

[2.1 The role of Daily Leader 6](#_Toc405799124)

[3 External data sources and dependencies 7](#_Toc405799125)

[3.1 Exhibition IOS application 7](#_Toc405799126)

[3.2 Meglerweb 7](#_Toc405799127)

[3.3 EmProf 8](#_Toc405799128)

[3.4 Posten.no 8](#_Toc405799129)

[3.5 DreamService 8](#_Toc405799130)

[3.6 MeglerfrontBackendService 8](#_Toc405799131)

[3.7 Bank – Tipsflyt alt. MSCRM/Dynamics 8](#_Toc405799132)

[3.7.1 Sending leads 8](#_Toc405799133)

[3.7.2 Receiving lead status 10](#_Toc405799134)

[3.7.3 Receiving leads 11](#_Toc405799135)

[4 Code structure 11](#_Toc405799136)

[4.1 Background 11](#_Toc405799137)

[4.2 Layers 11](#_Toc405799138)

[4.2.1 Frontend 11](#_Toc405799139)

[4.2.2 Business logic 12](#_Toc405799140)

[4.2.3 Data Access 12](#_Toc405799141)

[4.3 Shared 13](#_Toc405799142)

[4.4 Proxy projects 13](#_Toc405799143)

[4.5 Timer project 14](#_Toc405799144)

[5 Timer jobs 14](#_Toc405799145)

[5.1 AD-Sync 14](#_Toc405799146)

[5.2 AssessmentImport 14](#_Toc405799147)

[5.3 AutoprospectImport / FinnImport 15](#_Toc405799148)

[5.4 Queue job 16](#_Toc405799149)

[5.5 Residens 18](#_Toc405799150)

[6 Lead processing module 18](#_Toc405799151)

[6.1 Step validation/qualification 18](#_Toc405799152)

[6.2 Customer card (“Kundekort”) 19](#_Toc405799153)

[6.2.1 Qualify 19](#_Toc405799154)

[6.3 Current real estate (“Nåværende bolig”) 19](#_Toc405799155)

[6.3.1 Status 19](#_Toc405799156)

[6.3.2 Responsible (“Ansvarlig”) 19](#_Toc405799157)

[6.3.3 Qualify 19](#_Toc405799158)

[6.4 Dream match (“Ønskat bolig”) 20](#_Toc405799159)

[6.4.1 Qualify 20](#_Toc405799160)

[6.5 Economical advice (“Økonomisk rådgivning”) 20](#_Toc405799161)

[6.5.1 Qualify 20](#_Toc405799162)

[7 Qualify processing module 20](#_Toc405799163)

[7.1 Interest step (“Interesse”) 21](#_Toc405799164)

[7.2 Status 21](#_Toc405799165)

[7.3 Bidding 21](#_Toc405799166)

[7.4 Level of interest 21](#_Toc405799167)

[7.5 Source 21](#_Toc405799168)

[7.5.1 Qualify 22](#_Toc405799169)

[8 Merge 22](#_Toc405799170)

[8.1 Duplicate issue 22](#_Toc405799171)

[8.2 Merge lead 22](#_Toc405799172)

[8.2.1 Rules 23](#_Toc405799173)

[8.3 Merge persons 23](#_Toc405799174)

[9 Meglerfront pages 23](#_Toc405799175)

[9.1 Start page (“Øversikt”) 24](#_Toc405799176)

[9.2 Leads page (“Kundestrøm”) 24](#_Toc405799177)

[9.3 Real estate page (“Eiendom”) 24](#_Toc405799178)

[9.4 Person page 24](#_Toc405799179)

[9.4.1 Customer log tab (“Kundelogg”) 24](#_Toc405799180)

[9.4.2 Interest lists (“Interessentlister”) 25](#_Toc405799181)

[9.4.3 Accommodation (“Bolig”) 25](#_Toc405799182)

[9.4.4 Dream match (“Ønskat bolig”) 25](#_Toc405799183)

[9.5 Accommodation page 25](#_Toc405799184)

[9.5.1 All (“Alle”) 25](#_Toc405799185)

[9.5.2 Not contacted (“Ikke kontaktet”) 25](#_Toc405799186)

[9.5.3 Interested (“Interesserte”) 25](#_Toc405799187)

[9.5.4 Bidders (“Budgivere”) 25](#_Toc405799188)

[9.5.5 Not interested (“Ikke interesserte”) 26](#_Toc405799189)

[9.5.6 Match 26](#_Toc405799190)

[9.5.7 New construction (“Enheter”) 26](#_Toc405799191)

[9.5.8 Bidding (“Budgivning”) 26](#_Toc405799192)

[9.5.9 Exhibitions (“Visninger”) 26](#_Toc405799193)

[9.5.10 Statistics (“Statistikk”) 26](#_Toc405799194)

[10 Meglerfront lists 26](#_Toc405799195)

[10.1 Differentials in lists 26](#_Toc405799196)

[10.2 List items 27](#_Toc405799197)

[10.2.1 Differentials in list items 27](#_Toc405799198)

[11 Calendar integration 27](#_Toc405799199)

[11.1 Calendar uses 27](#_Toc405799200)

[11.2 The technical details 28](#_Toc405799201)

[11.3 Known issues 29](#_Toc405799202)

[11.3.1 Changing/deleting an appointment 29](#_Toc405799203)

[12 Exhange 29](#_Toc405799204)

[13 Logging 30](#_Toc405799205)

[13.1 Change log 30](#_Toc405799206)

[13.2 Trace log 30](#_Toc405799207)

[14 Statistics 31](#_Toc405799208)

[14.1 Statistics on Accommodation page 32](#_Toc405799209)

[15 Common support question 32](#_Toc405799210)

[15.1 Updating new construction 32](#_Toc405799211)

[15.2 Duplicate EmProfId 32](#_Toc405799212)

[16 Meglerfront flowchart 33](#_Toc405799213)

[17 Words and synonyms 34](#_Toc405799214)

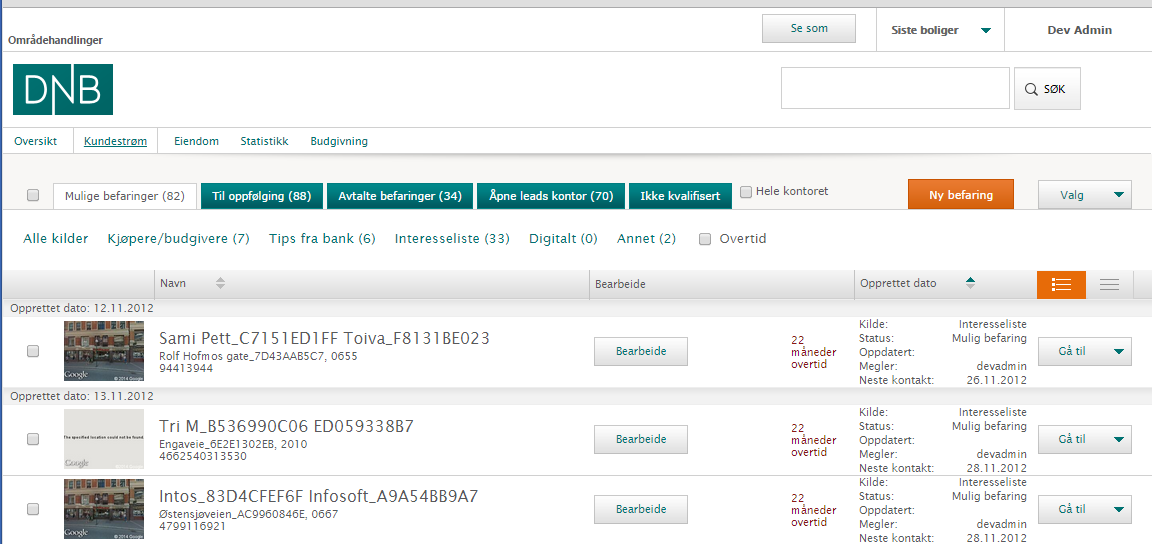
[17.1 Lead 34](#_Toc405799215)

[17.2 Accommodation 34](#_Toc405799216)

# Background

Meglerfront is an intranet developed in SharePoint 2010 for DNB Eiendom. Its main task is keeping track of leads and to make it easy for brokers to process leads in several steps. A lead is simply a seller or buyer of a property or real estate. The steps include possible leads, leads to contact, booked meetings, open leads (leads that are “up for grabs”) and not fully processed leads (or unqualified leads). In the last step users can find leads that the system believes are in need for contact.

These leads are collected in many ways and the remainder of this documentation will attempt to explain all of them in technical detail.

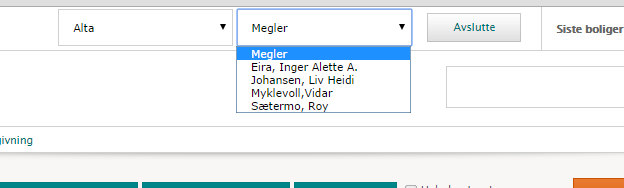


# Meglerfront access

Meglerfront servers exist in an enclosed environment and access is granted only directly from DNB Eiendom intranet or via secure VPN connections. The Meglerfront site uses windows authentication for its users. Meglerfront has a rule of thumb to not let other systems initiate data requests; instead Meglerfront is the one initiating the cross domain web requests. However exceptions have been made and in time of writing Meglerweb (see 3.2 for more information) and the internal exhibition IOS application (see 3.1) can send data to Meglerfront.

## The role of Daily Leader

There are some functions in Meglerfront only visible for a certain type of users called “Daily Leader”. These functions include “Se som” where the Daily Leader can view the site as another broker and see its data, however any log that is generated in this state will refer to the logged in user making the user that made the changes responsible for them.



A user is primary marked as Daily Leader in Active Directory and there is usually one Daily Leader for every DNB Eiendom office. Users can also be marked as a Daily Leader through the SharePoint group called “Daglige Ledere” and in such case should also be linked to a specific office in the SharePoint list called Meglerfront Daglige Lederes kontor.

# External data sources and dependencies

## Exhibition IOS application

An iOS app developed in Sencha. It’s brought to exhibitions by the broker and is used to register visitors. After the exhibition, a sync is manually initiated by the broker through the app. All communication is made through a separate website hosted on the same server as Meglerfront. The service is called MFJsonService, and as the name applies, it’s a restful service that manages requests and responses with Json. It serves the app with information about persons, interests and accommodations. Authentication is made with windows auth. A virtual directory on the website contains a ping-file, which is used by the app to check connectivity.

Sometimes when visitors are sent from the app to Meglerfront it encounters some problems causing the request to crash. This will also often result in an invalid state in the app where the visitors are lost. Luckily a SMS is always sent to the visitors upon registering. This way we can check the SMS log for the SMS-service to identify the visitors.

## Meglerweb

A direct translation of the word Meglerweb is Broker web.

Except from Exhibition IOS application, Meglerweb is the only external application allowed to initiate requests to Meglerfront. All requests are received by the WCF-service MeglerwebService.svc. The currently used methods are:

* UpdateOwnershipStatus – Called every time an ownership is updated in EmProf. This is managed by the Queue-job.
* UpdateExternalId – When an accommodation is transferred to EmProf we will get a temporary EmProfId. This method is used to update the EmProfId when a permanent EmProfId is created.

Meglerweb acts as a bridge between EmProf and Meglerfront since EmProf doesn’t initiate any requests to Meglerfront itself. Instead, all requests are made by Meglerweb. Meglerweb also hosts a service which is used by Meglerfront to fetch data about exhibitions. Meglerweb is hosted locally on DNB Eiendoms’ environment.

## EmProf

EmProf acts as master regarding data for persons, interests, accommodations and ownerships. In the extent that is possible, Meglerfront will always try to keep data synchronized between the two of them. Therefore Meglerfront will always synchronize person data when a user enters a person-page. Also, when a person, accommodation or ownership is updated in Meglerfront we will send the updates to EmProf.

EmProf is hosted locally on DNB Eiendoms environment.

## Posten.no

Meglerfront uses Posten.no for getting cities for a specific zip code. This is managed by the GUI when a user enters a zip code. Meglerfront will then make an asynchronous call to Posten and fetch the city.

## DreamService

A web service hosted by DNB Eiendom External Web. It serves Meglerfront with data regarding dreamsubscription, also called Boligkjöperegister, Drömmebolig and in some regards also “Matchning”. DNB Eiendom External Web is considered the owner of the data, however Meglerfront have full access to edit, delete or add data regarding dreamsubscription. It’s added in Meglerfront as a service reference in BusinessLogic.

## MeglerfrontBackendService

A web service hosted on the same server as DNB Eiendom External Web, but on a different website than the web. It serves Meglerfront with data for Autoprospect, Assessment (Verdivurderingar) and certain object data. DNB Eiendom external web has given a DLL to Meglerfront which serves as a contract and contains the models used.

## Bank – Tipsflyt alt. MSCRM/Dynamics

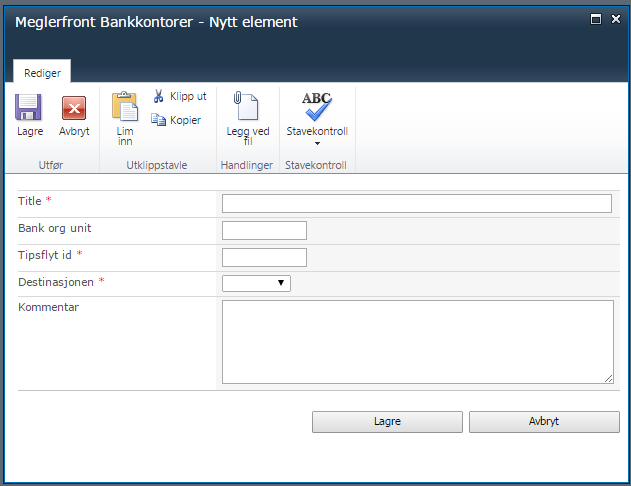
### Sending leads

One of the most important features in Meglerfront is the possibility to send a lead to the bank (DNB) for economical advice, also known as “Økonomisk rådgivning” (see chapter 6.5). When Meglerfront was built DNB didn’t have a centralized system for managing leads, and therefore they used a third party software called Tipsflyt (called TF3 in code). Tipsflyt has endpoints that provide Meglerfront with both bank offices and employees.

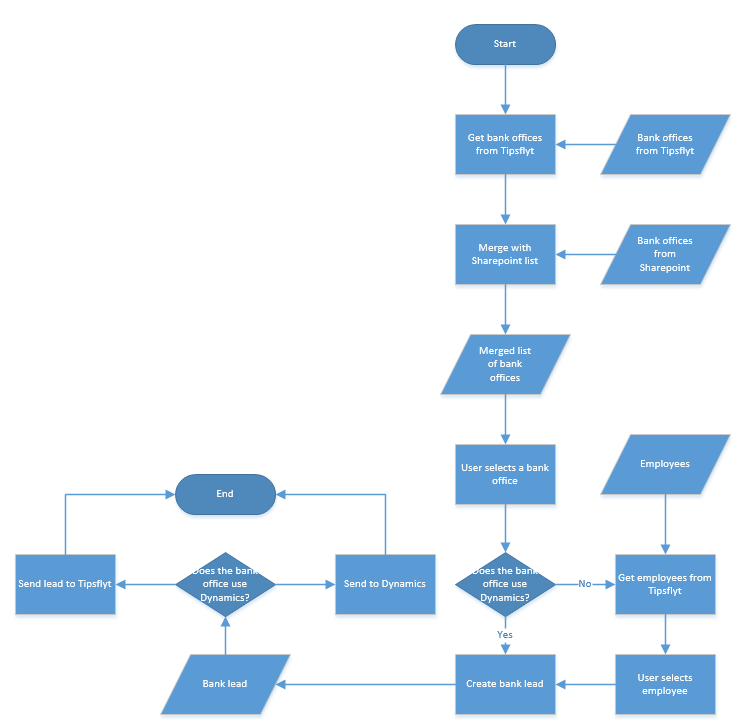
Currently DNB is implementing Microsoft Dynamics (also called Konsernfelles CRM or MS CRM), and has created an endpoint to which Meglerfront can send leads, but Meglerfront cannot fetch bank offices from Microsoft Dynamics. During the roll out of Microsoft Dynamics Meglerfront will be responsible for determine if a bank office is using Tipsflyt or Microsoft Dynamics. A SharePoint list has been created for this purpose. The list can be found at:

[host]/Lists/BankOfficeList/AllItems.aspx

Brukerstötte will be updating this list with new offices when a bank office has been migrated to Microsoft Dynamics.

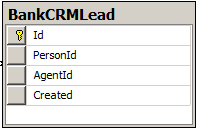


As seen by the screen dump above the administrator (technician or brukerstötte) can create new bank offices. When a new bank office that is created this way it will automatically overwrite the information Meglerfront receives from Tipsflyt. The goal is to remove all dependencies to Tipsflyt in the future.



### Receiving lead status

When a lead has been sent to the bank Meglerfront can also check the current status for the lead, eg if the bank employee has booked a meeting or if they signed a contract. This is managed a little differently depending on where the lead was sent (Tipsflyt or Microsoft Dynamics). When communicating with Microsoft Dynamics Meglerfront will be responsible for creating and keeping track of the lead’s id, which is called actionid. For this purpose Meglerfront will save all actionids in a table called BankCRMLeads.



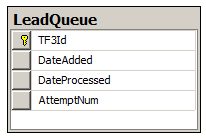
As seen in the image above each sent lead is saved with its actionid, personid, agentid and the creation date. However, please note that agentid isn’t a foreign key. That’s because agents can quit, but we want the information about the lead to persist.

When communicating with Tipsflyt Meglerfront will use its own personid. Then Tipsflyt is responsible for keeping track of which leads has been sent based upon the personid. Therefore Meglerfront doesn’t save any information about the leads sent to Tipsflyt. The only way to keep track of this is by reading the log.

### Receiving leads

The bank can send leads to Meglerfront. This is often people wanting the sell their accommodations. At the time of writing this is only implemented through Tipsflyt, however, in the future this will be processed directly with Microsoft Dynamics.

This is managed by a scheduled job called BankTipsTimerJob. It fetches all new bankleads from Tipsflyt and saves them in a table called LeadQueue. And then process them one by one.



The purpose of the table is keeping track of all imported (and not imported) leads from the bank. This way we can see which id’s that Meglerfront wasn’t able to import.

# Code structure

## Background

Meglerfront follows a layered structure where each layer is abstracted from all others. Simplified, the layers are: frontend, service, business and data. Frontend only calls the service layer, service only calls business layer etc. In addition there is the common project - where utilities and other shared code are stored. There is also a couple of other projects which handles external data sources and timer jobs, these acts as the service layer and only calls function in the business layer.

## Layers

### Frontend

The frontend layer in the solution includes two projects, Portal and Functionality. Portal contains SharePoint specifics such as Content Types and List Definitions. It also contains the CSS and all the frontend logic (javascript) - which is a heavy part of the project. Functionality includes all HTML whether it’s WebParts or UserControls (ControlTemplates). This is also were you find the folder Services which acts as the service layer and takes requests from the client code in Portal via AJAX and passes it on to the business layer.

### Business logic

This is a single project call BusinessLogic and contains models, mappers, managers and some business logic specific utilities.

#### Model classes

None of these include any logic. Most of these model classes acts as data contracts and are mapped from the data layers and sent via the functionality services to the front end logic.

#### Mapper classes

These are all responsible for correctly mapping models and lists of models from other layers (data layer and external sources).

#### Manager classes

This is where all the logic exists and it is basically the brain of the server side code. There is, roughly, one manager for each of the major model classes such as Accommodation model and Person model. There is also a manager for any other major functionality that is possible to categorize such as Dream match and EmProf Contact.

#### Utility classes

These classes are often generic with the purpose to minimize the code for managers. Only code that is specific for Business logic is to be placed here, code with more usability is to be placed in the Common project.

### Data Access

This layer’s purpose is to communicate with the databases - the Meglerfront content database and the Meglerfront SharePoint database. While most of the logic is in the business layer the data layer includes some where it is faster or makes more sense to communicate the logic to the database. The communications is done with LinqToSQL - and occasionally with stored procedures but the stored procedures are only used when queries need to be optimized and using database views, the site search functions are examples of this.

#### Repositories

Repositories for Meglerfront include SharePoint and SQL. The repository classes available in the Repository folder are responsible for creating and abstracting connections. Whether the connection is for a stored procedure, a database table or a SharePoint list, the repositories should always abstract them and make them available as simple function calls.

#### Access models

The access models contain representations of the SQL and SharePoint repository objects. These model classes are usually what are returned to the business layer - if the result is more complicated than a basic data type, which is where the mapper classes in the business layers comes in.

#### Services

The services folder includes the classes which uses the repository connections to the data sources. The functions in these classes are exclusively called from the business layer and ultimately use LinqToSQL-queries to get results from or to do other operations to the data sources.

## Shared

Code that is shared across projects in the solution is placed in the project named Common. This is where general utility functions are stored. These are, just as manager classes, one class per major model or function such as AccommodationUtils, PersonUtils.

The Common project is also where extensions, for data types, and all the enums are placed.

## Proxy projects

Meglerfront abstracts the communication with external sources to their own projects. For example all communication with the bank begins in BankManager in the business layer (This is where the logic of collecting data to send from Meglerfront as well as handling the response from the bank is made) and proceeds in project called BankCrm service, where all the mapping to their API is done. This way the business layer only needs to send the commonly used Lead model to the BankCrm service.

The following proxy projects are available:

* **Meglerfront.WCF.BankCrm.Contract** – Responsible for: creating the proxy object used by the business layer to access BankCrm.Service, all models that the BankCrm.Service needs and specification for the BankCrm.Service interface.
* **Meglerfront.WCF.BankCrm.Service** – Handles all calls to the bank.
* **Meglerfront.WCF.Meglerweb.Contract** – Same role as BankCrm.Contract
* **Meglerfront.WCF.Meglerweb.Service** – Handles all calls to Meglerweb
* **Meglerfront.WCF.MeglerwebIncoming.Contract** – Same role as other contracts.
* **Meglerfront.WCF.MeglerwebIncoming.Service** – Handles all possible calls from Meglerweb. Note that this isn’t used in the production environment. On production environment we still use *Meglerfront.WCF.Meglerfront.Service* for incoming Meglerweb calls. The reason for this is that we couldn’t get the authentication to work in time. It’s probably due to the lack of correct host-name.
* **Meglerfront.WCF.Contract** – Same role as other contracts.
* **Meglerfront.WCF.Meglerfront.Service** – Handles calls to EmProf and from Exhibition IOS Application. This is the first service project for Meglerfront which is why it handles multiple external parties. Calls to EmProf are made from MeglerfrontService.svc.cs and calls from Exhibition IOS Application are handled in MFJsonService.svc.cs.

## Timer project

The timer job project, called Timer, is responsible for executing the scheduled SharePoint Timer jobs. The project has two classes for every timer job.

The classes which names end with TimerJob inherits from the SharePoint class SPJobDefinition and overrides the Execute function called by the SharePoint framework. The overridden Execute function calls the other class to do the work which delegates this to the business layer.

# Timer jobs

Note that most logic in Meglerfront assumes that we have correct credentials for the logged on user. So when creating new timer jobs they need to be thoroughly tested, since they’re being executed in a different context. Calls to EmProf cannot be made within a timer job, since it’s not executed within the scope of an agent. Also, use the ChangeLogManager with care, since we always want to know who made a specific log.

## AD-Sync

Keeps offices and agents synchronized with DNB Eiendom Active Directory. It runs approx. every 30 minutes and fetches offices based upon a local XML and by their OU’s. It adds all offices that aren’t defined as excluded in the XML, such as “Administration”. For each OU it also adds all agents within that OU.

When all agents and all offices have been added it inactivates all agents that isn’t in the AD. Previously the job tried to remove agents that no longer existed in the AD. But since most agents have foreign keys in the Accommodation-table they cannot be removed. That’s why they are marked as inactivated instead. We almost never remove agents from the Agent-table, just because of the foreign keys.

## AssessmentImport

An assessment is a lead that wants to know the value of his/her accommodation. The job runs every 5 minutes and fetches new assessments (verdivurderingar) from DNB External Web (Meglerfront Backendservce). New assessments are saved as leads in Meglerfront. Meglerfront can receive assessments with or without an agent, but an office is needed. If the assessment doesn’t have an agent, Meglerfront will give the assessment to the Daily Leader or to a random agent within the office.

It keeps track of duplicates by writing imported assessments to the ChangeLog. Even though this works, it’s very slow since the ChangeLog is large. This timer job should be moved to the Queue Timer Job in the future. It should also be known that AssessmentImport crash fairly often. This is mostly because of faulty data. This can often be seen in the TraceLog as ”Error when importing assessment (ID: {0}) from external web (GetAndSaveAssessmentsByDate()). Exception follows”. The most common reason is that the office is a franchise office.

## AutoprospectImport / FinnImport

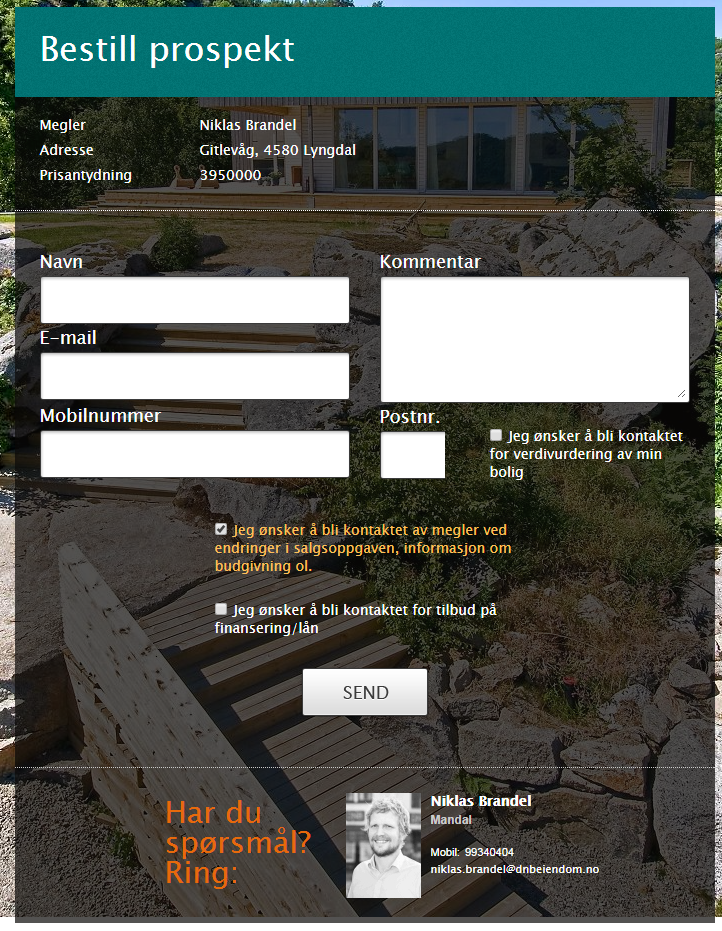
When the FinnImport-project was created it was mainly focused on the third party site Finn.no. However, the specs changed a lot during development and therefore some things are still called Finn, but it’s actually regarding Autoprospect.

It’s a job that runs every 5 minutes and makes a request to DNB Eiendom External Web and fetches the latest autoprospects. Then it saves them to The Queue, for later processing. This way we have a log for actually imported autoprospects and can easily check for duplicates.

An Autoprospect serves several purposes:

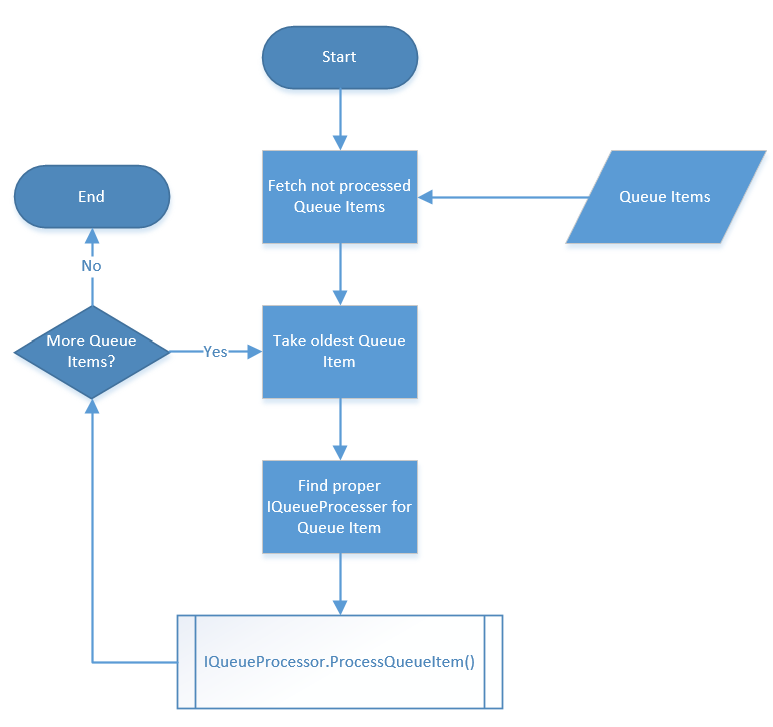
* Send a prospect to the user (done from External Web).
* Send an email to the broker telling him/her about the autoprospect
* Save user data to Meglerfront
* Create Boligkjöperegister (DreamMatch, Drömmebolig).
* Send to bank

Note that everything isn’t done all the time, it depends on what the user choses when he/she creates the autoprospect. See screen dump below.



## Queue job

A job that runs every 5 minutes and processes a Queue stored in the database. The Queue job itself doesn’t contain much logic; instead it identifies the actual IQueueProcessor responsible to process a Queue Item. It’s not threaded and each time the job runs it fetches all Queue Items that hasn’t failed more than 4 times.



Implementations so far:

* Autoprospect - Imports new Autoprospects.
* OwnershipManager - Sync with EmProf and adds new construction units.
* Residens - Not used at the moment.

It’s mainly a system for processing things in the SharePoint Timer Service instead of processing in the IIS-worker process. Preferably, it includes that which can be processed in the background but don’t need its own SharePoint Timer Job.

**Error handeling**

Once an autoprospect has been added from the external web you can check the database db.dnb.eiendom\_web\_live, and in the table avtAutoprospect. Search for the object that was added from the external web. Then copy the ItemId. Go to Meglerfronts database: Meglerfront\_Database and the table dbo.Queue and search for the ItemId. It should have QueueType 4, which is an enum for autoprospect. If it has the message: "Succeeded, new interest added, it should lie in the database table dbo.Accomondation. Search for it with the data you have from the object and then copy the ID. Then you go to meglerfront and type this url:

<http://meglerfront/Sider/accommodation.aspx?aid=5FAA6849-8701-4617-AACB-FE40CE924E84>

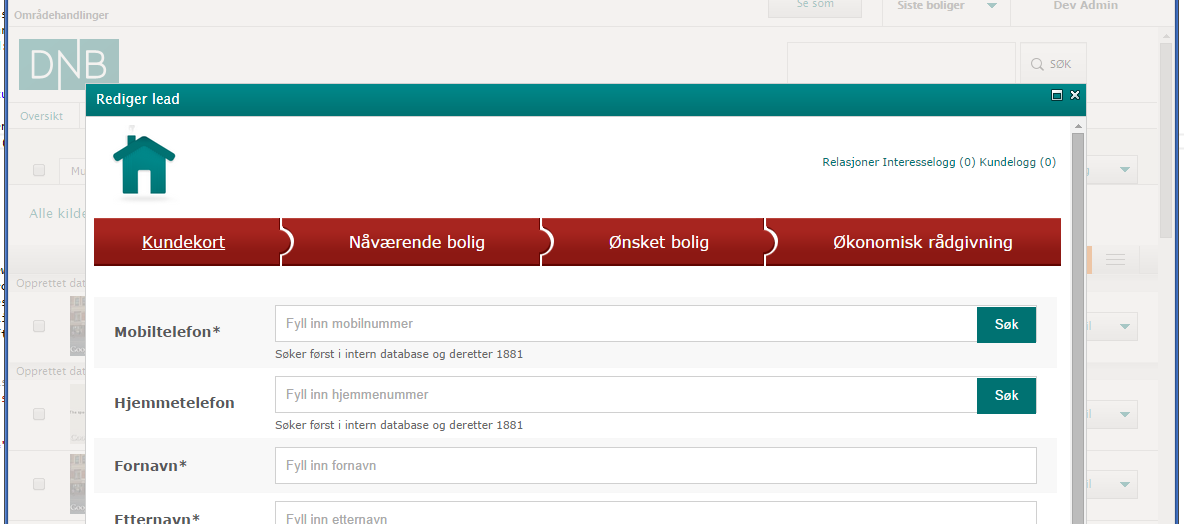
The yellow marked is the id from the database. If the autoprospect has been added to the accomondation database, everything is fine.

## Residens

Not in production at the time of writing. It imports interests for “residens” accommodations that are not yet publicly accessible. It’s a sort of a VIP-list, which gives certain users possibility to view an accommodation before others.

# Lead processing module

Called “Bearbeide” in Meglerfront and is one of the most important features in the system. The goal of it is to take a lead through several steps with the end goal of meeting the needs of a client.



## Step validation/qualification

The four different steps, detailed explanation later, is validated which is display by colors green (valid) and red (invalid). The validation is also reflected in the lists where the lead processing module is opened, this is a fast way for a user to see if a lead is fully processed. A fully processed lead is, in theory, a lead where the user has done what they can to close a deal. It is also possible to sort the lists on the level of qualification which makes it easy for the user to find the leads that are in need for attention.

## Customer card (“Kundekort”)

This is a basic form that describes a person in Meglerfront. By using the search functionality (“Sok”-buttons) in this form after entering the mobile or home number the system will firstly check the database for a matching person and secondly call on the API of Norway’s number service 1881.no. If a person is found the form displays that person and if that person came from Meglerfront database any change will prompt the user to confirm the change. Additionally if a user is already loaded in the form a search is made the same prompt is made to confirm change of an existing user.

### Qualify

The customer card is qualified just by saving it successfully.

## Current real estate (“Nåværende bolig”)

This step determines whether a lead owns real estate. It is also where appointments with leads are specified. What is entered here is determines how the lead appears in different lists for follow up and further process. Since a person can have several ownerships there is a button in this view to create a new ownership. All ownerships of a person is viewed in a list in the bottom of the same view.

### Status

Upon choosing status Booked (“Avtalt befaring”) the calendar will open (see Calendar Integration for more information) and the selected date and time will be saved on the lead. Setting the status to any other option than possible (“Mulig befaring”) will also change the field of ownership to yes.

### Responsible (“Ansvarlig”)

Every lead has a responsible agent (Meglerfront user) who will follow up the lead. This is assigned here. Connected to this field is a functional called “Foreslå”, Suggest in English, which will use the ownership zip code to find an office and assign the Daily Leader of that office to the field.

### Qualify

The Current real estate step is qualified when a conclusion of whether the lead owns property has been made by the user. In practice it means that saving the field “Eier du bolig?”, no matter in what state, is enough for the qualification.

## Dream match (“Ønskat bolig”)

Registering for a Dream match requires a mobile number or an email. This value will be saved in the database of the external web (see External data sources and dependencies > DreamService for more information). If one tries to change the email address of person that is registered with a dream match the user will be prompted to first remove the dream match. If this security measure wasn’t in place the Meglerfront GUI would tell the user that the person doesn’t have a dream match but the external web would still in other functions say otherwise. That issue has caused problems in the past where people complained about receiving unwanted real estate information.

### Qualify

One of the options yes (“Ja”) or no (“Nei”) need to be saved in order for this step to be qualified, saving “Not set” (“Ikke spurt”) will render this step unqualified.

## Economical advice (“Økonomisk rådgivning”)

This step will send information to a bank office (field “Bankfilial”). The option “Onsker å bli kontaktet av” decides whether the information is to be sent to a specific office which is only open on daytime or the client center (also called phone bank) which will contact the client after work hours.

Choosing a local office is in the user’s perspective very straight forward but not behind the scenes as there is an ongoing transition of where the information gets delivered. Before the transition started all office to receive the information used a system called Tipsflyt but in the time of writing some offices use MSCRM. For more information read paragraph [3.7](#_Bank_–_Tipsflyt).

### Qualify

Same as for dream match step: One of the options yes (“Ja”) or no (“Nei”) need to be saved in order for this step to be qualified, saving “Not set” (“Ikke spurt”) will render this step unqualified.

# Qualify processing module

Called “Kvalifisere” in Meglerfront and is very similar to Lead processing module but instead of processing leads through a number of steps, this is about processing interests. An interest is a potential buyer of real estate and is connected to specific accommodation, this means that if a person is interested in two different accommodations there will be two separate interest objects for this person.

This processing module contains the same steps as the Lead processing module with one addition, the interest step. Processing the other steps than the interest step in this module will have the same impact and result as doing so in the Lead processing module since the entered values still refer to the person and ownership objects.

## Interest step (“Interesse”)

In this tab the information about the interest is registered. First it is the grade of the interest, on a scale from 1 to 5, where 5 is very interested and 1 is moderately interested. After that, the status of the interest should be choosen. The only statuses that can be manually changed is “Ikke kontaktet” (“not contacted”), “Intressert” (“interested”) and “Ikke intressert” (“not interested”), where the other statuses is governed by the system (see paragraph below, [7.2](#_Status)).

There is also the possibility to set which source (“Kilde” in Norwegian) where the interest comes from. This is often set automatically when i.e. importing the interest from the external web or an exhibition. The broker can also add notes, remove the interest and go to active bidding (when bidding is active).

## Status

The interest step is mainly used for setting the status of an interest. Different status will affect the system in different ways as it is reflected in the tabs on the accommodation page (where the module is opened). There are status which isn’t available to choose for the user, bidder and buyer. These are instead governed and set by the Meglerweb system and can be synced in different ways.

## Bidding

The button “Til budgivning” (To bidding) will open a new module which displays the Meglerweb bidding system in an iFrame. Upon closing this iFrame Meglerfront will sync the current interest with the Meglerweb system and the status may be changed to bidder or buyer.

## Level of interest

In this step it is also possible to set a level of interest as a grade between 1 and 5 where 1 means low interest and 5 means a very high interest. This setting will be displayed in the list of interests and since it is possible to sort the list on this very property it makes it a powerful tool in finding the right interests to contact when there is a large set of them.

## Source

The Source (“Kilde”) option is available to keep track of where the interest came from.

This view also includes a button for deleting the interest. This will remove the interest from the system and accommodation lists but will not affect the person or any other related object.

### Qualify

Setting the interest status to any other than “not contacted” (“Ikke kontaktet”) will qualify this tab.

# Merge

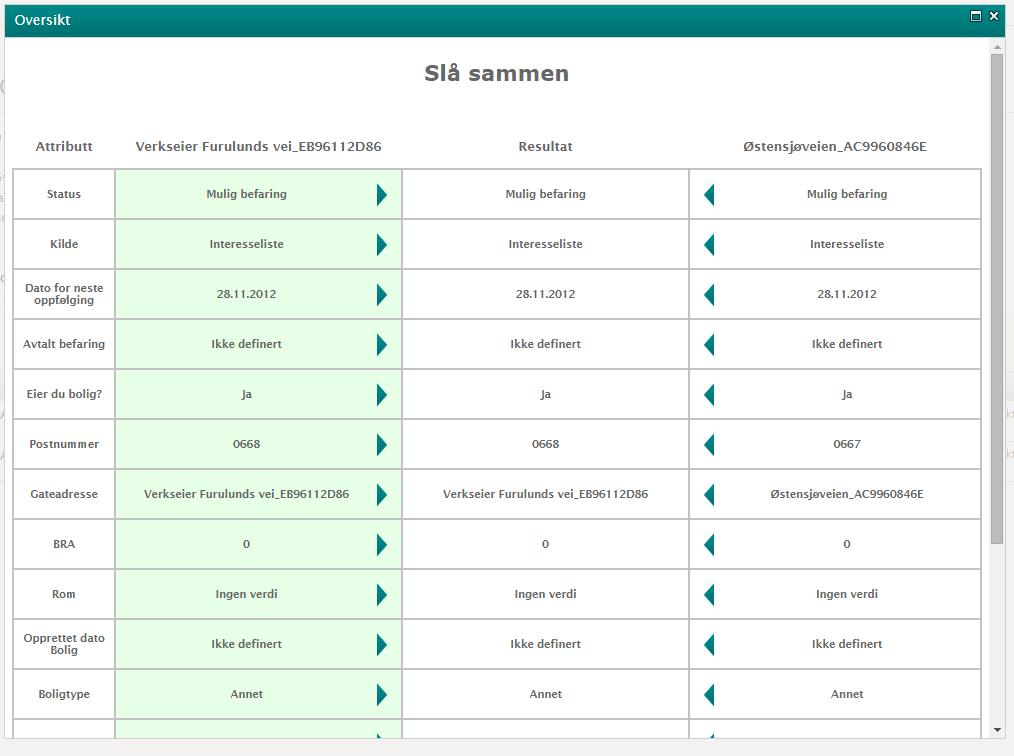
Issues related to persons and accommodations being added to the system more than once caused the need for a merge function where users of Meglerfront can merge lead objects. However as a security measure this isn’t always possible since, in certain scenarios, the probability of something being wrongly merged is too high.

## Duplicate issue

Since leads and persons have several sources of creation in Meglerfront it is not unusual that duplicates are created. Because an accommodation can be sold more than once automatic merges based solely on street addresses cannot be made. However, this is how EmProf treats accommodations and when sync is initiated between Meglerfront and EmProf for a duplicate object the very important EmProf ID will exists more than once I Meglerfront. When that happens, a merge is not allowed and careful manual change of this ID from a technician is required. The merge function is therefore only usable when a user identifies the duplication before the EmProf ID is obtained.

## Merge lead

Merging (“Slå sammen”) a lead is done from the tab “Bolig” (Living) on a person’s page. This is where lead objects or ownerships of a person are listed. Merging is done by checking the checkbox of two leads in this lists and after that the merge button. This will bring up the merge module (see image below) where everything in the left column in preselected. Clicking any field will cause that field to be selected and the opposite field to be displayed as unselected. Behind the scenes, the referenced lead in the left column will have all its values overwritten by the resulting field in the middle whilst the right column will be deleted upon saving.



### Rules

There are several rules for merging a lead which are in place to prevent users to manipulate objects in a way that could be unwanted for external parties. For example, if a lead has come to Meglerfront as a tip from a bank the bank should get credit for this tip and therefore it isn’t possible to overwrite this ID.

If both the ownerships are up for sale they cannot be merged and if one of them is up for sale, the for-sale status will automatically be in the resulting ownership. The same rule applies for the EmProf-ID property and, as mentioned, the bank ID (“fra bank” or “TF3Id” as called in the database) property.

## Merge persons

Although this feature is fully implemented it is not available to the users due to implications of - for example - the lack of group handling.

# Meglerfront pages

This section will give an overview of the different pages and their views in Meglerfront.

## Start page (“Øversikt”)

The start page contains two items. A big view of the calendar - for a quick overview of what the logged in user has on its schedule - and quick links - for fast access to tabs in the leads page.

## Leads page (“Kundestrøm”)

This page contains 5 tabs, 2 of which correspond to the most important lead statuses: Possible leads (“Mulige befaringer”) and Booked leads (“Avtalte befaringer”). The followup (“Til uppfolging”) tab contains the possible leads where the next contact date on a lead exists. The open leads (Åpne leads kontor) tab is a collection of leads relevant to the office where the user is placed, the leads displayed here are possible leads that have not been processed for some time.

The last tab, Not qualified (“Ikke kvalifisert”), is a search function where the user can, based on a range of zip codes, search for leads with the sold status (must have been sold at least one workday ago) and that have not yet been qualified. The purpose of this is to make sure most of the leads been given the questions of ownership end economical advice.

## Real estate page (“Eiendom”)

This is the overview page of the real estates that is in a sale stage and is connected to the logged in user. This overview provides fast access to the accommodation page and the person page that are currently interesting as the first two tabs are Upcoming sales (“Kommende salg”) and ongoing sales (“Aktive salg”). The other tabs display lists of previously sold accommodation which are Sold this year (“Solgt dette år”) and All my sales (“Alle mine salg”).

This page also features an alphabetic filter which filters the accommodation on their street address.

## Person page

The person page displays an image the persons living location and a summary of most important information. To the right of this information are simple tasks: Lookup relationships to other persons (“Relasjoner”), edit person information (“Rediger kunde”), send email (“Send e-post”) and send text message (“Send SMS”).

Below the basic information are tabs with additional details.

### Customer log tab (“Kundelogg”)

This tab displays all logs connected to the person and includes a function to filter the logs accordingly related accommodation, Meglerfront user and log type. Sorting of logs is always done on descending date and time.

### Interest lists (“Interessentlister”)

Effectively displays a list of the accommodations where the person is an interest.

### Accommodation (“Bolig”)

Displays a list of ownerships for the person, this tab includes the merge function. Viewing this tab also enables the user to create new ownerships and associate them with the person.

### Dream match (“Ønskat bolig”)

In this tab, a user can setup a dream match for the person which will result in the person showing up in the dream match tab on Accommodation page when a search matches the persons dream match. Dream match includes a lot of parameters which are saved in the extern web database associated with the person’s email or phone number. This tab is effectively the same as in the processing modules (“Bearbeide” and “Kvalifisere”).

## Accommodation page

The accommodation page displays a google street view image and the street address of the accommodation. It shows a summation of the information about the accommodation, with i.e. accommodation type, bedrooms count, living area, owner, responsible broker, “Oppdragsnummer” (which is the ID in EmProf) and the status of the lead associated with the accommodation.

Below is the information about the tabs visible on the page.

### All (“Alle”)

Shows all types of interests in this accommodation.

### Not contacted (“Ikke kontaktet”)

Shows all interests that have not yet been contacted by the responsible broker.

### Interested (“Interesserte”)

Shows all interests that have been contacted and have the status interested.

### Bidders (“Budgivere”)

Shows all interests that have been bidding on the accommodation.

### Not interested (“Ikke interesserte”)

Shows all interests that have changed their interest and is no longer interested in the accommodation.

### Match

In this tab it is possible to search for persons that is interested in similar accommodations. There is more filter alternatives than just similar accommodations. It is possible to change the range of the price of accommodations that have interests that can be interested in the current accommodation. There is also the possibility to change the bedroom count, district, accommodation type and if it should filter and only show persons that have an interest in newly constructed accommodations.

### New construction (“Enheter”)

If the accommodation is a new construction project. The existing sub accommodations are listen in this tab.

### Bidding (“Budgivning”)

Shows the current bidding process with the current bids. There is also a link to the current bidding in the bidding system “OnBroker”.

### Exhibitions (“Visninger”)

Shows the current booked exhibitions.

### Statistics (“Statistikk”)

Shows statistics about the current accommodation.

# Meglerfront lists

All lists in Meglerfront follow the same table layout. With a few exceptions, all columns are sortable. Also good to know is that the metadata column has various sort properties. Some places sorts on last updated, some on created date and so on.

## Differentials in lists

Depending on what data the list shows it uses different list items. There is some difference between person list items and accommodation list items (see differences of list items in paragraph [10.2.1](#_Differentials_in_list)).

As stated above, almost all of the lists have the same layout. One of the examples that do not have the same layout as the others is the customer log. The information there is more condensed with only timestamp, message and responsible user. Here the filtering is difference than the others. For more information about the customer log, see paragraph [9.4.1](#_Customer_log_tab) and [12.1](#_Change_log)

## List items

List items in Meglerfront are different in different places but follow the same template. Basically all list item has an image, a title, a couple of subtitles, metadata and navigation to the list items corresponding person page or accommodation page.

### Differentials in list items

One of the differences between list items is that the Interest list items shows extra information. This information is how many steps are qualified in the Qualify (“Kvalifisere”) popup window. It is displayed as [count of qualified steps] / [total steps count], i.e. 2/5.

Another difference is that accommodation list items (i.e. the tab “Bolig” on the person page) shows the information about when the next contact should be at the latest. This information is not shown on person list items.

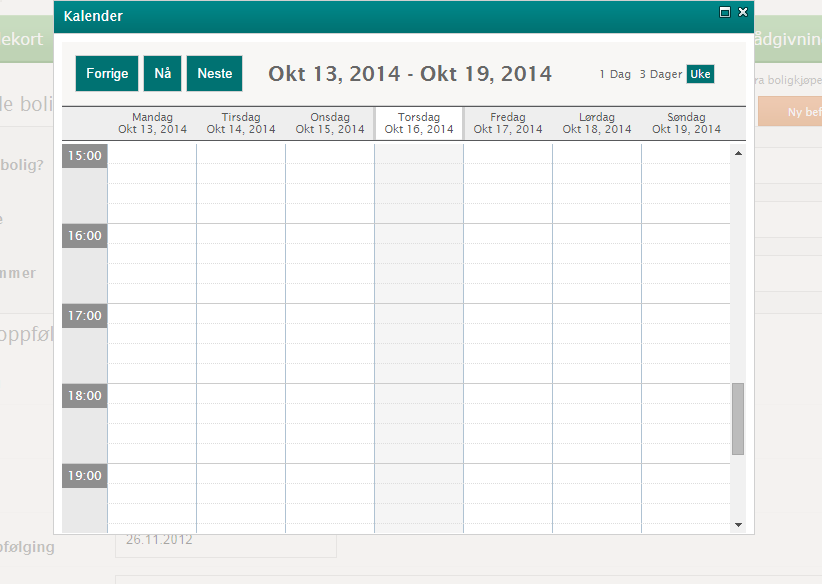
# Calendar integration

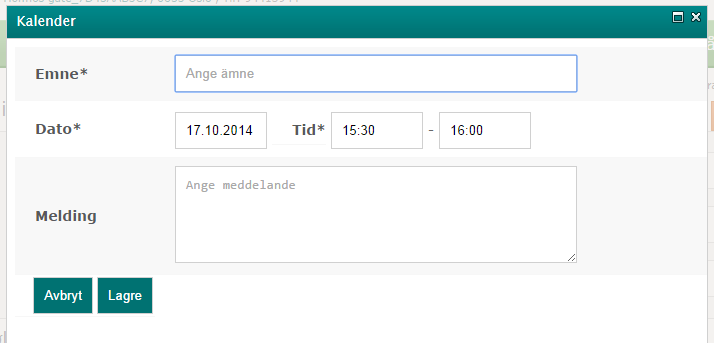
Meglerfront uses JQuery weekcalender for calendar visuals and frontend calendar logic and combines it with Microsoft Exchange Web Services to view, create and make changes to Exchange objects.

## Calendar uses

Meglerfront start page displays a large web part which basically includes the calendar. This gives the user a quick way of viewing, changing and creating the current week events such as meetings with clients.

The most important use of the calendar is when a user is processing a lead and is about to book a meeting. Upon choosing the booked status of a client, the calendar for the logged in user opens and the user can save an appointment. This appointment is saved on the lead’s database object with a reference to the exchange object created by the user.





## The technical details

A custom JavaScript is used to initialize and interact with the jQuery weekcalendar API. The function renderCalendar initializes the weekcalendar object and hooks functions to the events in the weekcalendar. These functions will in one way or another contact the backend with an Exchange ID. This ID is created by the Exchange Web Service and will reference a real calendar object in the users Exchange calendar. The ID is then saved to the database on the lead that was interacted with and thus permanently connecting the lead object to the actual Exchange calendar appointment.

## Known issues

### Changing/deleting an appointment

If a user changes a calendar object, either directly in exchange or on the Meglerfont start page, the lead’s database object is not updated resulting in a difference between the database appointment date and the appointment in the exchange environment. Therefore users should always interact with the appointments in the lead processing module (see paragraph 6 - Lead processing module for more information.

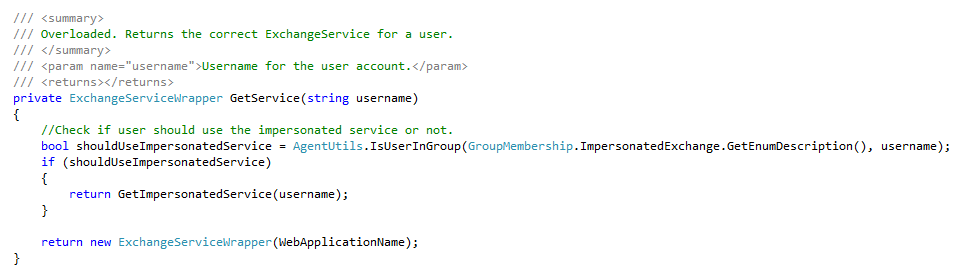
# Exhange

Meglerfront has an integration with two DNB Exchange servers. There are two ways of authenticating the user:

* Passing credentials of the logged on user.
* Using an impersonated account.

The reason why we use both is that DNB was migrating to a new environment during the fall of 2014. During that time Meglerfront had to be able to connect to two different Exchange servers while the migration was in process. The new server allowed impersonation, the old didn’t. Currently all users use impersonation and connect to the new server. But the code for passing the credentials of the logged on user is still in the project.

To determine which server to use Meglerfront has a group called “Exchange impersonifieringsgrupp”. If the user belonged in this group Meglerfront would use the new Exchange server (ERF01). If not, it would use the old Exchange server (Caax). All this is managed by ExchangeManager in BusinessLogic.



When accessing the new Exchange server Meglerfront need to logon with an account that has impersonating rights. For security reasons the password for accessing the new Exchange server has been salted and encrypted. Encryption key is stored in web.config. The encrypted value is stored in a propertybag. Salt is hardcoded. Avantime proposed a more secure solution with encryption by certificate. But due to lack of time the current solution was implemented instead.

Exchange is used for fetching/creating appointments, tasks and sending emails. URIs to the Exchangeservers are stored in connection strings. Note that they are also used on Meglerfront External and in Timer jobs.

# Logging

The logging in Meglerfront includes a lot of logic which is necessary in order to keep track of the events of the system. This is important as questions of what has happened often arise. There is two types of logs, change log and trace log. The change log saves changes on different objects. This do not have to be any kind of error log, most often it’s only to traces changes made (i.e. changed email address on a user), most of these logs are visible in Meglerfront. The trace log is used more as a debugging and error log, which is not shown any other place than in the database.

## Change log

The change log saves all changed made to persons, accommodations, ownerships, interests and bank tips. Which makes it easy to trace when something have been updated wrongly or correctly. Some SMS and Emails are also logged in the same log.

Change logs are saved in different ways depending on what type of object is logged (i.e. person, accommodation etc.). The logs have a property called “Scope” and a property called “Metadata”. With the help of these it is easy to decide which type of object have been logged. I.e. when updating the “Economical Advice” on a person, the log belongs to the scope “Person”, and the metadata “EconomicalAdvice”. The value will be the new choice of advice. The logs can also be saved as notes. These notes will be shown on different places on Meglerfront. Notes have the metadata “Note”, and can belong to different scopes (i.e. Person, Ownership, etc.)

The change log will be shown in the log tab on persons (“Kundelogg”). Notes on the other hand will be shown in Lead management (“Bearbeide”) on the corresponding tab. I.e. Notes about the person will be shown in the customer card (“Kundekort”), and notes about the customers current living will be shown in the current living tab (“Nåværende bolig”). Notes will also be shown in the persons log (“Kundelogg”), but with the prefix “Notat”.

## Trace log

The trace log saves three different types of severity of debugging and error logs, Info, Warning and Error.

Logs that are saved with the severity “Info” is just that, informational logging. One example of what is logged as Info is when Meglerfront are checking if an accommodation exists in EmProf. The information log also saves when timer jobs starts and ends and information on where in the process the job currently is.

Logs that are saved with the severity “Warning” logs errors that Meglerfront expect can happened and are handled as a specific error, with a specific error message. These errors are most often non-critical and handled in the code itself. Often when processing items in different timer jobs. These errors will often be tried to solve again when the timer job runs next time. I.e. Meglerfront logs errors when queue items cannot correctly be processed, or when there is an error when sending a notification to responsible broker when adding an auto prospect interest.

Logs that are saved with the severity “Error” log more critical errors which is not handled in the code. These logs are most often logged with a stack trace and the auto generated error message. For example service calls that fail when the service is offline or when objects that should not be null, is null.

Logs are saved with the Severity, message, possible stack trace and timestamp. When possible the logs are also saved with logged in user.

# Deploying

Meglerfront is deployed with Powershell. There are three scripts used:

* Add-settings – Used for inserting/updating configuration values such as propertybags and groups.
* Deploy-utility –A utility script for deploying WSPs to Sharepoint.
* Upgrade – The script actually used for deploying. It defines the WSPs to be deployed and used “Deploy-utility” to deploy them. After all WSPs are deploy it uses “add-settings” to update configurations.

Deploy on production environment is always made on Caas035n (app server). Note that the Powershell scripts only deploy the sharepoint projects. Proxies or external websites need to be deployed separately.

Deploy folder is located in C:\Deploy.

Usually a release to production environment is done on Wednesdays. A release window needs to be ordered in advance, and usually a release date is set about a month in advance.

Release notes for each deploy exists in each deploy folder. The purpose of the release notes is transparency, so that any developer can see what was done in the previous deploy.

VPN is needed for connecting to the production environment. VPN devices can be ordered by Fredrik Sveijer or Cecilie Våge.

# Statistics

There is two types of statistics in Meglerfront. One is statistics about current accommodation (when the broker is on a accommodation page), which is gathered from external sources. The other one, which is at the time of writing this documentation, does not exists in the production environment. But it will be statistics showing data about tips to bank.

## Statistics on Accommodation page

There is a statistics (“Statistikk”) tab to the right on the accommodation page. This statistics is fetched from the external web and Finn.no. This statistics shows the number of sent dream match mail (“Drömmeboligmail”) and the number of received autoprospect.

These statistics will only we shown if the accommodation have an EmProf ID and therefore can be linked to the real estate on the external web. If the EmProf ID is empty, the statistics tab will not contain anything.

# Common support question

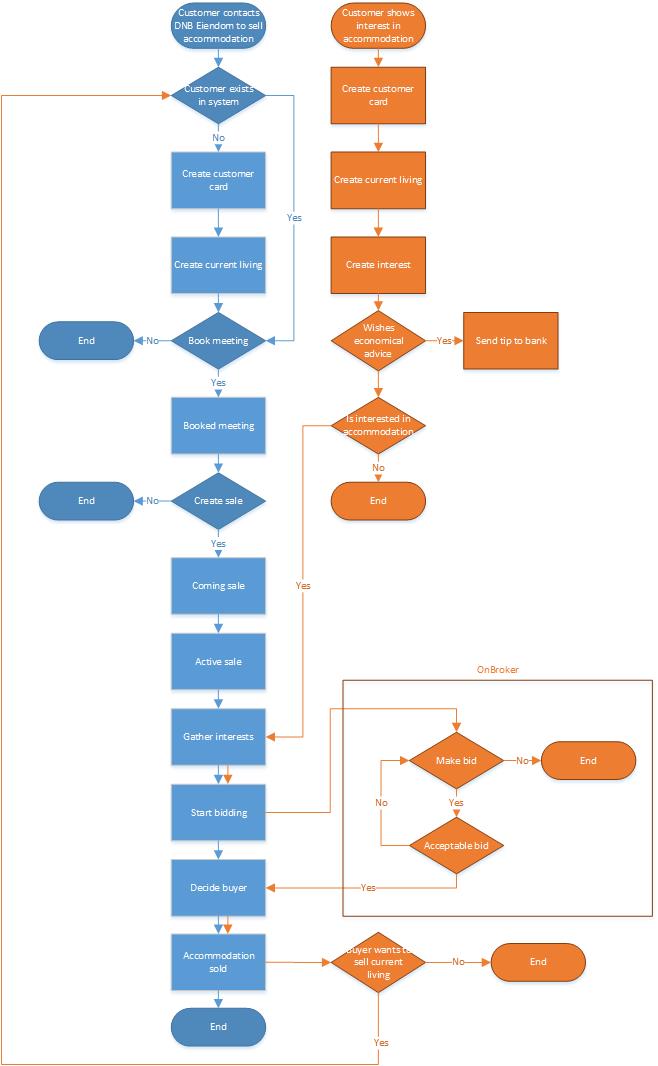
## Updating new construction

Since new construction is a fairly new feature in Meglerfront there has been some issues for the agents when they started using it. The main issue is that the accommodation often is created before new construction has implemented. Because of this, the accommodation often has the wrong accommodation type. This results a few extra errors. First of all, when we’re importing the sub-accommodations/units it will fail, since they cannot be connected to the parent accommodation (because of the wrong accommodation type). Secondly, it often results in the agent adding interests to the sub-accommodation instead of the project. Therefore Avantime if often asked to move the sub accommodations to the parent accommodation. When doing so, we also need to move the interests and make sure they have the correct status on the corresponding accommodation.

## Duplicate EmProfId

As mentioned in the chapter regarding Merging Meglerfront has an issue with duplicate IDs from EmProf. This is fairly common and is often solved by manually removing the EmProfId on one of the accommodations. Some caution is necessary when doing so. For example we should always check for interests on the accommodation. If the accommodation has any interests the interests will most likely also exist in EmProf. By removing the EmProfId you will then cause a “un-sync” between Meglerfront and EmProf regarding the specific accommodation. Mostly this is not a problem, however, the agent should always be informed about it.

# Meglerfront flowchart



# Words and synonyms

An issue for developers in this project is keeping track of what’s what when referring to different concepts within the project.

(DISCLAIMER: This is a developer’s perception of the definitions and the real ones may vary in some aspects).

## Lead

A lead is often referred to as:

* Befaring – The Norwegian word for lead in a certain stage. When a lead has real estate to sell, it is a “Befaring”
* Ownership – Lead is either called Lead or Ownership in code (Owns in database)
* Oppdrag – The Norwegian word for Assignment. It is often used in Helpdesk related matters when a user of Meglerfront reports a problem related to leads.

## Accommodation

Accommodation is often referred to as:

* Accomodation – A spelling error in the database.
* RealEstate – As the accommodation is a real estate (often called realEstate in the external web).
* Living – As the direct translation of the Norwegian word “Bolig” means living, it has been translated as such on some places.
* Bolig – The Norwegian word for accommodation and/or real estate.