Gardenzilla development notes

Peter Mezei

Tuesday 5th May, 2020

*G*ardenzilla development notes, questions and future development directions. The main goal of this document is to document the development steps and collect all the ideas for future evaluation.

Contents

1 Introduction

2 Modules

2.1 Customers

We manage customers as independent data objects, and manage a customer object through its implemented methods, or via public functions inside core::customer module.

Figure 1: Customer object

```
pub struct Customer {
  id: String,
  related_users: Vec<String>,
  name: String,
  tax_number: String,
  address: InvoiceAddress,
  phone: String,
  email: String,
  date_created: DateTime<Utc>,
  created_by: String,
}
```

Where:

id
Customer unique identifier

Describe and set the way we manage IDs.

related_users
vector of related user ID. User and Customer objects are independent of each other, as we can have customer without a user, and we can have - at least in theory - a user without a customer. Customer can purchase, User can login and interact withing the client. To manage Customer <-> User relations, we store them as a related ID list in each other. Here in a Customer object thats why we have this related_users vector.

name
name as string of the customer

tax_number
tax number as string

address
address as InvoiceAddress object

display InoviceAddress

phone
phone number as string

email
: email as string

date_created
: date as chrono::DateTime<Utc>

created_by
: userid as string, who created this object

describe datecreated and created by somewhere as we use them everywhere

2.1.1 Methods

new(. . .) -> Self
: creates a new instance of Customer. ID cannot be changed after creation

get_id() -> &str
: returns customer id as &str

get_name() -> &str
: returns customer name as &str

set_name(name: String)
: set customer name to the given new one

has_user() -> bool
: returns bool if there is at least one related user

get_users() -> &Vec<String>
: returns related user vector reference

remove_user(userid: &str)
: remove a user by its userid

should return bool, or result

get_tax_number() -> &str
: returns tax number as &str

set_tax_number(tax_number: String)
: set tax number

Todo list

*G*ardenzilla
Development Notes