Klad documentatie lagen

Laatst ge-update: 11/04/2023

Inhoud

[Extra Info 2](#_Toc132104803)

[Klasses 3](#_Toc132104804)

[HuidigeGebruiker 3](#_Toc132104805)

[ProjectInstellingen 4](#_Toc132104806)

[GebruikerInstellingen 5](#_Toc132104807)

[ProgrammaInstellingenManager 6](#_Toc132104808)

## Extra Info

* De Termen Manager en Service zijn verwisselbaar. In het effectieve programma en klasse diagram werd Manager als naamgeving gebruikt en in deze doc spreek ik over Service objecten. De reden dat er voor Manager werd gekozen is om verwarring te voorkomen in de toekomst (een REST API bevat meestal ‘service’ in de naam).
* Als een naam start met ‘\_’ is het een privé attribuut/property.

## Interfaces

### IGebruikerRepo

#### Attributen:

* /

#### Methodes:

* SetLogRegistratie (gebruiker: HuidigeGebruiker, appVersie: string): bool
  + Lege Constructor.

#### Relaties:

ProgrammaInstellingenManager: is een relatie tussen domein object (GebruikerInstellingen) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor aanpassen of het ophalen van GebruikerInstellingen.

#### Implementaties/Overerving:

Geen

## Klasses

### HuidigeGebruiker

Representeert de gebruiker die het programma momenteel gebruikt.

#### Attributen:

* ProdServer: string
  + Locatie van productieserver, het moet ergens opgeslagen worden zodat het correcte data kan ophalen van de db.
* StrServer: string
  + Server naam in string vorm, later gebruikt voor een connectiestring op te bouwen. (Productie, TestServer, etc.)
* StrDatabase: string
  + Database naam in string vorm, later gebruikt voor een connectiestring op te bouwen.
* ProjectVariablen: ProjectInstellingen
  + Klasse die algemene project instellingen bevat van het programma.
* GebruikerVariablen: GebruikerInstellingen
  + Klasse die gebruiker instellingen bevat van huidige gebruiker.
* PersoneelsId: int
  + Id van huidige gebruiker. (MpiNr en ArbovId worden niet opgeslagen).
* ProjectId: int
  + Id van huidig project, later gebruikt voor extra info zoals naamgeving in titelbalk.
* Gebruikersnaam: string
  + Samenvoeging van de gebruiker zijn namen (Voornaam.Achternaam).
* IsAdministrator: bool
  + Boolean die aanduidt of gebruiker een admin is.
* Toegangsrechten: List<Recht>
  + Lijst van rechten die weergeeft of een gebruiker toegang heeft tot het programma.

#### Methodes:

* HuidigeGebruiker(prodServer: string)
  + Constructor die productieserver inneemt, werd zo gedaan zodat als productieserver locatie verandert het programma niet moest hercompileren. Momenteel wordt het meegegeven via een App.config bestand.
* GetProductieServer(): string
  + Returned de property ProdServer.
* GetServerNaam(): string
  + Returned de property StrServer.

#### Relaties:

ProjectInstellingen: 1 op 1 relatie tussen HuidigeGebruiker en ProjectInstellingen, een gebruiker kan maar 1 instelling bevatten.

GebruikerInstellingen: 1 op 1 relatie tussen HuidigeGebruiker en GebruikerInstellingen, een gebruiker kan maar 1 instelling bevatten.

ProgrammaInstellingenManager: is een relatie tussen domein object (HuidigeGebruiker) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor het ophalen van ProjectInstellingen en GebruikerInstellingen.

GebruikerManager: is een relatie tussen domein object (HuidigeGebruiker) en Service object (GebruikerManager). Is verantwoordelijk voor het aanmaken van registraties met huidige gebruiker als object.

Recht: 1 op veel relatie tussen HuidigeGebruiker en Recht. Gebruiker kan meerdere rechten bevatten.

#### Implementaties/Overerving:

Geen

### ProjectInstellingen

Representeert de project instellingen die het programma bevat.

#### Attributen:

* ProjectNaam: string
  + Naam van huidige project, is gelinked met ProjectId in HuidigeGebruiker.
* ProjectDatabase: Db
  + Enum
* ProjectServers: List<string>
  + Lijst van servernamen die werden opgehaald (Productie en 4 test servers)

#### Methodes:

* ProjectInstellingen()
  + Lege Constructor.

#### Relaties:

ProgrammaInstellingenManager: is een relatie tussen domein object (ProjectInstellingen) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor aanpassen of het ophalen van ProjectInstellingen.

#### Implementaties/Overerving:

Geen

### GebruikerInstellingen

Representeert de project instellingen die het programma bevat.

#### Attributen:

* GebreInstId: int
  + /
* Emailadres: string
  + Emailadres van huidige gebruiker.
* InstNmVnm: bool
  + Instelling om te zien of ‘Naam – Voornaam’ of ‘Voornaam – Naam’ getoond moet worden.
* InstToner: bool
  + /
* Printer
  + /
* InstDefaultServerMode: int
  + Geeft aan welke server de huidige gebruiker heeft geselecteerd.

#### Methodes:

* GebruikerInstellingen ()
  + Lege Constructor.

#### Relaties:

ProgrammaInstellingenManager: is een relatie tussen domein object (GebruikerInstellingen) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor aanpassen of het ophalen van GebruikerInstellingen.

#### Implementaties/Overerving:

Geen

### ProgrammaInstellingenManager

Een service object die verantwoordelijk is voor het ophalen of wijzigen van ProjectInstellingen of GebruikerInstellingen.

#### Attributen:

* \_repo: IProgrammaInstellingenRepo
  + Interface die een data laag klasse implementeert. Interface zorgt ervoor dat de huidige Service object met de databank kan communiceren.

#### Methodes:

* ProgrammaInstellingenManager (repo: IProgrammaInstellingenRepo)
  + Constructor die ProgrammaInstellingenManager aanmaakt en een IProgrammaInstellingenRepo injecteert.

#### *Relaties*:

ProjectInstellingen: is een relatie tussen domein object (ProjectInstellingen) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor het ophalen van ProjectInstellingen.

GebruikerInstellingen: is een relatie tussen domein object (GebruikerInstellingen) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor het ophalen van GebruikerInstellingen.

HuidigeGebruiker: is een relatie tussen domein object (HuidigeGebruiker) en Service object (ProgrammaInstellingenManager). Is verantwoordelijk voor het ophalen van ProjectInstellingen en GebruikerInstellingen.

#### Implementaties/Overerving:

Geen

### GebruikerManager

Een service object die verantwoordelijk is voor het ophalen of wijzigen van ProjectInstellingen of GebruikerInstellingen.

#### Attributen:

* \_repo: IGebruikerRepo
  + Interface die een data laag klasse implementeert. Interface zorgt ervoor dat de huidige Service object met de databank kan communiceren.

#### Methodes:

* GebruikerManager (repo: IGebruikerRepo)
  + Constructor die GebruikerManager aanmaakt en een IGebruikerRepo injecteert.
* SetLogRegistratie(gebruiker: HuidigeGebruiker, string: appVersie)

#### *Relaties*:

HuidigeGebruiker: is een relatie tussen domein object (HuidigeGebruiker) en Service object (GebruikerManager). Bevat gebruiker info die weggeschreven moet worden naar de databank.

#### Implementaties/Overerving:

Geen