



 **BRONSCODE**

Ontologie Q-highschool

versie 2023

Inhoud

1	Introductie	3
2	Concepten	4
2.1	Primitieven	5
2.2	Constantes	6
2.3	Gebruiker	7
2.4	Vak	9
2.5	Module	10
2.6	Groep	11
2.7	Bijeenkomst	12
2.8	Planning	13
2.9	Planningrij	13
2.10	Inschrijving	14
2.11	Beoordeling	15
2.12	Activiteit	16
2.13	PTA	17
2.14	PTA regel	17
2.15	Parcours	18
3	Relatiediagram	19
4	Discussie	20
4.1	Vakken en Parcours	20
4.2	Groepen met één module	20
4.3	Constantes	21
4.4	Dag van een Groep	21
4.5	Overige inconsistenties	22

1 Introductie

Het woord “ontologie” verwijst naar een tak van de filosofie die zich bezighoudt met de studie van het zijn of bestaan. Het komt voort uit het Griekse woord “ontos”, wat “zijn” betekent, en “logos”, wat “studie” of “kennis” betekent. Ontologie onderzoekt fundamentele vragen over de aard van de werkelijkheid, het bestaan van entiteiten, de aard van relaties tussen entiteiten en de categorisatie van verschillende soorten zijn.

In de context van applicatieontwikkeling verwijst het woord “ontologie” naar een gestructureerde en formele beschrijving van de kennis over een bepaald domein. Het is een manier om de essentiële concepten, relaties en eigenschappen van een domein te modelleren en vast te leggen.

Het moment waarop de belevingswereld van de klant en de modelwereld van de applicatie uit elkaar drijven, is precies het moment waarop er pijnpunten in zowel de ontwikkeling als het gebruik van de applicatie ontstaan.

Voor applicatieontwikkeling is een ontologie dus van onschatbare waarde omdat het een gemeenschappelijke taal en een gedeeld begrip creëert tussen ontwikkelaars en gebruikers. Het stelt ontwikkelaars in staat om een applicatie te bouwen die nauwkeurig aansluit op de behoeften van het domein en de belevingswereld van de klant.

In dit document wordt een globale beschrijving van de huidige ontologie van de Q-highschool applicatie vastgelegd. Het bereik van dit document is beperkt tot een beschrijving van de concepten zoals daar op dit moment zowel in de echte wereld over wordt gedacht als hoe dit in de applicatie is gemodelleerd. De precieze interacties tussen de concepten en de interacties met andere applicaties worden bewaard voor een toekomstige uitbreiding.



2 Concepten

In dit hoofdstuk zullen één voor één alle concepten uit de applicatie worden benoemd. Elke paragraaf heeft als titel het concept zoals daar in de belevingswereld van de gebruikers over gesproken wordt, met in de kantlijn – in `monospace` font – de naam die dit concept heeft gekregen in de modelwereld van de applicatie. Wanneer dit monospace font ook elders in het document gebruikt wordt, is dit altijd om te refereren naar een term in de modelwereld.

2.1 Primitieven

Primitieven zijn simpele concepten die niet verder reiken dan één stukje informatie. Bijvoorbeeld een dag of datum. Deze primitieven zullen veelvuldig worden gebruikt bij het geven van de eigenschappen van een ingewikkelder concept. Hieronder een lijst primitieven met een korte beschrijving:

Datum : een kalenderdatum, dus dag, maand en jaar. Bijvoorbeeld 17 mei 2023.

Dag : Een dag in maandag t/m vrijdag, of “onbekend”.

Tijd : Een tijd op de klok, in 24 uren formaat. Bijvoorbeeld 15:00.

Periode : Elk jaar heeft vier periodes van zeven weken. Mogelijke waarden zijn “Blok 1” tot en met “Blok 4”.

Schoollocatie : Straatnaam en huisnummer van één van de locaties van een school.

Niveau : VWO, HAVO, MAVO, VMBO, OPUS, Jan Ligthart of Praktijkschool.

Schooljaar : Bijvoorbeeld 2022/2023.

QDate

Day

Time

Period

SchoolLocation

Level

SchoolYear

2.2 Constantes

De Q-highschool applicatie bevat een aantal constantes die herhaaldelijk worden gebruikt en elke zoveel tijd veranderen. De constantes zijn op dit moment:

Huidig schooljaar : Het huidige schooljaar.

Huidige periode : De periode waarvan de groepen op dit moment bezig zijn.

Inschrijfperiode : De periode waarvoor de inschrijving op dit moment bezig is.

Constants

`currentSchoolYear`

`currentPeriod`

`enrollmentPeriod`

2.3 Gebruiker

Een gebruiker ontstaat wanneer iemand voor het eerst inlogt in de app door middel van een Microsoft account. Hierbij wordt direct bepaalde informatie van het Microsoft account overgenomen.

Een gebruiker heeft de volgende eigenschappen:

Naam : Volledige naam zoals dit bij het Microsoft account staat.

Authenticatie email : E-mail adres zoals dit bij het Microsoft account staat.

Rol : De rol bepaalt wat een gebruiker wel en niet kan doen binnen de applicatie. De mogelijke waarden worden hieronder beschreven.

Beschikbare rollen : Indien er hier meerdere zijn, kan deze gebruiker van rol wisselen.

School : School van de gebruiker

Leerjaar : Leerjaar, tussen de 1 en 6.

Niveau : Zie niveau.

Voorkeusemail : E-mail adres waar de gebruiker e-mails op wilt ontvangen.

Profiel : Bijvoorbeeld "Natuur en Techniek".

Schoollocatie : Op welke locatie deze gebruiker voornamelijk verblijft.

Opmerkingen : Opmerkingen bij deze leerling.

Mobiel nummer : Telefoonnummer van de leerling.

Examenrechten : Mogelijke extra rechten voor bij een examen, zoals extra tijd.

Groepen : Een lijst referenties naar groepen waar deze gebruiker aan deelneemt. Voor een beheerder is dit altijd de hele lijst groepen.

Parcours : Een lijst met parcours die deze gebruiker volgt.

Inschrijvingen : Een lijst referenties naar groepen waarvoor deze gebruiker zich heeft ingeschreven.

User

```
displayName
email
role

availableRoles
school

year

level

preferedEmail

profile

schoolLocation

remarks

phoneNumber

examRights

participatingGroups

examSubjects

enrolled
```


Beoordelingen : Een lijst met beoordelingen die deze gebruiker heeft gekregen.

De mogelijke rollen zijn als volgt:

Leerling Volgt onderwijsaanbod van de Q-highschool.

Docent Geeft les aan groepen binnen de Q-highschool.

Contactpersoon Heeft toegang tot de gegevens van één school.

Beheerder Heeft toegang tot alle gegevens.

evaluations

student

teacher

grade_admin

admin

2.4 Vak

De vakken in Q-highschool zijn zowel de schoolvakken die leerlingen op hun middelbare schooldiploma kunnen zetten, als de overige categorieën van onderwijs die aangeboden worden, zoals de Avonturen en de Trajecten.

Een vak heeft de volgende eigenschappen:

Naam : De volledige naam van het vak, zoals "Informatica" of "Trajecten".

Afkorting : Een afkorting van de naam van het vak, zoals "wisD" voor "Wiskunde D".

Beschrijving : Een omschrijving van het vak.

Examenmogelijkheid : Een waarde die aangeeft of het vak een examenvak is, en het dus voor een diploma kan gelden.

Centraal examen : Een waarde die aangeeft of er een centraal examen is voor dit vak.

Subject

`name`

`abbreviation`

`description`

`canDoExam`

`hasCentralExam`



2.5 Module

Een module is een ondeelbaar stuk lesmateriaal binnen een bepaald vak over een bepaald onderwerp, waarvoor het mogelijk is om één beoordeling te geven.

Een module heeft de volgende eigenschappen:

Naam : De volledige naam van de module, zoals "Basis van Programmeren met Python".

Vak : Een referentie naar een vak.

Beschrijving : Een beschrijving van waar de module over gaat.

Opmerkingen : Opmerkingen over de module, zoals de voorkennis die nodig is om de module te volgen.

Studietijd : De totale hoeveelheid uren gereserveerd voor de module per blok.

Course

`name`

`subject`

`description`

`remarks`

`studyTime`

2.6 Groep

Een groep is een verzameling gebruikers die een bepaalde eigenschap gemeen hebben, bijvoorbeeld dat ze dezelfde module(s) volgen.

Een groep heeft de volgende eigenschappen:

Naam : De naam van de groep. Als een groep maar één module heeft, is dit altijd de naam van die module.

Periode : Het blok/de periode waarin de groep actief is.

Schooljaar : Het schooljaar waarin de groep actief is.

Modules : Een lijst van de module(s) die de gebruikers in de groep volgen.

Beschrijving Beschrijving van de groep: wat zullen leerlingen in deze groep gaan doen.

Opmerkingen Extra informatie die de leerling moet weten voor het inschrijven.

Doelgroep : Een korte beschrijving van de doelgroep waarvoor deze groep is bestemd.

Dag : Dag die als uitgangspunt wordt gebruikt bij het plannen van de bijeenkomsten.

Bijeenkomsten : Een geordende lijst bijeenkomsten.

Inschrijvingen : Een lijst inschrijvingen.

Deelnemers : Een lijst gebruikers die zijn toegelaten tot de groep. Deze lijst kan zowel leerlingen als docenten bevatten.

Beoordeling : Een lijst van beoordelingen van de leerlingen die de groep af hebben gerond.

Activiteit : De activiteit tabel van deze groep, die bijhoudt hoeveel inzet elke leerling heeft getoond.

Formulier : Eventuele link naar een extern formulier voor inschrijvingen met extra informatie.

Planning : Een eventuele referentie naar een planning voor het synchronizeren van de bijeenkomsten van deze groep.

Group

name
period
schoolYear
courses
description
remarks
enrollableFor
day
lessons
enrollments
participants
evaluations
presence
form
schedule

2.7 Bijeenkomst

Een bijeenkomst is een moment waarop de deelnemers van een groep bezig kunnen zijn met de leerstof van één of meerdere van de modules van die groep. Mogelijk vindt dit plaats op een bepaalde school in een bepaald lokaal met een leraar erbij, of op afstand met een leraar, of op afstand maar zelfstandig.

Een bijeenkomst heeft de volgende eigenschappen:

Datum Datum van de bijeenkomst.

Begintijd Begintijd van de bijeenkomst

Eindtijd Eindtijd van de bijeenkomst.

Locatie : Een Schoollocatie, of een andere aangepaste locatie zoals "Online".

Lokaal : Indien er een Schoollocatie is aangegeven, het lokaal op die school waar deze bijeenkomst plaatsvindt.

Onderwerp : Een korte beschrijving van de leerstof voor deze bijeenkomst.

Type : Of de bijeenkomst bijvoorbeeld fysiek of online plaatsvindt.

Aanwezigheid : "Aanbevolen" of "Optioneel".

Lesson

date
startTime
endTime
location
room
topic
kind
recommend

2.8 Planning

Een planning is een lijst data die gebruikt kan worden om snel de bijeenkomsten van een groep te vullen.

Een planning heeft de volgende eigenschappen:

Naam : Naam van de planning.

Concept : Als een planning een concept is, kunnen er nog geen groepen aan gekoppeld worden.

Bijeenkomsten : Een geordende lijst planningrijen.

2.9 Planningrij

Een planningrij heeft de volgende eigenschappen, waarvan de eerste vijf hetzelfde zijn als in een bijeenkomst:

Datum : Datum van de betreffende bijeenkomsten.

Starttijd : Starttijd van de betreffende bijeenkomsten.

Eindtijd : Eindtijd van de betreffende bijeenkomsten.

Type : Of de bijeenkomst bijvoorbeeld fysiek of online plaatsvindt.

Aanbevolen : "Aanbevolen" of "Optioneel".

Planning : Een referentie naar de planning waar deze rij toe behoort.

Schedule

```
name
isDraft
rows
```

ScheduleRow

```
date
startTime
endTime
kind
recommend
schedule
```


2.10 Inschrijving

Een inschrijving ontstaat wanneer een leerling zich inschrijft voor een groep. De inschrijving moet door een beheerder goedgekeurd worden voordat de gebruiker een deelnemer wordt.

Een inschrijving heeft de volgende eigenschappen:

Status : "Ingeschreven", "Wachlijst", "In behandeling", "Goedgekeurd" of "Afgekeurd".

Groep : Een referentie naar de groep waar deze inschrijving toe behoort.

Gebruiker : Een referentie naar de gebruiker die deze inschrijving heeft gemaakt.

Enrollment

status

group

user

2.11 Beoordeling

Soort : Categorie van de beoordeling. De mogelijke categorieën worden hieronder uitgewerkt.

Waarde : Waarde van de beoordeling. De mogelijke waarden worden hieronder uitgewerkt.

Toelichting : Uitleg van de docent.

Eindproducten : De hoeveelheid eindproducten die de leerling heeft ingeleverd.

Module : Een referentie naar de module waarvoor deze beoordeling geldt.

Leerling : Een referentie naar de gebruiker die deze beoordeling heeft gekregen.

Docent : Een referentie naar de gebruiker die deze beoordeling heeft ingevuld.

De mogelijke categorieën en bijbehorende waarden zijn als volgt:

Vink : Mogelijke waarden "Gehaald" (`passed`) of "Niet Gehaald" (`failed`).

Stapsgewijs : Mogelijke waarden "Goed" (`G`), "Voldoende" (`V`) of "Onvoldoende" (`O`).

Cijfer : Cijfer van 0 tot 10, afgerond op één cijfer na de komma. Dus bijvoorbeeld 9,3.

Bijzonder : Mogelijke waarden:

Niet deelgenomen (`ND`) : Leerling heeft niet actief deelgenomen aan de module of is (bewust) gestopt met de module.

Niet ingeleverd (`NI`) : Leerling heeft bij het invullen van de beoordelingen (nog) geen eindproduct of toets gemaakt/ingeleverd.

Plagiaat (`PG`) : Leerling heeft aantoonbaar plagiaat gepleegd (of het onderzoek daarnaar loopt nog).

Vrijstelling (`VRIJ`) : Leerling heeft een vervangbaar vak gedaan en heeft voor dit vak vrijstelling gekregen.

Evaluation

`type`

`assessment`

`explanation`

`submissions`

`course`

`user`

`updatedByUser`

`check`

`stepwise`

`decimal`

`special`



2.12 Activiteit

De activiteit tabel van een groep houdt bij welke leerlingen wel en niet actief zijn geweest bij elke bijeenkomst.

Een rij in deze tabel heet een Activiteit en heeft de volgende eigenschappen:

Status : "Actief" of "Niet Actief", zoals ingevuld door de docent.

Gebruiker status : "Actief" of "Niet Actief", zoals aangegeven door de leerling.

Toelichting Korte toelichting waarom de leerling niet actief was.

Bijeenkomst : Een referentie naar de betreffende bijeenkomst.

Gebruiker : Een referentie naar de betreffende leerling.

Presence

`status`

`userStatus`

`explanation`

`lesson`

`user`

2.13 PTA

De afkorting PTA staat voor “Programma Toetsing en Afsluiting” en geeft een overzicht van de modules die een leerling afgerond moet hebben om een vak op hun diploma te zetten. Evenals de wegingen van deze modules.

Een PTA heeft de volgende eigenschappen:

Naam : De naam van een PTA volgt een bepaald formaat, dat hieronder beschreven wordt.

Vak : Een referentie naar het vak waarvoor dit PTA bedoeld is.

Niveau : Niveau waarvoor dit PTA bedoeld is.

Start cohort : Een schooljaar, die het start cohort aangeeft.

Eind cohort : Een schooljaar, die het eind cohort.

Persoonlijk : Een waarde die aangeeft of dit PTA is aangepast voor een specifieke leerling.

Concept : Een concept PTA kan niet gekoppeld worden.

Regels : Een geordende lijst met PTA regels.

De naam van het PTA volgt het formaat “vak_niveau_start_eind” waarbij “start” en “eind” zijn afgekort naar twee getallen. Als een PTA een start cohort 2021/2022 en eind cohort 2023/2024 heeft wordt dit bijvoorbeeld afgekort naar “21” en ‘24”. In het geval van een persoonlijk PTA komt de naam van de leerling ook nog aan het eind: “vak_niveau_start_eind_leerling”.

2.14 PTA regel

Een PTA regel heeft de volgende eigenschappen:

Weging : De weging van het cijfer voor deze regel.

Modules : Een lijst met referenties naar modules die voor deze regel voldoen.

PTA

```
name
subject
level
cohortStart
cohortEnd
isPersonal
isDraft
rows
```

PTARow

```
weight
courses
```


2.15 Parcours

Een parcours is een opvolging van modules binnen een vak die een samenhangend geheel aan leerstof vormen, en gevolgd worden door een bepaalde leerling. Een parcours geeft zo de koppeling tussen een leerling en een vak, meestal in combinatie met een PTA. Een leerling "volgt" een parcours, en dit kan alleen voor vakken met een exammogelijkheid.

Een parcours heeft de volgende eigenschappen:

Vak : Een referentie naar het betreffende vak.

Examen : Een waarde die aangeeft of de leerling examen wilt doen in dit vak.

Deel : Of de leerling dit parcours "in profiedeel" of als vrije keuze "in vrije deel" volgt.

Niveau : zie niveau.

Startjaar : Het schooljaar waarin de leerling is begonnen met dit parcours.

Eindjaar : Het schooljaar waarin de leerling ofwel poogt klaar te zijn met dit parcours ofwel dit parcours heeft afgerond.

Status : "geschrap", "geslaagd", "gestopt" of "bezig".

PTA : Een referentie naar de gekoppelde PTA.

ExamSubject

subject

doExam

ptaPart

level

startSchoolYear

endSchoolYear

status

pta

4 Discussie

4.1 Vakken en Parcours

Er bestaat wat verwarring over de termen “vak” en “parcours”. Vaak wordt “parcours” gebruikt om te refereren naar een vak met examenmogelijkheid - zoals “Informatica” - zodat er zich drie categorieën vormen in vakken: Parcours, Avonturen en Trajecten. Nu worden Avonturen en Trajecten allebei gemodelleerd als zijnde een vak, maar dan zonder examenmogelijkheid. Desalniettemin leven ze dus op hetzelfde niveau in het systeem als alle andere vakken.

Dit zorgt er ook voor dat er een kleine omweg nodig is om de drie genoemde categorieën te onderscheiden. Op de aanbod pagina worden de Avonturen en Trajecten onderscheiden van de andere vakken door te kijken of die groep specifiek het vak “Avonturen” of “Trajecten” heeft.

Met de definitie van een parcours zoals die in 2.15 gegeven is, is het inderdaad logisch om Avonturen en Trajecten strict niet als parcours te laten fungeren. De modules in deze vakken staan namelijk vooral op zichzelf. Aan de andere kant kan een klein aantal modules binnen een vak nooit het hele betreffende vakgebied bevatten. Op die manier vormt een parcours dus altijd een deelverzameling van een vak.

4.2 Groepen met één module

Wanneer een groep slechts één module heeft, ligt het voor de hand om die groep dezelfde naam en beschrijving te geven als die van de module. Echter zijn er meerdere manieren om dit in praktijk te brengen, afhankelijk van hoe deze informatie gekoppeld wordt. Beschouw de volgende mogelijkheden:

1. **Module** → **Groep: Eenmalige kopie** Op het moment dat een groep wordt aangemaakt met maar één module, worden de naam en beschrijving van die module gekopieerd naar de groep. Hierna is de informatie niet langer gekoppeld.
2. **Module** → **Groep: Herhaalde kopie** Wanneer de beschrijving of naam van een module wordt aangepast, worden alle groepen die dit als enige module hebben ook aangepast. Hierdoor worden ook groepen uit het verleden veranderd, wat niet wenselijk is. Een groep die al afgerond is zou niet van naam moeten veranderen. Ook geeft het een situatie waarin veranderingen aan de kant van de groep overschreven worden wanneer de module verandert.
3. **Module** ↔ **Groep: Tweezijdse kopie** Het bovenstaande, maar veranderingen aan de naam of beschrijving van een van de

groepen wordt ook teruggekopiëerd naar de module, en vervolgens naar alle andere groepen. Dit betekent dat een verandering aan één groep een gigantische verandering vormt in de rest van de app.

4. **Module ⇒ Groep: Directe link** Het is ook mogelijk om de naam en beschrijving niet op te slaan in de groep, maar altijd de naam en beschrijving van de module op te halen bij het laden van de groep. Dit betekent dat bij het ophalen van de groep ook meteen de modules opgehaald moeten worden.
5. **Module ⇒ Groep: Directe link met override** Hetzelfde als de directe link, maar met de mogelijkheid om de beschrijving van de groep te laten afwijken, zodat de naam en beschrijving alleen opgehaald worden uit de module als die van de groep leeg zijn. Dit betekent een extra check, en ook dat de beheerder, wanneer die de beschrijving van een groep op deze manier wil laten afwijken, maar bijvoorbeeld met slechts één zin, alsnog met de hand de beschrijving van de module moet kopiëren naar het beschrijving veld van de groep.

Deze mogelijkheden lopen van simpel naar steeds ingewikkelder. Omdat alle mogelijkheden behalve 3.2.1 significante nadelen meebrengen raden we aan om te gaan voor de eerste optie.

4.3 Constantes

Op dit moment is er een losse constante voor de inschrijfperiode, maar niet voor het inschrijfjaar, dus als de huidige periode “Blok 4” is, maar de inschrijfperiode zou “Blok 1” van het volgende jaar moeten zijn, hebben we ook een soort “inschrijfjaar” constante nodig. Het huidige blok en huidig jaar worden verder gebruikt om te bepalen welke groepen er vooraan komen te staan op de pagina “Mijn Groepen”, terwijl de inschrijfperiode (en dan ook het inschrijfjaar) gebruikt wordt om te bepalen welke groepen er vooraan komen te staan op de Aanbod pagina, en ook welke groepen er open staan voor inschrijving.

4.4 Dag van een Groep

De dag eigenschap van een Groep heeft eigenlijk geen ontologische betekenis. Het is slechts een aanwijzing of hint naar de weekdag waarop de bijeenkomsten meestal plaats zouden vinden, maar de daadwerkelijke momenten van de bijeenkomsten worden bepaald door de (gekoppelde) planning. De dag eigenschap kan op “maandag” gezet worden terwijl alle bijeenkomsten op dinsdag plaatsvinden.

Wellicht zou deze eigenschap verplaatst moeten worden naar de planning. Of zou het een andere naam moeten krijgen, zoals “hoofddag”. Als derde mogelijkheid zou de dag afgeleid kunnen worden uit de momenten waarop de bijeenkomsten gepland zijn.



4.5 Overige inconsistenties

Naast bovenstaande discussiepunten zijn er nog een aantal inconsistenties. Waarschijnlijk zijn dit gewoon fouten of overblijfsels. De vraag is of ze op de voor de hand liggende manier verholopen kunnen worden.

Toelichtingen en opmerkingen De gebruikers en groepen hebben nog eigenschappen zoals “toelichting” of “opmerkingen” die in de app niet meer gebruikt lijken te worden.

Docent vs Expert Op één plek in de app, namelijk bij het interface voor het toevoegen van een deelnemer aan een groep, wordt een Docent een “Expert” genoemd. Dit lijkt volledig incorrect. Toch verschijnt deze term nog hier en daar in de communicatie.

TypeScript definitives

```
type Day =
  | "onbekend"
  | "maandag"
  | "dinsdag"
  | "woensdag"
  | "donderdag"
  | "vrijdag"
  | "zaterdag"
  | "zondag"
  | "niet van toepassing";

type SchoolYear =
  | "2016/2017"
  | "2017/2018"
  | "2018/2019"
  | "2019/2020"
  | "2020/2021"
  | "2021/2022"
  | "2022/2023"
  | "2023/2024"
  | "2024/2025";

type Level = "VWO" | "HAVO" | "VMBO";

interface Self extends BaseObject {
  role: string;
  user: Ref<User>;
  participatingGroups: List<Group>;
  enrollments: List<Group>;
}

interface User extends BaseObject {
  displayName: string;
  email: string;
  role: string;
  availableRoles: string;
  school: string;
  year: string;
```

```
  level: Level;
  preferredEmail: string; //fix this spelling soon!
  profile: string;
  schoolLocation: SchoolLocation;
  remarks: string;
  phoneNumber: string;
  examRights: string;
  participatingGroups: List<Ref<Group>>;
  examSubjects: List<ExamSubject>;
  enrolled: List<Ref<Group>>;
  evaluations: List<Evaluation>;
}

interface Subject extends BaseObject {
  name: string;
  abbreviation: string;
  description: text;
  canDoExam: boolean;
}

interface Course extends BaseObject {
  name: string;
  subject: Ref<Subject>;
  description: text;
  remarks: string;
  studyTime: string;
}

interface Group extends BaseObject {
  name: string;
  period: Period;
  schoolYear: SchoolYear;
  courses: List<Ref<Course>>;
  description: text;
  remarks: text;
  enrollableFor: string;
  day: Day;
```



```

lessons: OrderedList<Lesson>;
enrollments: List<Enrollment>;
participants: List<User>;
evaluations: List<Evaluation>;
presence: List<Presence>;
form: FormLink;
}

type LessonKind = string;

interface Lesson extends OrderedListMember {
    date: QDate;
    startTime: string;
    endTime: string;
    location: SchoolLocation;
    room: text;
    topic: text;
    kind: LessonKind;
    recommend: "required" | "optional";
}

interface LinkedLesson extends OrderedListMember {
    readonly date: QDate;
    readonly startTime: string;
    readonly endTime: string;
    location: SchoolLocation;
    room: text;
    topic: text;
    readonly kind: LessonKind;
    readonly recommend: "required" | "optional";
}

interface Schedule extends BaseObject {
    name: string;
    isDraft: boolean;
    rows: OrderedList<ScheduleRow>;
}

interface ScheduleRow extends OrderedListMember {
    date: QDate;

```

```

    startTime: Time;
    endTime: Time;
    kind: LessonKind;
    recommend: "required" | "optional";
    schedule: Ref<Schedule>;
}

interface PTA extends BaseObject {
    name: string;
    subject: Ref<Subject>;
    level: Level;
    cohortStart: SchoolYear;
    cohortEnd: SchoolYear;
    isPersonal: boolean;
    isDraft: boolean;
    rows: OrderedList<PTARow>;
}

interface PTARow extends OrderedListMember {
    weight: integer;
    courses: List<Ref<Course>>;
}

interface ExamSubject extends BaseObject {
    readonly subject: Ref<Subject>;
    doExam: boolean;
    ptaPart: "profile" | "free" | "-";
    level: Level;
    readonly startSchoolYear: SchoolYear;
    readonly endSchoolYear: SchoolYear;
    readonly status: "geschrapt" | "geslaagd" |
        "gestopt" | "bezig";
    readonly pta: Ref<PTA>;
}

interface ExamSubject extends BaseObject {
    readonly subject: Ref<Subject>;
    doExam: boolean;
    ptaPart: "profile" | "free" | "-";
    level: Level;

```



```

    startSchoolYear: SchoolYear;
    endSchoolYear: SchoolYear;
    status: "geschrappt" | "geslaagd" |
           "gestopt" | "bezig";
    pta: Ref<PTA>;
}

interface Evaluation extends BaseObject {
    type: "decimal" | "stepwise" | "check";
    assessment: string;
    explanation: text;
    submissions: integer;
    course: Ref<Course>;
    user: Ref<User>;
}

interface Enrollment extends BaseObject {
    status: "new" | "pending" | "rejected" |
           "waitlist" | "accepted";
}

interface Presence extends BaseObject {
    status: "present" | "absent";
}

```

```

    userStatus: "present" | "absent";
    explanation: text;
    // lesson: Ref<Lesson>; // doesnt work
    user: Ref<User>;
}

interface ConnectionLogEntry extends BaseObject {
    issuer: Ref<User> | "system";
    connectionName: string;
    stackTrace: string;
    args: string;
    response: string | null;
    status: "open" | "sent" |
           "error" | "success";
    createdAt: string;
}

interface Constants extends BaseObject {
    currentSchoolYear: SchoolYear;
    currentPeriod: Period;
    enrollmentPeriod: Period;
}

```




Kunstenaarsverklaring

"I can never make something as perfect as an iPhone, but Apple could never make something as flawed as what I do." (Tom Sachs)

Willekeur is een venijnig beestje, zeker in een wereld die wordt geregeerd door controle, waarin alles perfect gepresenteerd moet worden. Aan de ene kant lijkt een computer misschien willekeur te bezitten, veel meer dan een mens, maar toch is zijn werk al bij voorbaat vastgelegd. Een perfect middel voor haarfijne presentatie. Vraag een mens daarentegen naar een willekeurig getal onder de 100, en daar blijkt weinig willekeur in te zitten. Toch is de mens (als individu, maar vooral als groep) in staat de meest bizarre dingen te bedenken. Uiteindelijk is willekeur een uitermate menselijk iets, iets dat het leven interessant maakt en iets dat we delen. Het omarmen van willekeur is daarom, misschien, ook het omarmen van het menselijk leven.

Dit werk, een serie aan willekeurig gegenereerde afbeeldingen, zoekt het raakvlak op tussen de menselijke behoefte aan orde en de schijnbare willekeur van de computer. Het spiegelt hierin de inhoud van het document: ontologie, de zoektocht naar een waarheid in een o zo willekeurige wereld. Een zoektocht die op zichzelf kan inslaan en steeds dieper gaat. Eentje die misschien wat hobbelig verloopt.

Ik heb mijn waarheid (gelukkig?) nog niet gevonden. Op het moment houd ik mij bezig met beeldende kunst en crossmedia design. Ik onderzoek graag hoe de computer het creatieve proces kan dienen en inspireren. Vanaf september zal ik te vinden zijn bij de MA Experimental Publishing aan de Willem de Kooning Academie.

Thijs van Loenhout