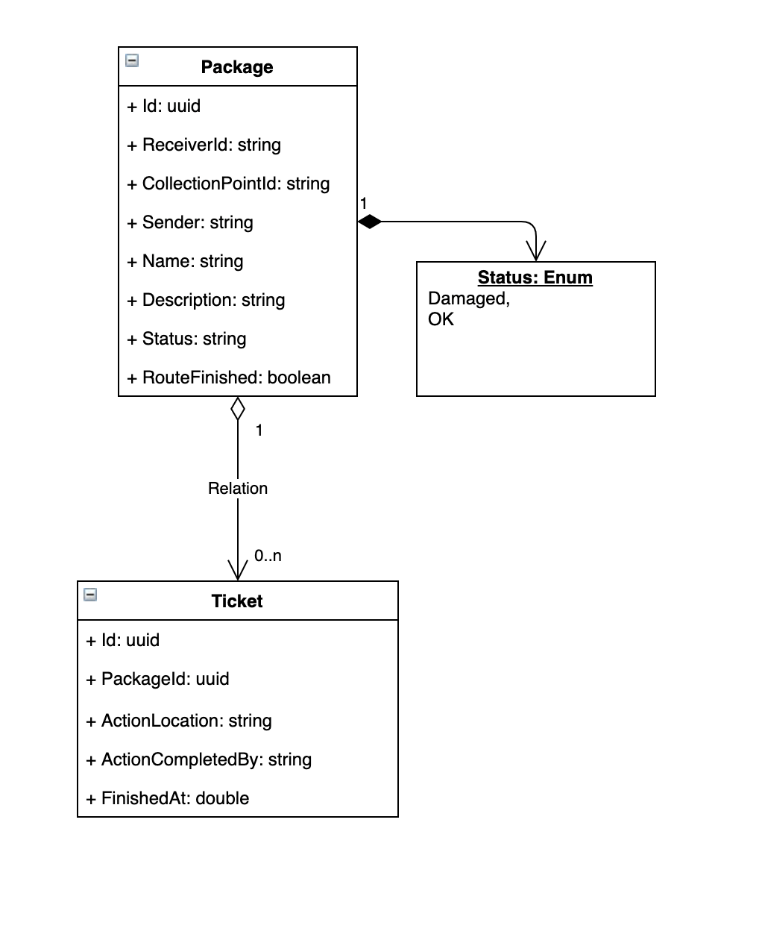
Code (C4)

**Pakket Service Model**



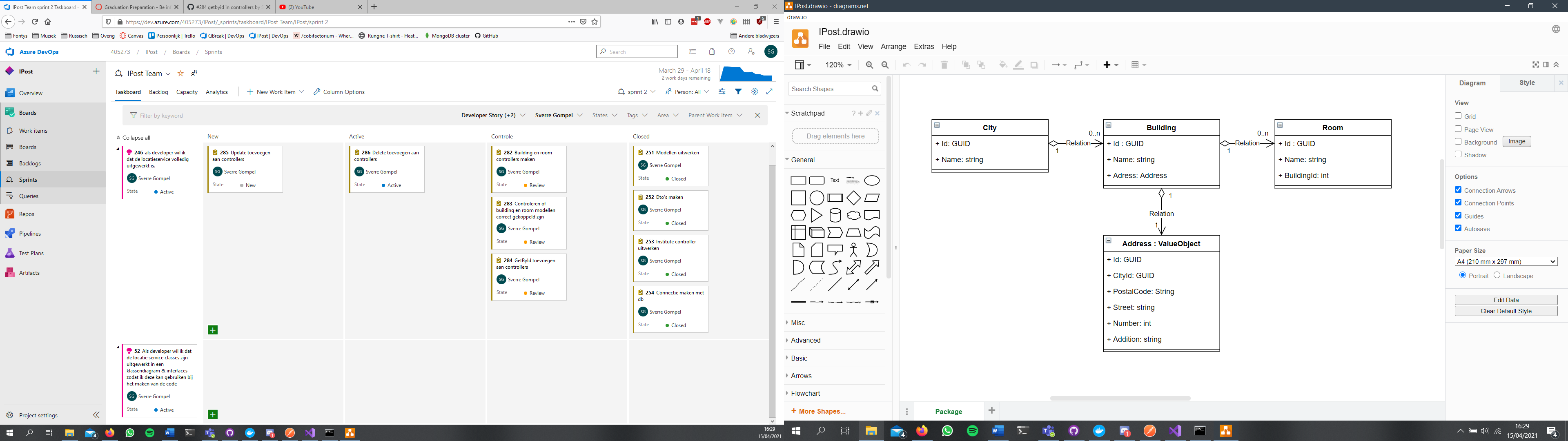
Pakket service bestaat uit 2 domein objecten: Package en Ticket. Package is een object met data over een pakket. Ticket is een moment opname van een pakket en heeft hierbij een Package.

Interfaces

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| GetPackageById | GET | /{id} |
| GetAllPackages | GET | / |
| PostPackage | POST | / |
| UpdatePackage | PUT | /{id} |
| DeletePackage | DELETE | /{id} |

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| AddTicketToPackage | POST | /{packageId} |

**Locatie Service Model**



Locatie service bestaat uit 3 domein objecten: City, Building, en Room. Een gebouw behoort tot een stad en kan meerdere ophaalpunten hebben. Een gebouw bevat een ValueObject: *Address.* Dit object slaat adresgegevens op in de database. Elke ruimte bevat een buildingId, waarmee het bijbehorende gebouw opgehaald wordt in de backend.

Interfaces

***City***

Endpoint: /city

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| CreateCity | POST | / |
| GetAllCities | GET | / |
| GetCityById | GET | /{id} |
| DeleteCityById | DELETE | /{id} |

**CreateCity**

Creëer een nieuwe stad.

**GetAllCities**

Geeft een collectie van steden terug.

**GetCityById**

Geeft een stad object terug dat bij het Id hoort.

**DeleteCityById**

Verwijder de stad die bij het Id hoort.

***Building***

Endpoint: /building

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| CreateBuilding | POST | / |
| GetAllBuildings | GET | / |
| GetBuildingById | GET | /{id} |
| DeleteBuildingById | DELETE | /{id} |

**CreateBuilding**

Creëer een nieuw gebouw.

**GetAllBuildings**

Geeft een collectie van gebouwen terug.

**GetBuildingById**

Geeft een gebouw object terug dat bij het Id hoort.

**DeleteBuildingById**

Verwijder het gebouw dat bij het Id hoort.

***Room***

Endpoint: /room

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| CreateRoom | POST | / |
| GetAllRooms | GET | / |
| GetRoomById | GET | /{id} |
| DeleteRoomById | DELETE | /{id} |

**CreateRoom**

Creëer een nieuw gebouw.

**GetAllRooms**

Geeft een collectie van gebouwen terug.

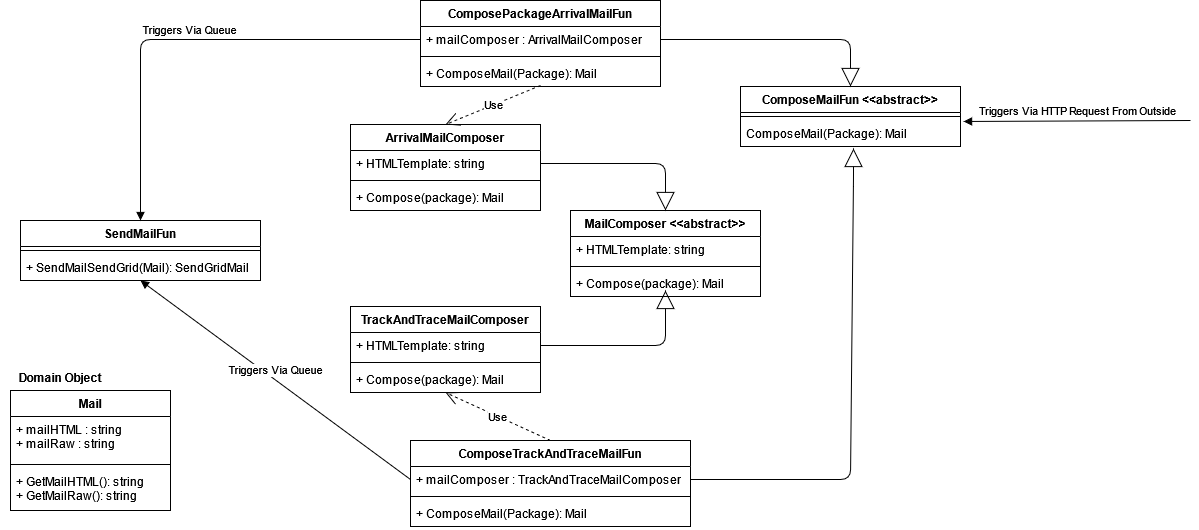
**GetRoomById**

Geeft een gebouw object terug dat bij het Id hoort.

**DeleteRoomById**

Verwijder het gebouw dat bij het Id hoort.

**Mail Service Class Diagram**



De mail service bestaat uit een eigen domeinobject, en maakt ook gebruik van de domeinobjecten van de andere services zoals het pakket om email te composeren. De mailservice werkt op basis van een serverless architectuur met azure functions en sendgrid. Het aanspreekpunt van buitenaf is ComposeMailFun. Deze wordt aangesproken met een HTTP Trigger via de API Gateway. Mails worden gecomposed in de MailComposers en vervolgens doorgestuurd als mailobjecten naar de SendMailFun via een queue. Deze handelt de logica van het versturen van de mail via SendGrid.

**Interfaces**

(sidenote: dit is niet een service wie REST conform responses terug hoeft te sturen naar de gebruiker)

|  |  |  |
| --- | --- | --- |
| **Naam** | **Methode** | **Endpoint** |
| ComposeArrivalMail | POST | /ArrivalMail |
| ComposeTrackAndTraceMail | POST | /TrackAndTraceMail |
| SendMail | Triggers Via Queue | N/A |

**ComposeArrivalMail**

Maakt een mail aan die wordt verstuurd wanneer een pakket op een afhaalpunt is aangekomen

**ComposeTrackAndTraceMail**

Maakt een mail aan wanneer een pakket is geregistreerd by Fontys met track en trace link

**SendMail**

Verstuurt mail naar een ontvanger via SendGrid.