

# Styles

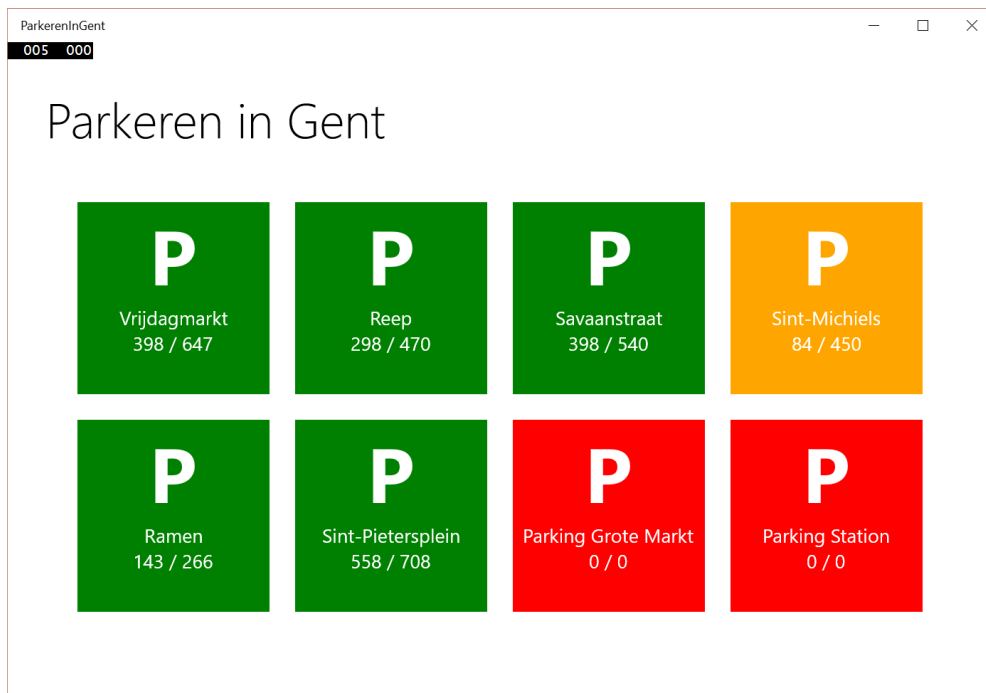
## Inleiding

Zoals in de theorie les uitgelegd is het de bedoeling dat u een applicatie maakt die de huidige parkeerstatus in Gent weergeeft.

U kan deze data terugvinden via volgende link:

<http://datatank.stad.gent/4/mobiliteit/bezettingparkingsrealtime.json>

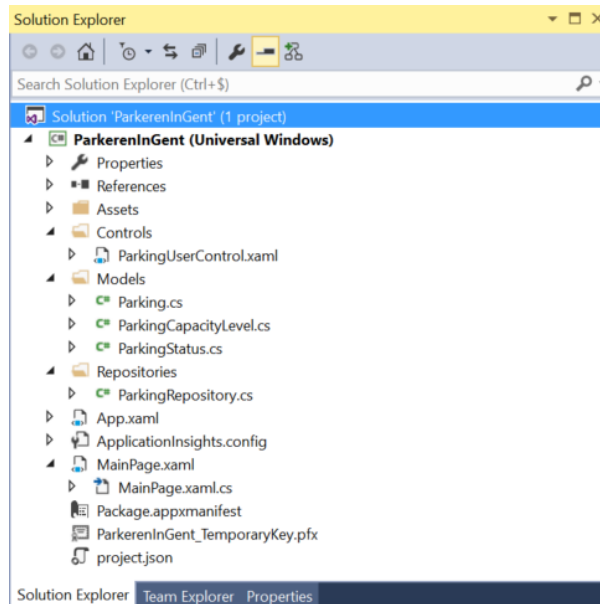
Deze data is een JSON file die u heel wat data ter beschikking stelt. Het is aan u om deze data als volgt weer te geven:



## Uitwerking

### Project Structuur

Wanneer u een groot project maakt is het belangrijk dat je elk onderdeel gemakkelijk kan terug vinden. Maak daarom volgende structuur aan in Visual Studio:



## Globale Settings

Er zijn een aantal variabelen / instellingen die je wil hergebruiken in uw applicatie. In onze applicatie zijn volgende variabelen interessant om globaal aan te maken:

- URL naar de JSON file
- Tussen welke waarde kleuren we ons Parking Element oranje

Maak hiervoor 2 resource objecten aan in de **app.xaml**. Deze kan je dan in uw applicatie ophalen via volgende code:

```
string url = Application.Current.Resources["parkingUrl"] as string;
```

## Het Parking object

Wanneer u de link volgt krijg je json zien die u een 8-tal parking objecten geeft. Bekijken we daar één parking object van ziet deze er zo uit:

```
{
  "id": 10,
  "name": "P01 Vrijdagmarkt",
  "description": "Vrijdagmarkt",
  "latitude": 51.057,
  "longitude": 3.726,
  "address": "Vrijdagmarkt 1\n9000 Gent",
  "contactInfo": "...",
  "blurAvailability": false,
  "city": {
    "id": 1,
    "name": "Gent"
  },
  "parkingServer": {
    "id": 2,
    "name": "ITG Gent"
  }
}
```

```
"suggestedFreeThreshold": 3,
"suggestedFullThreshold": 3,
"capacityRounding": 1,
"openingTimes": [
  {
    "days": [ "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY"],
    "from": "00:00",
    "to": "23:59"
  }
],
"openingTimesInfo": {
  "id": 1372234,
  "code": "24h_24",
  "text": "Open 24h/24"
},
"parkingStatus": {
  "availableCapacity": 398,
  "totalCapacity": 647,
  "open": true,
  "suggestedCapacity": "ACTUAL",
  "activeRoute": "P01 hoofd aanrijroute"
}
}
```

Wij zijn echter geïnteresseerd in volgende eigenschappen:

```
{
  "id": 10,
  "name": "P01 Vrijdagmarkt",
  "description": "Vrijdagmarkt",
  "latitude": 51.057,
  "longitude": 3.726,
  "address": "Vrijdagmarkt 1\n9000 Gent",
  "parkingStatus": {
    "availableCapacity": 398,
    "totalCapacity": 647,
    "open": true
  }
}
```

Om deze JSON om te zetten naar C# objecten via de Newtonsoft module moet je een c# klasse maken die al deze eigenschappen bevat. In dit geval zullen we 2 objecten nodig hebben:

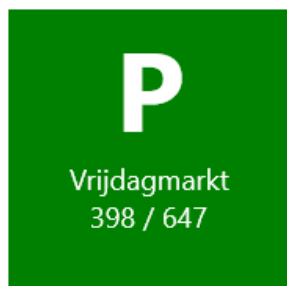
- Parking
- ParkingStatus

## Repositories

Voorzie een ParkingRepository waar je een methode voorziet die een verzameling van ParkingObjecten teruggeeft.

*De code om een webservice aan te spreken hebben jullie in week 01 cadeau gekregen. U mag deze code natuurlijk kopiëren en hergebruiken.*

## ParkingUserControl



De bedoeling is dat je een vierkant blokje (zie links) maakt waar volgende gegevens op staan:

- De letter P
- Description property
- AvailableCapacity / TotalCapacity

Schrijf hiervoor de nodige XAML code. Belangrijk is dat er geen enkele UI gerelateerde property in uw XAML code te vinden is. Alles wat met de UI te maken heeft maak je aan in een Style object.

U mag zelf de dimensies van het blokje bepalen. Maak hiervoor een resource object aan van het type double. Deze resource kunnen we dan gebruiken in onze stijl. Zo kunnen we deze makkelijk wijzigen indien nodig doordat deze maar op 1 plaats gedefinieerd is.

**Maak uw stijl code zo efficient mogelijk. Zo gebruik je style inheritance (via basedOn), default styles en resources om uw stijlen zo eenvoudig mogelijk te beheren.**

## Parking Capaciteit weergave

Als je naar de huidige applicaties kijkt, en dan zeker op een smartphone, wordt alles aangeduid adhv. kleuren. Zorg dat je volgende stijlen definieert:

- Groene achtergrond wanneer er nog meer als 30% van de totale capaciteit beschikbaar is
- Orange achtergrond wanneer er minder als 30% van de totale capaciteit beschikbaar is
- Rode achtergrond indien er geen plaats meer vrij is of wanneer de data niet beschikbaar is

Deze 30% is een variabele die je in de app.xaml gedefinieerd hebt.

### Extra

Maak een enumeratie aan die deze statussen weergeeft. Dit bevordert de leesbaarheid van je code.

```
public enum ParkingCapacityStatus
{
    AVAILABLE,
    SWARMING,
    FULL
}
```

## MainPage

Gebruik een gridview control om al onze data grafisch weer te geven. Voorzie boven de dat ook een titel. Deze titel wordt gestyled met één van **de ingebouwde stijlen van je operating system**.

### Extra

Indien u snel met deze oefening klaar bent kan u volgende uitbreidingen voorzien:

- Voorzie een mogelijkheid om parking te zoeken.
- Voorzie een mogelijkheid om 1 parking te selecteren en alle data weer te geven
- Voorzie een mogelijkheid om op beschikbare capaciteit te sorteren
- Voorzie een mogelijkheid om enkel de parkings te tonen waar er nog plaats is

Bij elke uitbreiding zorg je voor de juiste Style objecten.