

# Documentatie Drempeelvrij app

Code beschikbaar op: <http://code.google.com/p/wheelchair-routing/>

## Weergeven van de kaart

### Initialiseren van de kaart

De constructor van de Map class heeft twee parameters nodig. De eerste parameter bepaald op welk punt de kaart gecentreerd wordt en moet van het type LatLng zijn. De tweede parameter geeft aan in welk <div> element de kaart getekend moet worden.

```
$map = new Map(new LatLng(4.4758699, 51.9067910), 'map_canvas');
```

### Stratenbestand laden

De loadStreets methode heeft als enige parameter het pad naar het stratenbestand.

```
$map->loadStreets('streetsfile.php');
```

### Kaart tekenen

Dit moet gebeuren in het <head> gedeelte van de pagina. In de toekomst zal de draw methode een parameter krijgen waarin een array met opties voor het tekenen van de kaart meegegeven kan worden.

```
echo $map->draw();
```

## Opties voor het tekenen van de kaart

### Weergeven van kruisingen en wegdelen

Indien nodig kunnen de kruisingen en wegdelen worden weergegeven op de kaart met de drawJunctions en drawSections methodes. Deze methodes moeten worden uitgevoerd voor de draw methode.

```
$map->drawJunctions();  
$map->drawSections();  
echo $map->draw();
```

## Route berekenen

Voordat een route berekend kan worden moet er eerst een instantie gemaakt worden van de Map class en moet het stratenbestand worden ingeladen.

```
$map = new Map(new LatLng(4.4758699, 51.9067910), 'map_canvas');  
$map->loadStreets('streetsfile.php');
```

### Initialiseren van de Routing class

De Routing class moet weten met welk straten/kruising/wegdelen er gewerkt moet worden. De straten uit het stratenbestand worden meegegeven aan de constructor van de Routing class.

```
$routing = new Routing($map->streets);
```

### Alle mogelijke routes uitrekenen

Voordat een "beste" of "snelste" route gekozen kan worden moeten eerst alle mogelijke verbindingen berekend worden. Dit gebeurt met de calculate methode van de Routing class. Deze methode geeft een

array terug met de gekozen start en eind kruising van de route. De calculate methode heeft twee parameters nodig. De eerste parameter is het begin adres b.v. “Calandstraat+7,+Rotterdam,+Nederland” en de tweede parameter is het eind adres b.v. “Westerkade+14,+Rotterdam,+Nederland”.

```
$c = $routing->calculate('Calandstraat+7,+Rotterdam,+Nederland',  
'Westerkade+14,+Rotterdam,+Nederland');
```

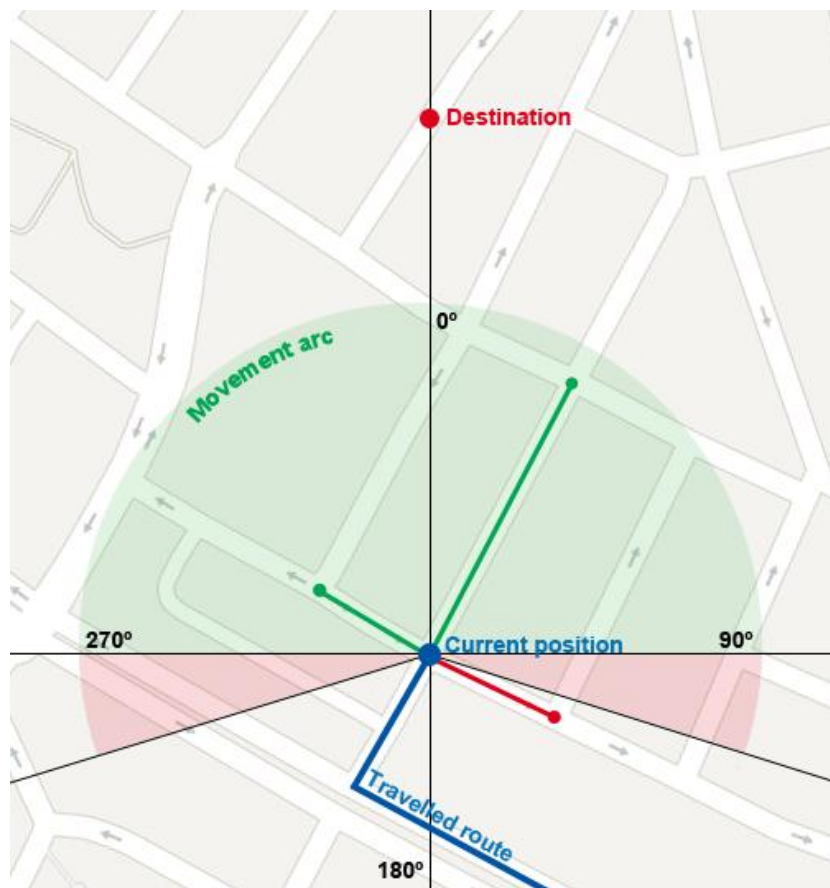
## Ophalen van de snelste of beste route

De getShortest en getBest methode van de Routing class geven beiden een Route object terug.

```
$route = $routing->getShortest();  
$route = $routing->getBest();
```

## Route algoritme

Het route algoritme berekend alle mogelijke routes van punt A naar punt B door te kijken naar de richting van een wegdeel ten opzichte van het einddoel.



In dit voorbeeld valt het rode wegdeel af omdat het zich buiten de “movement arc” bevindt. Als een wegdeel zich in het rode gedeelte van de “movement arc” bevindt dan wordt de “movement arc” voor het volgende wegdeel in die route verkleind om uitwaaiering te voorkomen. Het blauwe wegdeel valt ook af omdat we hier vandaan komen. Alleen de twee groene wegdelen zijn mogelijke routes vanaf deze kruising in dit voorbeeld.

# Stratenbestand

Het stratenbestand definieert alle kruisingen, straten en wegdelen.

## Kruising definitie:

De constructor van Junction heeft één parameter. Deze parameter bepaald het punt waarop de kruising zich bevindt en is van het type LatLong.

```
$junctions[0] = new Junction(new LatLong(51.9069300, 4.4733583));
```

## Straat definitie:

De constructor van Street heeft ook één parameter, namelijk: de naam van de straat.

```
$streets[0] = new Street('Parklaan');
```

## Wegdeel definitie:

Een wegdeel wordt toegevoegd aan een straat via de addSection methode van de Street class. De constructor van Section heeft twee parameters. Beide parameters accepteren een Junction object en beschrijven zo het begin en einde van het wegdeel.

```
$streets[0]->addSection(new Section($junctions[0], $junctions[1],  
'Tegels'));
```

## Huidige stratenbestand

