

## Common.Operations namespace

---

...PAGE 7

CommonService class	...PAGE 7
GetCountries( string language)	...PAGE 9
GetCategories( string language)	...PAGE 9
CompressPolyline( Polyline polyline)	...PAGE 7
UncompressPolyline( BinaryPolyline polyline)	...PAGE 9
ConvertCoordinates( Position[] positions,CoordinateTypes targetType,MapInfo mapInfo)	...PAGE 7

## Common.Types namespace

---

...PAGE 10

Area class	...PAGE 10
AreaByDisc class	...PAGE 10
AreaByHeightWidth class	...PAGE 10
AreaByLogicalScale class	...PAGE 12
AreaByLongLatBounds class	...PAGE 12
AreaByPhysicalScale class	...PAGE 12
AreaByPositions class	...PAGE 13
AreaRepresentations class	...PAGE 13
BasePicture class	...PAGE 15
BinaryPolyline class	...PAGE 15
BufferPicture class	...PAGE 17
Category class	...PAGE 17
Circle class	...PAGE 18
Country class	...PAGE 18
CustomerPicture class	...PAGE 19
Date class	...PAGE 19
Distance class	...PAGE 20
Label class	...PAGE 20
LongLatDistance class	...PAGE 22
LongLatPosition class	...PAGE 22
MapInfo class	...PAGE 22
MetersDistance class	...PAGE 24
MetersPosition class	...PAGE 24
PinPoint class	...PAGE 24
PixelDistance class	...PAGE 27
PixelPosition class	...PAGE 27
Polygon class	...PAGE 27
Polyline class	...PAGE 28
Position class	...PAGE 30
Projection class	...PAGE 30

Shape class	...PAGE 30
UrlPicture class	...PAGE 32
CollisionTypes enumeration	...PAGE 32
CoordinateTypes enumeration	...PAGE 32
HorizontalAlignments enumeration	...PAGE 34
ImageTypes enumeration	...PAGE 34
ProjectionTypes enumeration	...PAGE 34
ServiceTypes enumeration	...PAGE 36
UnitSystems enumeration	...PAGE 36
VerticalAlignments enumeration	...PAGE 36

---

## Core.Coder.Operations namespace

...PAGE 38

CoderService class	...PAGE 38
AddressToPoint( PostalAddress address,boolean filter)	...PAGE 38
CheckAddress( PostalAddress address,boolean filter)	...PAGE 38
PointToAddress( LongLatPosition origin,LocatorParameters option)	...PAGE 40

---

## Core.Coder.Types namespace

...PAGE 41

AddressParameters class	...PAGE 41
AddressResult class	...PAGE 41
CheckAddressResult class	...PAGE 41
CheckGeoResult class	...PAGE 43
CodedResult class	...PAGE 43
GeoCoderResult class	...PAGE 43
LocatorParameters class	...PAGE 45
LocatorResult class	...PAGE 45
PointResult class	...PAGE 45
PostalAddress class	...PAGE 46
GeoResultTypes enumeration	...PAGE 47

---

## Core.Itinerary.Operations namespace

...PAGE 48

ItineraryService class	...PAGE 48
ComputeRouteDefault( double lgStart,double ltStart,double lgEnd,double ltEnd)	...PAGE 48
ComputeRoute( RouteParameters routeParameters)	...PAGE 48
ComputeDistances( DistancesParameters distParameters)	...PAGE 48
GetRoadInfo( Point2RoadParameters pTRDParameters)	...PAGE 49

---

## Core.Itinerary.Types namespace

...PAGE 50

CustomSpeedProfile class	...PAGE 50
DistancesParameters class	...PAGE 50
DistancesResult class	...PAGE 52
DistanceStep class	...PAGE 52
ImageSettings class	...PAGE 52
ItineraryParameters class	...PAGE 54
OrientedWayPoint class	...PAGE 54
Point2RoadParameters class	...PAGE 56
Point2RoadResult class	...PAGE 56
RouteParameters class	...PAGE 58
RouteResult class	...PAGE 58
RouteStep class	...PAGE 59
RouteSummary class	...PAGE 62
SpeedCamera class	...PAGE 62
StepTollInfo class	...PAGE 64
WayPoint class	...PAGE 64
AreaRepresentationsOptions enumeration	...PAGE 64
ItineraryModes enumeration	...PAGE 66
ItineraryTypes enumeration	...PAGE 66
OptimizationTypes enumeration	...PAGE 66
Orientations enumeration	...PAGE 67
RankingTypes enumeration	...PAGE 67
RoadTypes enumeration	...PAGE 68
RouteStepModes enumeration	...PAGE 68
RouteStepTypes enumeration	...PAGE 70
SectionTypes enumeration	...PAGE 70
SpeedProfiles enumeration	...PAGE 71
UrbanValues enumeration	...PAGE 71

---

## Core.Map.Operations namespace ...PAGE 72

MapService class	...PAGE 72
DrawMapDefault( double lg,double lt)	...PAGE 72
DrawMap( MapParameters mapParameters)	...PAGE 72
DrawMaps( MapsParameters mapsParameters)	...PAGE 73
DrawMapFromId( string mapId)	...PAGE 73

---

## Core.Map.Types namespace ...PAGE 74

BaseMapParameters class	...PAGE 74
ImageSettings class	...PAGE 74
MapParameters class	...PAGE 74
MapResult class	...PAGE 75

MapsParameters class	...PAGE 77
MapStyle class	...PAGE 77
ImageResultTypes enumeration	...PAGE 79
LayerFlags enumeration	...PAGE 79
RotateAngles enumeration	...PAGE 80

---

## Core.Poi.Operations namespace

PoiService class	...PAGE 81
LookAroundDefault( double longitude,double latitude,int radius)	...PAGE 82
CountLookAround( LookAroundParameters lookAroundParameters)	...PAGE 81
LookAround( LookAroundParameters lookAroundParameters)	...PAGE 82
CountLookAlong( LookAlongParameters lookAlongParameters)	...PAGE 81
LookAlong( LookAlongParameters lookAlongParameters)	...PAGE 82
CountLookInside( LookInsideParameters lookInsideParameters)	...PAGE 81
LookInside( LookInsideParameters lookInsideParameters)	...PAGE 84

---

## Core.Poi.Types namespace

LookAlongParameters class	...PAGE 85
LookAroundParameters class	...PAGE 85
LookInsideParameters class	...PAGE 87
LookupConstraint class	...PAGE 87
LookupParameters class	...PAGE 89

---

## Extensions.Weather.Operations namespace

WeatherService class	...PAGE 90
WeatherForecast( ForecastParameters forecastParameters)	...PAGE 90
WeatherLayer( LayerParameters layerParameters)	...PAGE 90
WeatherLayerFromId( string layerId)	...PAGE 91

---

## Extensions.Weather.Types namespace

ForecastParameters class	...PAGE 92
ForecastResults class	...PAGE 92
ForecastSummary class	...PAGE 92
LayerParameters class	...PAGE 94
TypeLayer enumeration	...PAGE 94

---

## Partner.Operations namespace

DatabaseService class	...PAGE 97
-----------------------	------------

GetTables( )	...PAGE 99
GetPrimaryKey( string tablename)	...PAGE 99
GetCoordFields( string tablename)	...PAGE 99
GetSchema( string tablename)	...PAGE 99
ReadDefault( )	...PAGE 101
Read( ReadParameters readParameters)	...PAGE 101
CountRead( ReadParameters readParameters)	...PAGE 97
PartnerService class	...PAGE 101
ProximityService class	...PAGE 103
GetTables( )	...PAGE 105
GetPrimaryKey( string tablename)	...PAGE 104
GetCoordFields( string tablename)	...PAGE 104
GetSchema( string tablename)	...PAGE 104
SearchAroundDefault( double longitude,double latitude,int radius)	...PAGE 105
SearchAround( ProximityParameters proximityParameters)	...PAGE 105
CountSearchAround( ProximityParameters proximityParameters)	...PAGE 103
SearchService class	...PAGE 105
CountSearchAround( SearchAroundParameters searchAroundParameters)	...PAGE 107
SearchAround( SearchAroundParameters searchAroundParameters)	...PAGE 108
CountSearchAlong( SearchAlongParameters searchAlongParameters)	...PAGE 106
SearchAlong( SearchAlongParameters searchAlongParameters)	...PAGE 107
CountSearchInside( SearchInsideParameters searchInsideParameters)	...PAGE 107
SearchInside( SearchInsideParameters searchInsideParameters)	...PAGE 108
SpatialAccessExtension class	...PAGE 108
GetInitializer( Type serviceType)	...PAGE 108
GetInitializer( LogicalMethodInfo methodInfo,SoapExtensionAttribute attribute)	...PAGE 109
Initialize( Object initializer)	...PAGE 109
ProcessMessage( SoapMessage message)	...PAGE 109
<b>Partner.Operations.Configuration namespace</b>	...PAGE 111
<hr/>	
IpElement class	...PAGE 111
IpElementCollection class	...PAGE 111
SpatialAccessSection class	...PAGE 111
<b>Partner.Types namespace</b>	...PAGE 114
<hr/>	
CoordFields class	...PAGE 114
PartnerParameters class	...PAGE 114
PartnerResults class	...PAGE 115
<b>Partner.Types.Database namespace</b>	...PAGE 116
<hr/>	

ReadParameters class	...PAGE 116
ReadResults class	...PAGE 116
<b>Partner.Types.Proximity namespace</b>	...PAGE 119
<hr/>	
ProximityParameters class	...PAGE 119
ProximityResults class	...PAGE 119
<b>Partner.Types.Search namespace</b>	...PAGE 122
<hr/>	
SearchAlongParameters class	...PAGE 122
SearchAlongResults class	...PAGE 122
SearchAroundParameters class	...PAGE 123
SearchAroundResults class	...PAGE 125
SearchInsideParameters class	...PAGE 125
SearchInsideResults class	...PAGE 126

# 1. Common.Operations namespace

## 1.1. CommonService class

The Maporama CommonService offers methods to perform common operations out off business operations scope

Constuctors	
Constructor	Description
CommonService( )	Initializes a new instance of the Common.Operations.CommonService class.

### 1.1.1. CompressPolyline( Polyline polyline) method

Compresses the given Polyline using a proprietary binary format. The resulting BinaryPolyline object can be used with other Maporama Web Services.

Parameters			
Parameter	Type	Direction	Description
polyline	Common.Types.Polyline		A polyline to compress.

  

Returns	
Type	Description
Common.Types.BinaryPolyline	A compressed polyline.

### 1.1.2. ConvertCoordinates( Position[] positions,CoordinateTypes targetType,MapInfo mapInfo) method

Convert a collection of Positions of any type to any coordinate type. Note: All positions to convert must have the same coordinate type.

Parameters			
Parameter	Type	Direction	Description
mapInfo	Common.Types.MapInfo		Information about the Map (used to convert from and to Pixels coordinates).
positions	Common.Types.Position[]		The collection of Positions to convert.
targetType	Common.Types.CoordinateTypes		The Coordinate Type to convert to.

## Returns

Type	Description
Common.Types.Position[]	The converted collection of Positions of type 'targetType'.



### 1.1.3. GetCategories( string language) method

Returns the list of categories used by Maporama POI database for a given language (2-letter code).

Parameters			
Parameter	Type	Direction	Description
language	string		A 2-letter language code.

  

Returns	
Type	Description
Common.Types.Category[]	The list of categories.

### 1.1.4. GetCountries( string language) method

Returns the list of countries covered by Maporama and their associated services for a given language (2-letter code).

Parameters			
Parameter	Type	Direction	Description
language	string		A 2-letter language code.

  

Returns	
Type	Description
Common.Types.Country[]	The list of countries and the associated available services.

### 1.1.5. UncompressPolyline( BinaryPolyline polyline) method

Uncompresses the given Polyline using a proprietary binary format. A BinaryPolyline can be returned by different Maporama Web Services. The resulting Polyline object can be used with other Maporama Web Services.

Parameters			
Parameter	Type	Direction	Description
polyline	Common.Types.BinaryPolyline		A polyline to uncompress.

  

Returns	
Type	Description
Common.Types.Polyline	An uncompressed polyline.

## 2. Common.Types namespace

### 2.1. Area class

Abstract base type to define an geographic area on the globe. It can be used to render a Map or is returned when computing an itinerary.

### 2.2. AreaByDisc class

Defines an Area represented by a WGS84 Center and a Radius.

#### **Inherits :**

Common.Types.AreaByDisc

Common.Types.Area

Properties				
Property	Type	Description	Get	Set
Center	Common.Types.LongLatPosition	WGS84 Center Coordinates.	•	•
Radius	Common.Types.Distance	A Radius defining the Area. The possible distance types are: LongLat, Kilometers, Meters and Miles.	•	•

### 2.3. AreaByHeightWidth class

Defines an Area represented by a WGS84 Center and Width and Height in the given distance unit (default is Kilometers).

#### **Inherits :**

Common.Types.AreaByHeightWidth

Common.Types.Area

Properties				
Property	Type	Description	Get	Set
Center	Common.Types.LongLatPosition	WGS84 Center Coordinates.	•	•
Height	Common.Types.Distance	Height in the specified distance unit. The possible distance types are: LongLat, Kilometers, Meters and Miles.	•	•
		Width in the specified distance unit. The possible distance types are:		

Width	Common.Types.Distance	LongLat, Kilometers, Meters and Miles.	.	.
-------	-----------------------	--	---	---

## 2.4. AreaByLogicalScale class

Defines an Area represented by a WGS84 Center and a Logical Scale (from 1 to 12).

### **Inherits :**

Common.Types.AreaByLogicalScale

Common.Types.Area

Properties				
Property	Type	Description	Get	Set
Center	Common.Types.LongLatPosition	WGS84 Center Coordinates.	•	•
LogicalScale	int	A Logical Scale. Valid values are from 1 to 12:	•	•

## 2.5. AreaByLongLatBounds class

Defines an Area represented by a WGS84 Bounding Rectangle.

### **Inherits :**

Common.Types.AreaByLongLatBounds

Common.Types.Area

Properties				
Property	Type	Description	Get	Set
NorthEast	Common.Types.LongLatPosition	WGS84 Coordinates of the North-East corner of the Bounding Rectangle.	•	•
SouthWest	Common.Types.LongLatPosition	WGS84 Coordinates of the South-West corner of the Bounding Rectangle.	•	•

## 2.6. AreaByPhysicalScale class

Defines an Area represented by a WGS84 Center and a Physical Scale.

### **Inherits :**

Common.Types.AreaByPhysicalScale

Properties				
Property	Type	Description	Get	Set
Center	Common.Types.LongLatPosition	WGS84 Center Coordinates.	•	•
PhysicalScale	double	A Physical Scale, from 1/2,000 (5E-4) to 1/20,000,000 (5E-8).	•	•

## 2.7. AreaByPositions class

Defines an Area represented by a collection of WGS84 Positions. It can be used to ensure that most of the positions in the given collection will be visible on a single map.

### Inherits :

Common.Types.AreaByPositions

Common.Types.Area

Properties				
Property	Type	Description	Get	Set
HorizontalMarge	Common.Types.Distance	Optional Horizontal Marge. Note: Currently only WGS84 Distances are supported.	•	•
Positions	Common.Types.LongLatPosition[]	Collection of WGS84 Coordinates.	•	•
VerticalMarge	Common.Types.Distance	Optional Vertical Marge. Note: Currently only WGS84 Distances are supported.	•	•

## 2.8. AreaRepresentations class

Defines a single Area with 4 different representations.

Properties				
Property	Type	Description	Get	Set
AreaByHeightWidth	Common.Types.AreaByHeightWidth	The single Area represented by a WGS84 Center and its Width and Height, in Kilometers.	•	•

AreaByLogicalScale	Common.Types.AreaByLogicalScale	The single Area represented by a WGS84 Center and a Logical Scale.	•	•
AreaByLongLatBounds	Common.Types.AreaByLongLatBounds	The single Area represented by its WGS84 Bounds.	•	•
AreaByPhysicalScale	Common.Types.AreaByPhysicalScale	The single Area represented by a WGS84 Center and a Physical Scale.	•	•

## 2.9. BasePicture class

Base type for the picture to be used in PinPoint shapes.

Properties				
Property	Type	Description	Get	Set
IsTransparent	boolean	When set to true the Transparency color will be transparent.	•	•
Transparency	int	RGB Color to be used as the transparent color.	•	•

## 2.10. BinaryPolyline class

A binary polyline is used to represent an itinerary trace (its buffer uses a private format).

### Inherits :

Common.Types.BinaryPolyline

Common.Types.Shape

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
Buffer	byte[]	Buffer holding the collection of points composing the polyline stored in a private binary format.	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
UncompressedBufferSize	int	Holds the uncompressed buffer size.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2,	•	•

		and so on...		
--	--	--------------	--	--



## 2.11. BufferPicture class

Defines a picture using a buffer.

### **Inherits :**

Common.Types.BufferPicture

Common.Types.BasePicture

Properties				
Property	Type	Description	Get	Set
Buffer	byte[]	Image buffer. The format can be of one of the following types: GIF, BMP, JPEG, PNG.	•	•
IsTransparent	boolean	When set to true the Transparency color will be transparent.	•	•
Transparency	int	RGB Color to be used as the transparent color.	•	•

## 2.12. Category class

Describes the category associated to a category id

Properties				
Property	Type	Description	Get	Set
Class	string	Category class	•	•
Description	string	Category name	•	•
Id	int	Category identity	•	•

Constructors	
Constructor	Description
Category( )	Default Constructor.

## 2.13. Circle class

Defines a circle shape represented by a Center and a Radius. Note: Currently only WGS84 Positions are supported for the Center.

### **Inherits :**

Common.Types.Circle

Common.Types.Shape

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
Center	Common.Types.Position	Center of the circle. Note: Currently only WGS84 Positions are supported.	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
Radius	Common.Types.Distance	Radius of the circle.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.14. Country class

Defines the services associated to a country.

Properties				
Property	Type	Description	Get	Set
CountryCode	string	The Country Code.	•	•
CountryName	string	The Country Name.	•	•
Services	Common.Types.ServiceTypes	The Service Types (mask) available for this Country.	•	•

Constructors	
Constructor	Description
Country( )	Default constructor.
Country( string countryName,string countryCode,ServiceTypes types)	Parameter constructor.

## 2.15. CustomerPicture class

Defines a picture which have been uploaded via the Partner Hub.

### **Inherits :**

Common.Types.CustomerPicture

Common.Types.BasePicture

Properties				
Property	Type	Description	Get	Set
CustomerName	string	Partner Hub login.	•	•
ImageFileName	string	Image file name with its extension.	•	•
IsTransparent	boolean	When set to true the Transparency color will be transparent.	•	•
Transparency	int	RGB Color to be used as the transparent color.	•	•

## 2.16. Date class

Defines a date time structure.

Properties				
Property	Type	Description	Get	Set
Day	int	The actual day number of Date.	•	•
Month	int	The actual month number of Date.	•	•
Year	int	The actual year number of Date.	•	•

## 2.17. Distance class

Defines a distance using the specified Coordinate Type.

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
Value	double	The actual value of this Distance.	•	•

## 2.18. Label class

Defines a text label shape. Note: Currently only LongLat Position are supported.

### Inherits :

Common.Types.Label

Common.Types.Shape

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
Background	boolean	If true, the frame will be filled with the background color.	•	•
BackgroundColor	int	Background RGB Color when Background is set to true.	•	•
Bold	boolean	Text in bold.	•	•
BorderWidth	int	Size of the border. If positive, a frame is drawn around the text.	•	•
Color	int	RGB Color for the text (e.g. Red=#FF0000, Blue=RGB(0,0,255)).	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
FontFamily	string	Name of the font family.	•	•
HAlign	Common.Types.HorizontalAlignments	Horizontal alignment.	•	•
Height	int	Height of the text in pixels.	•	•
Highlight	boolean	Text is highlighted.	•	•

HighlightColor	int	RGB Color for the highlight effect when Highlight is set to true.	•	•
HighlightWidth	int	Highlight Width in pixels when Highlight is set to true. Minimum: 1 pixel ; Maximum = 3 pixels.	•	•
HOffset	int	When a Picture and a Text are both specified in a PinPoint, horizontal distance in pixels between the Picture and the shifted Text.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
Italic	boolean	Text in italic.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
Padding	int	Distance in pixels between the text and the frame.	•	•
Position	Common.Types.Position	Position of the top-left corner of this label. Note: Currently only LongLat Position are supported.	•	•
Round	int	Diameter in pixels of the round corners of the frame (default is 0).	•	•
Text	string	Text to be drawn on the map.	•	•
VAlign	Common.Types.VerticalAlignments	Vertical alignment.	•	•
VOffset	int	When a Picture and a Text are both specified in a PinPoint, vertical distance in pixels between the Picture and the shifted Text.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.19. LongLatDistance class

A Distance in Longitude or Latitude.

### **Inherits :**

Common.Types.LongLatDistance

Common.Types.Distance

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
Value	double	The actual value of this Distance.	•	•

## 2.20. LongLatPosition class

A Position where coordinates are Longitude and Latitude.

### **Inherits :**

Common.Types.LongLatPosition

Common.Types.Position

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
X	double	The X Coordinate or Longitude.	•	•
Y	double	The Y Coordinate or Latitude.	•	•

## 2.21. MapInfo class

Holds information about a rendered map. This object can be used to convert coordinates from and to pixels with `CommonService.ConvertCoordinates()`.

Properties				

Property	Type	Description	Get	Set
Center	Common.Types.LongLatPosition	WGS84 coordinates of the center of the rendered map.	•	•
Height	int	Height in pixels of the rendered image.	•	•
LogicalScale	int	The Logical Scale used to render the rendered map.	•	•
PhysicalScale	double	The Physical Scale used to render the rendered map.	•	•
Projection	Common.Types.Projection	Projection used to render the rendered map.	•	•
Width	int	Width in pixels of the rendered image.	•	•

## 2.22. MetersDistance class

A Distance in Meters.

### **Inherits :**

Common.Types.MetersDistance

Common.Types.Distance

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
Value	double	The actual value of this Distance.	•	•

## 2.23. MetersPosition class

A Position where coordinates are Meters.

### **Inherits :**

Common.Types.MetersPosition

Common.Types.Position

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
X	double	The X Coordinate or Longitude.	•	•
Y	double	The Y Coordinate or Latitude.	•	•

## 2.24. PinPoint class

Defines a pin point shape. Note: Currently only LongLat Position are supported.

### **Inherits :**

Common.Types.PinPoint

Common.Types.Shape



Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
CollisionType	Common.Types.CollisionTypes	Specify the collision type to prevent overlapping between PinPoints.	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
HAlign	Common.Types.HorizontalAlignments	Horizontal alignment.	•	•
HOffset	int	When a Picture and a Text are both specified in a PinPoint, horizontal distance in pixels between the Picture and the shifted Text.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
Label	Common.Types.Label	A Label Shape to associate to this PinPoint.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
Link	boolean	When set to true, a line is drawn between the pointed position and the icon/text.	•	•
LinkColor	int	ARGB Color for the link between the pointed position and the label and/or picture.	•	•
LinkWidth	int	Width in pixels of the link.	•	•
MaxCollisionResearch	int	Used when CollisionType is FixedPictureFloatingText or FloatingPictureAndText. When two PinPoints overlap (their circumscribed rectangles intersect), a new available position is searched between MinCollisionResearch pixels and MaxCollisionResearch pixels from the original position.	•	•
MinCollisionResearch	int	Used when CollisionType is FixedPictureFloatingText or FloatingPictureAndText. When two PinPoints overlap (their circumscribed rectangles intersect), a new available position is searched between MinCollisionResearch pixels and MaxCollisionResearch pixels from the original position.	•	•

Picture	Common.Types.BasePicture	A Picture Shape to associate to this PinPoint.	•	•
Position	Common.Types.Position	Pointed position. Note: Currently only LongLat Position are supported.	•	•
VAlign	Common.Types.VerticalAlignments	Vertical alignment.	•	•
VOffset	int	When a Picture and a Text are both specified in a PinPoint, vertical distance in pixels between the Picture and the shifted Text.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.25. PixelDistance class

A Distance in Pixels (integers).

### **Inherits :**

Common.Types.PixelDistance

Common.Types.Distance

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
Value	double	The actual value of this Distance.	•	•

## 2.26. PixelPosition class

A Position where coordinates are Pixels (integers).

### **Inherits :**

Common.Types.PixelPosition

Common.Types.Position

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
X	double	The X Coordinate or Longitude.	•	•
Y	double	The Y Coordinate or Latitude.	•	•

## 2.27. Polygon class

Class to define a polygon shape, which can be filled. Note: Currently only WGS84 Positions are supported.

### **Inherits :**

Common.Types.Polygon

Common.Types.Shape

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
Points	Common.Types.Position[]	The collection of points representing this polyline/polygon.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.28. Polyline class

Class to define a polyline shape (Fill and FillColor have no effects). Note: Currently only WGS84 Positions are supported.

### **Inherits :**

Common.Types.Polyline

Common.Types.Polygon

Common.Types.Shape

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•
Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•

LineWidth	int	Width of the outer line of the shape.	•	•
Points	Common.Types.Position[]	The collection of points representing this polyline/polygon.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.29. Position class

Defines a Position using the specified Coordinate Type.

Properties				
Property	Type	Description	Get	Set
Type	Common.Types.CoordinateTypes	The Type of this Position.	•	•
X	double	The X Coordinate or Longitude.	•	•
Y	double	The Y Coordinate or Latitude.	•	•

## 2.30. Projection class

Defines a Geometric Projection.

Properties				
Property	Type	Description	Get	Set
South	boolean	Flag used if Projection Type is UTM, to specify North (false) or South (true) hemisphere.	•	•
TimeZone	int	Time Zone used if Projection Type is UTM.	•	•
Type	Common.Types.ProjectionTypes	The Type of this Projection.	•	•

## 2.31. Shape class

Abstract base type for the different shapes which can be rendered on a map.

Properties				
Property	Type	Description	Get	Set
AltText	string	Alternative Text to be associated to the shape.	•	•

Fill	boolean	When this flag is set, the FillColor is used to fill the shape when possible.	•	•
FillColor	int	ARGB color to fill the shape.	•	•
Hyperlink	string	Hyperlink to be associated to the shape.	•	•
LineColor	int	ARGB color of the outer line of the shape.	•	•
LineWidth	int	Width of the outer line of the shape.	•	•
ZOrder	int	Drawing order when several shapes are specified. Starts at 0, then 1, 2, and so on...	•	•

## 2.32. UriPicture class

Defines a picture using a URL.

### **Inherits :**

Common.Types.UriPicture

Common.Types.BasePicture

Properties				
Property	Type	Description	Get	Set
IsTransparent	boolean	When set to true the Transparency color will be transparent.	•	•
Transparency	int	RGB Color to be used as the transparent color.	•	•
Url	string	Image URL. The format can be of one of the following types: GIF, BMP, JPEG, PNG.	•	•

## 2.33. CollisionTypes enumeration

Collision types used to prevent overlapping between PinPoints.

Fields	
Field	Description
FixedPictureAndText	Fixed picture and text (default value).
FixedPictureFloatingText	Fixed picture, floating text.
FloatingPictureAndText	Floating picture and text.

## 2.34. CoordinateTypes enumeration

Defines the Coordinate Types.

Fields	
Field	Description



Kilometers	Real/projected coordinates in Kilometers.
LongLat	Longitude and Latitude (for WGS84 coordinates).
Meters	Real/projected coordinates in Meters.
Miles	Real/projected coordinates in Miles.
Pixel	Pixel coordinates (to locate a point in a 2D picture for instance).

## 2.35. HorizontalAlignments enumeration

Horizontal Alignment types.

Fields	
Field	Description
Center	Centered.
Left	Aligned on the left.
Right	Aligned on the right.

## 2.36. ImageTypes enumeration

Image types which can be used to render a Map, Bar code, any image to draw.

Fields	
Field	Description
BMP	Bitmap (no compression).
GIF	GIF (256 colors only, very good compression, allows basic transparency).
JPEG	JPEG (good compression).
PNG	PNG (good compression, allows transparency).
TIFF	TIFF (lossless, no compression).
WBMP	WBMP (no compression).

## 2.37. ProjectionTypes enumeration

The different Types a Projection can use.

Fields	
Field	Description
L2E	Lambert II Etendu.

Mercator	World Mercator.
PC	Plate Carrée.
Unknown	Invalid Projection Type.
UTM	Universal Transverse Mercator.

## 2.38. ServiceTypes enumeration

The different Maporama Service types.

Fields	
Field	Description
All	All Service Types.
Barcode	Barcode.
DirectionsMP3	Directions.MP3.
Geocoder	Geocoder.
Itinerary	Itinerary/Route/Ranking.
Map	Map.
None	No Services.
ReverseGeocoder	Reverse Geocoder.
Sms	SMS (Short Messaging System).
Traffic	Traffic.
Weather	Weather.

## 2.39. UnitSystems enumeration

Defines the different Unit System which can be used for the scale of a Map or in a route plan of an Itinerary.

Fields	
Field	Description
Imperial	Imperial System (Miles, mph)
Metric	Metric System (Meters, Kilometers, Km/h).
MetricAndImperial	Both Metric and Imperial Systems.

## 2.40. VerticalAlignments enumeration

Vertical Alignment types.

<b>Fields</b>	
<b>Field</b>	<b>Description</b>
Bottom	Aligned on the bottom.
Center	Centered.
Top	Aligned on the top.

## 3. Core.Coder.Operations namespace

### 3.1. CoderService class

The Maporama Coder Web Service offers the method to perform geolocalization and reverse geolocalization operations.

Constuctors	
Constructor	Description
CoderService( )	Initializes a new instance of the Core.Coder.Operations.CoderService class.

#### 3.1.1. AddressToPoint( PostalAddress address,boolean filter) method

Returns the possible coordinates of a given address.

Parameters			
Parameter	Type	Direction	Description
address	Core.Coder.Types.PostalAddress		The postal address to geocode.
filter	boolean		When set, returns only the results with score greater than 80.

Returns	
Type	Description
Core.Coder.Types.PointResult	A collection of addresses.

#### 3.1.2. CheckAddress( PostalAddress address,boolean filter) method

Checks the validity of a given address.

Parameters			
Parameter	Type	Direction	Description
address	Core.Coder.Types.PostalAddress		The postal address to check.
filter	boolean		When set, returns only the results with score greater than 80.

Returns	
Type	Description

Core.Coder.Types.CheckAddressResult

A collection of checked addresses.

### 3.1.3. PointToAddress( LongLatPosition origin,LocatorParameters option) method

Returns the possible addresses at the given coordinates.

Parameters			
Parameter	Type	Direction	Description
option	Core.Coder.Types.LocatorParameters		Options.
origin	Common.Types.LongLatPosition		WGS84 Coordinates to locate.

  

Returns	
Type	Description
Core.Coder.Types.AddressResult	A collection of addresses.



## 4. Core.Coder.Types namespace

### 4.1. AddressParameters class

Contains parameters to point an address

Properties				
Property	Type	Description	Get	Set
Filter	boolean	Only results with score greater than 80 are kept.	•	•
Location	Core.Coder.Types.PostalAddress	Postal Address to locate.	•	•

### 4.2. AddressResult class

Result of a Point To Address operation.

**Inherits :**

Core.Coder.Types.AddressResult

Core.Coder.Types.CodedResult

Properties				
Property	Type	Description	Get	Set
ErrorCode	long	An optional error might be set even if a result is returned.	•	•
Results	Core.Coder.Types.LocatorResult[]	The collection of Locator results.	•	•

### 4.3. CheckAddressResult class

Result of a Check Address operation.

**Inherits :**

Core.Coder.Types.CheckAddressResult

Core.Coder.Types.CodedResult

Properties				
Property	Type	Description	Get	Set

ErrorCode	long	An optional error might be set even if a result is returned.	•	•
Results	Core.Coder.Types.CheckGeoResult[]	Collection of checked addresses.	•	•

#### 4.4. CheckGeoResult class

Result of a Check Address operation.

Properties				
Property	Type	Description	Get	Set
PostalAddress	Core.Coder.Types.PostalAddress	The checked Postal Address.	•	•
Score	double	Score of the result.	•	•
Type	short	Result type.	•	•

#### 4.5. CodedResult class

Abstract base type for coded Results.

Properties				
Property	Type	Description	Get	Set
ErrorCode	long	An optional error might be set even if a result is returned.	•	•

#### 4.6. GeoCoderResult class

Describes a result returned by the GeoCoding engine.

Properties				
Property	Type	Description	Get	Set
AreaRepresentation	Common.Types.AreaRepresentations	Area Representations of this Location. It defines the bounds containing the city for a city level geocoding or the street for a street level geocoding. An Area can be used to render a map for instance.	•	•
		Maximum distance in meters between the geocoded location and the result. For example, for a street level		

ErrorDistance	double	geocoding of "main street", the returned coordinates will be centered at the middle of "main street" and error distance will measure half the length of "main street".	•	•
Level	short	Precision level (1=City, 2=Zip Code or Street, 3=Address Number or Point of Interest).	•	•
Location	Core.Coder.Types.PostalAddress	The geocoded postal address.	•	•
Position	Common.Types.LongLatPosition	WGS84 Coordinates of the Location.	•	•
PositionIsValid	boolean	Indicates if the result could be displayed as it on a map.	•	•
Score	double	Confidence level for the result (0=Poor to 100=Exact).	•	•
Type	Core.Coder.Types.GeoResultTypes	Indicates the type of the result.	•	•

## 4.7. LocatorParameters class

Contains parameters to Locate a point

Properties				
Property	Type	Description	Get	Set
DistanceMax	int	Max distance to search into in meters	•	•
NbMaxResult	short	Number of result to return	•	•

## 4.8. LocatorResult class

Result of a Locator operation.

Properties				
Property	Type	Description	Get	Set
CityCenter	Common.Types.LongLatPosition	The position of the nearest city center.	•	•
CityPositionRepresentation	Common.Types.AreaRepresentations	Area Representations of the City Position of this Result. An Area can be used to render a map for instance.	•	•
DistanceToAddress	double	Distance in meter from the position to the postal address.	•	•
DistanceToCityCenter	double	Distance in meter from the position to the nearest city center.	•	•
Location	Core.Coder.Types.PostalAddress	The postal address.	•	•
Position	Common.Types.LongLatPosition	The position of the address.	•	•
StreetPositionRepresentation	Common.Types.AreaRepresentations	Area Representations of the Street Position of this Result. An Area can be used to render a map for instance.	•	•

## 4.9. PointResult class

Result of a Address to Point operation.

**Inherits :**

Core.Coder.Types.PointResult

Core.Coder.Types.CodedResult

Properties				
Property	Type	Description	Get	Set
ErrorCode	long	An optional error might be set even if a result is returned.	•	•
Results	Core.Coder.Types.GeoCoderResult[]	The collection of GeoCoder results.	•	•

**4.10. PostalAddress class**

Contains all required information to geocode a location.

Properties				
Property	Type	Description	Get	Set
Address	string	Street Type, Street Name and Street Number, in any order. The Name of a Point Of Interest (POI) may also be supplied (e.g. Eiffel Tower).	•	•
AddressId	string	Internal unique identifier for the Address. May be used for quick reference.	•	•
City	string	Name of the City. Mandatory if Zip Code and State are omitted.	•	•
CityId	string	Internal unique identifier for the City. May be used for quick reference.	•	•
Country	string	International two-character Country Code based on Internic standard (see Appendix).	•	•
State	string	Two-letter State in the USA and Canada, three-letter State in Australia. Recommended wherever applicable.	•	•
ZipCode	string	Zip Code (or Postal Code) of the location. Mandatory if City and State are omitted.	•	•

## 4.11. GeoResultTypes enumeration

Returned Point Address Types.

Fields	
Field	Description
City	Indicates that the returned Address is a City.
POI	Indicates that the returned Address is a POI (Point of Interest).
State	Indicates that the returned Address is a State.
Street	Indicates that the returned Address is a Street.
Unknown	Indicates that the returned Address has an Unknown Type.
ZipCode	Indicates that the returned Address is a Zip Code.

## 5. Core.Itinerary.Operations namespace

### 5.1. ItineraryService class

The Maporama Itinerary Web Service exposes the methods to compute itineraries and distances.

Constructors	
Constructor	Description
ItineraryService( )	Initializes a new instance of the Core.Itinerary.Operations.ItineraryService class.

#### 5.1.1. ComputeDistances( DistancesParameters distParameters) method

Computes the distances between the given positions.

Parameters			
Parameter	Type	Direction	Description
distParameters	Core.Itinerary.Types.DistancesParameters		The parameters to be used to compute the distances.

  

Returns	
Type	Description
Core.Itinerary.Types.DistancesResult	The resulting distance matrix.

#### 5.1.2. ComputeRoute( RouteParameters routeParameters) method

Computes an itinerary using the given parameters.

Parameters			
Parameter	Type	Direction	Description
routeParameters	Core.Itinerary.Types.RouteParameters		The parameters to be used to compute the route.

  

Returns	
Type	Description
Core.Itinerary.Types.RouteResult	The resulting route and its steps.

#### 5.1.3. ComputeRouteDefault( double lgStart,double ltStart,double lgEnd,double ltEnd) method



Computes an itinerary between the given locations using the default parameters.

Parameters			
Parameter	Type	Direction	Description
lgEnd	double		Longitude of the ending position.
lgStart	double		Longitude of the starting position.
ltEnd	double		Latitude of the ending position.
ltStart	double		Latitude of the starting position.

  

Returns	
Type	Description
Core.Itinerary.Types.RouteResult	The resulting route and its steps.

#### 5.1.4. GetRoadInfo( Point2RoadParameters pTRDParameters) method

Get road info from longlat position.

Parameters			
Parameter	Type	Direction	Description
pTRDParameters	Core.Itinerary.Types.Point2RoadParameters		The parameters to be used to get the road info.

  

Returns	
Type	Description
Core.Itinerary.Types.Point2RoadResult	Road info and score.

## 6. Core.Itinerary.Types namespace

### 6.1. CustomSpeedProfile class

Allows to define custom speed values in Km/h for each speed category (8).

Properties				
Property	Type	Description	Get	Set
SpeedValue1	int	Speed Value in Km/h for Speed Category 1.	•	•
SpeedValue2	int	Speed Value in Km/h for Speed Category 2.	•	•
SpeedValue3	int	Speed Value in Km/h for Speed Category 3.	•	•
SpeedValue4	int	Speed Value in Km/h for Speed Category 4.	•	•
SpeedValue5	int	Speed Value in Km/h for Speed Category 5.	•	•
SpeedValue6	int	Speed Value in Km/h for Speed Category 6.	•	•
SpeedValue7	int	Speed Value in Km/h for Speed Category 7.	•	•
SpeedValue8	int	Speed Value in Km/h for Speed Category 8.	•	•

### 6.2. DistancesParameters class

Class to define the parameters to compute a route.

#### Inherits :

Core.Itinerary.Types.DistancesParameters

Core.Itinerary.Types.ItineraryParameters

Properties				
Property	Type	Description	Get	Set
AvoidHighways	boolean	Avoid highways.	•	•
AvoidTollRoads	boolean	Avoid toll roads.	•	•
CustomSpeedProfile	Core.Itinerary.Types.CustomSpeedProfile	Speed Values for each Speed Categories.	•	•
	Core.Itinerary.Types.ItineraryM	The mean to use to compute the		

ItineraryMode	odes	itinerary (vehicle, pedestrian...).	•	•
OptimizationType	Core.Itinerary.Types.OptimizationTypes	The optimization type, either by distance or by time.	•	•
PrioritizeMajorRoads	boolean	Prioritize major roads.	•	•
RankingType	Core.Itinerary.Types.RankingTypes	Specifies the ranking type.	•	•
RealTimeTrafficInfo	boolean	Use real time traffic info optimization.	•	•
SpeedProfile	Core.Itinerary.Types.SpeedProfiles	Pre-defined Speed Profile.	•	•
UnitSystem	Common.Types.UnitSystems	The unit system to be used for speeds and distances.	•	•
WayPoints	Core.Itinerary.Types.WayPoint[]	Way points (start, vias(optionals) and end).	•	•

### 6.3. DistancesResult class

Holds the result matrix of a distance computation.

Properties				
Property	Type	Description	Get	Set
DistanceSteps	Core.Itinerary.Types.DistanceStep[]	The collection of steps defining the distance results.	•	•

### 6.4. DistanceStep class

Informations about a single distance result.

Properties				
Property	Type	Description	Get	Set
Distance	int	Distance of the current step. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
EndName	string	Name of the End Position.	•	•
EndPosition	Common.Types.LongLatPosition	WGS84 Coordinates of the End Position.	•	•
StartName	string	Name of the Start Position.	•	•
StartPosition	Common.Types.LongLatPosition	WGS84 Coordinates of the Start Position.	•	•
Time	int	Time for this Step.	•	•
TrafficClosed	boolean	Indicates if the Traffic is closed for this Step.	•	•
TrafficTime	int	Time with Traffic for this Step.	•	•

### 6.5. ImageSettings class

Settings relative to the preparation of an itinerary Map rendering.

Properties				
------------	--	--	--	--

Property	Type	Description	Get	Set
Height	int	Height in pixels of the image of map to render.	•	•
ItineraryTraceLineColor	int	Itinerary trace line color.	•	•
ItineraryTraceLineWidth	int	Itinerary trace line width.	•	•
MapletHeight	int	Height in pixels of the image of maplet to render.	•	•
MapletWidth	int	Width in pixels of the image of maplet to render.	•	•
Width	int	Width in pixels of the image of map to render.	•	•

## 6.6. ItineraryParameters class

Base class to define the parameters to compute an itinerary. Specialized class are RouteParameters and DistancesParameters.

Properties				
Property	Type	Description	Get	Set
AvoidHighways	boolean	Avoid highways.	•	•
AvoidTollRoads	boolean	Avoid toll roads.	•	•
CustomSpeedProfile	Core.Itinerary.Types.CustomSpeedProfile	Speed Values for each Speed Categories.	•	•
ItineraryMode	Core.Itinerary.Types.ItineraryModes	The mean to use to compute the itinerary (vehicle, pedestrian...).	•	•
OptimizationType	Core.Itinerary.Types.OptimizationTypes	The optimization type, either by distance or by time.	•	•
PrioritizeMajorRoads	boolean	Prioritize major roads.	•	•
RealTimeTrafficInfo	boolean	Use real time traffic info optimization.	•	•
SpeedProfile	Core.Itinerary.Types.SpeedProfiles	Pre-defined Speed Profile.	•	•
UnitSystem	Common.Types.UnitSystems	The unit system to be used for speeds and distances.	•	•
WayPoints	Core.Itinerary.Types.WayPoint[]	Way points (start, vias(optionals) and end).	•	•

## 6.7. OrientedWayPoint class

A specialized WayPoint allowing to specify an orientation.

**Inherits :**

Core.Itinerary.Types.OrientedWayPoint

Core.Itinerary.Types.WayPoint

Properties				
Property	Type	Description	Get	Set
CityLevel	boolean	Tell if this point is used to locate a city (true) or a more precise location, like a	•	•

		street (false).		
Name	string	Display name of this WayPoint.	•	•
Orientation	double	Allow to specify an angle in degrees when searching for an edge.	•	•
Position	Common.Types.LongLatPosition	WGS84 Coordinates for this WayPoint.	•	•

## 6.8. Point2RoadParameters class

Class to define the parameters to get road info.

Properties				
Property	Type	Description	Get	Set
CourseOverGround	double	Gps orientation (Course Over Ground.).	•	•
CurrentPosition	Common.Types.LongLatPosition	Default constructor.	•	•
EndPoint	Common.Types.LongLatPosition	End point coordinates	•	•
GenerateRoadTrace	boolean	If set to true, the Road Trace in the PointToRoad Result will be generated as a object.	•	•
PreviousCourseOverGround	double	Previous Gps orientation (Course Over Ground.).	•	•
PreviousPoint	Common.Types.LongLatPosition	previous position coordinates	•	•
PreviousRoadType	Core.Itinerary.Types.SectionTypes	Road Class	•	•
PreviousSpeed	int	Previous road speed category	•	•
RoadTypeExpected	Core.Itinerary.Types.SectionTypes	Previous road class.	•	•
Speed	int	Road speed category	•	•

## 6.9. Point2RoadResult class

Holds the Point2Road result.

Properties				
Property	Type	Description	Get	Set
FormOfWay	Core.Itinerary.Types.RoadTypes	Indicates the Road type for this Step. It identifies certain aspects of the physical form that a road takes.	•	•
GlobalDirection	string	Global Direction of this road.	•	•
	Core.Itinerary.Types.Orientatio			



GlobalOrientation	ns	Global Orientation of this road.	•	•
ParamScore	int	Get parameters score	•	•
ResultScore	int	Get result score.	•	•
RoadClass	Core.Itinerary.Types.SectionTypes	Indicates the Road class.	•	•
RoadElementInAdministrativeArea	string	Identifies which particular Order-8 Area a Road Element belongs to.	•	•
RoadEndPoint	Common.Types.LongLatPosition	Road End point coordinates	•	•
RoadName	string	Get road name.	•	•
RoadNumber	string	Route Number The route number is the official identifying number of the road and is designated by the administrative authorities. A "route number" is usually alphanumeric such as "I-10", "US-50", or "A3".	•	•
RoadStartPoint	Common.Types.LongLatPosition	Road Start point coordinates	•	•
RoadTrace	Common.Types.BinaryPolyline	The BinaryPolyline object can be used to render the Road trace on a map.	•	•
Speed	int	Get road speed category	•	•
UrbanValue	Core.Itinerary.Types.UrbanValues	Indicates if this road id urban or not.	•	•

## 6.10. RouteParameters class

Class to define the parameters to compute a route.

### Inherits :

Core.Itinerary.Types.RouteParameters

Core.Itinerary.Types.ItineraryParameters

Properties				
Property	Type	Description	Get	Set
AreaRepresentationOption	Core.Itinerary.Types.AreaRepresentationsOptions	Set the AreaRepresentations generation option.	•	•
AvoidHighways	boolean	Avoid highways.	•	•
AvoidTollRoads	boolean	Avoid toll roads.	•	•
CustomSpeedProfile	Core.Itinerary.Types.CustomSpeedProfile	Speed Values for each Speed Categories.	•	•
GenerateItineraryTrace	boolean	If set to true, the Itinerary Trace in the Itinerary Result will be generated as a object.	•	•
ItineraryMapSettings	Core.Itinerary.Types.ImageSettings	Itinerary image settings.	•	•
ItineraryMode	Core.Itinerary.Types.ItineraryModes	The mean to use to compute the itinerary (vehicle, pedestrian...).	•	•
Language	string	The language of the step route sentences (2 letter code).	•	•
OptimizationType	Core.Itinerary.Types.OptimizationTypes	The optimization type, either by distance or by time.	•	•
PrioritizeMajorRoads	boolean	Prioritize major roads.	•	•
RealTimeTrafficInfo	boolean	Use real time traffic info optimization.	•	•
SpeedProfile	Core.Itinerary.Types.SpeedProfiles	Pre-defined Speed Profile.	•	•
UnitSystem	Common.Types.UnitSystems	The unit system to be used for speeds and distances.	•	•
WayPoints	Core.Itinerary.Types.WayPoint[]	Way points (start, vias(optionals) and end).	•	•

## 6.11. RouteResult class

Holds the itinerary result of a route computation.

Properties				
Property	Type	Description	Get	Set
RouteSteps	Core.Itinerary.Types.RouteStep []	The collection of steps defining the route.	•	•
Summary	Core.Itinerary.Types.RouteSummary	Summary of the route.	•	•
XmlRouteTest	string	For test purposes only. <b>Obsolete : For test purposes only.</b>	•	•

## 6.12. RouteStep class

Informations about a single step of a route.

Properties				
Property	Type	Description	Get	Set
AreaRepresentations	Common.Types.AreaRepresentations	Generated Areas to represent the area covered by the step. These Areas can be used to render a map or perform a local search.	•	•
Country	string	Two-letter country code for this step.	•	•
CumulDistance	int	Cumulative distance from the first step of the route including this step. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
CumulTime	int	Cumulated time in minutes from the first step of the route including this step.	•	•
CumulTrafficTime	int	Cumulated Traffic Time in minutes from the first step of the route including this step.	•	•
Distance	int	Distance of the current step. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
EndCity	string	Name of the city at the end of this step (optional).	•	•
EndOrientation	Core.Itinerary.Types.Orientations	Orientation at the end of this step.	•	•
EndPosition	Common.Types.LongLatPosition	Coordinates of the last point of this step.	•	•

ExitNumber	string	Number of the exit road to take (optional).	•	•
IsStart	boolean	For Transit and Pedestrian steps, indicates a start.	•	•
IsTollRoad	boolean	Indicates if the Step is a Toll Road.	•	•
Name	string	Name of the step. This is the name of the Road for Road Steps and the name of the Line for Transit Steps.	•	•
RoadNumber	string	Gives the official Road Number for this step, it can be empty.	•	•
RoadType	Core.Itinerary.Types.RoadTypes	Indicates the Road type for this Step. It identifies certain aspects of the physical form that a road takes.	•	•
RoundaboutExitNumber	int	Exit number to take when the road type of this step is Roundabout.	•	•
SectionType	Core.Itinerary.Types.SectionTypes	Indicates the Section type for this Step. This is the hierarchical classification of a road network or specify a particular section (Transit, Ferry).	•	•
Sentence	string	A sentence of the maneuver to take using the specified language.	•	•
SignPost	string	Signpost information associated to this Step (optional).	•	•
SpeedCameras	Core.Itinerary.Types.SpeedCamera[]	A collection of speed cameras on this route segment. If no speed camera, the array will be null.	•	•
StartCity	string	Name of the city at the beginning of this step (optional).	•	•
StartOrientation	Core.Itinerary.Types.Orientations	Orientation at the beginning of this step.	•	•
StartPosition	Common.Types.LongLatPosition	Coordinates of the first point of this step.	•	•
StepMode	Core.Itinerary.Types.RouteStepModes	Indicates the Transport Mode for this Step.	•	•
StepType	Core.Itinerary.Types.RouteStepTypes	Indicates the Type of this Step.	•	•
Time	int	Time in minutes for this step.	•	•
TollInfos	Core.Itinerary.Types.StepTollInfo[]	Holds information about a Toll Road or a Toll Booth. If no toll information, the object will be null.	•	•
TrafficClosed	boolean	Indicates if the traffic is closed on this step.	•	•
TrafficEvents	int[]	A collection of traffic events on this route segment.	•	•
TrafficTime	int	Traffic Time in minutes for this step.	•	•

TransitDirection	string	For Transit steps, supplies the name of the direction to take.	•	•
Urban	boolean	This flag is set when this step is located in an urban area.	•	•

## 6.13. RouteSummary class

Summary of a route.

Properties				
Property	Type	Description	Get	Set
AreaRepresentations	Common.Types.AreaRepresentations	Generated Areas to represent the area covered by the itinerary.	•	•
AverageSpeed	double	Average speed for complete route. Unit is feet/seconds if UnitSystem was set to Imperial otherwise meters/seconds are used.	•	•
HighwayDistance	int	Total Highway Distance. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
HighwayTime	int	Total Highway Time, in seconds.	•	•
ItineraryTrace	Common.Types.BinaryPolyline	The BinaryPolyline object can be used to render the itinerary trace on a map.	•	•
Language	string	The language of the step route sentences (2 letter code).	•	•
TotalDistance	int	Total Distance. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
TotalTime	int	Total Time, in seconds.	•	•
TrafficClosed	boolean	Will be set if a portion of the route is closed.	•	•
TrafficTime	int	Total Time when using Traffic Infos, in seconds.	•	•

## 6.14. SpeedCamera class

Holds information about a Speed Camera.

Properties				
Property	Type	Description	Get	Set
City	string	Name of the closest city.	•	•
		Calculated driving distance when		

Distance	int	crossing this Speed Camera. Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
OfficialDistancePoint	int	Official Distance on the Road ("Point Kilométrique" in France). Unit is feet if UnitSystem was set to Imperial otherwise meters are used.	•	•
SpeedLimit	double	Speed Limit, in Km/h.	•	•
Time	int	Calculated driving time when crossing this Speed Camera.	•	•

## 6.15. StepTollInfo class

Vehicle Step TollInfo.

Properties				
Property	Type	Description	Get	Set
CumulDistance	int	Cumul distance of toll info.	•	•
CumulTime	int	Cumul time of toll info.	•	•
Name	string	Name of the Toll Booth.	•	•
TollFeeSincePrevious	double	Toll Fee from the previous Toll Info.	•	•

## 6.16. WayPoint class

Holds informations about a given position (For Start, End or Via Way Points...).

Properties				
Property	Type	Description	Get	Set
CityLevel	boolean	Tell if this point is used to locate a city (true) or a more precise location, like a street (false).	•	•
Name	string	Display name of this WayPoint.	•	•
Position	Common.Types.LongLatPosition	WGS84 Coordinates for this WayPoint.	•	•

## 6.17. AreaRepresentationsOptions enumeration

Defines the possible options for the AreaRepresentations generation.

Fields	
Field	Description
Itinerary	Generate AreaRepresentations for Itinerary only.
ItineraryAndSteps	Generate AreaRepresentations for both Itinerary and Steps.



None	Don't generate AreaRepresentations at all (default).
Steps	Generate AreaRepresentations for Steps only (for "Maplets").

## 6.18. ItineraryModes enumeration

Defines the mode of the itinerary.

Fields	
Field	Description
Pedestrian	Itinerary for pedestrians.
PublicTransport	Itinerary for pedestrians with public transport.
Vehicle	Itinerary for vehicles (default).

## 6.19. ItineraryTypes enumeration

Defines the type of the itinerary.

Fields	
Field	Description
DefaultValue	Default value for this enum type.
Navigation	Route Plan for Navigation Solution.
Rank	Ranking/Distance (One-to-Many and Many-to-Many).
Route	Driving Directions Route Plan.
valuesCount	Number of values in this enum.

## 6.20. OptimizationTypes enumeration

Defines the type of graph optimization.

Fields	
Field	Description
Distance	Minimize the distance.
Time	Minimize the speed (default).

## 6.21. Orientations enumeration

Defines the different possible orientations of a step.

Fields	
Field	Description
E	East.
ENE	East-North-East.
ESE	East-South-East.
N	North.
NE	North-East.
NNE	North-North-East.
NNW	North-North-West.
NW	North-West.
S	South.
SE	South-East.
SSE	South-South-East.
SSW	South-South-West.
SW	South-West.
W	West.
WNW	West-North-West.
WSW	West-South-West.

## 6.22. RankingTypes enumeration

Defines the type of ranking computation to be used by ComputeDistances.

Fields	
Field	Description
ManyToMany	Many to Many. The way points are used both as start points and as end points.
OneToMany	One to Many (default). The first way point is used as the start point and the remaining points are used as end points.

## 6.23. RoadTypes enumeration

Road Types identify certain aspects of the physical form that a road takes.

Fields	
Field	Description
CircularIntersection	A circular intersection can be assimilated to a roundabout with special cases.
DeliveryAccess	Roads that allow delivery access but not auto access.
Highway	Highways, roads with controlled access.
MultipleOpposingLanes	Roads with multiple opposing lines which are not highways.
PedestriansOnly	Pedestrians only represents a walkway. In Europe, this may represent a bicycle only lane.
Ramp	A ramp is a road section to go from one road to another. Typically used to access highways.
Roundabout	Roundabout (contiguous loop with consistent one-way traffic).
ServiceAccessWithoutParking	Indicates that the road connects the main road network to services but without parking service.
ServiceAccessWithParking	Indicates that the road connects the main road network to services and features parking, garage or park.
ServiceRoad	A Service Road is a local road that run parallel to and usually contain the name(s) and addresses of a road with a higher traffic flow.
SingleOpposingLanes	Classic roads, with 2 direction lanes.
Undefined	Undefined road type.
UndefinedTrafficSquare	Undefined Traffic Square is an unstructured traffic square but where there are no legally defined traffic paths.

## 6.24. RouteStepModes enumeration

Defines the different modes a Route Step can take.

Fields	
Field	Description
Pedestrian	A Pedestrian Route Step.
PublicTransport	A Public Transport Route Step.

Vehicle	A Vehicle Route Step.
---------	-----------------------

## 6.25. RouteStepTypes enumeration

Defines the possible types for a Route Step.

Fields	
Field	Description
CityToCityEnter	An Exit Step of a City-to-City Itinerary.
CityToCityExit	An Exit Step of a City-to-City Itinerary.
MultiEnd	A End Step of a Multi End Itinerary.
Other	Other Step Type.
Via	Via Step.

## 6.26. SectionTypes enumeration

Section type is the hierarchical classification of a road network or specify a particular section (Transit, Ferry).

Fields	
Field	Description
DefaultValue	FourthClassRoad is the DefaultValue.
FerrySection	Ferry boat section.
FirstClassRoad	1st class road.
FourthClassRoad	4th class road (lower level).
MainRoad	Main road class (higher level: highways...).
NoThroughTraffic	Used for sections where traffic is not allowed.
SecondClassRoad	2nd class road.
ThirdClassRoad	3rd class road.
TransitSection	Public transit section (bus, metro...).
Undefined	-1 when SectionType is not defined

## 6.27. SpeedProfiles enumeration

Defines some predefined Speed Profiles.

Fields	
Field	Description
Fast	Fast speeds.
FastAndFurious	Very fast speeds.
Moderate	Moderate speeds (default).
Slow	Slow speeds.
SlowMotion	Very slow speeds.

## 6.28. UrbanValues enumeration

Navigateq reference: Urban (29). This attribute identifies Road Elements or Ferry Connections located with in a Built-up Area.

Fields	
Field	Description
DefaultValue	Default value is not urban road.
NotUrban	Not urban road
Urban	Urban road.
ValuesCount	values count of this enum.

## 7. Core.Map.Operations namespace

### 7.1. MapService class

The Maporama Map Web Service exposes the method to render or prepare maps and utility methods to convert coordinates.

Constructors	
Constructor	Description
MapService( )	Initializes a new instance of the Core.Map.Operations.MapService class.

#### 7.1.1. DrawMap( MapParameters mapParameters) method

Render or prepare a Map using the given parameters.

Parameters			
Parameter	Type	Direction	Description
mapParameters	Core.Map.Types.MapParameters		Parameters used to render or prepare the Map.

  

Returns	
Type	Description
Core.Map.Types.MapResult	A MapResult object, with the URL or the Image Buffer and other extra information.

#### 7.1.2. DrawMapDefault( double lg,double lt) method

Render a map at street-level centered on the given location using the default parameters.

Parameters			
Parameter	Type	Direction	Description
lg	double		Longitude of the center point of the Map to render.
lt	double		Latitude of the center point of the Map to render.

  

Returns	
Type	Description
byte[]	The Image Buffer of the rendered Map.



### 7.1.3. DrawMapFromId( string mapId) method

Render a map from a stored Map Request represented by its ID.

Parameters			
Parameter	Type	Direction	Description
mapId	string		A Map ID.

  

Returns	
Type	Description
Core.Map.Types.MapResult	The Image Buffer of the rendered Map.

### 7.1.4. DrawMaps( MapsParameters mapsParameters) method

Render or prepare one or more Maps using the given parameters.

Parameters			
Parameter	Type	Direction	Description
mapsParameters	Core.Map.Types.MapsParameters		Parameters used to render or prepare the Maps.

  

Returns	
Type	Description
Core.Map.Types.MapResult[]	A collection of MapResult object, with the URL or the Image Buffer and other extra information.

## 8. Core.Map.Types namespace

### 8.1. BaseMapParameters class

Abstract base type to define the parameters to render a map.

Properties				
Property	Type	Description	Get	Set
ImageSettings	Core.Map.Types.ImageSettings	Settings of the image representing the rendered maps.	•	•
Shapes	Common.Types.Shape[]	A collection of shapes to be drawn over the maps to render.	•	•
Style	Core.Map.Types.MapStyle	Defines the style of the maps to render.	•	•

### 8.2. ImageSettings class

Settings relative to the image generated to render a Map.

Properties				
Property	Type	Description	Get	Set
Height	int	Height in pixels of the image to render.	•	•
ImageQuality	int	Compression rate of the image to render. Used only with ImageTypes.GIF, ImageTypes.PNG and ImageTypes.JPEG.	•	•
ImageType	Common.Types.ImageTypes	Image type of the image to render.	•	•
ResultType	Core.Map.Types.ImageResultTypes	Specify the format of the resulting Map.	•	•
Width	int	Width in pixels of the image to render.	•	•

### 8.3. MapParameters class

Parameters to render a single map.

**Inherits :**

Core.Map.Types.MapParameters

Properties				
Property	Type	Description	Get	Set
Area	Common.Types.Area	An Area to be represented by a map. It can be of type AreaByPhysicalScale, AreaByLogicalScale, AreaByHeightWidth, AreaByLongLatBounds or AreaByPositions.	•	•
ImageSettings	Core.Map.Types.ImageSettings	Settings of the image representing the rendered maps.	•	•
Shapes	Common.Types.Shape[]	A collection of shapes to be drawn over the maps to render.	•	•
Style	Core.Map.Types.MapStyle	Defines the style of the maps to render.	•	•

## 8.4. MapResult class

Holds the rendered map and some extra informations about it. The map can be represented by a Buffer, an ID or an URL, depending on the ImageResultTypes set in ImageSettings.

Properties				
Property	Type	Description	Get	Set
AreaRepresentations	Common.Types.AreaRepresentations	Holds different Area types representing the resulting map. They can be used to generate a new map with a new logical or physical scale for instance.	•	•
ImageBuffer	byte[]	Buffer containing the image of the rendered map. It is set when ImageSettings.ResultType is set to ImageBuffer.	•	•
ImageId	string	ID of the image map to render, it can be used with MapService.DrawMapFromId(). It is set when ImageSettings.ResultType is set to ImageUrl.	•	•
ImageUrl	string	URL pointing to the image map to render. It is set when ImageSettings.ResultType is set to ImageUrl.	•	•
MapInfo	Common.Types.MapInfo	Holds some information about the	•	•

		resulting map.		
MimeType	string	The MIME Type for the format of the resulting map.	•	•
Warnings	string[]	May contain a collection of warning messages.	•	•

## 8.5. MapsParameters class

Parameters to render multiple maps.

### Inherits :

Core.Map.Types.MapsParameters

Core.Map.Types.BaseMapParameters

Properties				
Property	Type	Description	Get	Set
Areas	Common.Types.Area[]	A collection of Areas to be represented by a map. An Area can be of type AreaByPhysicalScale, AreaByLogicalScale, AreaByHeightWidth, AreaByLongLatBounds or AreaByPositions.	•	•
ImageSettings	Core.Map.Types.ImageSettings	Settings of the image representing the rendered maps.	•	•
Shapes	Common.Types.Shape[]	A collection of shapes to be drawn over the maps to render.	•	•
Style	Core.Map.Types.MapStyle	Defines the style of the maps to render.	•	•

## 8.6. MapStyle class

Defines the Style of a Map to render.

Properties				
Property	Type	Description	Get	Set
Angle	Core.Map.Types.RotateAngles	Rotate the map to the given angle. Useful for fax/print.	•	•
DisableAntialias	boolean	Toggle anti-aliasing.	•	•
DisplayTheme	string	Display Theme. Selects a predefined set of points of interest categories displayed on the map (see appendix for a listing of available themes).	•	•
GrayScale	boolean	Flag to reduce the colors to obtain a gray scale map. Useful for fax/print.	•	•

HideCopyright	boolean	Show/Hide the copyrights at the bottom left of the map.	•	•
HideScale	boolean	Show/Hide the scale at the bottom right of the map.	•	•
Language	int	Language Resource Index. Belgium: 0=French, 1=Flemish. Greece: 0=Latin, 1=Greek. Czech Republic: 0=Latin, 1=Czech.	•	•
Layers	Core.Map.Types.LayerFlags	Allow to activate specific layers to be rendered or not. If the surfaces are not rendered, the background will be transparent when the ImageType is set to PNG.	•	•
MapTemplate	string	Map Template. Selects a predefined set of colors and fonts for the map (see appendix for a listing of available templates).	•	•
Projection	Common.Types.Projection	A specific cartographic projection may be set with this property, by default the World Mercator projection is used.	•	•
ScaleUnit	Common.Types.UnitSystems	Unit system to be used to render the scale (Metric, Imperial or both).	•	•

## 8.7. ImageResultTypes enumeration

Enumeration of the different result type when generating a Map.

Fields	
Field	Description
Buffer	The actual buffer of the image representing the Map is returned. The corresponding result variable is the byte array <code>MapResult.ImageBuffer</code> .
Id	An ID which can be used with <code>MapService.DrawMapFromId()</code> . The Map is actually rendered only when <code>MapService.DrawMapFromId()</code> is called. The corresponding result variable is the string <code>MapResult.ImageId</code> .
Url	Returns an URL to the Map to generate. The Map is actually rendered only when the URL is navigated. The corresponding result variable is the string <code>MapResult.ImageUrl</code> .

## 8.8. LayerFlags enumeration

Flags use to select which layer to render on a map.

Fields	
Field	Description
All	Everything.
Default	Default filter: Geometry and Texts.
Geometry	Geometry (include Linears, Surfaces and Symbols).
Linears	Linear elements: roads, rivers... This flag is active with the Geometry flag.
None	No layer.
Surfaces	Surface elements: islands, buildings... This flag is active with the Geometry flag.
Symbols	Symbols/POIs. This flag is active with the Geometry flag.
Text	Texts and Labels.
Traffic	Real Time Traffic information (if the information is available for the requested map).

## 8.9. RotateAngles enumeration

Angles that can be used to rotate a map.

Fields	
Field	Description
Rotate0	No rotation.
Rotate180	Rotation of 180 degrees.
Rotate270	Rotation of 270 degrees.
Rotate90	Rotation of 90 degrees.



## 9. Core.Poi.Operations namespace

### 9.1. PoiService class

The Maporama Poi Web Service offers the method to perform Lookup operations.

Constuctors	
Constructor	Description
PoiService( )	Initializes a new instance of the Core.Poi.Operations.PoiService class.

#### 9.1.1. CountLookAlong( LookAlongParameters lookAlongParameters) method

Count the number of POI available for given search along parameters.

Parameters			
Parameter	Type	Direction	Description
lookAlongParamet ers	Core.Poi.Types.LookAlongParamet ers		

  

Returns	
Type	Description
int	

#### 9.1.2. CountLookAround( LookAroundParameters lookAroundParameters) method

Count the number of POI available for given search around parameters.

Parameters			
Parameter	Type	Direction	Description
lookAroundParame ters	Core.Poi.Types.LookAroundParam eters		

  

Returns	
Type	Description
int	

#### 9.1.3. CountLookInside( LookInsideParameters lookInsideParameters) method

Count the number of POI available for given search inside parameters.

Parameters			
Parameter	Type	Direction	Description
lookInsideParameters	Core.Poi.Types.LookInsideParameters		

  

Returns	
Type	Description
int	

#### 9.1.4. LookAlong( LookAlongParameters lookAlongParameters) method

Look up for POI for given search along parameters.

Parameters			
Parameter	Type	Direction	Description
lookAlongParameters	Core.Poi.Types.LookAlongParameters		

  

Returns	
Type	Description
string	

#### 9.1.5. LookAround( LookAroundParameters lookAroundParameters) method

Look up for POI for given search around parameters.

Parameters			
Parameter	Type	Direction	Description
lookAroundParameters	Core.Poi.Types.LookAroundParameters		

  

Returns	
Type	Description
string	

#### 9.1.6. LookAroundDefault( double longitude,double latitude,int radius) method

Look up for POI around a given location using default parameters.

Parameters			

<b>Parameter</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
latitude	double		
longitude	double		
radius	int		

Returns	
Type	Description
string	

9.1.7. LookInside( LookInsideParameters lookInsideParameters) method

Look up for POI for given search inside parameters.

Parameters			
Parameter	Type	Direction	Description
lookInsideParameters	Core.Poi.Types.LookInsideParameters		

Returns	
Type	Description
string	

## 10. Core.Poi.Types namespace

### 10.1. LookAlongParameters class

Contains parameters to process LookAlong a polyline for a given distance in POI database

#### **Inherits :**

Core.Poi.Types.LookAlongParameters

Core.Poi.Types.LookupParameters

Properties				
Property	Type	Description	Get	Set
Constraint	Core.Poi.Types.LookupConstra inst	Constraint to restrict POI research	•	•
Distance	Common.Types.Distance	Search distance from the polyline Note: You can only use Meters, Kilometers or Miles	•	•
NbMaxResult	int	Number of result to return	•	•
Offset	int	Max distance to search into in meters	•	•
Polyline	Common.Types.Polyline	Polyline representing the line of the search	•	•

### 10.2. LookAroundParameters class

Contains parameters to process LookAround a center in POI database

#### **Inherits :**

Core.Poi.Types.LookAroundParameters

Core.Poi.Types.LookupParameters

Properties				
Property	Type	Description	Get	Set
Constraint	Core.Poi.Types.LookupConstra inst	Constraint to restrict POI research	•	•
NbMaxResult	int	Number of result to return	•	•
Offset	int	Max distance to search into in meters	•	•
Position	Common.Types.LongLatPositio n	Position representing the origin (center) of the search	•	•
		Search radius from the origin point		

Radius	Common.Types.Distance	Note: You can only use Meters, Kilometers or Miles	.	.
--------	-----------------------	---	---	---

### 10.3. LookInsideParameters class

Contains parameters to process LookInside a polygon in POI database

#### Inherits :

Core.Poi.Types.LookInsideParameters

Core.Poi.Types.LookupParameters

Properties				
Property	Type	Description	Get	Set
Constraint	Core.Poi.Types.LookupConstra inst	Constraint to restrict POI research	•	•
NbMaxResult	int	Number of result to return	•	•
Offset	int	Max distance to search into in meters	•	•
Polygon	Common.Types.Polygon	Positions representing the polygon to search	•	•

### 10.4. LookupConstraint class

Contains word or category collections to allow or reject in your search

Properties				
Property	Type	Description	Get	Set
CategoriesBan	System.Collections.Generic.Lis t{Common.Types.Category}	Category collection to ban in your search	•	•
CategoriesMatch	System.Collections.Generic.Lis t{Common.Types.Category}	Category collection to match in your search	•	•
WordsBan	string[]	Word collection to ban in your search	•	•
WordsMatch	string[]	Word collection to match in your search	•	•

Constructors	
Constructor	Description

LookupConstrainst( )	Initializes a new instance of the Core.Poi.Types.LookupConstrainst class.
----------------------	---



## 10.5. LookupParameters class

Base parameters, defines parameters to perform a POILookUp operation.

Properties				
Property	Type	Description	Get	Set
Constraint	Core.Poi.Types.LookupConstrains	Constraint to restrict POI research	•	•
NbMaxResult	int	Number of result to return	•	•
Offset	int	Max distance to search into in meters	•	•

## 11. Extensions.Weather.Operations namespace

### 11.1. WeatherService class

The Maporama Weather Web Service exposes the method to forecast weather for 6 days and to render or prepare map using weather layer.

Constructors	
Constructor	Description
WeatherService( )	Initializes a new instance of the Extensions.Weather.Operations.WeatherService class.

#### 11.1.1. WeatherForecast( ForecastParameters forecastParameters) method

Forecast the weather summary for next 6 days for a given parameters.

Parameters			
Parameter	Type	Direction	Description
forecastParameters	Extensions.Weather.Types.ForecastParameters		Parameters used to set position, language and unit.

  

Returns	
Type	Description
Extensions.Weather.Types.ForecastResults	The weather forecast summary.

#### 11.1.2. WeatherLayer( LayerParameters layerParameters) method

Render or prepare the map using the requested layer by given parameters.

Parameters			
Parameter	Type	Direction	Description
layerParameters	Extensions.Weather.Types.LayerParameters		Parameters used to set map settings et layer information.

  

Returns	
Type	Description
Core.Map.Types.MapResult	A MapResult object, with the URL or the Image Buffer and other extra information.

### 11.1.3. WeatherLayerFromId( string layerId) method

Render a map layered from a stored WeatherLayer Request represented by its ID.

Parameters			
Parameter	Type	Direction	Description
layerId	string		A layer ID.

  

Returns	
Type	Description
Core.Map.Types.MapResult	The Image Buffer of the rendered Map Layer.

## 12. Extensions.Weather.Types namespace

### 12.1. ForecastParameters class

Parameters to set weather forecast.

Properties				
Property	Type	Description	Get	Set
Language	string	Language for result. Available languages are : 'US', 'FR', 'ES' and 'BR'.	•	•
Position	Common.Types.LongLatPosition	WGS84 Coordinates of the requested point.	•	•
Unit	Common.Types.UnitSystems	Forecast unit used: imperial or metric.	•	•

### 12.2. ForecastResults class

Describes weather forecast results.

Properties				
Property	Type	Description	Get	Set
Altitude	int	Altitude in meters of the requested point.	•	•
ForecastSummaryCollection	Extensions.Weather.Types.ForecastSummary[]	Summary of weather forecast by day.	•	•

### 12.3. ForecastSummary class

Describes weather forecast for one day.

Properties				
Property	Type	Description	Get	Set
Date	Common.Types.Date	Date of weather forecast.	•	•
Description	string	Weather description of current date.	•	•
Humidity	float	Relative humidity percent of current	•	•

		date.		
IconId	int	Id of the icon for the weather of current date.	•	•
IconImage	byte[]	Image of icon for the weather of current date.	•	•
PrecipitationProbability	float	Precipitation probability percent of current date.	•	•
RainAmount	float	Amount of rain of current date.	•	•
ShortDescription	string	Weather short description of current date.	•	•
TemperatureMax	short	Temperature maximum of current date.	•	•
TemperatureMin	short	Temperature minimum of current date.	•	•
WindChill	short	Felt chill of the wind of current date.	•	•
WindDirection	string	Direction of the wind of current date.	•	•
WindSpeed	short	Speed of the wind of current date.	•	•

## 12.4. LayerParameters class

Weather Layer Parameters.

### Inherits :

Extensions.Weather.Types.LayerParameters

Extensions.Weather.Types.ForecastParameters

Properties				
Property	Type	Description	Get	Set
Day	int	Day to display: 1(today), 2, 3, 4 or 5	•	•
ImageSettings	Core.Map.Types.ImageSettings	Settings of the image representing the rendered maps.	•	•
Language	string	Language for result. Available languages are : 'US', 'FR', 'ES' and 'BR'.	•	•
Layer	Extensions.Weather.Types.TypeLayer	Weather layer requested	•	•
Opacity	int	Layer opacity in percent	•	•
Position	Common.Types.LongLatPosition	WGS84 Coordinates of the requested point.	•	•
Unit	Common.Types.UnitSystems	Forecast unit used: imperial or metric.	•	•

## 12.5. TypeLayer enumeration

Defines the layer to display.

Fields	
Field	Description
CloudCover	Display the cloud cover layer
CloudCoverAndPressure	Display CloudCover and Pressure on the same layer
CloudCoverAndWind	Display CloudCover and Wind on the same layer
Humidity	Display a humidity layer
HumidityAndPressure	Display Humidity and Pressure on the same layer
HumidityAndWind	Display Humidity and Wind on the same layer

MaxTemp	Display a max temperature layer
MaxTempAndPressure	Display MaxTemp and Pressure on the same layer
MaxTempAndWind	Display MaxTemp and Wind on the same layer
MinTemp	Display a min temperature layer
MinTempAndPressure	Display MinTemp and Pressure on the same layer
MinTempAndWind	Display MinTemp and Wind on the same layer
None	Default layer, display nothing
Pressure	Display the pressure layer
RainAmount	Display a rain amount layer
RainAmountAndPressure	Display RainAmount and Pressure on the same layer
RainAmountAndWind	Display RainAmount and Wind on the same layer
RainProbability	Display a rain probability layer
RainProbabilityAndPressure	Display RainProbability and Pressure on the same layer
RainProbabilityAndWind	Display RainProbability and Wind on the same layer
SeaTemp	Display the sea temperature layer, center must not be to far from coast
WaveDirection	Display the wave direction layer, center must not be to far from coast
WaveHeight	Display the wave height layer, center must not be to far from coast
Wind	Display the wind layer





## 13. Partner.Operations namespace

### 13.1. DatabaseService class

The Database Web Service exposes the methods to access and manage the items in customer database.

#### Inherits :

Partner.Operations.DatabaseService

Partner.Operations.PartnerService

Maporama.WebServices.Base.WebServiceBase

Properties				
Property	Type	Description	Get	Set
CustomerId	System.Guid		•	
SpatialAccess	boolean		•	•

Fields			
Field	Type	Description	Static
_clientPartner	Maporama.Business.Partner.WebClient.PartnerClient		

Constructors	
Constructor	Description
DatabaseService( )	Initializes a new instance of the Partner.Operations.DatabaseService class.

#### 13.1.1. CountRead( ReadParameters readParameters) method

Counts the number of items corresponding to requested read parameters

Parameters			
Parameter	Type	Direction	Description
readParameters	Partner.Types.Database.ReadParameters		the read parameters

Returns	
Type	Description
int	number of items

### 13.1.2. GetCoordFields( string tablename) method

retrieves the field name of longitude and latitude column name of param table.

Parameters			
Parameter	Type	Direction	Description
tablename	string		name of the table

  

Returns	
Type	Description
Partner.Types.CoordFields	Coordinate fields name

### 13.1.3. GetPrimaryKey( string tablename) method

retrieves the field name of the primary key of param table.

Parameters			
Parameter	Type	Direction	Description
tablename	string		name of the table

  

Returns	
Type	Description
string	primary key field name

### 13.1.4. GetSchema( string tablename) method

retrieves the schema of param table. Note: the schema is a XML object written in a string.

Parameters			
Parameter	Type	Direction	Description
tablename	string		name of the table

  

Returns	
Type	Description
string	xml schema

### 13.1.5. GetTables( ) method

retrieves all tables in database.

Returns	
Type	Description
string[]	collection of tablename

### 13.1.6. Read( ReadParameters readParameters) method

Reads the items in database following read parameters. Note: Tablename is mandatory in parameter.

Parameters			
Parameter	Type	Direction	Description
readParameters	Partner.Types.Database.ReadParameters		the read parameters

  

Returns	
Type	Description
Partner.Types.Database.ReadResults	the XML string of result

### 13.1.7. ReadDefault( ) method

Reads the ten first items in the first table found in database.

Returns	
Type	Description
string	the XML string of result

## 13.2. PartnerService class

Properties				
Property	Type	Description	Get	Set
CustomerId	System.Guid		•	
SpatialAccess	boolean		•	•

Fields			
Field	Type	Description	Static
_clientPartner	Maporama.Business.Partner.WebClient.PartnerClient		

Constructors	

<b>Constructor</b>	<b>Description</b>
PartnerService( )	Initializes a new instance of the Partner.Operations.PartnerService class.

### 13.3. ProximityService class

The Proximity Web Service exposes the methods to perform proximity search of the items in customer database.

#### Inherits :

Partner.Operations.ProximityService

Partner.Operations.PartnerService

Maporama.WebServices.Base.WebServiceBase

Properties				
Property	Type	Description	Get	Set
CustomerId	System.Guid		•	
SpatialAccess	boolean		•	•

Fields			
Field	Type	Description	Static
_clientPartner	Maporama.Business.Partner.WebClient.PartnerClient		

Constructors	
Constructor	Description
ProximityService( )	Initializes a new instance of the Partner.Operations.ProximityService class.

#### 13.3.1. CountSearchAround( ProximityParameters proximityParameters) method

Counts the number of items corresponding to requested proximity parameters.

Parameters			
Parameter	Type	Direction	Description
proximityParameters	Partner.Types.Proximity.ProximityParameters		The proximity parameters.

Returns	
Type	Description
int	Number of items.

### 13.3.2. GetCoordFields( string tablename) method

retrieves the field name of longitude and latitude column name of param table.

Parameters			
Parameter	Type	Direction	Description
tablename	string		name of the table

  

Returns	
Type	Description
Partner.Types.CoordFields	Coordinate fields name

### 13.3.3. GetPrimaryKey( string tablename) method

Retrieves the field name of the primary key of param table.

Parameters			
Parameter	Type	Direction	Description
tablename	string		Name of the table.

  

Returns	
Type	Description
string	Primary key field name.

### 13.3.4. GetSchema( string tablename) method

Retrieves the schema of param table. Note: the schema is a XML object written in a string.

Parameters			
Parameter	Type	Direction	Description
tablename	string		Name of the table.

  

Returns	
Type	Description
string	XML Schema.



### 13.3.5. GetTables( ) method

Retrieves all tables in database.

Returns	
Type	Description
string[]	Collection of tablename.

### 13.3.6. SearchAround( ProximityParameters proximityParameters) method

Searches the items in database following proximity parameters. Note: Tablename, position and distance is mandatory in parameter.

Parameters			
Parameter	Type	Direction	Description
proximityParameters	Partner.Types.Proximity.ProximityParameters		The proximity parameters.

  

Returns	
Type	Description
Partner.Types.Proximity.ProximityResults	The XML string of result.

### 13.3.7. SearchAroundDefault( double longitude,double latitude,int radius) method

Searches ten first items (if any) in database around on the given location center inside the specified radius.

Parameters			
Parameter	Type	Direction	Description
latitude	double		Latitude of the center point to search around.
longitude	double		Longitude of the center point to search around.
radius	int		Radius in meters to search in.

  

Returns	
Type	Description
string	The XML string of result.

## 13.4. SearchService class

The Search Web Service exposes the methods to perform advanced spatial search of the items in customer database.

**Inherits :**

Partner.Operations.SearchService

Partner.Operations.PartnerService

Maporama.WebServices.Base.WebServiceBase

Properties				
Property	Type	Description	Get	Set
CustomerId	System.Guid		•	
SpatialAccess	boolean		•	•

Fields			
Field	Type	Description	Static
_clientPartner	Maporama.Business.Partner.WebClient.PartnerClient		

Constructors	
Constructor	Description
SearchService( )	Initializes a new instance of the Partner.Operations.SearchService class.

13.4.1. CountSearchAlong( SearchAlongParameters searchAlongParameters) method

Counts the number of items corresponding to requested given parameters.

Parameters			
Parameter	Type	Direction	Description
searchAlongParameters	Partner.Types.Search.SearchAlongParameters		search along parameters.

Returns	
Type	Description
int	The XML string of result.

### 13.4.2. CountSearchAround( SearchAroundParameters searchAroundParameters) method

Counts the number of items corresponding to requested given parameters.

Parameters			
Parameter	Type	Direction	Description
searchAroundParameters	Partner.Types.Search.SearchAroundParameters		search around parameters.

  

Returns	
Type	Description
int	The XML string of result.

### 13.4.3. CountSearchInside( SearchInsideParameters searchInsideParameters) method

Counts the number of items corresponding to requested given parameters.

Parameters			
Parameter	Type	Direction	Description
searchInsideParameters	Partner.Types.Search.SearchInsideParameters		search inside parameters.

  

Returns	
Type	Description
int	The XML string of result.

### 13.4.4. SearchAlong( SearchAlongParameters searchAlongParameters) method

Searches the items in database following search along parameters. Note: Tablename, polyline and distance are mandatory in parameter.

Parameters			
Parameter	Type	Direction	Description
searchAlongParameters	Partner.Types.Search.SearchAlongParameters		search along parameters.

  

Returns	
Type	Description
Partner.Types.Search.SearchAlongResults	The XML string of result.

### 13.4.5. SearchAround( SearchAroundParameters searchAroundParameters) method

Searches the items in database following search around parameters. Note: Tablename, position and distance are mandatory in parameter.

Parameters			
Parameter	Type	Direction	Description
searchAroundParameters	Partner.Types.Search.SearchAroundParameters		search around parameters.

  

Returns	
Type	Description
Partner.Types.Search.SearchAroundResults	The XML string of result.

### 13.4.6. SearchInside( SearchInsideParameters searchInsideParameters) method

Searches the items in database following search inside parameters. Note: Tablename and polygon are mandatory in parameter.

Parameters			
Parameter	Type	Direction	Description
searchInsideParameters	Partner.Types.Search.SearchInsideParameters		search inside parameters.

  

Returns	
Type	Description
Partner.Types.Search.SearchInsideResults	The XML string of result.

## 13.5. SpatialAccessExtension class

Defines authorization to allow access to spatial data in Customer Databases

Constructors	
Constructor	Description
SpatialAccessExtension( )	Initializes a new instance of the Partner.Operations.SpatialAccessExtension class.

### 13.5.1. GetInitializer( Type serviceType) method

### Overridden initializer

Parameters			
Parameter	Type	Direction	Description
serviceType	System.Type		

  

Returns	
Type	Description
System.Object	

### 13.5.2. GetInitializer( LogicalMethodInfo methodInfo, SoapExtensionAttribute attribute) method

#### Overridden initializer

Parameters			
Parameter	Type	Direction	Description
attribute	System.Web.Services.Protocols.SoapExtensionAttribute		
methodInfo	System.Web.Services.Protocols.LogicalMethodInfo		

  

Returns	
Type	Description
System.Object	

### 13.5.3. Initialize( Object initializer) method

#### Initialize method set config

Parameters			
Parameter	Type	Direction	Description
initializer	System.Object		

  

Returns	
Type	Description
System.Void	

### 13.5.4. ProcessMessage( SoapMessage message) method

#### Check authorization on AfterDeserialize Stage

Parameters			
------------	--	--	--

Parameter	Type	Direction	Description
message	System.Web.Services.Protocols.SoapMessage		

Returns	
---------	--

Type	Description
System.Void	

## 14. Partner.Operations.Configuration namespace

### 14.1. IpElement class

Defines the Authorized IP addresses.

Properties				
Property	Type	Description	Get	Set
Address	string	IP address or IP set using '*' (e.g: 192.168.20.*).	•	•

Constructors	
Constructor	Description
IpElement( )	Initializes a new instance of the Partner.Operations.Configuration.IpElement class.

### 14.2. IpElementCollection class

Collection of authorized addresses allowed to access spatial data.

Properties				
Property	Type	Description	Get	Set
Item	Partner.Operations.Configuration.IpElement	Returns the IpElement element for a given address.	•	

Constructors	
Constructor	Description
IpElementCollection( )	Default constructor.

### 14.3. SpatialAccessSection class

Properties				
------------	--	--	--	--

Property	Type	Description	Get	Set
AllowedAddresses	Partner.Operations.Configuration.IpElementCollection	Collection of Authorized IP address.	•	



Fields			
Field	Type	Description	Static
SECTION_NAME	string	the value of 'spatialAccess'	•

Constructors	
Constructor	Description
SpatialAccessSection( )	Initializes a new instance of the Partner.Operations.Configuration.SpatialAccessSection class.

## 15. Partner.Types namespace

### 15.1. CoordFields class

Coordinate fields for data of PartnerBase

Properties				
Property	Type	Description	Get	Set
Latitude	string	Latitude column name	•	•
Longitude	string	Longitude column name	•	•

Constructors	
Constructor	Description
CoordFields()	Initializes a new instance of the Partner.Types.CoordFields class.

### 15.2. PartnerParameters class

Parameters for Partner request to PartnerBase.

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•
Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Tablename	string	Partner table name.	•	•

### 15.3. PartnerResults class

Results of Partner request to PartnerBase.

Properties				
Property	Type	Description	Get	Set
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial. An Area can be used to render a map for instance.	•	•
XmlResult	string	Items of the PartnerBase in XML format.	•	•

## 16. Partner.Types.Database namespace

### 16.1. ReadParameters class

Parameters to set read access to PartnerBase.

#### **Inherits :**

Partner.Types.Database.ReadParameters

Partner.Types.PartnerParameters

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•
Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Tablename	string	Partner table name.	•	•

### 16.2. ReadResults class

Results of read access to PartnerBase.

#### **Inherits :**

Partner.Types.Database.ReadResults

Partner.Types.PartnerResults

Properties				
Property	Type	Description	Get	Set
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial. An Area can be used to render a map for instance.	•	•

XmlResult	string	Items of the PartnerBase in XML format.	.	.
-----------	--------	---	---	---



## 17. Partner.Types.Proximity namespace

### 17.1. ProximityParameters class

Parameters to set proximity search to PartnerBase.

#### **Inherits :**

Partner.Types.Proximity.ProximityParameters

Partner.Types.PartnerParameters

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•
Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Position	Common.Types.LongLatPosition	Position representing the origin (center) of the search	•	•
Radius	Common.Types.MetersDistance	Search radius from the origin point Note: You can only use Meters, Kilometers or Miles	•	•
Tablename	string	Partner table name.	•	•

### 17.2. ProximityResults class

Results of proximity search in PartnerBase.

#### **Inherits :**

Partner.Types.Proximity.ProximityResults

Partner.Types.PartnerResults

Properties				
Property	Type	Description	Get	

				<b>Set</b>
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial. An Area can be used to render a map for instance.	•	•
XmlResult	string	Items of the PartnerBase in XML format.	•	•





## 18. Partner.Types.Search namespace

### 18.1. SearchAlongParameters class

Parameters to set search along into PartnerBase.

#### **Inherits :**

Partner.Types.Search.SearchAlongParameters

Partner.Types.PartnerParameters

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
Distance	Common.Types.MetersDistance	Distance from the polyline Note: You can only use Meters, Kilometers or Miles	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•
Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Polyline	Common.Types.Polyline	Polyline representing the positions along the search	•	•
Tablename	string	Partner table name.	•	•

Constuctors	
Constructor	Description
SearchAlongParameters( )	Initializes a new instance of the Partner.Types.Search.SearchAlongParameters class.

### 18.2. SearchAlongResults class

Results of search along in PartnerBase.

**Inherits :**

Partner.Types.Search.SearchAlongResults

Partner.Types.PartnerResults

Properties				
Property	Type	Description	Get	Set
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial. An Area can be used to render a map for instance.	•	•
XmlResult	string	Items of the PartnerBase in XML format.	•	•

Constructors	
Constructor	Description
SearchAlongResults( )	Initializes a new instance of the Partner.Types.Search.SearchAlongResults class.

**18.3. SearchAroundParameters class**

Parameters to set search around into PartnerBase.

**Inherits :**

Partner.Types.Search.SearchAroundParameters

Partner.Types.PartnerParameters

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•

Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Position	Common.Types.LongLatPosition	Position representing the origin (center) of the search	•	•
Radius	Common.Types.MetersDistance	Search radius from the origin point Note: You can only use Meters, Kilometers or Miles	•	•
Tablename	string	Partner table name.	•	•

Constuctors	
Constructor	Description
SearchAroundParameters( )	Initializes a new instance of the Partner.Types.Search.SearchAroundParameters class.

## 18.4. SearchAroundResults class

Results of search around in PartnerBase.

### Inherits :

Partner.Types.Search.SearchAroundResults

Partner.Types.PartnerResults

Properties				
Property	Type	Description	Get	Set
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial. An Area can be used to render a map for instance.	•	•
XmlResult	string	Items of the PartnerBase in XML format.	•	•

Constuctors	
Constructor	Description
SearchAroundResults( )	Initializes a new instance of the Partner.Types.Search.SearchAroundResults class.

## 18.5. SearchInsideParameters class

Parameters to set search inside into PartnerBase.

### Inherits :

Partner.Types.Search.SearchInsideParameters

Partner.Types.PartnerParameters

Properties				
Property	Type	Description	Get	Set
Clause	string	Restricted clause to retrieve items.	•	•
DisplayFields	string[]	Partner table fields to retrieve, set to null for all.	•	•
GetAreaRepresentation	boolean	Set true, to add an area representation in the response of requested parameters. If chosen table is not spatial, AreaRepresentation in response is null.	•	•
Nbltems	int	Number of items to retrieve.	•	•
Offset	int	Start offset to retrieve.	•	•
OrderFields	string[]	Partner table fields order to sort retrieved values, set to null to inactive.	•	•
Polygon	Common.Types.Polygon	Polygon representing the positions inside to search	•	•
Tablename	string	Partner table name.	•	•

Constuctors	
Constructor	Description
SearchInsideParameters( )	Initializes a new instance of the Partner.Types.Search.SearchInsideParameters class.

## 18.6. SearchInsideResults class

Results of search inside in PartnerBase.

### Inherits :

Partner.Types.Search.SearchInsideResults

Partner.Types.PartnerResults

Properties				
Property	Type	Description	Get	Set
AreaByPositions	Common.Types.AreaByPositions	Area Representation of this Results. It defines the positions of the returned items if the requested table is spatial.	•	•

		An Area can be used to render a map for instance.		
XmlResult	string	Items of the PartnerBase in XML format.	•	•

<b>Constutors</b>	
<b>Constructor</b>	<b>Description</b>
SearchInsideResults( )	Initializes a new instance of the Partner.Types.Search.SearchInsideResults class.