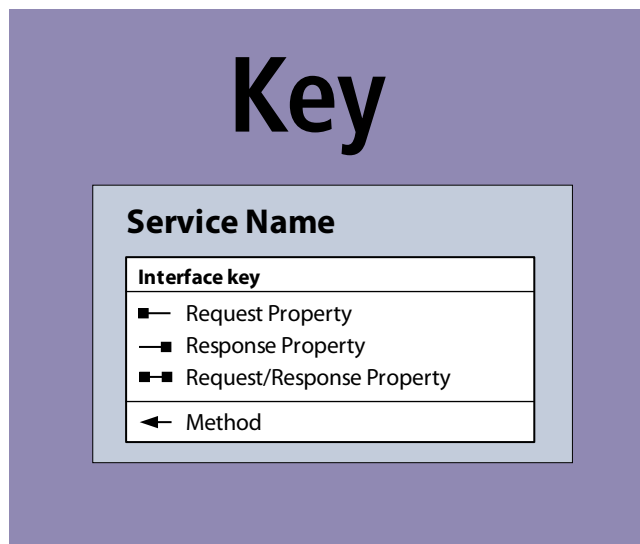


ArcWebSM Services 2006 SOAP Object Model



www.esri.com/arcwebservice
800-447-9778

Copyright © 2006 ESRI. All rights reserved. ESRI, the ESRI globe logo, ArcWeb, and www.esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Common Objects

Common	
AttributeDesc <ul style="list-style-type: none">— length: integer— name: string— precision: integer— required: string— shortDesc: string— type: string	KeyValue <ul style="list-style-type: none">■ key: string■ value: string
Circle <ul style="list-style-type: none">■ center: Point■ radius: double■ radiusUnits: string■ trueCircle: boolean	MapImageSize <ul style="list-style-type: none">■ height: integer■ width: integer
CoordSys <ul style="list-style-type: none">■ datumTransformation: string■ projection: string	OrderBy <ul style="list-style-type: none">■ descending: boolean■ field: string
DateTimeRange <ul style="list-style-type: none">■ end: long■ start: long	Point <ul style="list-style-type: none">■ coordSys: CoordSys■ x: double■ y: double
Envelope <ul style="list-style-type: none">■ coordSys: CoordSys■ maxX: double■ maxY: double■ minX: double■ minY: double	ResultSet <ul style="list-style-type: none">— fields: FieldDesc[]— rows: RowData[]— totalCount: integer
FieldDesc <ul style="list-style-type: none">— length: integer— name: string— precision: integer— shortDesc: string— type: string	ResultSetRange <ul style="list-style-type: none">■ count: integer■ startIndex: integer
Geometry <ul style="list-style-type: none">■ coordSys: CoordSys■ parts: integer[]■ points: SimplePoint[]■ type: integer	RowData <ul style="list-style-type: none">— fieldValues: string[]
	SearchOptions <ul style="list-style-type: none">■ orderByList: OrderBy[]■ resultSetRange: ResultSetRange■ returnFields: string[]■ whereClause: string
	SimplePoint <ul style="list-style-type: none">■ x: double■ y: double
	SpatialBuffer <ul style="list-style-type: none">■ bufferDistance: double■ bufferUnits: string

Data Management

Data Manager Service

DataManager

- ← addFeatures(
 - dataFileName: string,
 - features: Feature[],
 - rollbackOnError: boolean,
 - token:string): FeatureIDInfo[]
- ← createDataFile(
 - dataFile: DataFile,
 - token: string): void
- ← deleteDataFile(
 - dataFileName: string,
 - token:string): void
- ← deleteFeatures(
 - dataFileName: string,
 - featureIDs: string[],
 - token:string): void
- ← findFeatures(
 - dataFileName: string,
 - featureSearchOptions: FeatureSearchOptions,
 - token:string): FeatureResultSet
- ← geocodeFeatures(
 - dataFileName: string,
 - featureIDs: string[],
 - geocodeOptions: GeocodeOptions,
 - token:string): GeocodeInfo[]
- ← getAvailableDataFileAttributes (
 - token: string): AttributeDesc[]
- ← getDataFilesInfo(
 - dataFileNames: string[],
 - token:string): DataFileInfo[]
- ← getUniqueValues(
 - dataFileName: string,
 - fieldName: string,
 - resultSetRange: ResultSetRange,
 - token:string): StringSet
- ← getVersion: string
- ← renameField(
 - dataFileName: string,
 - oldFieldName: string,
 - newFieldName: string,
 - token:string): void
- ← updateDataFile(
 - dataFileName: string,
 - dataFile: DataFile,
 - token:string): void
- ← updateFeatures(
 - dataFileName: string,
 - featureUpdateOptions: FeatureUpdateOptions,
 - featureIDs: string[],
 - token:string): void

AddressFields

- city: string
- country: string
- houseNumber: string
- intersection: string
- postalCode: string
- stateProvince: string
- street: string

DataFile

- addressFields: AddressFields
- attributes: KeyValue[]
- coordSys: CoordSys
- fieldInfo: FieldInfo[]
- geometryType: integer
- name: string

DataFileInfo

- dataFile: DataFile
- dateCreated: long
- extent: Envelope
- featureCount: integer
- geographicExtent: Envelope
- lastModified: long
- maxScale: long
- minScale: long
- size: long

Feature

- fieldValues: KeyValue[]
- geometry: Geometry

FeatureIDInfo

- errorMessage: string
- id: string

FeatureInfo

- dateCreated: long
- feature: Feature
- geocodeDataSource: string
- geocodeStatus: string
- id: string
- lastModified: long

FeatureResultSet

- features: FeatureInfo[]
- totalCount: integer

FeatureSearchOptions

- intersectGeometry: Geometry
- returnGeometry: boolean
- searchOptions: SearchOptions
- spatialBuffer: SpatialBuffer

FeatureUpdateOptions

- fieldValues: KeyValue[]
- geometry: Geometry

FieldInfo

- isIndexed: boolean
- label: string
- name: string
- type: integer

GeocodeCandidate

- address: Address
- desc1: string
- desc2: string
- extent: Envelope
- matchType: string
- point: Point
- score: double
- type: string

GeocodeInfo

- candidates: GeocodeCandidate[]
- errorMessage: string
- hasMore: boolean
- matchType: string
- totalCount: integer

GeocodeOptions

- dataSource: string
- extendedPostalCode: boolean
- formatAddressFields: boolean
- partialAddress: boolean

StringSet

- strings: string[]
- totalCount: integer

Find

Common

Address

- city: string
- country: string
- houseNumber: string
- intersection: string
- postalCode: string
- stateProvince: string
- street: string

AddressFinderInfo

- countryCodes: KeyValue[]
- dataSource: string
- extendedPostalCode: boolean
- partialAddress: boolean
- reverseGeocode: boolean
- snapTypes: string[]

AddressFinderOptions

- dataSource: string
- extendedPostalCode: boolean
- partialAddress: boolean
- resultSetRange: ResultSetRange
- snapType: string

GeocodeCandidate

- address: Address
- desc1: string
- desc2: string
- extent: Envelope
- matchType: string
- point: Point
- score: double
- type: string

GeocodeInfo

- candidates: GeocodeCandidate[]
- errorMessage: string
- hasMore: boolean
- matchType: string
- totalCount: integer

PlaceFinderInfo

- countryCodes: KeyValue[]
- dataSource: string
- types: KeyValue[]

PlaceFinderOptions

- dataSource: string
- filterCountry: string
- filterExtent: Envelope
- filterType: string
- resultSetRange: ResultSetRange
- searchType: string

Address Finder Service

AddressFinder

- ← findAddressByPoint(
point: Point,
addressFinderOptions:
AddressFinderOptions,
token: string): Address
- ← findLocationByAddress(
address: Address,
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo
- ← findLocationByIP(
ipAddress: string,
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo
- ← findLocationByPhoneNumber(
phoneNumber: string,
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo
- ← getInfo(
dataSources: string[],
token: string): AddressFinderInfo[]
- ← getVersion: string

Address Manager Service

AddressManager

- ← findAddressesByPoint(
points: Point[],
addressFinderOptions:
AddressFinderOptions,
token: string): AddressInfo[]
- ← findLocationsByAddress(
addresses: Address[],
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo[]
- ← findLocationsByIP(
ipAddresses: string[],
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo[]
- ← findLocationsByPhoneNumber(
phoneNumbers: string[],
addressFinderOptions:
AddressFinderOptions,
token: string): GeocodeInfo[]
- ← getInfo(
dataSources: string[],
token: string): AddressFinderInfo[]
- ← getVersion: string

AddressInfo

- address: Address
- errorMessage: string

Find (Continued)

Content Finder Service

ContentFinder

- ← getAvailableAttributes(
 contentType: string,
 token: string): AttributeDesc[]
- ← getAvailableServices(
 token: string): string[]
- ← getDependentContent(
 contentFinderRecordID:
 ContentFinderRecordID,
 token: string): ContentFinderRecord[]
- ← getRecord(
 contentFinderRecordID:
 ContentFinderRecordID,
 token: string): ContentFinderRecord
- ← getSubContent(
 contentFinderRecordID:
 ContentFinderRecordID,
 token: string): ContentFinderRecord[]
- ← getUniqueValues(
 contentType: string,
 attributeName: string,
 token: string): string[]
- ← getVersion: string
- ← searchRecords(
 contentFinderSearchOptions:
 ContentFinderSearchOptions,
 token: string): ContentFinderResultSet

ContentFinderRecord

- attributes: KeyValue[]
- contentLastUpdated: long
- contentProperties: KeyValue[]
- extent: Envelope
- id: ContentFinderRecordID

ContentFinderRecordID

- contentType: string
- name: string
- service: string

ContentFinderResultSet

- records: ContentFinderRecord[]
- totalCount: integer

ContentFinderSearchOptions

- contentType: string
- dataScale: long
- dataType: integer
- extent: Envelope
- orderBy: string
- orderDescending: boolean
- ownerType: string
- resultSetRange: ResultSetRange
- returnAttributes: string[]
- returnContentProperties: boolean
- service: string
- spatialOperator: string
- whereClause: string

Place Finder Service

PlaceFinder

- ← findPlace(
 placeName: string,
 placeFinderOptions: PlaceFinderOptions,
 token: string): GeocodeInfo
- ← getInfo(
 dataSources: string[],
 token: string): PlaceFinderInfo[]
- ← getVersion: string

Place Finder Sample Service

PlaceFinderSample

- ← findPlace(
 placeName: string,
 placeFinderOptions: PlaceFinderOptions):
 GeocodeInfo
- ← getInfo(): PlaceFinderInfo
- ← getVersion: string

Map & Route

Common

CircleDesc

- antialiasing: string
- boundaryColor: string
- boundaryTransparency: double
- boundaryType: string
- circle: Circle
- fillColor: string
- fillTransparency: double
- overlapLabels: string
- thickness: integer

LabelDesc

- angle: double
- antialiasing: string
- blackoutColor: string
- caps: string
- font: string
- fontColor: string
- fontSize: integer
- fontStyle: string
- glowing: boolean
- hAlignment: string
- interval: integer
- outlineColor: string
- overlapLabels: string
- transparency: double
- vAlignment: string

Layer

- name: string
- visibility: string

LayerInfo

- legends: LayerLegend[]
- name: string
- visibility: string

LayerLegend

- layerClassName: string
- symbolURL: string

LineDesc

- antialiasing: string
- color: string
- endType: string
- geometry: Geometry
- jointType: string
- lineType: string
- overlapLabels: string
- thickness: integer
- transparency: double

MapArea

- center: Point
- extent: Envelope
- scale: long

MapImageInfo

- layers: LayerInfo[]
- legendURL: string
- mapArea: MapArea
- mapURL: string
- outputCoordSys: CoordSys

MapImageOptions

- backgroundColor: string
- circles: CircleDesc[]
- dataSource: string
- displayLayers: Layer[]
- lines: LineDesc[]
- mapImageFormat: string
- mapImageSize: MapImageSize
- mapLegend: MapLegend
- markers: MarkerDesc[]
- outputCoordSys: CoordSys
- polygons: PolygonDesc[]
- returnLayers: boolean
- scaleBars: ScaleBarDesc[]
- styleSheet: string

MapLegend

- antialiasing: string
- autoextend: boolean
- backgroundColor: string
- columns: integer
- font: string
- height: integer
- layerFontSize: integer
- splitText: string
- swatchHeight: integer
- swatchWidth: integer
- title: string
- titleFontSize: integer
- valueFontSize: integer
- width: integer

MarkerDesc

- color: string
- iconDataSource: string
- label: string
- labelDesc: LabelDesc
- location: Point
- name: string
- size: integer

PixelCoord

- x: integer
- y: integer

PolygonDesc

- antialiasing: string
- boundary: string
- boundaryColor: string
- boundaryJointType: string
- boundaryTransparency: double
- boundaryType: string
- boundaryWidth: integer
- fillColor: string
- fillInterval: integer
- fillTransparency: double
- fillType: string
- geometry: Geometry
- overlapLabels: string

ScaleBarDesc

- barColor: string
- barLength: integer
- barPixelLocation: PixelCoord
- barUnits: string
- barWidth: integer
- font: string
- fontColor: string
- fontSize: integer

Map & Route (Continued)

Map Image Service

MapImage

- ← convertMapCoordsToPixelCoords(
mapArea: MapArea,
mapSize: MapImageSize,
mapCoords: Point[],
token:string): PixelCoord[]
- ← convertPixelCoordsToMapCoords(
mapArea: MapArea,
mapSize: MapImageSize,
mapClickPoints: PixelCoord[],
token:string): Point[]
- ← getBestMap(
mapImageOptions: MapImageOptions,
bufferPercent: double,
token: string): MapImageInfo
- ← getBestMapArea(
mapImageOptions: MapImageOptions,
bufferPercent: double,
token:string): MapArea
- ← getCustomThematicMap(
mapArea: MapArea,
mapImageOptions: MapImageOptions,
thematicData: ThematicData,
thematicOptions: ThematicOptions,
thematicColors: ThematicColors,
token: string): MapImageInfo
- ← getESRIThematicMap(
mapArea: MapArea,
mapImageOptions: MapImageOptions,
thematicField: string,
thematicOptions: ThematicOptions,
thematicColors: ThematicColors,
token: string): MapImageInfo
- ← getFusedMapImage(
mapFusionOptions: MapFusionOptions[],
outputImageFormat: string,
token: string): string
- ← getMaps(
mapAreas: MapArea[],
mapImageOptions: MapImageOptions[],
token: string): MapImageInfo[]
- ← getMarkerNames(
iconDataSource: string,
token: string): string[]
- ← getSavedMap(
savedMapID: string,
token: string): MapImageInfo
- ← getThematicFields(
thematicDataSource: string,
token: string): string[]
- ← getValueMap(
mapArea: MapArea,
mapImageOptions: MapImageOptions,
thematicData: ThematicData,
codeColorValues: KeyValue[],
thematicOptions: ThematicOptions,
token: string): MapImageInfo
- ← getVersion: string
- ← saveMap(
mapArea: MapArea,
mapImageOptions: MapImageOptions,
token: string): string

MapFusionOptions

- mapURL: string
- transparency: double

ThematicColors

- classColors: string[]
- colorPalette: string

ThematicData

- data: KeyValue[]
- precision: integer
- sign: string

ThematicOptions

- classificationMethod: string
- classLabels: string[]
- numClasses: integer
- thematicOnlyLegend: boolean

Route Finder Service

RouteFinder

- ← findRoute(
routeStops: RouteStop[],
routeFinderOptions: RouteFinderOptions,
token: string): RouteInfo
- ← getLanguages(
token: string): string[]
- ← getVersion: string

RouteDisplayOptions

- color: string
- thickness: integer
- transparency: double

RouteFinderOptions

- avoidTraffic: boolean
- dataSource: string
- language: string
- returnDirections: boolean
- returnGeometry: boolean
- returnMap: boolean
- returnTurnByTurnMaps: boolean
- routeDisplayOptions: RouteDisplayOptions
- routeMapOptions: MapImageOptions
- routeOptions: RouteOptions
- trafficDataSource: string
- trafficSeverity: integer
- turnByTurnMapOptions: MapImageOptions
- units: string

RouteInfo

- optimizedStopOrder: integer[]
- routeMap: MapImageInfo
- routePath: Geometry
- segmentDescs: SegmentDesc[]
- totalDesc: RouteSummary
- turnByTurnMaps: MapImageInfo[]

RouteOptions

- filterExtent: Envelope
- hwyPref: integer
- optimizeStops: boolean
- precision: double
- returnExtents: boolean
- routeBarriers: Point[]
- routeType: string

RouteStop

- desc: string
- point: Point

RouteSummary

- distanceUnits: string
- numericDistance: double
- numericTime: double
- routeExtent: Envelope
- totalDistance: string
- totalTime: string

SegmentDesc

- descriptiveDirections: string
- descriptiveDistance: string
- extent: Envelope
- numericDistance: double
- numericTime: double

Report Services

Report Service

Report

- ← createPDFDocument(
 templateName: string,
 data: KeyValue[],
 token: string): string
- ← getAvailableTemplateName(
 token: string): string[]
- ← getCustomSiteReport(
 site: Site,
 reportOptions: ReportOptions[],
 variables: CustomReportVariables[],
 token: string): ReportInfo
- ← getCustomStandardGeographyReport(
 standardGeographies: StandardGeography[],
 reportOptions: ReportOptions,
 variables: CustomReportVariables[],
 token: string): ReportInfo
- ← getCustomVariables(
 variableDataSource: string,
 token: string): CustomVariableInfo[]
- ← getDatabases(
 token: string): string[]
- ← getGeographyIDs(
 geographyIDOptions: GeographyIDOptions,
 token: string): ResultSet
- ← getReportHeaderKeys(
 methodName: string,
 token: string): KeyValue[]
- ← getSiteReports(
 site: Site,
 reportOptions: ReportOptions[],
 combineReports: boolean,
 token: string): ReportInfo[]
- ← getStandardGeographyReport(
 standardGeographies: StandardGeography[],
 reportOptions: ReportOptions,
 token: string): ReportInfo
- ← getTables(
 database: string,
 token: string): string[]
- ← getThematicMapReport(
 sites: Site[],
 reportOptions: ReportOptions,
 thematicReportOptions: ThematicReportOptions,
 thematicVariable: string,
 token: string): ReportInfo
- ← getThematicVariables(
 database: string,
 table: string,
 token: string): ThematicVariableInfo[]
- ← getVersion: string
- ← searchThematicVariables(
 keyword: string,
 token: string): ThematicVariableInfo[]

CustomReportVariables

- variableDataSource: string
- variableNames: string[]

CustomVariableInfo

- database: string
- fieldDesc: FieldDesc

GeographyIDOptions

- geographyIDFilter: string
- geographyLevelFilter: string
- keyword: string
- outputGeographyLevel: string
- resultSetRange: ResultSetRange

ReportInfo

- dataSource: string
- reportFormat: string
- reportURL: string

ReportOptions

- dataSource: string
- reportFormat: string
- reportHeader: KeyValue[]

Site

- calcType: string
- polygons: Geometry[]
- ringRadii: double[]
- ringUnits: string
- siteLocation: Point
- siteName: string

StandardGeography

- geographyIDs: string[]
- geographyLevel: string

ThematicReportOptions

- classificationMethod: string
- colorPalette: string
- geographyLevel: string
- numClasses: integer

ThematicVariableInfo

- database: string
- shortDesc: string
- table: string
- variable: string

Spatial Query

Spatial Query Service

SpatialQuery

- ← findFeaturesByExtent(
 extent: Envelope,
 spatialQueryOptions: SpatialQueryOptions,
 token: string): ResultSet
- ← findFeaturesByGeometry(
 geometry: Geometry,
 spatialQueryOptions: SpatialQueryOptions,
 token: string): ResultSet
- ← findFeaturesByPoint(
 point: Point,
 spatialQueryOptions: SpatialQueryOptions,
 token: string): ResultSet
- ← getAvailableFieldNames(
 dataSources: string[],
 token: string): SpatialQueryInfo[]
- ← getVersion: string

SpatialQueryInfo

- availableFields: FieldDesc[]
- dataSource: string

SpatialQueryOptions

- dataSource: string
- distanceUnits: string
- returnDistance: boolean
- returnGeometry: boolean
- searchOptions: SearchOptions
- spatialBuffer: SpatialBuffer

Utility

Account Info Service

AccountInfo <ul style="list-style-type: none">← getServices(accountInfoOptions: AccountInfoOptions, token: string): ServiceSummary[]← getUsage(accountInfoOptions: AccountInfoOptions, dateTimeRange: DateTimeRange, token: string): AccountSummary← getVersion: string	DataSourceUsage <ul style="list-style-type: none">— name: string— title: string— usage: UsageInfo
AccountInfoOptions <ul style="list-style-type: none">— adminOptions: AdminOptions— returnDataSources: string[]— returnDetailedInfo: boolean— returnServices: string[]	ServiceInfo <ul style="list-style-type: none">— dataSourcesInfo: DataSourceInfo[]— desc: string— name: string— wsd: string
AccountSummary <ul style="list-style-type: none">— groupUsage: AccountUsage— usageByUser: UserUsage[]	ServiceSummary <ul style="list-style-type: none">— servicesInfo: ServiceInfo[]— user: User
AccountUsage <ul style="list-style-type: none">— averageDailyCreditUsage: double— creditsRemaining: double— creditsUsed: double— dailyStorageCost: double— diskStorageQuota: long— diskStorageUsed: long— estimatedEndDate: long— expiration: long— totalStorageCost: double	ServiceUsage <ul style="list-style-type: none">— dataSourcesUsage: DataSourceUsage[]— name: string— usage: UsageInfo
AdminOptions <ul style="list-style-type: none">— returnGroupUsage: boolean— returnUsers: string[]	UsageInfo <ul style="list-style-type: none">— creditsUsed: double— dateLastUsed: long— numRequests: long
DataSourceInfo <ul style="list-style-type: none">— name: string— summary: string— supportsArcXML: boolean— supportsWMS: boolean— title: string	User <ul style="list-style-type: none">— emailAddress: string— firstName: string— isActive: boolean— isAdmin: boolean— lastName: string— username: string
	UserUsage <ul style="list-style-type: none">— accountUsage: AccountUsage— servicesUsage: ServiceUsage[]— user: User

Authentication Service

Authentication <ul style="list-style-type: none">← getCustomExpirationToken(username: string, password: string, expiration: integer): string← getToken(username: string, password: string): string← getVersion: string← validateToken(token: string): integer

Utility Service

Utility <ul style="list-style-type: none">← convertCircleToGeometry(circle: Circle, numPoints: integer, token: string): Geometry← convertCircularArcToGeometry(arc: CircularArc, numPoints: integer, token: string): Geometry← convertEllipseToGeometry(ellipse: Ellipse, numPoints: integer, token: string): Geometry← getBestProjection(extent: Envelope, token: string): Envelope← getDataCollectionDate(dataSource: string, extent: Envelope, mapSize: mapImageSize, token: string): string← getDistance(location1: Point, location2: Point, units: string, numPoints: integer, returnGeometry: boolean, token: string): DistanceInfo← getDriveTime(dataSource: string, point: Point, interval: integer, units: string, token: string): Geometry← getSupportedProjections(type: string, token: string): KeyValue[]← getVersion: string← projectExtent(extent: Envelope, projectTo: CoordSys, token: string): Envelope← projectGeometry(geometry: Geometry, projectTo: CoordSys, token: string): Geometry← projectPoint(point: Point, projectTo: CoordSys, token: string): Point
CircularArc <ul style="list-style-type: none">— center: Point— endAngle: double— innerRadius: double— outerRadius: double— radiusUnits: string— startAngle: double— trueCircle: boolean
DistanceInfo <ul style="list-style-type: none">— distance: double— geometry: Geometry
Ellipse <ul style="list-style-type: none">— axisUnits: string— center: Point— majorAxis: double— minorAxis: double— rotation: double— trueEllipse: boolean

Wireless Location

Wireless Location Service

WirelessLocation

```
← getDeviceLocation(  
  devices: MobileDevice[],  
  deviceLocationOptions:  
    DeviceLocationOptions,  
  token: string): DeviceLocationInfo[]  
← getDeviceStatus(  
  devices: MobileDevice[],  
  token: string): DeviceStatus[]  
← getDeviceStops(  
  deviceStopOptions: DeviceStopOptions,  
  token: string): DeviceStopInfo[]  
← getTripInfo(  
  deviceTripOptions: DeviceTripOptions,  
  token: string): DeviceTripInfo  
← getVersion: string  
← sendSMS(  
  devices: MobileDevice[],  
  message: string,  
  token: string): string[]
```

DeviceLocationInfo

```
—■ accuracy: string  
—■ errorMessage: string  
—■ point: Point  
—■ radius: double  
—■ timeStamp: long
```

DeviceLocationOptions

```
■— accuracy: string  
■— radiusUnits: string
```

DeviceStatus

```
—■ errorMessage: string  
—■ status: string
```

DeviceStopInfo

```
—■ dateTimeRange: DateTimeRange  
—■ point: Point
```

DeviceStopOptions

```
■— dataFileName: string  
■— dateTimeRange: DateTimeRange  
■— maxRadius: double  
■— minDuration: integer  
■— radiusUnits: string
```

DeviceTripInfo

```
—■ angle: double  
—■ distance: double  
—■ path: Geometry  
—■ speed: integer
```

DeviceTripOptions

```
■— dataFileName: string  
■— dateTimeRange: DateTimeRange  
■— returnAngle: boolean  
■— returnDistance: boolean  
■— returnPath: boolean  
■— returnSpeed: boolean  
■— units: string
```

MobileDevice

```
■— dataSource: string  
■— phoneNumber: string
```