

ArcWebSM 2006 Object Model

Data Management

Find

Map & Route

Report

Spatial Query

Utility

Wireless

| Common | |
|-----------------------|--|
| AttributeDesc | <ul style="list-style-type: none">— length: integer— name: string— precision: integer— required: integer— shortDesc: string— type: string |
| Circle | <ul style="list-style-type: none">— center: Point— radius: double— radiusUnits: string— trueCircle: boolean |
| CoordSys | <ul style="list-style-type: none">— datumTransformation: string— projection: string |
| DateTimeRange | <ul style="list-style-type: none">— end: long— start: long |
| Envelope | <ul style="list-style-type: none">— coordSys: CoordSys— max: double— max: double— minX: double— minY: double |
| FieldDesc | <ul style="list-style-type: none">— length: integer— name: string— precision: integer— shortDesc: string— type: string |
| Geometry | <ul style="list-style-type: none">— coordSys: CoordSys— parts: integer[]— points: SimplePoint[]— type: integer |
| KeyValue | <ul style="list-style-type: none">— key: string— value: string |
| MapImageSize | <ul style="list-style-type: none">— height: integer— width: integer |
| OrderBy | <ul style="list-style-type: none">— descending: boolean— field: string |
| Point | <ul style="list-style-type: none">— coordSys: CoordSys— x: double— y: double |
| ResultSet | <ul style="list-style-type: none">— fields: FieldDesc[]— rows: RowData[]— rowCount: integer |
| ResultSetRange | <ul style="list-style-type: none">— count: integer— startIndex: 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 Manager Service | |
|--|--|
| DataManager <ul style="list-style-type: none"> ➦ <code>addFeature()</code> <ul style="list-style-type: none"> dataFileName: string, features: Feature[], rollbackOnError: boolean, tokenString: FeatureInfo[] ➦ <code>createDataFile()</code> <ul style="list-style-type: none"> dataFile: DataFile, token: string: void ➦ <code>deleteDataFile()</code> <ul style="list-style-type: none"> dataFileName: string, tokenString: void ➦ <code>deleteFeatures()</code> <ul style="list-style-type: none"> dataFileName: string, featuresId: string[] tokenString: void ➦ <code>findFeatures()</code> <ul style="list-style-type: none"> dataFileName: string, featureSearchOptions: FeatureSearchOptions, tokenString: FeatureResultSet ➦ <code>geocodeFeatures()</code> <ul style="list-style-type: none"> dataFileName: string, featuresId: string[] geocodeOptions: GeocodeOptions, tokenString: GeocodeInfo[] ➦ <code>getAvailableDataFileAttributes()</code> <ul style="list-style-type: none"> token: string: AttributeDesc[] ➦ <code>getDataFileInfo()</code> <ul style="list-style-type: none"> dataFileNames: string[] tokenString: DataFileInfo[] ➦ <code>getUniqueValues()</code> <ul style="list-style-type: none"> dataFileName: string, fieldName: string, resultSetRange: ResultSetRange, tokenString: StringSet ➦ <code>getVersion()</code> <ul style="list-style-type: none"> tokenString: StringSet ➦ <code>renameField()</code> <ul style="list-style-type: none"> dataFileName: string, oldFieldName: string, newFieldName: string, tokenString: void ➦ <code>updateDataFile()</code> <ul style="list-style-type: none"> dataFile: DataFile, tokenString: void ➦ <code>updateFeatures()</code> <ul style="list-style-type: none"> dataFile: DataFile, tokenString: void ➦ <code>updateFeatures()</code> <ul style="list-style-type: none"> dataFile: DataFile, tokenString: void ➦ <code>updateFeaturesOptions()</code> <ul style="list-style-type: none"> featureUpdateOptions: FeatureUpdateOptions, featuresId: string[] tokenString: void | DataFileInfo <ul style="list-style-type: none"> ➦ <code>dataFile: DataFile</code> ➦ <code>dateCreated: long</code> ➦ <code>extent: Envelope</code> ➦ <code>featureCount: Integer</code> ➦ <code>geographicExtent: Envelope</code> ➦ <code>lastModified: long</code> ➦ <code>maxScale: long</code> ➦ <code>minScale: long</code> ➦ <code>size: long</code> Feature <ul style="list-style-type: none"> ➦ <code>fieldValues: KeyValue[]</code> ➦ <code>geometry: Geometry</code> FeatureInfo <ul style="list-style-type: none"> ➦ <code>errorMessage: string</code> ➦ <code>id: string</code> FeatureInfo <ul style="list-style-type: none"> ➦ <code>dateCreated: long</code> ➦ <code>featureData: Feature</code> ➦ <code>geocodeDataSource: string</code> ➦ <code>geocodeStatus: string</code> ➦ <code>id: string</code> ➦ <code>lastModified: long</code> FeatureResultSet <ul style="list-style-type: none"> ➦ <code>features: FeatureInfo[]</code> ➦ <code>totalCount: Integer</code> FeatureSearchOptions <ul style="list-style-type: none"> ➦ <code>returnGeometry: Geometry</code> ➦ <code>returnGeometry: boolean</code> ➦ <code>searchOptions: SearchOptions</code> ➦ <code>spatialBuffer: SpatialBuffer</code> FeatureUpdateOptions <ul style="list-style-type: none"> ➦ <code>fieldValues: KeyValue[]</code> ➦ <code>geometry: Geometry</code> FieldInfo <ul style="list-style-type: none"> ➦ <code>indexed: boolean</code> ➦ <code>label: string</code> ➦ <code>name: string</code> ➦ <code>type: Integer</code> GeocodeCandidate <ul style="list-style-type: none"> ➦ <code>address: Address</code> ➦ <code>desc1: string</code> ➦ <code>desc2: string</code> ➦ <code>extent: Envelope</code> ➦ <code>matchType: string</code> ➦ <code>point: Point</code> ➦ <code>score: double</code> ➦ <code>type: string</code> GeocodeInfo <ul style="list-style-type: none"> ➦ <code>candidates: GeocodeCandidate[]</code> ➦ <code>errorMessage: string</code> ➦ <code>hashCode: boolean</code> ➦ <code>matchType: string</code> ➦ <code>totalCount: Integer</code> GeocodeOptions <ul style="list-style-type: none"> ➦ <code>dataSource: string</code> ➦ <code>extendedPostalCode: boolean</code> ➦ <code>formatAddressFields: boolean</code> ➦ <code>partialAddress: boolean</code> StringSet <ul style="list-style-type: none"> ➦ <code>strings: string[]</code> ➦ <code>totalCount: Integer</code> |
| AddressFields <ul style="list-style-type: none"> ➦ <code>city: string</code> ➦ <code>country: string</code> ➦ <code>houseNumber: string</code> ➦ <code>intersection: string</code> ➦ <code>postalCode: string</code> ➦ <code>stateProvince: string</code> ➦ <code>street: string</code> | |
| DataFile <ul style="list-style-type: none"> ➦ <code>addresses: AddressFields</code> ➦ <code>attributes: KeyValue[]</code> ➦ <code>coordsBy: Country</code> ➦ <code>fieldInfo: FieldInfo[]</code> ➦ <code>geometryType: Integer</code> ➦ <code>name: string</code> | |

| Common | |
|-----------------------------|--|
| Address | <ul style="list-style-type: none"> city: string country: string houseNumber: string intersection: string postalCode: string stateProvince: string street: string |
| AddressFinderInfo | <ul style="list-style-type: none"> countryCodes: Key<Value>[] dataSource: string extendedPostalCode: boolean partialAddress: boolean reverseGeocode: boolean snappyTypes: string |
| AddressFinderOptions | <ul style="list-style-type: none"> dataSource: string extendedPostalCode: boolean partialAddress: boolean resultRange: ResultSetRange snappyType: string |
| GeocodeCandidate | <ul style="list-style-type: none"> address: Address desc1: string desc2: string extent: Envelope matchType: string point: Point score: double type: string |
| GeocodeInfo | <ul style="list-style-type: none"> candidates: GeocodeCandidate[] errorMessage: string hasNone: boolean matchType: string totalCount: integer |
| PlaceFinderInfo | <ul style="list-style-type: none"> countryCodes: Key<Value>[] dataSource: string types: Key<Value>[] |
| PlaceFinderOptions | <ul style="list-style-type: none"> dataSource: string filterCountry: string filterExtent: Envelope filterType: string resultSetRange: ResultSetRange searchType: string |
| Address Finder Service | |
| AddressFinder | <ul style="list-style-type: none"> findAddressByPoint: point: Point, addressFinderOptions: AddressFinderOptions, token: string): Address findLocationByAddress: address: Address, addressFinderOptions: AddressFinderOptions, token: string): GeocodeInfo findLocationByIP: ipAddress: string, addressFinderOptions: AddressFinderOptions, token: string): GeocodeInfo findLocationByPhoneNum: findLocationByPhoneNum(|

| Content Finder Service | |
|----------------------------|--|
| ContentFinder | <ul style="list-style-type: none">← getAvailableAttributes(): contentType: string; token: string; AttributeDesc[]← getAvailableServices(): token: string; string[]← getDependentContent(): ContentFinderRecordID; ContentFinderRecordID← getRecord(): ContentFinderRecordID; token: string; ContentFinderRecord← getSubContent(): ContentFinderRecordID; token: string; ContentFinderRecord[]← getUniqueValues(): contentType: string; attributeName: string; token: string; string[]← getVersion(): string← searchContent(): ContentFinderSearchOptions; ContentFinderSearchOptions; token: string; ContentFinderResultSet |
| ContentFinderRecord | <ul style="list-style-type: none">← attributes: KeyValue[]← contentTypeUpdated: long← contentProperties: KeyValue[]← extent: Envelope← id: ContentFinderRecordID |
| ContentFinderRecordID | <ul style="list-style-type: none">← name: string← name: string← service: string |
| ContentFinderResultSet | <ul style="list-style-type: none">← records: ContentFinderRecord[]← totalCount: integer |
| ContentFinderSearchOptions | <ul style="list-style-type: none">← 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 | <ul style="list-style-type: none">← findPlace(): placeName: string; placeName: string; PlaceFinderOptions; token: string; GeoCoordinate← getPlace(): dataSources: string[]; token: string; PlaceFinderInfo[]← getVersion(): string |

Address Finder Service

```
AddressFinder
├─ findAddressByPoint
│   point: Point;
│   addressFinderOptions:
│       AddressFinderOptions,
│       token: string;
│   findLocalAddressByAddress
│       address: Address;
│       addressFinderOptions:
│           AddressFinderOptions,
│           token: string;
│   findLocalAddressByIP
│       ipAddress: string;
│       addressFinderOptions:
│           AddressFinderOptions,
│           token: string;
│   findLocalAddressByPhoneNumbers
│       phoneNumbers: string[];
│       addressFinderOptions:
│           AddressFinderOptions,
│           token: string;
│   getAddressInfo
│       dataSources: string[];
│       token: string;
│   getAddressInfoByToken
│       token: string;
└─ getAddressInfoByToken
```

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

Address Manager Service

```
AddressManager
├── findAddressesByPoint
│   │   Point: Point;
│   │   addressFinderOptions:
│   │       token: string;
│   │       addressInfo[];
│   └── findLocationsByAddress
│       │   addresses: Address[];
│       │   addressFinderOptions:
│       │       token: string;
│       │       geoCentroid[];
│       └── findLocationsByIP
│           │   (pAddresses: string);
│           │   addressFinderOptions:
│           │       token: string;
│           │       addressFinderOptions:
│           │           token: string;
│           │           phoneNumbers: string[];
│           │           addressFinderOptions:
│           │               token: string;
│           │               addressFinderInfo[];
│           └── getGeoInfo
│               │   dataSources: string[];
│               │   token: string;
│               │   addressFinderInfo[];
│               └── geoVersion: string
```

AddressInfo

```
├── address: Address;
└── errorMessage: string
```

| Common | |
|--------------------|---|
| CircleDesc | <ul style="list-style-type: none"> — antialiasing: string — boundaryColor: string — boundaryTransparency: double — boundaryType: string — circle: Circle — fillColor: string — fillTransparency: double — overlapLabels: string — thickness: integer |
| LabelDesc | <ul style="list-style-type: none"> — angle: double — antialiasing: string — blockoutColor: string — caps: string — font: string — fontColor: string — fontSize: integer — fontStyle: string — glowing: boolean — align: string — interval: integer — outlineColor: string — overlapLabels: string — transparency: double — vAlign: string |
| Layer | <ul style="list-style-type: none"> — name: string — visibility: string |
| LayerInfo | <ul style="list-style-type: none"> — legends: LayerLegend[] — name: string — visibility: string |
| LayerLegend | <ul style="list-style-type: none"> — layerClassName: string — symbolURL: string |
| LineDesc | <ul style="list-style-type: none"> — antialiasing: string — color: string — endType: string — geometry: Geometry — jointType: string — lineType: string — overlapLabels: string — thickness: integer — transparency: double |
| MapArea | <ul style="list-style-type: none"> — center: Point — extent: Envelope — scale: long |
| MapInfo | <ul style="list-style-type: none"> — layers: LayerInfo[] — legendURL: string — mapArea: MapArea — mapURL: string — outputCoords: CoordSys |

| | |
|--------------------------------|------------------------------|
| MapImageOptions | |
| — backgroundColor: string | circle: Circle |
| — dataSource: string | displayLayers: Layer[] |
| — displayLayers: Layer[] | lines: Line[] |
| — lines: Line[] | mapImageFormat: string |
| — mapImageFormat: string | mapImageSize: MapImageSize |
| — mapImageSize: MapImageSize | mapLegend: MapLegend |
| — mapLegend: MapLegend | markers: MarkerDesc[] |
| — markers: MarkerDesc[] | outputCoords: CoordSys |
| — outputCoords: CoordSys | polygons: PolygonDesc[] |
| — polygons: PolygonDesc[] | returnLayers: boolean |
| — returnLayers: boolean | scalesAs: ScalesDesc[] |
| — scalesAs: ScalesDesc[] | stylesheet: string |
| — stylesheet: string | |
| MapLegend | |
| — antialiasing: string | autoExtend: boolean |
| — autoExtend: boolean | backgroundColor: string |
| — backgroundColor: string | columns: integer |
| — columns: integer | font: string |
| — font: string | height: integer |
| — height: integer | layerFontSize: integer |
| — layerFontSize: integer | splitText: string |
| — splitText: string | swatchHeight: integer |
| — swatchHeight: integer | swatchWidth: integer |
| — swatchWidth: integer | title: string |
| — title: string | titleFontSize: integer |
| — titleFontSize: integer | valueFontSize: integer |
| — valueFontSize: integer | width: integer |
| — width: integer | |
| MarkerDesc | |
| — color: string | iconDataSource: string |
| — iconDataSource: string | label: string |
| — label: string | labelDesc: LabelDesc |
| — labelDesc: LabelDesc | location: Point |
| — location: Point | name: string |
| — name: string | size: integer |
| — size: integer | |
| PixelCoord | |
| — x: integer | y: integer |
| — y: integer | |
| PolygonDesc | |
| — antialiasing: string | boundary: string |
| — boundary: string | boundaryColor: Color |
| — boundaryColor: Color | boundaryType: string |
| — boundaryType: string | boundaryTransparency: double |
| — boundaryTransparency: double | boundaryWidth: integer |
| — boundaryWidth: integer | fillColor: string |
| — fillColor: string | fillIntensity: integer |
| — fillIntensity: integer | fillTransparency: double |
| — fillTransparency: double | fillType: string |
| — fillType: string | geometry: Geometry |
| — geometry: Geometry | overlapLabels: string |
| — overlapLabels: string | |
| ScalesDesc | |
| — barColor: string | barLength: integer |
| — barLength: integer | barWidthColor: PixelCoord |
| — barWidthColor: PixelCoord | barUnits: string |
| — barUnits: string | barWidth: integer |
| — barWidth: integer | font: string |
| — font: string | fontColor: string |
| — fontColor: string | fontSize: integer |
| — fontSize: integer | |

MapImage Service

Mapimage

← convertMapCoordsToPixelCoords

mapArea: MapArea;

mapSize: MapImageSize;

mapCoords: Point[];

token:string; PixelCoords: Point[];

← convertPixelCoordsToMapCoords

mapArea: MapArea;

mapSize: MapImageSize;

mapCoordsToPixelCoords: Point[];

token:string; PixelCoords: MapArea;

← getBstMap

mapImageOptions: MapImageOptions;

bufferColors: Double;

token:string; MapImageInfo

← getBstMapArea

mapImageOptions: MapImageOptions;

bufferColors: Double;

token:string; MapArea

← getCustomThematicMap

mapArea: MapArea;

mapImageOptions: MapImageOptions;

thematicData: ThematicData;

thematicOptions: ThematicOptions;

thematicColors: ThematicColors;

token:string; MapImageInfo

← getSRIThematicMap

mapArea: MapArea;

mapImageOptions: MapImageOptions;

thematicFields: string;

thematicOptions: ThematicOptions;

thematicColors: ThematicColors;

token:string; MapImageInfo

← getFusedMap

mapImageOptions: MapImageOptions;

outputImageFormat: string;

token:string; string;

← getMap

mapArea: MapArea[];

mapImageOptions: MapImageOptions;

token:string; MapImageInfo[];

← getMarkerName

iconData: string;

token:string; string;

← getSaveMap

saveMapId: string;

token:string; MapImageInfo

← getThematicData

thematicDataFields: string;

token:string; string;

← getValuemap

mapArea: MapArea;

mapImageOptions: MapImageOptions;

thematicData: ThematicData;

colorColorValues: KeyValue[];

thematicOptions: ThematicOptions;

token:string; MapImageInfo

← getVersion

string;

← saveMap

mapArea: MapArea;

mapImageOptions: MapImageOptions;

token:string; string;

MapImageOptions

← mapURL: string

← transparency: double

ThematicColors

← classColors: string[]

← colorPalette: string

ThematicData

← data: KeyValue[]

← precision: integer

← sig: string

ThematicOptions

← classificationMethod: string

← classificationSize: integer

← numClasses: integer

← thematicOnlyLegend: boolean

| Route Finder Service | |
|---|---|
| ← findRoute | <pre> routeStop: RouteStop[]; routeFinderOptions: RouteFinderOptions; token: string; routeInfo: getLanguage: token: string; string[]; getVersion: string; </pre> |
| <pre> color: Options thickness: integer transparency: double </pre> | |
| RouteFinderOptions | <pre> avoidTraffic: boolean dataSource: string language: string returnDirections: boolean returnGeometry: boolean returnMap: boolean turnByTurnMaps: boolean routeDisplayOptions: RouteDisplayOptions routeMapOptions: MapImageOptions routeOptions: RouteOptions trafficDataSource: string trafficSeverity: integer turnByTurnMapOptions: MapImageOptions units: string </pre> |
| RouteInfo | <pre> optimizedStopOrder: integer[] routeMap: MapImageInfo routePath: Geometry segmentDescs: SegmentDesc[] totalDesc: RouteSummary turnByTurnMaps: MapImageInfo[] </pre> |
| RouteOptions | <pre> filterExtent: Envelope isWayPnt: integer optimizeStops: boolean precision: double returnExtents: boolean routeBarriers: Point[] routeType: string </pre> |
| RouteStop | <pre> desc: string point: Point </pre> |
| RouteSummary | <pre> distanceUnits: string numericDistance: double numericTime: double routeExtent: Envelope totalDistance: string totalTime: string </pre> |
| SegmentDesc | <pre> descriptiveDirections: string descriptiveDistance: string extent: Envelope numericDistance: double numericTime: double </pre> |

| Report Service | |
|-----------------------|---|
| Report | <ul style="list-style-type: none"> ← createPDFDocument <ul style="list-style-type: none"> templateName: string; data: KeyValue[]; token: string; ← getAvailableTemplateName(s) <ul style="list-style-type: none"> tokens: string[]; string[] ← getCustomTemplateReport <ul style="list-style-type: none"> site: Site; reportOptions: ReportOptions[]; variables: CustomReportVariables[]; token: string; ← getCustomStandardGeographyReport <ul style="list-style-type: none"> standardGeographies: StandardGeography[]; reportOptions: ReportOptions; variables: CustomReportVariables[]; token: string; ← getCustomVariables <ul style="list-style-type: none"> variableDataSource: string; token: string; ← getDatabase(s) <ul style="list-style-type: none"> token: string; ← getGeographyDiD <ul style="list-style-type: none"> geographyOptions: GeographyDiOptions; token: string; ← getReportHeaderKeys <ul style="list-style-type: none"> methodName: string; token: string; ← getSiteReport(s) <ul style="list-style-type: none"> site: Site; reportOptions: ReportOptions[]; combineReports: boolean; token: string; ← getStandardGeographyReport(s) <ul style="list-style-type: none"> standardGeographies: StandardGeography[]; reportOptions: ReportOptions; token: string; ← getTable(s) <ul style="list-style-type: none"> database: string; token: string; ← getThematicMapReport(s) <ul style="list-style-type: none"> site: Site[]; reportOptions: ReportOptions; thematicReportOptions: ThematicReportOptions[]; token: string; ← getThematicVariables <ul style="list-style-type: none"> thematicVariable: string; table: string; token: string; ← getVersion: string; ← over(s)ThematicVariables <ul style="list-style-type: none"> keyWord: string; token: string; |
| CustomReportVariables | |
| | <ul style="list-style-type: none"> — variableDataSource: string; — variableNames: string[]; |
| CustomVariableInfo | |
| | <ul style="list-style-type: none"> — database: string; — fieldDesc: FieldDesc; |
| GeographyDiOptions | |
| | <ul style="list-style-type: none"> — geographyFilter: string; — geographyLevelFilter: string; — keyword: string; — reportGeographyLevel: string; — resultSetRange: ResultSetRange; |
| ReportInfo | |
| | <ul style="list-style-type: none"> — dataSource: string; — reportFormat: string; — reportURL: string; |
| ReportOptions | |
| | <ul style="list-style-type: none"> — dataSource: string; — reportFormat: string; — reportHeader: KeyValue[]; |
| Site | |
| | <ul style="list-style-type: none"> — calcType: string; — polygons: Geometry[]; — ringsRadii: double[]; — ringUnits: string; — siteLocation: Point; — siteName: string; |
| StandardGeography | |
| | <ul style="list-style-type: none"> — geographyDiDs: string[]; — geographyLevel: string; |
| ThematicReportOptions | |
| | <ul style="list-style-type: none"> — classificationMethod: string; — colorPalette: string; — geographyLevel: string; — numClasses: integer; |
| ThematicVariableInfo | |
| | <ul style="list-style-type: none"> — database: string; — shortDesc: string; — table: string; — variable: string; |

SpatialQueryService

SpatialQuery

- findFeaturesByExtent: Extent; Envelope;
- findQueryOptions: SpatialQueryOptions; string; string; ResultSet;
- findQueryOptionsGeometry: Geometry; Geometry;
- findQueryOptionsGeometry: SpatialQueryOptions; string; ResultSet;
- findFeaturesByPoint: Point; Point;
- findQueryOptions: SpatialQueryOptions; string; string; ResultSet;
- getAvailableFeatureNames: string; string; string; SpatialQueryInfo[];
- getVersion: string;

SpatialQueryInfo

- availableFields: FieldDesc[];
- dataSource: string;

SpatialQueryOptions

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

| |
|---|
| Account Info Service |
| AccountInfo |
| ← getServices: AccountInfoOptions: AccountInfoOptions; token: string; ServiceSummary: I ← getUsage: accountInfoOptions: AccountInfoOptions; dateTimeRange: DateTimeRange; token: string; AccountSummary ← getVersion: string |
| AccountInfoOptions |
| ■ adminOptions: AdminOptions ■ returnDataSources: string[] ■ returnDetails: boolean ■ returnServices: string[] |
| AccountSummary |
| ■ groupUsage: AccountUsage ■ userByUser: UserUsage[] |
| AccountUsage |
| ■ averageDailyCreditUsage: double ■ creditsRemaining: double ■ creditsUsed: double ■ dailyStorageCost: double ■ storageQuota: long ■ diskStorageUsed: long ■ estimatedEndDate: long ■ expiration: long ■ totalStorageCost: double |
| AdminOptions |
| ■ returnGroupUsage: boolean ■ returnUsers: string[] |
| DataSourceInfo |
| ■ name: string ■ summary: string ■ supportsArc4Mac: boolean ■ supportWMS: boolean ■ title: string |
| DataSourceUsage |
| ■ name: string ■ title: string ■ usage: UsageInfo |
| ServiceInfo |
| ■ dataSourceInfo: DataSourceInfo[] ■ desc: string ■ name: string ■ wsdl: string |
| ServiceSummary |
| ■ serviceInfo: ServiceInfo[] ■ user: User |
| ServiceUsage |
| ■ dataSourceUsage: DataSourceUsage[] ■ name: string ■ usage: UsageInfo |
| UsageInfo |
| ■ creditsUsed: double ■ dateLastUsed: long ■ numRequests: long |
| User |
| ■ email: Address: string ■ firstName: string ■ isActive: boolean ■ isAdmin: boolean ■ lastName: string ■ username: string |
| UserUsage |
| ■ accountUsage: AccountUsage ■ servicesUsage: ServiceUsage[] ■ user: User |

| Utility Service | |
|---------------------|---|
| Utility | |
| ← | convertCircleToGeometry circle: Circle numPoints: Integer tokens: string; Geometry |
| ← | convertCircularArcToGeometry arc: CircularArc numPoints: Integer tokens: string; Geometry |
| ← | convertEllipseToGeometry ellipse: Ellipse numPoints: Integer tokens: string; Geometry |
| ← | getBboxProjection extent: Envelope tokens: string; Envelope |
| ← | getDataSourceDateInfo dataSource: string envelope: Envelope mapSize: mapImageSize tokens: string; string |
| ← | getDistance location1: Point location2: Point units: string numPoints: Integer returnGeometry: boolean tokens: string; DistanceInfo |
| ← | getGeometry dataSource: string point: Point interval: Integer units: string tokens: string; Geometry |
| ← | getSupportedProjections type: string tokens: string; KeyValues[] |
| ← | projectExtent extent: Envelope project: CoordSys tokens: string; Envelope |
| ← | projectGeometry geometry: Geometry project: CoordSys tokens: string; Geometry |
| ← | projectPoint point: Point project: CoordSys tokens: string; Point |
| CircularArc | |
| ← | center: Point |
| ← | endAngle: double |
| ← | innerRadius: double |
| ← | outerRadius: double |
| ← | radiusUnits: string |
| ← | startAngle: double |
| ← | useCircle: boolean |
| DistanceInfo | |
| ← | distance: double |
| ← | geometry: Geometry |
| Ellipse | |
| ← | widthUnits: string |
| ← | center: Point |
| ← | majorAxis: double |
| ← | minorAxis: double |
| ← | rotation: double |
| ← | tuffEpsilon: boolean |

Wireless Location Service

WirelessLocation

getDeviceLocation():
Device: MobileDevice[];
deviceLocationOptions:
DeviceLocationOptions,
trackingInfo: DeviceLocationInfo[]

← getDeviceStatus():
Device: MobileDevice[];
token: string; DeviceStatus[]

← getDeviceStopInfo():
DeviceStopInfo: DeviceStopInfo[];
token: string; DeviceStopInfo[]

← getDeviceTripInfo():
DeviceTripOptions: DeviceTripOptions[];
token: string; DeviceTripInfo

← getVersion(): string

← sendSMS():
Device: 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

← dateTimeName: string

← dateTimeRange: DateTimeRange

← maxRadius: double

← minDuration: integer

← radiusUnits: string

DeviceTripInfo

← angle: double

← distance: double

← path: Geometry

← speed: Integer

DeviceTripOptions

← dateTimeName: string

← dateTimeRange: DateTimeRange

← returnAngle: boolean

← returnDistance: boolean

← returnPath: boolean

← returnSpeed: boolean

← units: string

MobileDevice

← dataSource: string

← phoneName: string

| Service Name |
|---|
| Interface key ■— Request Property —■ Response Property ■— Request/Response Property ← Method |

Key

