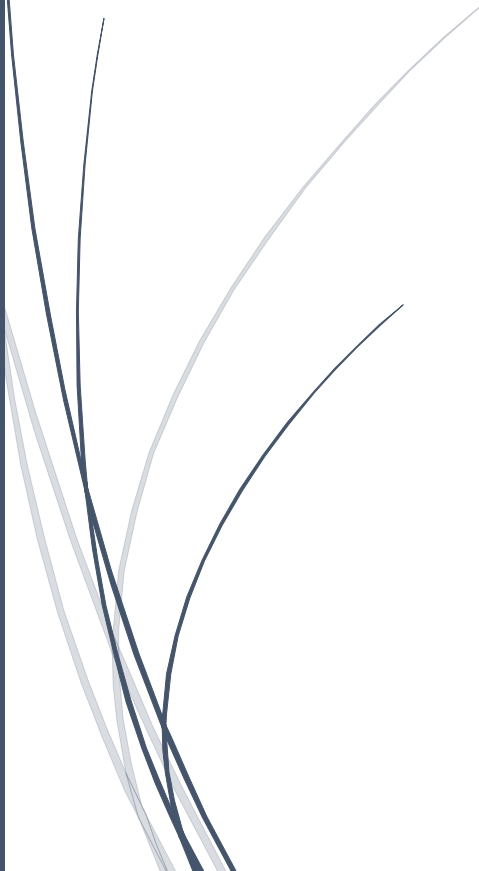


ثبت اطلاعات راهنما

مبتنی بر وب سرویس

راهنمای برنامه نویسان (Developer's Guide)

کد سند: C1.2



فهرست مطالب

مقدمه	۲
۱- چرخه کاری فراخوانی متدها	۲
۲- متد GetHashCode	۳
۱-۲- ساختار گروه راهنما (CMRCategory):	۴
۱-۱-۲- فیلدهای کلاس‌های مرتبط با گروه راهنما (CMRCategory)	۵
۲-۲- ساختار راهنما	۹
۱-۲-۲- فیلدهای جداول مرتبط با راهنما (CMR)	۹
۳-۲- ساختار قلم راهنما	۱۲
۴-۲- پارامترهای خروجی	۱۳
۳- متد Add	۱۴
۱-۳- پارامترهای ورودی	۱۴
۲-۳- پارامترهای خروجی	۱۴
۴- متد Void	۱۶
۱-۴- پارامترهای ورودی	۱۶
۲-۴- پارامترهای خروجی	۱۷
۵- کلاس‌های پایه	۱۸

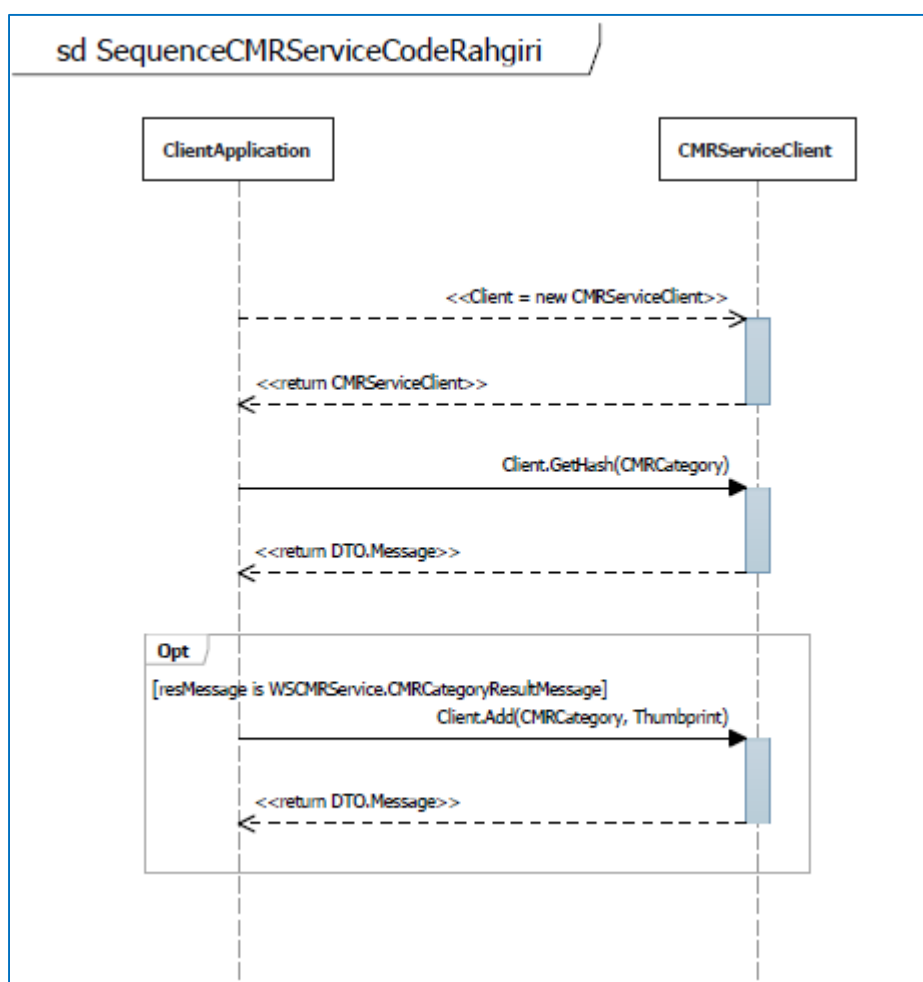
مقدمه

وب سرویس «ثبت اطلاعات راهنما» جهت استفاده شرکت‌های تولیدکننده نرم‌افزار توسط شرکت فن‌آوا سیستم ارائه شده است. با استفاده از این وب سرویس آن دسته از شرکت‌های نرم‌افزاری که برای شرکت‌های حمل و نقل، نرم‌افزار کاربردی تولید می‌کنند، قادر خواهند بود از امکان ثبت راهنما و ابطال آن در «سامانه راهنما برخط» بهره‌مند گردند. این وب سرویس از طریق آدرس زیر قابل دسترسی است:

<http://cmr.rmta.ir/WCFServices/CMRService.svc>

۱- چرخه کاری فراخوانی متدها

با استفاده از وب سرویس CMRService، برنامه نویسان به کلاس‌ها و متدهایی برای ثبت راهنما دسترسی خواهند داشت. متدهای مورد نظر در جدول (۱) ارائه شده است. ساختار کلاس‌های مورد نظر نیز در بخش (۲-۲)، (۳-۲) و (۴-۲) در جدول‌هایی ارائه شده است. برای گرفتن کد رهگیری باید ابتدا متد GetHashCode و سپس متد Add طبق نمودار توالی (۱) فراخوانی گردد.



نمودار ۱: نمودار توالی فراخوانی متدهای وب سرویس CMRService برای گرفتن کد رهگیری

جدول ۱: متدهای CMRService و پارامترهای ورودی و خروجی آنها

ردیف	نام متد	ورودی ها	خروجی	توصیف
۱	GetHash	DTO.CMRCategory T	DTO.Message	مقدار HashData برای CMRCategory را برمیگرداند. امضا (با استفاده از امضای دیجیتال) روی این مقدار انجام می شود.
۲	Add	DTO.CMRCategory input, String TP	DTO.Message	با استفاده از این متد کد رهگیری برای گروه راهنما ارسال شده دریافت می گردد.
۳	Void	(String CMRReferenceNo, String CMRReferenceNoSignature, String TP)	DTO.Message	با استفاده از این متد راهنما مورد نظر ابطال می گردد.

۲-متد GetHash

جهت استفاده از این متد باید تمامی فیلدهای مورد نیاز از ورودی DTO.CMRCategory T مقدار دهی گردد و سپس با استفاده از فراخوانی این متد CMRCatgeory.HashData و پس از آن با استفاده از توکن (سخت افزار امضای دیجیتال) مقدار CMRCategory.Signature مشخص گردد. در نمونه کد شکل (۱) این کارها انجام گرفته است:

```

// فرض می شود که تمامی فیلدهای لازم در
//CMRCategory
// از قبل مقدار دهی شده است
1 WSCMRService.CMRServiceClient client = new WSCMRService.CMRServiceClient();
2 WSCMRService.Message hashmsg = client.GetHash(CMRCategory);
3 if (hashmsg != null && hashmsg is WSCMRService.CMRCategoryResultMessage)
4 {
5     CMRCategory.HashData = (hashmsg as WSCMRService.CMRCategoryResultMessage).Text;
6     byte[] hash = Convert.FromBase64String(CMRCategory.HashData);
7     X509Store store = new X509Store(StoreName.My, StoreLocation.CurrentUser);
8     store.Open(OpenFlags.ReadOnly);
9     X509Certificate2Collection collection = (X509Certificate2Collection)store.Certificates;
10    X509Certificate2Collection fcollection = (X509Certificate2Collection)collection
11        .Find(X509FindType.FindByTimeValid, DateTime.Now, false);
12    X509Certificate2Collection scollection =
13        X509Certificate2UI.SelectFromCollection(fcollection
14            , "Test Certificate Select"
15            , "Select a certificate from the following list to get information on that certificate"
16            , X509SelectionFlag.MultiSelection);
17    var cert = scollection[0];
18    RSACryptoServiceProvider csp = null;
19    csp = (RSACryptoServiceProvider)cert.PrivateKey;
20    if (csp == null)
21        throw new Exception("No valid cert was found");
22    RSAPKCS1SignatureFormatter RSAFormatter = new RSAPKCS1SignatureFormatter(csp);
23    RSAFormatter.SetHashAlgorithm("SHA1");
24    byte[] signedHash = RSAFormatter.CreateSignature(hash);
25    CMRCategory.Signature = Convert.ToBase64String(signedHash);
26 }
27 else if (hashmsg is WSCMRService.ErrorMessage)
28 {
29     throw new Exception((hashmsg as WSCMRService.ErrorMessage).Text);
30 }

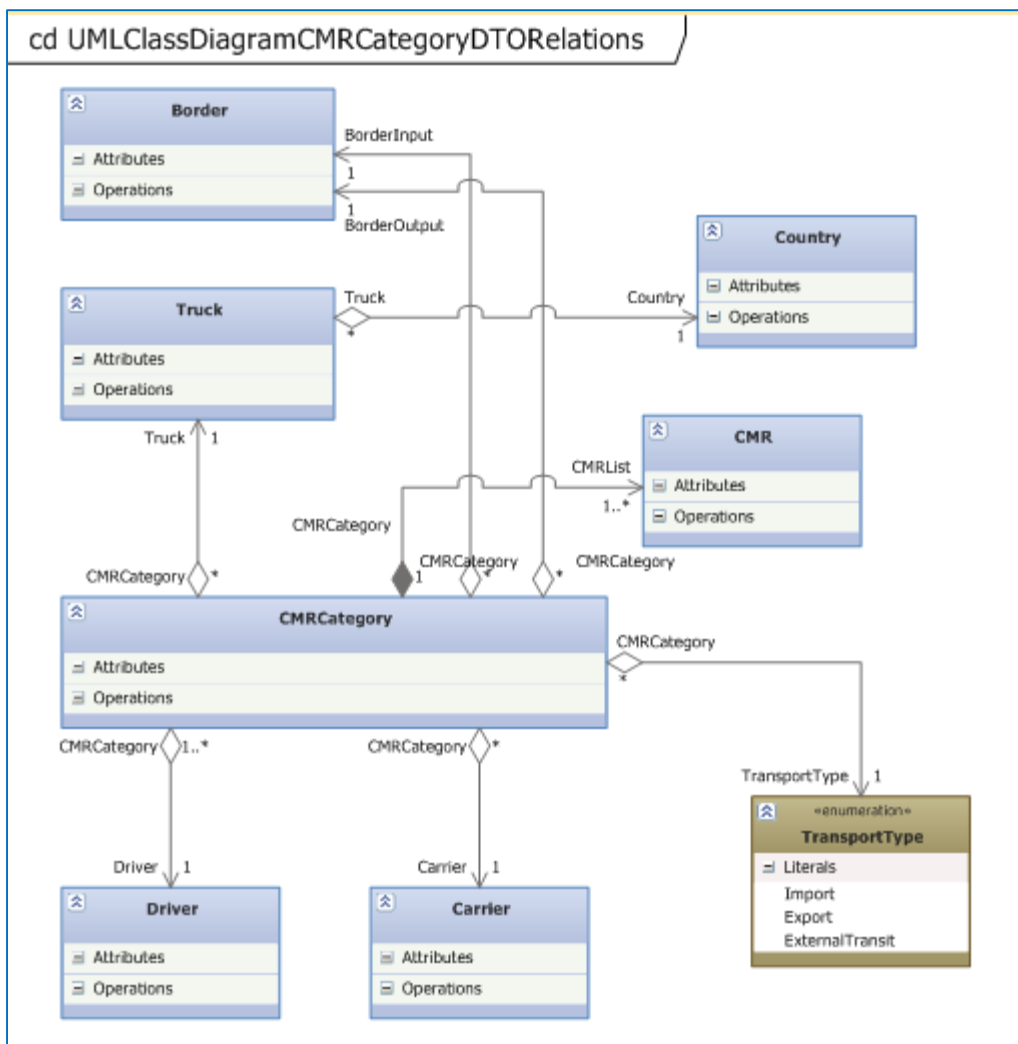
```

شکل ۱: نمونه کد سی شارپ فراخوانی متد *GetHash* و امضا کردن با استفاده از آن

در نمونه کد بالا فرض شده است که تمامی فیلدهای لازم شیء *CMRCategory* به جز فیلدهای *HashData* و *Signature* مقداردهی شده است. با فراخوانی *GetHash* فیلد *HashData* مقداردهی می شود (خط کدهای ۱، ۲، ۳، ۴ و ۵). فیلد *HashData* باید پیش از آن که امضا گردد به فرم *byte array* تبدیل گردد (خط کد ۶). سپس در صورت وصل بودن توکن به کامپیوتر کلاینت باید *Certificate* آن انتخاب گردد (خط کدهای ۷ تا ۱۷). پس از آن باید کلید خصوصی توکن با ورود پین مربوطه استخراج گردد (خط کدهای ۱۸ تا ۲۱). آن گاه باید *byte array* گفته شده امضا گردد. پس از انجام امضا، خروجی به صورت *byte array* خواهد بود که باید با استفاده از متد *Convert.ToBase64String* به فرمت یک رشته *Base64* برده شود و در فیلد *Signature* ذخیره گردد (خط کدهای ۲۲ تا ۲۵). بنابراین فیلد *Signature* نیز اکنون مقداردهی شده است و *CMRCategory* آماده برای ارسال به متد *Add* از وب سرویس *CMRService* است.

۲-۱- ساختار گروه راهنما (CMRCategory):

ساختار گروه راهنما در مقدار برگشتی در دیگرام کلاس (نمودار ۲) نمایش داده شده است. این دیگرام تنها مشخص کننده روابط شیء گروه راهنما است.



نمودار ۲: نمودار کلاس گروه راهنما مشخص کننده روابط بین اشیاء

۲-۱-۱- فیلدهای کلاس‌های مرتبط با گروه راهنما (CMRCategory)

در ادامه فیلدهای جداول مشخص شده در نمودار کلاس (۲) ارائه می شود. از بین این جداول برخی مشخص کننده اطلاعات پایه شامل مرزها (Border) و کشورها (Country) هستند و سایر جداول، جداول اصلی هستند.

جدول ۲: فیلدهای کلاس CMRCategory

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد	مرجع مقادیر
DocumentNo	شماره پرونده	String	
Signature	امضای راهنما	String	
FK_NationalityID	کشور محل صدور	Int?	ID از کلاس پایه Country (ایران ۱۰۳ می باشد)
FK_TransportTypeID	نوع حمل	Int	ID از کلاس پایه TransportType
FK_InputBorderID	مرز ورودی	Int?	ID از کلاس پایه Border
FK_OutputBorderID	مرز خروجی	Int?	ID از کلاس پایه Border
CMRList	لیست راهنماها	List<CMR>	List از کلاس CMR
Driver	راننده	Driver	کلاس Driver
Truck	کامیون	Truck	کلاس Truck
CMRCount	تعداد راهنماهای گروه راهنما	int	

جدول ۳: فیلدهای کلاس Driver

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد	مرجع مقادیر
NameE	نام (لاتین)	String	
NameP	نام (فارسی)	String	
LastNameE	نام خانوادگی (لاتین)	String	

	String	نام خانوادگی (فارسی)	LastNameP
	String	شماره پاسپورت	PassportNo
	String	شماره گواهی نامه	LicenceNo
	String	نام مادر (لاتین)	MotherNameE
	String	نام مادر (فارسی)	MotherNameP
	String	نام پدر (لاتین)	FatherNameE
	String	نام پدر (فارسی)	FatherNameP
	String	شماره کارت هوشمند	ExternalID
Country پایه ID از کلاس پایه	Int?	ملیت راننده	FK_CountryID

جدول ۴: فیلدهای کلاس Truck

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد	مرجع مقادیر
TruckNo	شماره پلاک شهربانی	String	
TransitNo	شماره پلاک ترانزیت	String	
ExternalID	شماره کارت هوشمند	String	
FK_CountryID	ملیت کامیون	Int?	Country پایه ID از کلاس پایه

جدول ۵: فیلدهای کلاس Border (اطلاعات پایه)

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد
ID	ID	int
Code8	کد هشت رقمی مرز	String

String	نام مرز	Name
String	نام استان	ProvinceName

جدول ۶: فیلدهای کلاس Country (اطلاعات پایه)

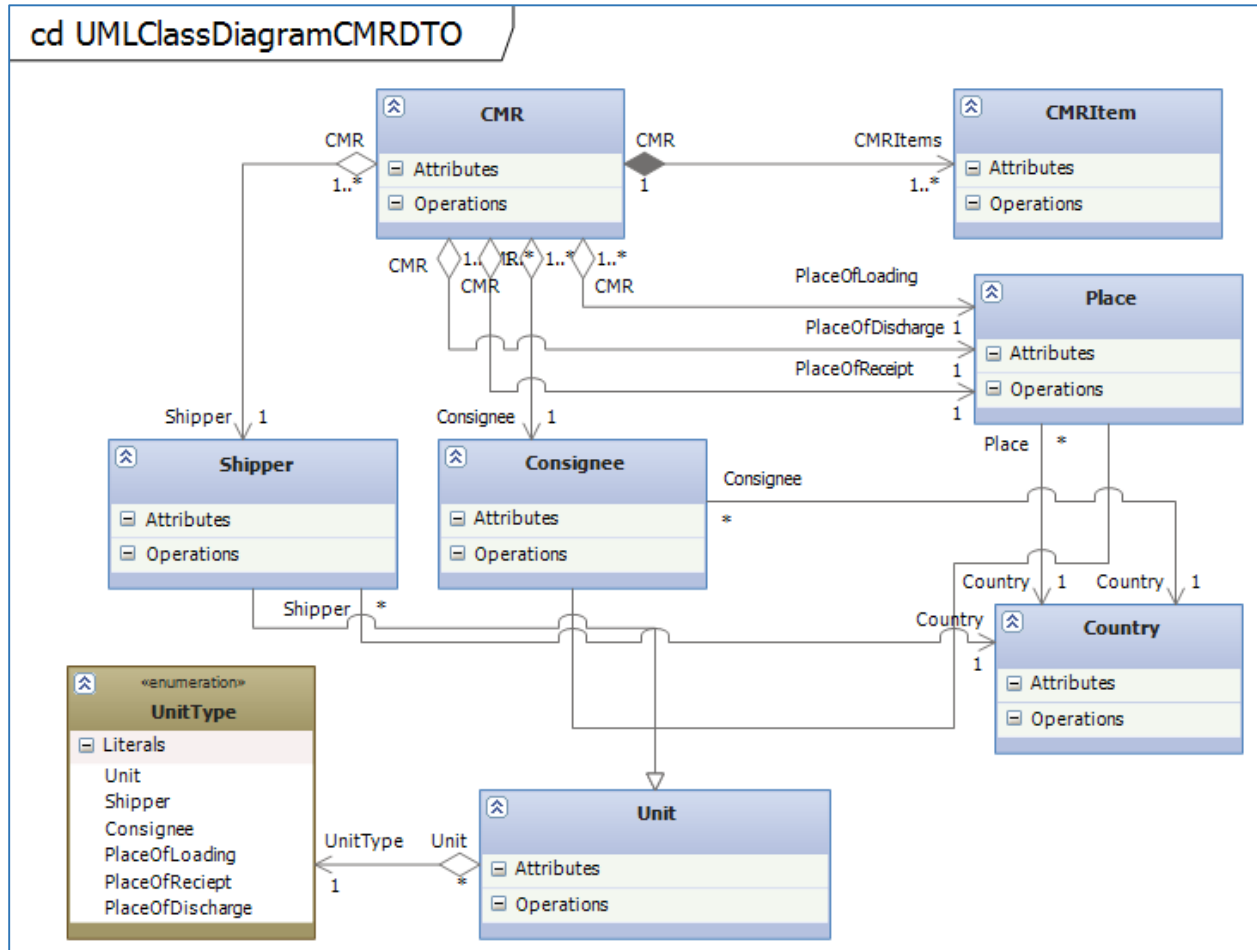
نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد
ID	ID	int
CountryCode	کد کشور	String
CountryNameE	نام کشور به انگلیسی	String
CountryNameP	نام کشور به فارسی	String

جدول ۷: فیلدهای کلاس TransportType (اطلاعات پایه)

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد
ID	عدد enum مربوط به نوع حمل: واردات (۱)، صادرات (۲) و ترانزیت (۴)	int
Name	نوع حمل به انگلیسی	String
Title	عنوان نوع حمل به فارسی	String

۲-۲-۲- ساختار راهنما

نمودار کلاس (۳) مشخص کننده کلاس CMR و روابط آن است.



نمودار ۳: نمودار کلاس راهنما مشخص کننده روابط بین اشیاء

۲-۲-۲-۱- فیلدهای جداول مرتبط با راهنما (CMR)

در ادامه فیلدهای کلاس‌های مشخص شده در نمودار کلاس (۳) ارائه می شود. از بین این کلاس‌ها برخی مشخص کننده اطلاعات پایه شامل کشورها (Country) هستند و سایر کلاس‌ها، کلاس‌های اصلی هستند.

جدول ۸: فیلدهای کلاس CMR

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد	مرجع مقادیر
BLNumber	شماره بارنامه	String	
CarnetTirNo	شماره کارنه تیر	String	
TotalWeight	وزن کلی (مربوط به حالتی است که وزن اقلام به جای آنکه به تفکیک مشخص شده باشد، به صورت وزن کلی شامل جمع وزن اقلام مشخص شده باشد)	Double?	
AttachmentDocuments	اسناد ضمیمه	String	
GoodsDescription	متنی که در راهنما در پایین بخش اقلام نوشته می شود. (فاقد باکس مشخص در راهنما چاپی)	String	
Remark	دستورات فرستنده	String	
IssueDate	تاریخ صدور	DateTime?	
LoadingDate	تاریخ بارگیری	DateTime?	
ReceiptDate	تاریخ دریافت	DateTime?	
PrintSerialNo	شماره سریال کاغذ	String	
ReferenceNo	کد رهگیری	String	
PlaceOfLoading	محل بارگیری	Place	کلاس Place
PlaceOfIssue	محل صدور	Place	کلاس Place

Place کلاس	Place	محل دریافت	PlaceOfReceipt
Place کلاس	Place	محل تخلیه	PlaceOfDischarge
Unit کلاس	Unit	صاحب کالا	Consignee
Unit کلاس	Unit	فرستنده	Shipper
CMRItem کلاس از List	List<CMRItem>	اقلام راهنما	CMRItems

جدول ۹: فیلدهای کلاس Place اثر برنده از کلاس Unit

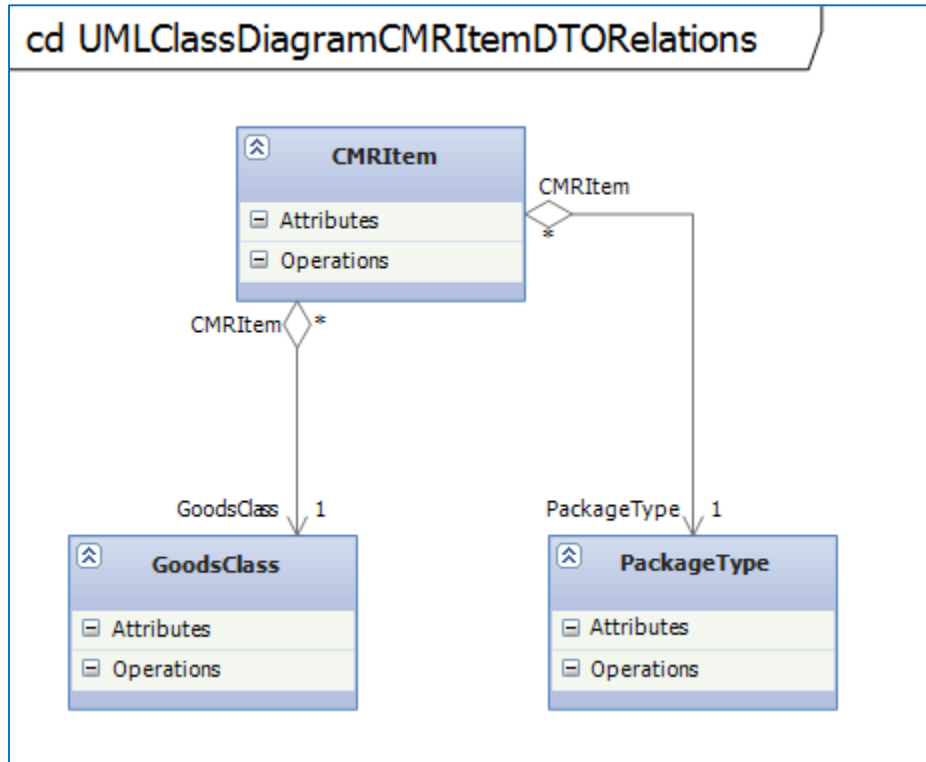
مرجع مقادیر	نوع فیلد	توضیح فیلد اطلاعاتی	نام فیلد اطلاعاتی
	String	توضیح محل صدور	Description
ID از کلاس پایه Country	Int?	کشور صدور	FK_CountryID

جدول ۱۰: فیلدهای کلاس Unit

مرجع مقادیر	نوع فیلد	توضیح فیلد اطلاعاتی	نام فیلد اطلاعاتی
	String	توضیح صاحب کالا	Description
ID از کلاس پایه Country	Int?	کشور صاحب کالا	FK_CountryID

۲-۳- ساختار قلم راهنما

نمودار کلاس (۴) مشخص کننده کلاس CMRItem و روابط آن است.



نمودار ۴: نمودار کلاس قلم راهنما مشخص کننده روابط بین اشیاء

جدول ۱۱: فیلدهای کلاس CMRItem

نام فیلد اطلاعاتی	توضیح فیلد اطلاعاتی	نوع فیلد	مرجع مقادیر
Quantity	تعداد بسته ها	Double?	
GrossWeight	وزن ناخالص	Double?	
Volume	حجم (مترمکعب)	Double?	
StatisticalNumber	شماره آماری	String	
Marks	علامت	String	
KotajNo	شماره کوتاژ	String	

	String	شرح کالا (نوع کالا)	GoodsDescription
PackageType پایه ID از کلاس	Int?	نوع بسته بندی	FK_PackageTypeID
GoodClass پایه ID از کلاس	Int?	کلاس کالای خطرناک	FK_GoodsClassID
	String	کد HS کالا	HSCODE

جدول ۱۲: کلاس PackageType (کلاس پایه)

نوع فیلد	توضیح فیلد اطلاعاتی	نام فیلد اطلاعاتی
int	ID	ID
String	کد نوع بسته بندی	PackageTypeCode
String	نام نوع بسته بندی	PackageTypeName

جدول ۱۳: فیلدهای کلاس GoodClass (کلاس پایه)

نوع فیلد	توضیح فیلد اطلاعاتی	نام فیلد اطلاعاتی
int	ID	ID
String	کد کلاس کالا	Code
String	نام کلاس کالا	Name

۲-۴- پارامترهای خروجی

با فراخوانی متد GetHashCode ، خروجی به صورت یک شیء Message (پیام) خواهد بود و این پیام یکی از انواع زیر را خواهد داشت:

۱- ErrorMessage: این پیام برگرداننده خطای اتفاق افتاده در سمت سرور در فیلد Text آن است.

۲- **CMRCategoryResultMessage**: در این پیام فیلد **CMRCategory** حاوی اطلاعات گروه راهنما از جمله کد (های) رهگیری دریافتی برای گروه راهنما است. در صورتی که پیغام خطا نباشد باید به **CMRCategoryResultMessage** تبدیل (cast) گردد، تا بتوان اطلاعات گروه راهنما استعمال شده را از درون آن استخراج نمود. ساختار انواع **Message** در نمودار کلاس (۵) نمایش داده شده است.

۳-متد Add

جهت ثبت راهنما از متد **Add** با پارمترهای مشخص شده در جدول زیر استفاده کنید. ورودی **input** از نوع **CMRCategory** است و ساختار آن در بخش (۲) متد **GetHash** توضیح داده شد.

۳-۱-پارمترهای ورودی

جدول ۱۴: پارامترهای ورودی متد **Add**

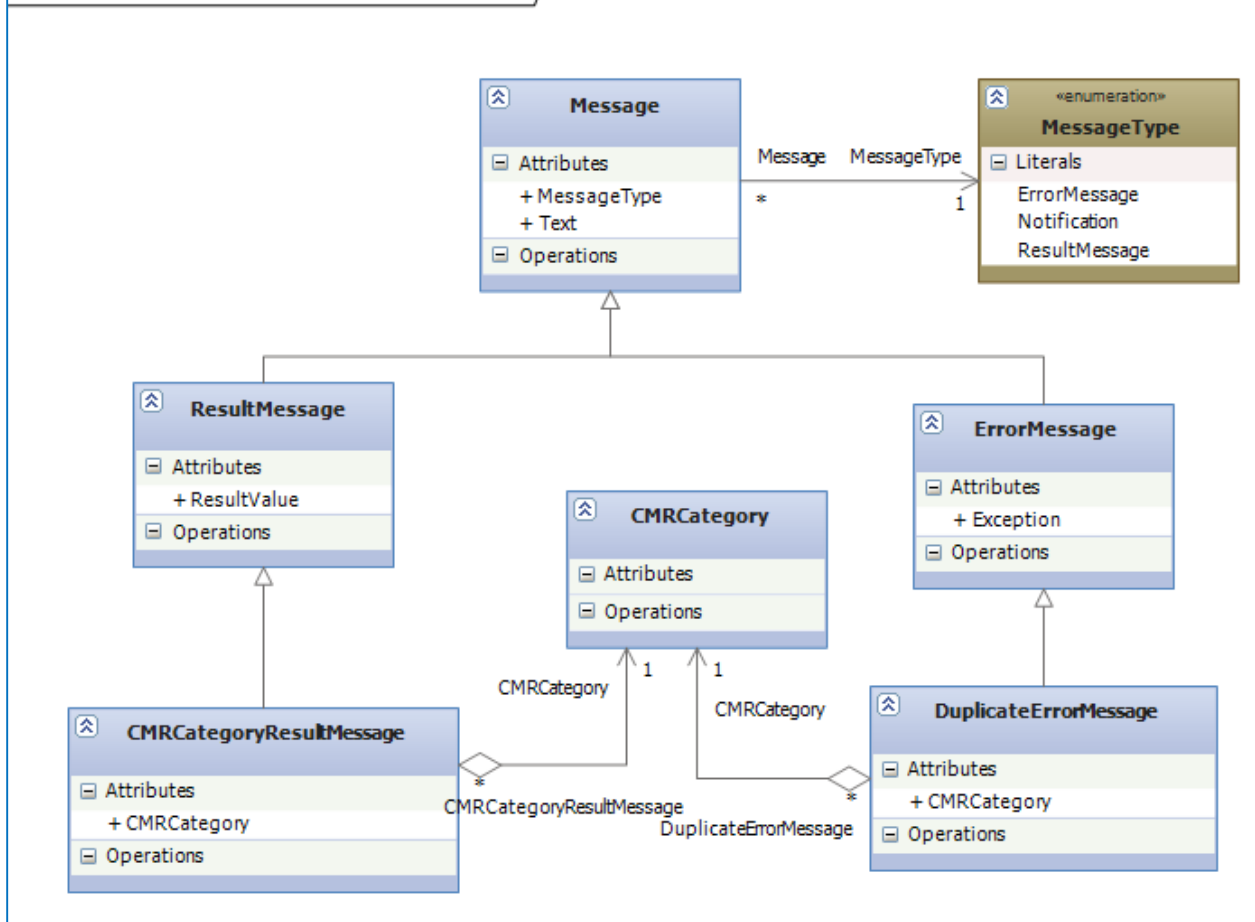
ردیف	پارامتر	نوع	توضیح
۱	input	DTO.CMRCategory	گروه راهنما به همراه تمامی راهنماها و فیلدهای آن
۲	TP	String	Thumbprint مشخصه توکن شرکت حمل و نقل

۳-۲-پارامترهای خروجی

با فراخوانی متد **Add**، خروجی به صورت یک شیء **Message** (پیام) خواهد بود و این پیام یکی از انواع زیر را خواهد داشت:

- ۱- **ErrorMessage**: این پیام برگرداننده خطای اتفاق افتاده در سمت سرور در فیلد **Text** آن است.
- ۲- **DuplicateErrorMessage**: این پیام به معنای تکراری بودن گروه راهنما ارسالی است. در فیلد **CMRCategory** از این **Message**، گروه راهنما تکراری موجود برگردانده می شود.
- ۳- **CMRCategoryResultMessage**: در این پیام فیلد **CMRCategory** حاوی اطلاعات گروه راهنما از جمله کد (های) رهگیری دریافتی برای گروه راهنما است. در صورتی که پیغام خطا نباشد باید به **CMRCategoryResultMessage** تبدیل (cast) گردد، تا بتوان اطلاعات گروه راهنما استعمال شده را از درون آن استخراج نمود. ساختار انواع **Message** در نمودار کلاس (۵) نمایش داده شده است.

cd UMLClassDiagramMessages



نمودار ۵: نمودار کلاس انواع *Message* ها

در شکل (۲) نمونه کد سی شارپ فراخوانی متد Add ارائه شده است.


```

// فرض می شود که تمامی فیلدهای لازم در
//CMRCategory
// از قبل مقدار دهی شده است
WSCMRService.Message resMsge = client.Add(CMRCategory, TP);
List<String> coderahgiris = new List<string>();
if (!(resMsge is WSCMRService.ErrorMessage))
{
    var rescategory = (resMsge as WSCMRService.CMRCategoryResultMessage).CMRCategory;
    foreach (var cmr in rescategory.CMRList)
    {
        coderahgiris.Add(cmr.ReferenceNo);
    }
    TCodeRahgiri.Text = coderahgiris.Aggregate((p, q) => p + Environment.NewLine + q);
}
else if (resMsge is WSCMRService.DuplicateErrorMessage)
{
    throw new Exception((resMsge as WSCMRService.DuplicateErrorMessage).Text);
}
else
{
    throw new Exception((resMsge as WSCMRService.ErrorMessage).Text);
}

```

شکل ۲: نمونه کد سی شارپ فراخوانی متد Add

۴-متد Void

با استفاده از متد Void می توان با داشتن کد رهگیری، راهنما صادر شده را ابطال نمود.

۴-۱- پارامترهای ورودی

جدول ۱۵: پارامترهای ورودی متد Void

ردیف	پارامتر	نوع	توضیح
۱	CMRReferenceNo	String	کد رهگیری
۲	CMRReferenceNoSignature	String	رشته‌ی امضای کد رهگیری با استفاده از امضای دیجیتال شرکت حمل
۳	TP	String	Thumbprint مشخصه توکن شرکت حمل و نقل

۴-۲- پارامترهای خروجی

با فراخوانی متد Void ، خروجی به صورت یک شیء Message (پیام) خواهد بود و این پیام یکی از انواع زیر را خواهد داشت:

۱- ErrorMessage: این پیام برگرداننده خطای اتفاق افتاده در سمت سرور در فیلد Text آن است.

۲- ResultMessage: در این پیام فیلد ResultValue حاوی اطلاعات گروه راهنما است.

در شکل (۳) نمونه کد سی شارپ فراخوانی متد Void ارائه شده است.

```
byte[] data = new byte[coderahgiri.Length * sizeof(char)];
System.Buffer.BlockCopy(coderahgiri.ToCharArray(), 0, data, 0, data.Length);
SHA1 sha = new SHA1CryptoServiceProvider();
byte[] result = sha.ComputeHash(data);
X509Certificate2Collection collection = (X509Certificate2Collection)store.Certificates;
X509Certificate2Collection fcollection = (X509Certificate2Collection)collection
    .Find(X509FindType.FindByTimeValid, DateTime.Now, false);
X509Certificate2Collection scollection =
    X509Certificate2UI.SelectFromCollection(fcollection
        , "Test Certificate Select"
        , "Select a certificate from the following list to get information on that certificate"
        , X509SelectionFlag.MultiSelection);
var cert = scollection[0];
RSACryptoServiceProvider csp = null;
csp = (RSACryptoServiceProvider)cert.PrivateKey;
if (csp == null)
    throw new Exception("No valid cert was found");
RSAPKCS1SignatureFormatter RSAFormatter = new RSAPKCS1SignatureFormatter(csp);
RSAFormatter.SetHashAlgorithm("SHA1");
byte[] signedHash = RSAFormatter.CreateSignature(result);
var signature = Convert.ToBase64String(signedHash);
WSCMRService.CMRServiceClient client = new WSCMRService.CMRServiceClient();
WSCMRService.Message voidMsg = client.Void(coderahgiri, signature, cert.Thumbprint);
if (voidMsg is WSCMRService.ResultMessage)
{
    var cmr = ((voidMsg as WSCMRService.ResultMessage).ResultValue) as WSCMRService.CMR;
    MessageBox.Show("با موفقیت انجام شد " + " " + cmr.ReferenceNo + " " + "ابطال",
        "ابطال", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    throw new Exception(((voidMsg as WSCMRService.ErrorMessage).Text));
}
```

شکل ۳: نمونه کد سی شارپ فراخوانی متد Void

۵- کلاس‌های پایه

همان‌طور که در ساختار مشخص شده، کلاس‌های پایه Country، Border، PackageType، GoodsClass دارای مقادیر (تقریباً) ثابتی هستند که ID آنها به عنوان فیلد برای موجودیت‌های دیگر همچون Country برای CMRCategory، Border برای CMRCategory، PackageType برای CMRItem و GoodsClass برای CMRItem در فیلدهای برگشتی موجود است. اما اگر بخواهیم بدانیم که مثلاً Border با ID= 24 مشخص کننده کدام مرز کشور است باید از وب سرویس‌هایی که برگرداننده مقادیر جداول پایه است استفاده کنیم. آدرس وب سرویس مورد نظر به صورت زیر است:

<http://cmr.rmta.ir/WCFServices/LookupService.svc>

متمدهای ارائه شده در این وب سرویس طبق جدول (۱۶) مشخص شده است:

جدول ۱۶: متمدهای برگرداننده مقادیر کلاس‌های پایه از LookupService

ردیف	نام متد	نوع خروجی	توصیف
۱	FindAllCountries()	List<Country>	اشیای تمامی کشورهای قابل انتخاب را برمی‌گرداند.
۲	FindAllBorders()	List<Border>	اشیای تمامی مرزهای قابل انتخاب را برمی‌گرداند.
۳	FindAllPackageTypes()	List<PackageType>	اشیای تمامی انواع بسته بندی قابل انتخاب را برمی‌گرداند.
۴	FindAllProvinces()	List<Province>	اشیای تمامی استان‌های قابل انتخاب را برمی‌گرداند.
۵	FindAllTransportTypes()	List<TransportType>	اشیای تمامی انواع ترافیک قابل انتخاب را برمی‌گرداند.
۶	FindAllCities()	List<City>	اشیای تمامی شهرهای قابل انتخاب را برمی‌گرداند.
۷	FindAllBorderTypes()	List<BorderType>	اشیای تمامی انواع مرزهای قابل انتخاب را برمی‌گرداند.
۸	FindAllGoodsClasses()	List<GoodClass>	اشیای تمامی کلاس‌های کالای خطرناک قابل انتخاب را برمی‌گرداند.
۹	FindAllGoods()	List<Goods>	اشیای تمامی کدهای HS قابل انتخاب را برمی‌گرداند.
	GetCurrentVersions()	List<LookupVersion>	اشیای ورژن فعلی جداول lookup را برمی‌گرداند. (هر تغییری که در داده‌های موجود در جداول پایه داده می‌شود ورژن آن جدول پایه تغییر می‌کند.)

در نمونه زیر با کد سی شارپ متد FindAllBorders فراخوانی شده است:

```
WSLookupService.LookupServiceClient client = new WSLookupService.LookupServiceClient();  
List<WSLookupService.Border> borders = client.FindAllBorders();
```

شکل ۴: نمونه کد فراخوانی متد FindAllBorders برگرداننده تمامی اشیای Border