Easybook.Net
Service API

Document information	
Document version:	1.37
Changed date:	30-09-2015
Print date:	30-09-2015
Author	Allan Rasmussen

Technical Software Engineering
Banegaardsgade 2, 6440 Augustenborg
http://www.tse.dk

Table of Contents

1 REVISION PAGE

Project information

System name:	Easybook.net
Project Id:	EB05020106

Change log:

Version	Date	Author	Comment
1.36	13-08-2014	Allan Rasmussen	Added new service method: RentalObjectsRooms (or just “room” for REST edition)
1.35	10-03-2014	Allan Rasmussen	Added support for time in ChangeTimeStamp input arguments See chapter 4.1.
1.34	10-02-2014	Allan Rasmussen	Edited OrderList service method chapter. DateFrom and DateTo filters on arrival date and not created date.
1.33	27-11-2013	Allan Rasmussen	Added a new service method: OrderList. This method is used to retrieve all orders generated from the partner calling the service method. This method works for partners with and without contract fulfillment obligations. Added a “ProvisionAmount” to the OrderWeb class returned when submitting a booking. This will enable partners to see what the provision of the booking is immediately after placing the booking.
1.32	22-03-2013	Peter Hoffmann	Added the property “PriceIntervalDate” to the class “ArticleBaseWeb”. This property is used to specify an alternative price interval for ticket articles in service methods “Booking” and “UpdateOrder” criteria arguments.
1.31	06-02-2013	Allan Rasmussen	Added the property “NumberOfPersons” to the class “AvailabilityCriteria”. This is used to correctly calculate person related discounts when available.
1.30	10-12-2012	Allan Rasmussen	Added new service method: RentalObjectsArticles. Extended the AvailabilityResult class with some comments text for the rental object. Included documentation on how to use delta enabled feeds (i.e. only get changes since DD-MM-YYYY).
1.29	10-11-2012	Allan Rasmussen	Added new “Attribute” services method. Returns all available rental object attributes. Improved the documentation regarding Booking with the REST enabled service.
1.28	01-11-2012	Allan Rasmussen	Updated Price feed and Availability service method with a new CleaningOptionsvalue that replaces the old CleaningMandatory value.
1.27	24-10-2012	Allan Rasmussen	Updated section 4.7 calendar feed with more detailed information about the two types of booking and blocked periods. Updated 3.2.3 Authentication with information about a required domain name when authenticating using a SOAP based client.
1.26	19-09-2012	Allan Rasmussen	All feeds and booking methods are now REST enabled. Updated the documentation to reflect this.

2 PURPOSE

The purpose of this document is to describe the service methods made available for partners to retrieve data and perform online bookings in EasyBook.NET on the URL test.easybook.net.

2.1 Online Documentation

An online documentation site is available. Please visit http://docs.appl.easybook.net for more details. A zip file containing some sample code is available here: http://docs.appl.easybook.net/content/samples.zip. The samples will typically demonstrate how to call the different service endpoints.

2.2 Concepts

Consumer:
The company whose rental houses are maintained in EasyBook.NET

Partner:
The company reselling rental houses on the behalf of a consumer.

3 COMMON CONSIDERATIONS

3.1 Standards

All partner feeds and booking related methods are available in 2 different editions:

- WCF/SOAP Web service
- REST Service

Sections 3.2 & 3.3 discuss the 2 different methods of accessing the partner services. Whether to combine the two possibilities or chose a single method of accessing the service endpoints is entirely left up to the partner.

3.2 WCF/SOAP Web service

The 2 service endpoints available, BookingWcfService and BookingAuthenticateWcfService, adhere to WS web service specification: “WS Basic Profile Version 1.1”. Both partner service endpoints also support the Microsoft WCF framework.

3.2.1 SERVICE BOOKINGWCFSERVICE

The following SOAP headers must be added before calling any methods on the service BookingWcfService:

1. ConsumerId, type integer. This header must specify the consumer for which to retrieve data.
2. Culture, type string. This header specifies the customer language code according to the ISO 639-1 standard. E.g. da=Danish, en=English, de=German etc. It is only relevant for specific service methods returning data in only one customer language. If this header is omitted, the customer language will default to English.
3. Accept-Encoding: gzip. See also section 3.5 Compression.

Please note that when adding headers to the request, the namespace “http://easybook.net/services/2007/01” must be provided. See Samples.zip for an example.

Service address:
http://test.easybook.net/BookingWcfService.svc/basicHttpBinding
WSDL for the service can be retriewed on the following address:
http://test.easybook.net/BookingWcfService.svc?wsdl

See also section 4 Partner Feeds.

3.2.2 SERVICE BOOKINGAUTHENTICATEWCFSERVICE

The following SOAP headers must be added before calling any methods on the service BookingAuthenticateWcfService:

1. ConsumerId, type integer. This header must specify the consumer for which to retrieve data. This header is mandatory and if it is omitted, the service will return an error se section “3.8 Error Handling”.
2. Culture, type string. This header specifies the customer language code according to the ISO 639-1 standard. E.g. da=Danish, en=English, de=German etc. It is only relevant for specific service methods returning data in only one customer language. If this header is omitted, the customer language will default to English.

Please note that when adding headers to the request, the namespace “http://easybook.net/services/2007/01” must be provided. See Samples.zip for an example.

Service address:
https://test.easybook.net/BookingAuthenticateWcfService.svc/basicHttpBinding
WSDL for the service can be retriewed on the following address:
https://test.easybook.net/BookingAuthenticateWcfService.svc?wsdl

3.2.3 AUTHENTICATION

BookingAuthenticateWcfService is only accessible via SSL(Secure Socket Layer) and the caller must identify oneself with a partner login, password and domain name using basic www-authentication. The domain name should be set to “ngintern” to ensure proper authentication.

3.2.4 ACCESSING SERVICES USING A WCF CLIENT

To get the full benefit of the WCF technology, the client application can use a Windows Communication Foundation Client to communicate with the EasyBook.NET services. To create a WCF client in C#, the Microsoft “svcutil.exe” program can be used:

Windows Command Prompt:

```
svcutil.exe /language:cs /out:MyProxy.cs /config:app.config
http://test.easybook.net/BookingWcfService.svc
http://test.easybook.net/BookingAuthenticateWcfService.svc
```

(Please note there are no line breaks in the above command statement)

3.3 REST Service

All partner feeds and booking related methods are also available as a simpler RESTful service. A RESTful service can be called using normal HTTP and requires no additional messaging layer like SOAP or WCF. Data feeds from a RESTful service can be retrieved using nothing else than a web browser. EasyBook.NET supports two different messageformats for Restful services:

- XML
- JSON

JSON is a more compact format, which results in a smaller amount of data transferred compared to XML. It is entirely up to the partner to descide which message format to use.

3.3.1 FEEDS

Data feeds located at http://test.easybook.net/feeds can be called without any special consideration. However, compression must be allowed by the caller. See section 3.5 Compression for details.

Example of a call to a data feed using HTTP GET

```
http://test.easybook.net/feeds/{format}/{Partner Code}/{Consumer Id}/{Method}?QuerystringArguments 0..n
http://test.easybook.net/feeds/json/Partner1/1/calendar?countryisocode=DK&culture=da
The above URL returns all calendars for rental houses located in Denmark for consumer ID 1 in the format: JSON.
```

The HTTP URL is comprised of four dynamic path elements:

- **Format**: The type of message format. Both XML and JSON are supported.
- **Partner Code**: A code uniquely identifying the partner calling the feed.
- **Consumer ID**: The company whose rental houses are maintained in EasyBook.NET.
- **Method**: The name of the feed or booking method requested.

Additionally, 0..n querystring arguments may be supplied to the feed to control what is returned from the method.

3.3.2 BOOKING

Methods related to booking are located at https://test.easybook.net/booking. These methods can only be called using HTTPS. In some cases the HTTP verbs POST/UPDATE/DELETE/ are also used.

Example of a call to a booking method using HTTPS

```
https://test.easybook.net/booking/{format}/{Partner Code}/{Consumer Id}/{Method}?QuerystringArguments 0..n
https://test.easybook.net/booking/json/Partner1/1/booking
```

3.3.3 BASIC ACCESS AUTHENTICATION

All booking related RESTful services are protected using Basic Authentication as well as SSL. To access these services, add an “Authorization” header to the HTTPS request. Example of an authorization header: “Authorization: Basic QWxhZGluOnNlc2FtIG9wZW4=”.

To learn more about how to access content protected by basic authentication, please visit the following wikipedia topic: http://en.wikipedia.org/wiki/Basic_access_authentication.

3.3.4 DOCUMENTATION

Besides the documentation in this document and the site http://docs.appl.easybook.net, All RESTful services are also documented at the following two locations:

- http://test.easybook.net/feeds/help
- https://test.easybook.net/booking/help (username/password is required).

See also section 4 Partner Feeds for a full list of HTTP/HTTPS URL’s and their corresponding arguments.

3.4 Language support

All services support multiple languages. For a list of supported languages please contact the data provider (consumer). A service always replies in the language requested in the service header. For WCF endpoints, this is done by inserting the “Culture” header value with the desired ISO 639-1 language code. See also section 3.2.1 & 3.2.2. For RESTful endpoints, the same can be achieved by appending the “&culture=xx” querystring argument to the end of the partner feed URL.

If the partner operates with multiple language support, then the webservice “Localization” in section 4.2 should be utilized. This service simply returns all localizable strings available from the partner feeds. So the actual feeds **must only** be called a single time with a single culture and subsequently all localizable strings can be retrieved separately using the more lightweight “Localization” service endpoint. For more details see section “4.2 Localization”.

3.5 Compression

The caller **must** request compression for the outgoing data on all service methods by adding the http header “Accept-Encoding: gzip”. Without this header, the server will deny any incoming request (Errortype: Bad Request). This is done to ensure efficient bandwidth usage on large requests. Please note that the server may opt **not** to compress outgoing data depending on the current server workload and size of the outgoing content. Currently, content smaller than 4096 bytes will not be compressed by the server. However, the caller must be prepared to handle both compressed and uncompressed content.

C# Example of a web request using automatic decompression:

```
System.Net.HttpWebRequest webRequest =
(System.Net.HttpWebRequest)System.Net.WebRequest.Create("http://test.easybook.net/feeds/json/IH/1/calendar");
webRequest.AutomaticDecompression = System.Net.DecompressionMethods.GZip;
System.Net.WebResponse webResponse = webRequest.GetResponse();
System.IO.StreamReader responseStream = new System.IO.StreamReader(webResponse.GetResponseStream());
```

3.6 Common base class for data contract classes

Class DataContractEntityWeb:
Common base for all data entities and are always identified via the Id field. The Id is unique across all instances of a class type.

3.7 Test environment

EasyBook.NET has a completely separate test environment where integration testing can be performed without disturbing the production environment.

The test environment is located at: http://test.easybook.net. All service endpoints and means of communicating with EasyBook.NET is duplicated on the test environment.

Service address:
http://test.easybook.net/BookingWcfService.svc/basicHttpBinding
https://test.easybook.net/BookingAuthenticateWcfService.svc/basicHttpBinding

WSDL for the test services can be retriewed on the following address:
http://test.easybook.net/BookingWcfService.svc?wsdl
https://test.easybook.net/BookingAuthenticateWcfService.svc?wsdl

The switch to production servers should only occur when the integration is verified to be working as intented.

3.8 Error Handling

If an error occurs during a call to a service method, the server will abort the call and return an WS Fault to the caller. This fault should be handled by the caller. Particularly in service methods where a fault can be expected, such as BookingAuthenticateWcfService.Booking. To help the caller differentiate between the different error types, the fault contains an error code. The error code is an enumeration of error codes (see table below).

Some errors are global and can be returned by any service method. Others are only returned by a specific service method. The error codes that are specific to a single service method can be easily identified since the service method name is prefixed to the error code name.

C# Example of calling BookingAuthenticateWcfService.Booking and handling errors:

```
try
{
    OrderWeb bookingResult = agent.Proxy.Booking(criteria);
}
catch (FaultException<Fault> ex)
{
    switch (ex.Detail.ErrorCode)
    {
        case ErrorCodeEnum.Error:
            //An unexpected error occured
            break;
        case ErrorCodeEnum.InvalidRentalObjectNo:
            //The provided rental object number was invalid
            break;
        case ErrorCodeEnum.InvalidCurrencyIsoCode:
            //Unknown currency code
            break;
        case ErrorCodeEnum.InvalidCountryIsoCode:
            //Unknown customer country code
            break;
        case ErrorCodeEnum.BookingInvalidBankInfo:
            //The provided bank account details could not be validated
            break;
        case ErrorCodeEnum.BookingInvalidCustomerPerson:
            //Invalid customer details if the customer is a person
            break;
        case ErrorCodeEnum.BookingInvalidCustomerCompany:
            ////Invalid customer details if the customer is a company
            break;
        case ErrorCodeEnum.BookingInvalidMediaCode:
            //Invalid or unknown media code
            break;
        default:
            break;
    }
}
```

3.8.1 ERROR CODES LIST

For a full list of error codes, please refer to the online documentation:
http://docs.appl.easybook.net/html/T_Tse_EasyBook_Server_DataContracts_Faults_ErrorCodeEnum.htm

4 PARTNER FEEDS

Feeds are service methods that return the data that partners must integrate into their respective backends. Feeds include rental houses details, prices and calendar data. Feeds only contains data for “active” rental objects. Active rental objects are defined as being available for booking from today and forward. Rental objects with a fully booked calendar are still considered active.

As discussed in section 3, EasyBook.NET supports 2 methods of accessing feeds:

- Using the methods exposed in the web service: BookingWcfService
- Using normal HTTP GET: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/{Method}

In both cases incoming compression must be turned on by the caller.

4.1 Delta enabled feeds

Most datafeeds support deltas. A delta means that the feed only returns data that have changed since the service was called the last time. This greatly diminishes both bandwith requirements as well as synchronization tasks on the caller side. TSE recommends that the delta feature is used whenever possible.

To use this feature, simply pass in the argument called “ChangeTimeStamp=dd-MM-yyyy HH:mm:ss” to only receive changes on or after this date and time. The time part is optional. The following feeds support deltas:

- Rental Objects
- Pictures
- Rental Object Text
- Rental Object Numbers
- Calendar

4.2 Localization

WCF/SOAP End Point: BookingWcfService. Localization	**Details** See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URI to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/localization
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/LocalizationJson XML: http://test.easybook.net/feeds/help/operations/LocalizationXml
Authentication required	No.
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_Localization.htm
Authorization required	No.
Input arguments	None.
Output	A LocalizationWeb object, see “Figure 1: Localization”.

See also section “3.4 Language support”. This method returns translated texts in all available languages for the entities containing string properties which are translated. Hence if a class is derived from DataContractEntityWeb and contains one or more translated text properties the translations are available as LocalizedEntityWeb list on Localization object. For example, if a translation is necessary for the property AttributeWeb.Name on a AttributeWeb object, the list Localization.AttributeNames will contain a LocalizedWeb item with the same id as the AttributeWeb object and the items list Localization.AttributeNames[x].Texts contains the translations for the name. For the details for AttributeWeb class refer to “Figure 3: RentalObjectWeb”.

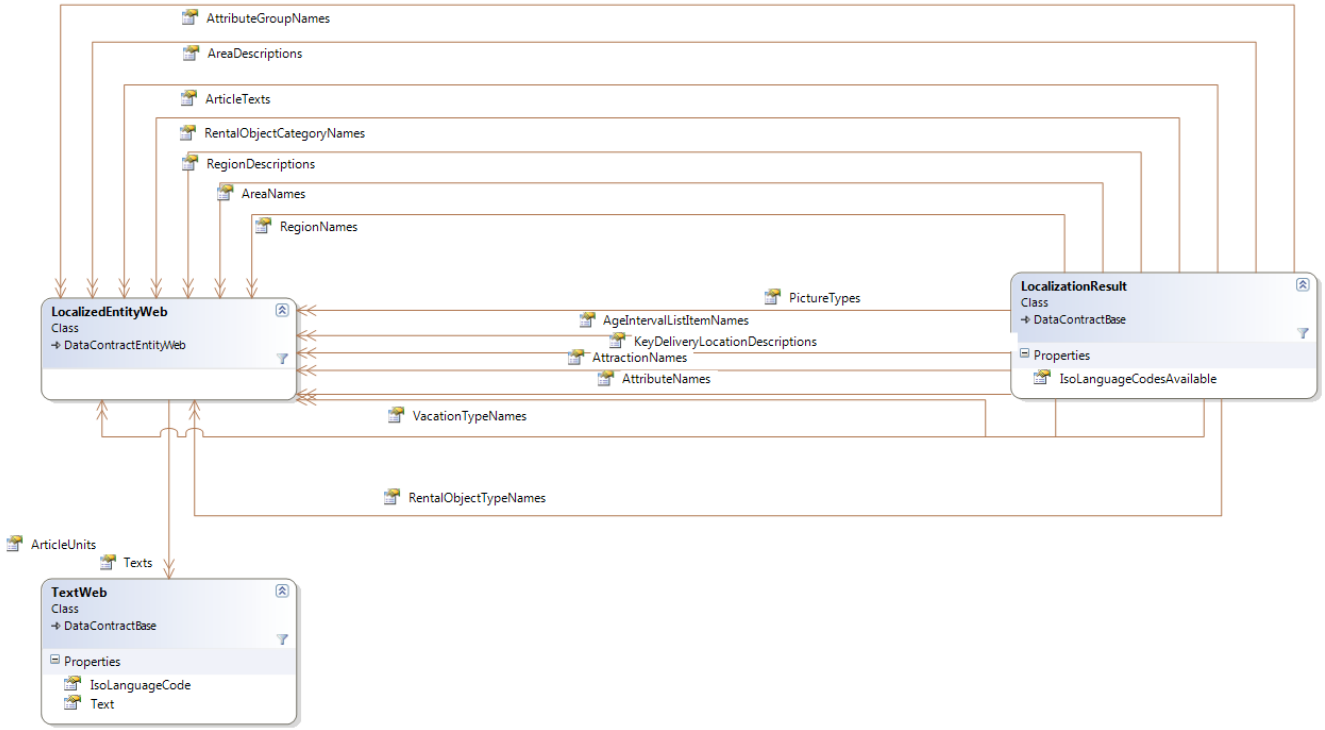

Figure 1: Localization

4.3 Rental Objects

WCF/SOAP End Point:	Details
BookingWcfService.RentalObjects	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/rentalobject?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/RentalObjectJson XML: http://test.easybook.net/feeds/help/operations/RentalObjectXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjects.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectWeb objects, one for each requested house. See “Figure 3: RentalObjectWeb”.

This method returns data for a range of houses containing basic house information.

The range of houses is determined by the properties on the base class RentalObjectsIntervalCriteria, see “Figure 2: RentalObjectsIntervalCriteria”.

Description of range properties:

Name	Description
DateFrom	Start date for the date interval for which to return house data. This property is mandatory.
DateTo	End date for the date interval for which to return house data. This property is mandatory.
CountryIsoCode	CountryIsoCode is the country code according to the ISO 3166-1 alpha 2 standard. This property is optional. If it is filled out, the range of houses, is limited to houses from the specified country.
RegionIds	A list of region ids for limiting the range of houses to specific regions. This property is optional. The possible region ids per country can be retrieved with the method Geography, see section “Geography”.
RentalObjectNoStart	The first number of a interval of rental object numbers. This property is optional. If it is filled out, the range of houses is limited to houses with rental object numbers occurring after the specified number in an ascending alphanumeric sorting of all rental object numbers. A list of all rental object numbers can be retrieved with the method RentalObjectsNumber, see section “Rental Object Number”.
RentalObjectNoEnd	The last number for a interval of rental object numbers This property is optional. If it is filled out, the range of houses is limited to houses with rental object numbers occurring before the specified number in an ascending alphanumeric sorting of all rental object numbers.

This date interval(DateFrom – DateTo) can overlap any number of seasons. The range properties are mutual excluding. E.g., if a country code for country A and region ids from country B are specified, no houses will be returned.

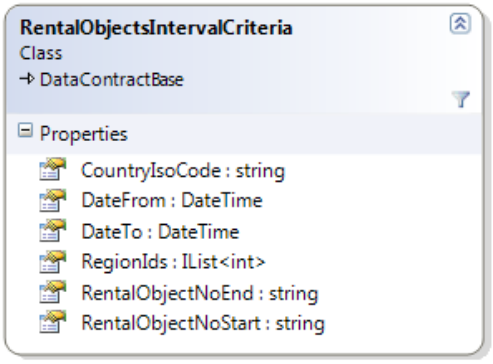

Figure 2: RentalObjectsIntervalCriteria

The AttributeValues returned by the method contains all attributes for a house. E.g. “maximum number of persons”, pets, distance to the ocean/sea etc. A specific attribute is identified by the ArticleAttributeValueWeb.Id property. The interpretation of the values can be determined by the properties ArticleAttributeValueWeb.ValueType and ArticleAttributeValueWeb.ComparerType.

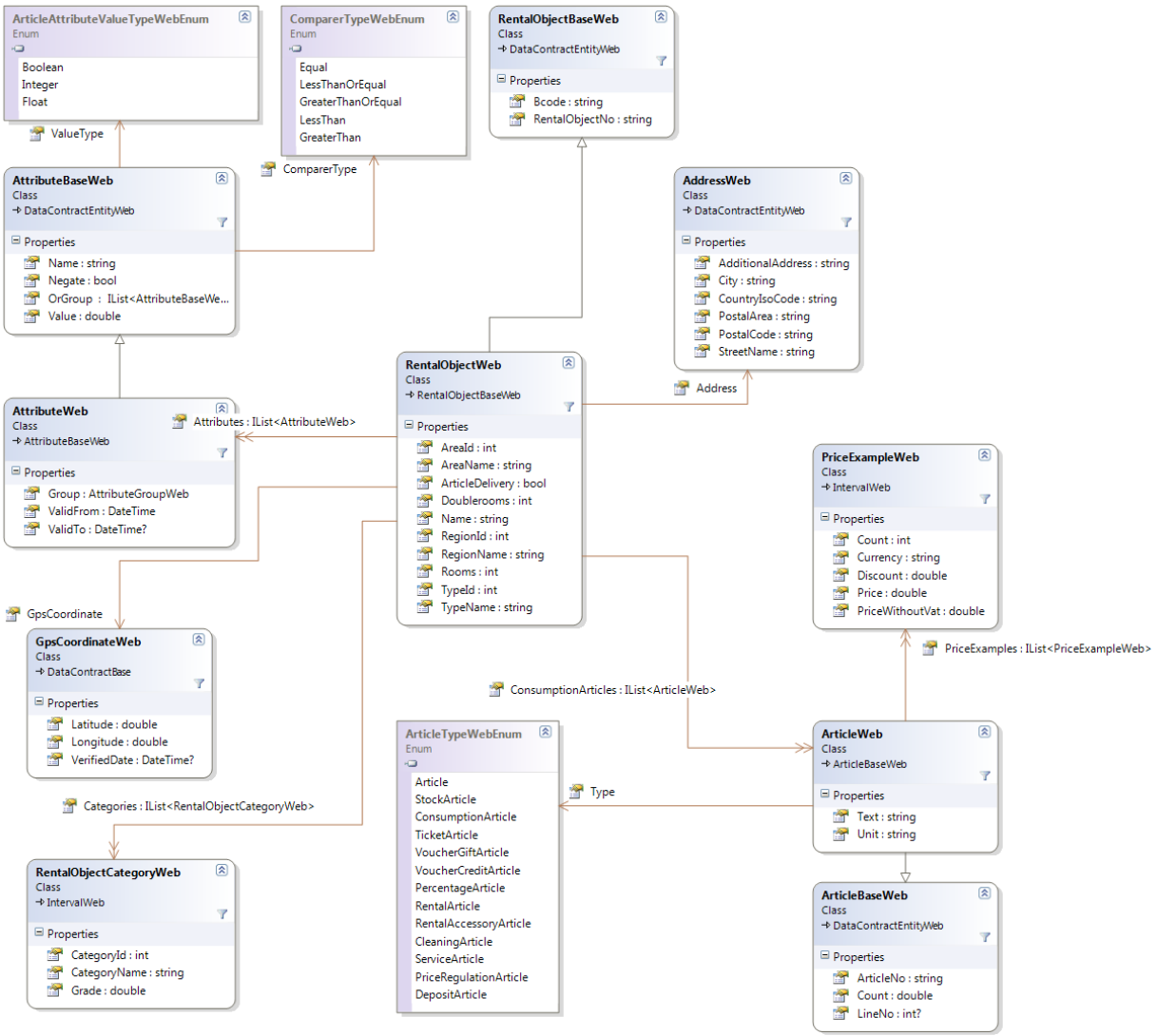

Figure 3: RentalObjectWeb

4.4 Pictures

WCF/SOAP End Point: BookingWcfService. RentalObjectsPictures	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/picture?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/RentalObjectPictureJson XML: http://test.easybook.net/feeds/help/operations/RentalObjectPictureXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjectsPictures.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectPictureWeb objects, one for each requested house. See “Figure 4: RentalObjectPictureWeb”.

This method returns a range of rental object pictures and identification numbers(ids) for all house available for renting today. A picture is only valid within the date interval specified in the “ValidFrom and “ValidTo” properties. This means that a picture can expire once the “ValidTo” date is exceeded. Only pictures that overlaps the requested date period are returned.

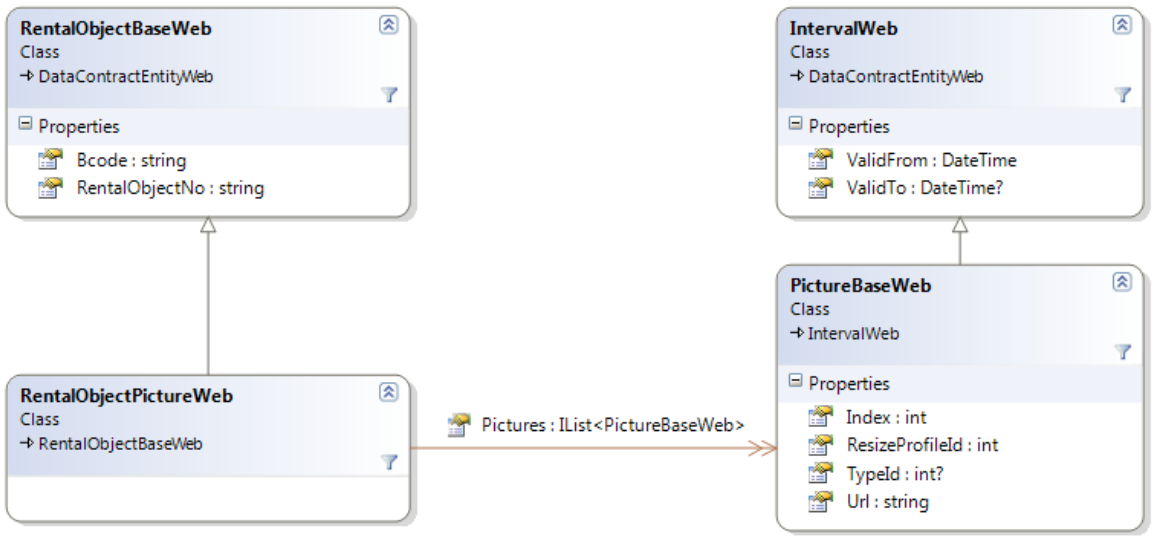

Figure 4: RentalObjectPictureWeb

4.5 Text

WCF/SOAP End Point: BookingWcfService. RentalObjectsText	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/text?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/RentalObjectTextJson XML: http://test.easybook.net/feeds/help/operations/RentalObjectTextXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjectsText.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectTextWeb objects, one for each requested house. See “Figure 5: RentalObjectTextWeb”.

This method returns a range of rental object text descriptions for all house available for renting today. A single rental object can hold multiple texts because each text is only valid within a specific date interval. If the property “ValidTo” is null however, the text has no expiration date. All text intervals overlapping the requested date period are returned.

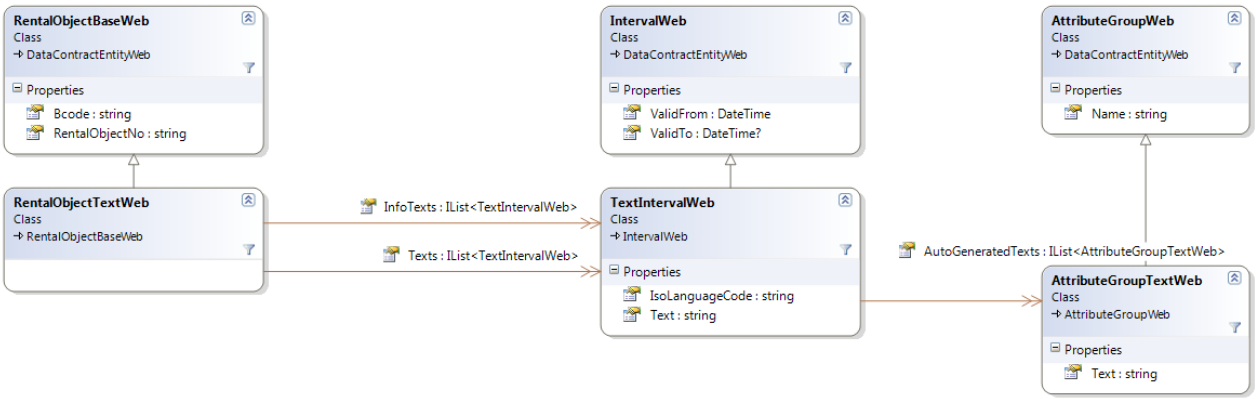

Figure 5: RentalObjectTextWeb

The class diagram illustrates the contents of the service reply. The TextIntervalWeb holds the prose text, while the collection “AutoGeneratedTexts” holds a list of autogenerated texts divided into one description for each applicable attribute group. Autogenerated texts are created based on the values of the associated attributes on a rental object. Thus, the resulting text is a product of the configured attributes on a rental object.It is unlikely that all attribute groups has an associated automatic text. Please note, that some rental objects may have elements in this list while the "Text" property is empty. Other rental objects may have data in both properties, and still others may only have a prose "Text" value.

4.6 Rooms

WCF/SOAP End Point: BookingWcfService. RentalObjectsRooms	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/room?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/RoomJson XML: http://test.easybook.net/feeds/help/operations/RoomXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjectsRooms.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectTextWeb objects, one for each requested house. See “Figure 5: RentalObjectTextWeb”.

This method returns a list of rooms for each requested rental object. Under each room any number of attributes can be listed which indicates what the room contains. This will mostly be used for describing the number of bedplaces (double beds, single beds etc) available in the different rooms. The ValidFrom/To periods on the RoomAttributeWeb determines the timeframe where this attribute applies in the given room. If ValidTo is null then it has no expiry date.

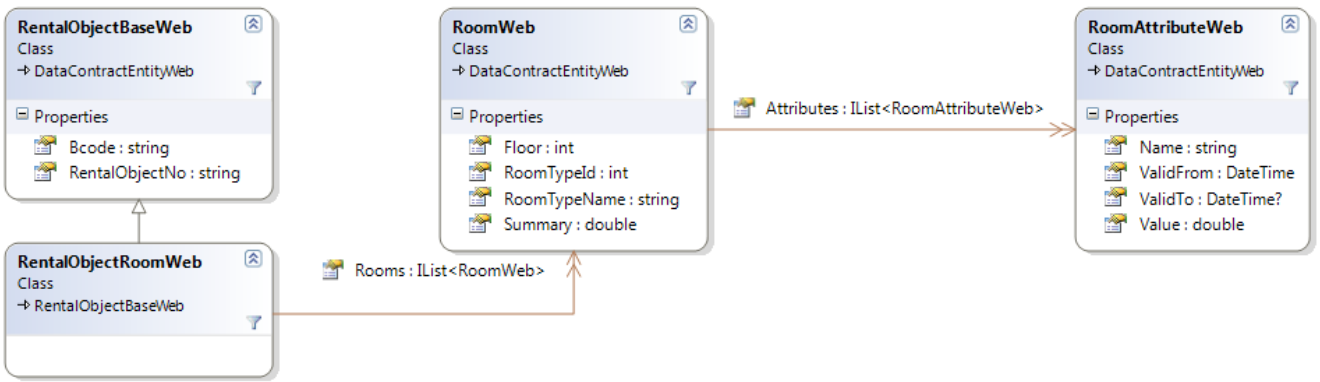

Figure 6: RentalObjectTextWeb

4.7 Prices

	Details
REST URI (HTTP GET)	Only available at this URL as either XML, JSON or CSV: http://test.easybook.net/feeds/compressed/{type}/{PartnerCode}/{ConsumerId}/price?include=
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/PriceJson XML: http://test.easybook.net/feeds/help/operations/PriceXml CSV: http://test.easybook.net/feeds/help/operations/PriceCsv
Online documentation	None
Authentication required	No.
Authorization required	No.
Input arguments	“include”. A full output (2-28 days of vacation length)or only a week based output where vacation lengths of 7, 14, 21 and 28 days are included.
Output	A pre-compressed (ZIP) file containing all possible arrivals and corresponding prices for all rental objects. See “Figure 7: RentalObjectPrices”.

This method is only available as a HTTP REST feed. The feed returns a pre-compressed file contanining all possible combinations of arrival dates, vacation lengths (2-28 days) and corresponding prices for all available rental objects. Prices are pre-genereted due to the complex nature of the EasyBook.NET price calculation algorithm. Prices in EasyBook.NET are calculated on a day to day basis and takes into consideration factors such as cancellations, owner discounts and vacation types when determining the rental house price price. Pre-calculating prices ensures the integrity of the prices presented in EasyBook.NET and in partner systems.

Due to the signifigantly large content size of this feed, the feed is available as 3 different output types which may be requested at will:

1. A zipped file containing an XML file
2. A zipped file containing a JSON file
3. A zipped file containing a CSV file

The feed has a “include=” input which can hold 2 different values:

1. Full: The default value. All possible price combinations for stays of 2-28 days.
2. Week: Only arrival/price combinations for stays of length 7,14,21 & 28 days are returned.

The structure of both the xml and json output can be seen on Figure 7: RentalObjectPrices. For CSV, the following line structure is outputted (example):

NUMBER,DAYS,ARRIVALSTART,ARRIVALEND,CLEANINGOPTIONS,PRICE_DKK,DISCOUNT_DKK,DEPOSIT_DKK,CLEANING_DKK,PRICE_EUR,DISCOUNT_EUR,DEPOSIT_EUR,CLEANING_EUR,PRICE_NOK,DISCOUNT_NOK,DEPOSIT_NOK,CLEANING_NOK

As can be seen, the price details is split into 4:

- Price
- Discount
- Deposit
- Cleaning*

Furthermore, these columns are then repeated for each currency of the consumer.

For the column “CLEANINGOPTIONS”, the following values are used:

1. Optional: Cleaning is optional. A cleaning price is supplied.
2. Included: Cleaning is included in the sales price. A cleaning price is not available.
3. On site: Cleaning must be ordered on site by the customer and cannot be ordered through the booking service. A cleaning price is not available, but *may* be listed in the rental object text descriptions.

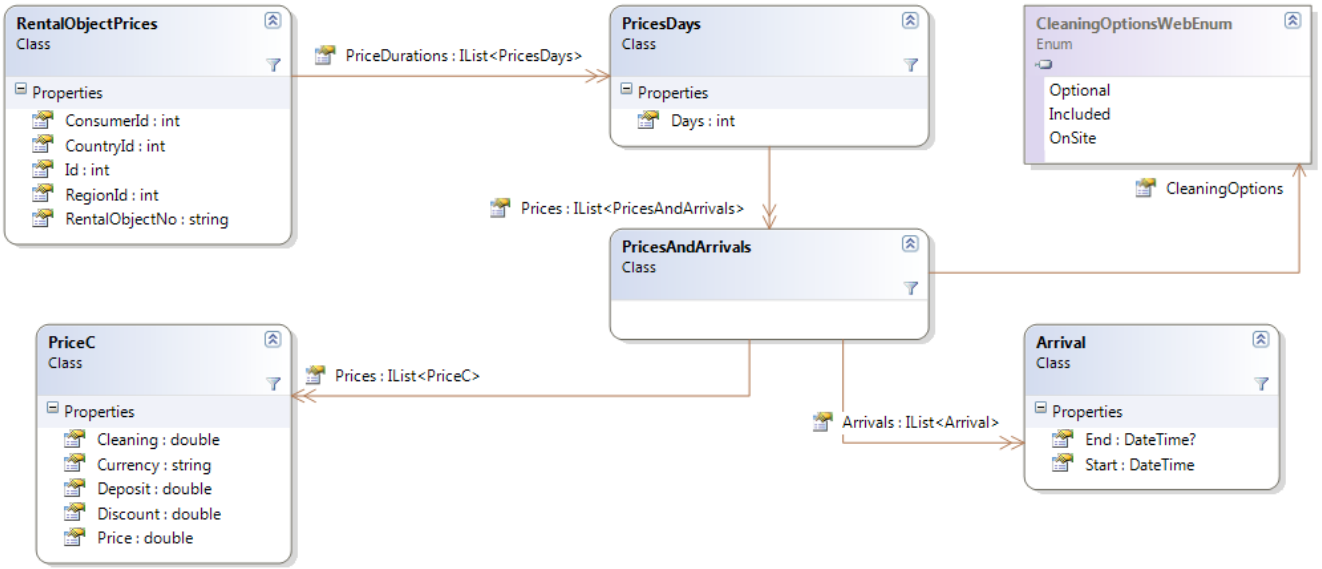

Figure 7: RentalObjectPrices

For each rental object (RentalObjectPrices), all possible vacation lengths are listed under the PriceDurations collection. This is normally ranging from 2-28 days but some rental objects may have a more restricted collection. For each PriceDays element, a list of PriceAndArrivals is listed. Each PriceAndArrivals element contains an arrival date or a span of valid arrival days. That is, all allowed arrivals for the vacation length specified in the PriceDays. Each PriceAndArrivals element also contains the total price, cleaning price, deposit price, discount in available currencies.

4.8 Rental Object Numbers

WCF/SOAP End Point:	Details
BookingWcfService. RentalObjectsNumber	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/rentalobjectnumber?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/RentalObjectNumberJson XML: http://test.easybook.net/feeds/help/operations/RentalObjectNumberXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjectsNumber.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectBaseWeb objects, one for each requested house. See “Figure 8: RentalObjectBaseWeb”.

This method returns a range of house rental objects numbers and identification numbers(ids) for all house available for renting today, but not necessary vacant today. The full range of all rental object numbers can be used to split calls to all methods with RentObjectsIntervalCriteria as input into a number of smaller chunks.

The property Bcode can in the future be used to eliminate newly acquired houses from the data feeds. For now it is always empty.

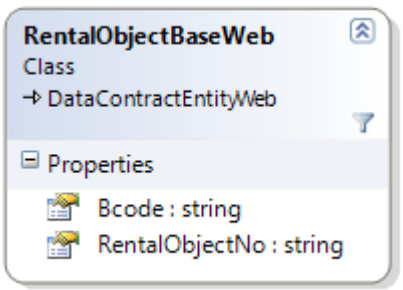

Figure 8: RentalObjectBaseWeb

4.9 Calendar

WCF/SOAP End Point: BookingWcfService. RentalObjectsCalendar	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/calendar?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=&ChangeTimeStamp=DD-MM-YYYY
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/CalendarJson XML: http://test.easybook.net/feeds/help/operations/CalendarXml
Documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_RentalObjectsCalendar.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsCalendarCriteria object specifying a range of houses and optionally if a full list of default weeks should be included. See “Figure 9: RentalObjectsCalendarCriteria”.
Output	A list of RentalObjectVacancyWeb objects, one for each requested house. See “Figure 10: RentalObjectVacanyWeb”.

This method returns data for a range of houses containing vacancy details. In the list “Calendar” all booked and blocked intervals are returned. These are periods where the house is *not* vacant. For booked periods, the following distinction should be made:

Type = “b”: The calendar is blocked in the provided interval. This does not count against any winter rental rules (39 week rule in Denmark)
Type = “B”: the calendar is booked in the provided interval. This should count against the aforementioned 39 week rule.

If the input argument “IncludeWeeks” is true, a list of weeks is returned in the collection “Weeks”. The week list expresses the default arrival days for the rental object as well as vacant and non vacant periods. To check if a particular week is available, check the “Type” property for either an “a” or “A” string.

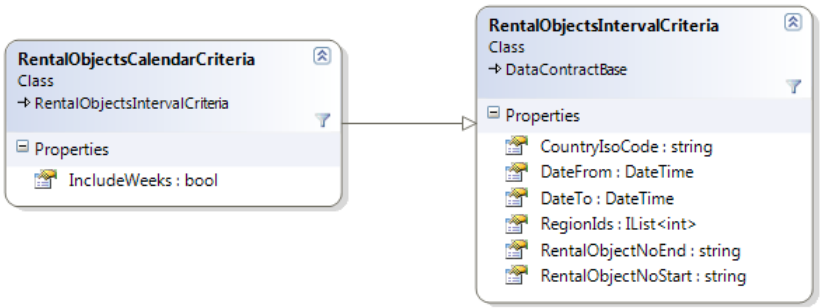

Figure 9: RentalObjectsCalendarCriteria

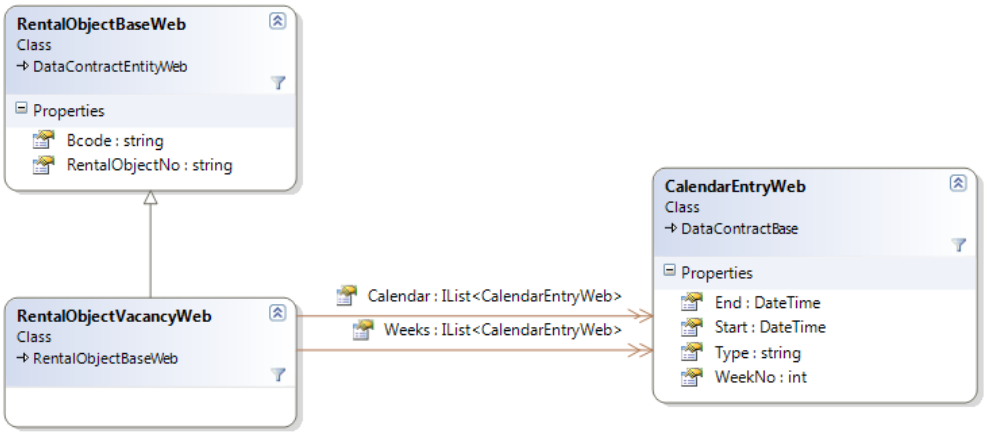

Figure 10: RentalObjectVacanyWeb

4.10 Geography

WCF/SOAP End Point: BookingWcfService. Geography	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/geography?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/GeographyJson XML: http://test.easybook.net/feeds/help/operations/GeographyXml
Documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_Geography.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of Region objects for the specified country. See “Figure 11: Region and Area”.

This method returns all regions and areas for the specified country.

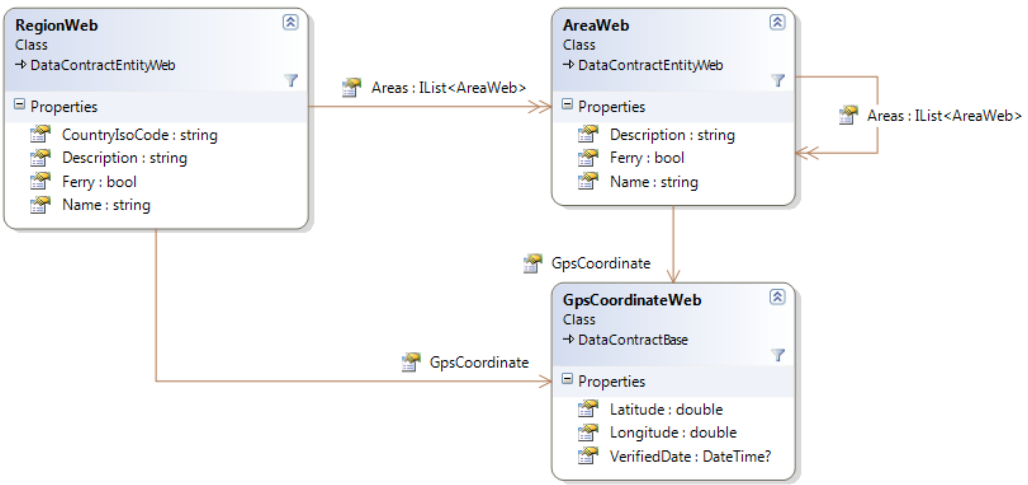

Figure 11: Region and Area

4.11 Attributes

WCF/SOAP End Point: BookingWcfService. RentalObjectsAttributes	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/attribute
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/AttributeJson XML: http://test.easybook.net/feeds/help/operations/AttributeXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_Attributes.htm
Authentication required	No.
Authorization required	No.
Input arguments	None.
Output	A list of AttributeWeb objects. See “Figure 12: AttributeWeb”.

This method returns all possible available attributes that are used across all rental objects. Only a subset of this list is available/registered on any rental object.

The properties “Negate”, “OrGroup”, “ComparerType”, “Value”, “Validfrom” & “ValidTo” can be ignored. These are used elsewhere by the system. The ValueType determines if the attribute holds a Boolean (yes/no) value or an integer/float value. Examples of Boolean values are “Flatscreen”, “Dishwasher” & “Stove”. Examples of integer attributes are “Distance to sea”, “Build year”, “Renovated year” & “Bedrooms”.

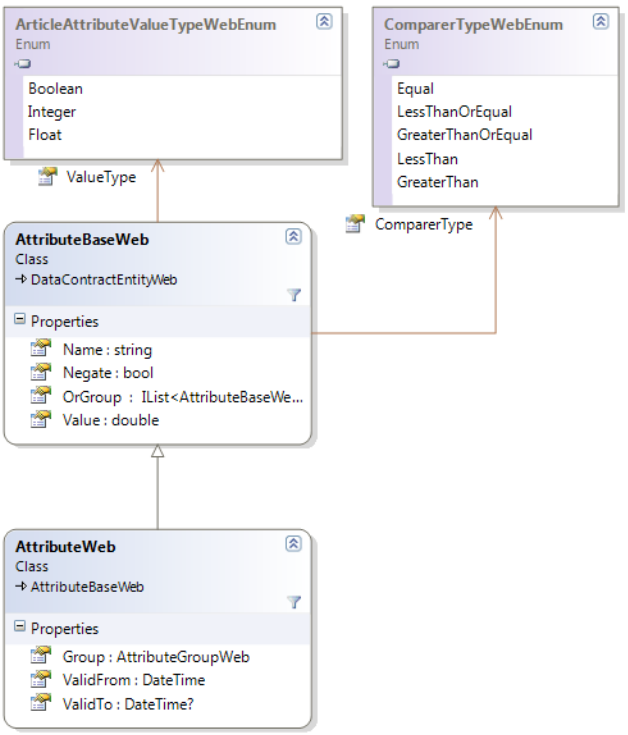

Figure 12: AttributeWeb

4.12 Articles

WCF/SOAP End Point: BookingWcfService. RentalObjectsArticles	Details
	See section 3.2.1 Service BookingWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/article?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=
REST Documentation	JSON: http://appl.easybook.net/feeds/help/operations/ArticleJson XML: http://appl.easybook.net/feeds/help/operations/ArticleXml
Online documentation	http://docs.appl.easybook.net//?topic=html/AllMembers_T_Tse_EasyBook_Server_DataContracts_RentalObjectsArticlesResult.htm
Authentication required	No.
Authorization required	No.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	RentalObjectsArticlesResult. See

This method returns all possible available articles that can be sold using the service methods “Booking” or “UpdateOrder”. For each rental object specified in the input criteria, a list of strings representing the article numbers are available. The article numbers refer to the list of articles also present in the RentalObjectsArticlesResult class.

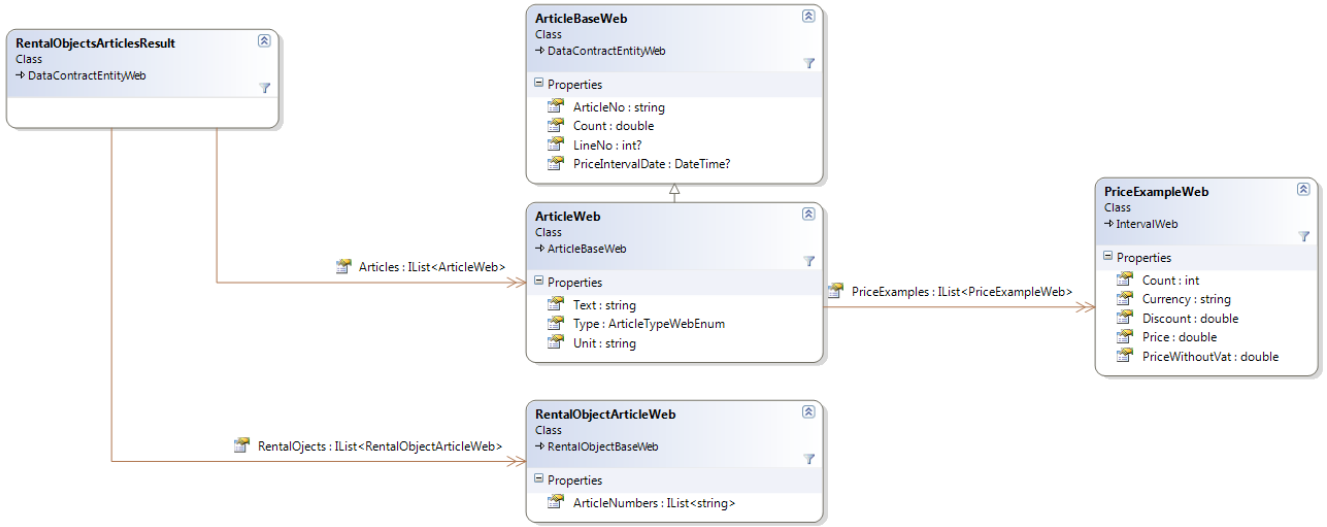

Figure 13: RentalObjectsArticlesResult

The result class holds 2 main collections: “Articles” is a list of all available articles across all rental objects. For each item in “RentalObjects”, the collection “ArticleNumbers” holds the link to the different items in the “Articles” list. Please note that the listed pridce in the “PriceExamples” list is a suggested price only based on the “unit” of the item. The actual price may differ based on the length of the booking. For a specific booking period, calling the “Availability” service method will always result in a correctly calculated price.

ArticleBaseWeb.PriceIntervalDate is always null when included in a RentalObjectsArticlesResult class instance or in any other result class instance returned by the server application. See section “5.2 Booking” for further description of this property.

5 BOOKING

This section contains information about the service methods that EasyBook.NET exposes in order to support partner bookings.

5.1 Availability Check

WCF/SOAP End Point:	Details
BookingWcfService. Availability	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTP GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: http://test.easybook.net/feeds/{type}/{PartnerCode}/{ConsumerId}/availability?rentalObjectNo=&currencyIsoCode=&arrival=&departure=&numberOfPersons=
REST Documentation	JSON: http://test.easybook.net/feeds/help/operations/AvailabilityJson XML: http://test.easybook.net/feeds/help/operations/AvailabilityXml
Documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingWcfService_Availability.htm
Authentication required	No.
Authorization required	No.
Input arguments	AvailabilityCriteria object. See “Figure 14: AvailabilityCriteria”.
Output	AvailabilityResult object. See “Figure 15: AvailabilityResult” .

This method returns the availability information for a specific house and date interval. May be called to gather last minute information and check the prices before an actual booking is submitted. Availability also contains a list of articles that can be added to the order, when placing a booking. The “Comments” list holds any relevant comments that should be displayed to the customer before submitting the booking. Use the “NumberOfPersons” property in the input criteria to accurately calculate any person related discounts that may be available for the rental object.

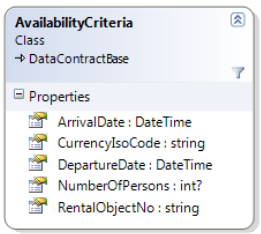

Figure 14: AvailabilityCriteria

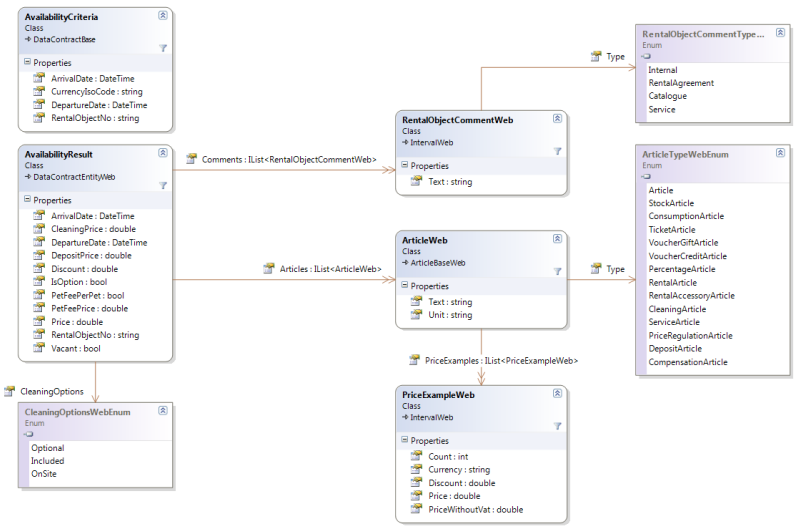

Figure 15: AvailabilityResult

5.2 Booking

WCF/SOAP End Point:	Details
BookingWcfService. Booking	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS POST)	Bookings can also be created by issuing a HTTPS POST to the following URI: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/booking Include the BookingCriteria as XML or JSON in the body of the HTTPS POST.
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/BookingJson XML: https://test.easybook.net/booking/help/operations/BookingXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_Booking.htm
Authentication required	Yes.
Authorization required	Yes.
Input arguments	BookingCriteria object. See “Figure 11: BookingCriteria”.
Output	OrderWeb object. See “Figure 12: OrderWeb”.

5.2.1 DESCRIPTION

Handles all orders from WEB. If a booking is successful, the server returns an OrderWeb data contract with detailed information about the booking. The OrderWeb also contains a customer profile ID as well as an order number. The Profile ID is a unique code of 32 letters (GUID) that identifies the customer. The profile ID is important since it is required to be able to retrieve details about the order through the web service “OrderInfo. This is to ensure that only the partner who created the order can gain access to it. The following subsections detail part of the datacontracts that make up the BookingCriteria.

Test Booking

On both production and test servers, a booking may be simulated by setting the boolean value “TestBooking” to “True”. When this is done, the server processes the booking as it would normally but finishes by performing a rollback of the transaction. Bookings with the flag “TestBooking” set to ”True” are therefore *not* registered in EasyBook.NET, and can be called repeatedly for integration and testing purposes.

PersonWeb/CompanyWeb

Dependant on whether the customer is a person or a company, the objects PersonWeb or CompanyWeb is provided and the CustomerType is set accordingly. For companies, the EAN or CVR number can be set. If the company is vat free the field “VatFree” is set to “true”.

The server will attempt to match the booking customer with the customer database on the server based on information such as name, address, email etc. If an existing customer is found to be a match, then that customer is used on the resulting order instead. The customer details are updated with the latest received data.

A language ISO Code can be added to either the PersonWeb or CompanyWeb. This ensures that the customer is registered with a correct language. If the language code is left blank, the server defaults to either the language code sent in with the SOAP Header or the customers countries default language.

FormOfPayment

FormOfPayment is provided to inform the server what payment method is to be used. By default, the value is set to “None”, which means payment is not done online but instead by invoicing. The customer receives an invoice including payment instructions.

BankInfoCriteria

If the customer selected invoicing as payment, the bank information can be provided here but it is not mandatory. Account details are used in case some left over deposit must be transferred back to the customer.

To validate the correctness of the bank info, please use the service method: BookingWcfService.CheckBankInfo. The booking service will validate bank info according to the same rules and will NOT work unless all validation rules are abided.

Valid combinations of the mandatory fields:

1. RegNo=Swift code, AccountNo=IBAN account number, CountryCode=Iso code for the country of the bank.
2. RegNo=Domestic registration number, AccountNo=Domestic bank account number, CountryCode=Iso code for the country of the bank. This combination is only valid for countries where the consumer has a domestic account. Please contact the consumer for the list of countries where a domestic transfer is possible.

IncludeCleaning

To include cleaning on the order, set this property to true. For rental objects, where cleaning is already included by default, this property does nothing.

Bedlinen included in price

For some rental objects, linen may be included in the rental price. This can be easily tested by checking what the linen price is in the available articles in the “Availability” service call. See also section “Availability”. If the price is 0 then the number of appropriately required linen articles should still be set in the BookingCriteria prior to calling “BookingWcfService.Booking”. Typically the number of persons renting the rental object should equal the number of linen required. Please note that this is *not* handled automatically by EasyBook.NET, and is solely the service callers responsibility.

NumberOfPersons

NumberOfPersons = number of adults staying in the house.

NumberOfChildrenAsAdults = number of kids 4-11 years old. If this information is not available simply provide the total number of persons in the “NumberOfPersons” field.

The server will add NumberOfPersons and NumberOfChildrenAsAdults when determining if the house has the capacity to accommodate the number of guests. If not, an error will be returned.

NumberOfChildren = number of children 0-3 years old.

NumberOfPets

NumberOfPets = The number of pets the guests will bring. In some cases bringing pets to the house will results in a fee. For some houses, the fee may be tied to the number of pets taken. This is why the exact pet count is important.

Articles

Additional articles as returned in the “Availability” service method may be purchased as part of the booking. Simply add the articles with an ID or number and the desired count of articles to the “Articles” list of the booking criteria. The articles will then be added to the order and invoice created for the booking. Please note that pet fee and cleaning is handled explicitly using the “IncludeCleaning” and “NumberOfPets” criteria values.
For ticket articles the ArticleBaseWeb.PriceIntervalDate can be used to specify an alternative price interval. If it is null the price interval containing the arrival date of the rental is chosen. But if for example 2 price intervals are available for an rental interval, the last price interval can be chosen by filling PriceIntervalDate with the start date of this interval.
PriceIntervalDate is only relevant for ticket articles and it is ignored for all other article types and it is always null in outgoing ArticleBase derived class instances.
Ticket articles are NOT returned in the “Availability” service method but are returned in the “Articles” service method.

Electronic Invoicing & Ean fields

For the use of EAN, the following properties are available according to the following source:
URL: www.virk.dk/efak, Document: “Electronic Invoicing Handbook”.

EAN number = BookingCriteria.CustomerCompany.EanLocationNumber, mandatory.
Order or requisition number = BookingCriteria.CustomerReferenceNumber, mandatory.
Personal or other reference = BookingCriteria.CustomerCompany.AttentionPerson, mandatory.
Internal accounting number = BookingCriteria.InternalAccountingNo, optional.

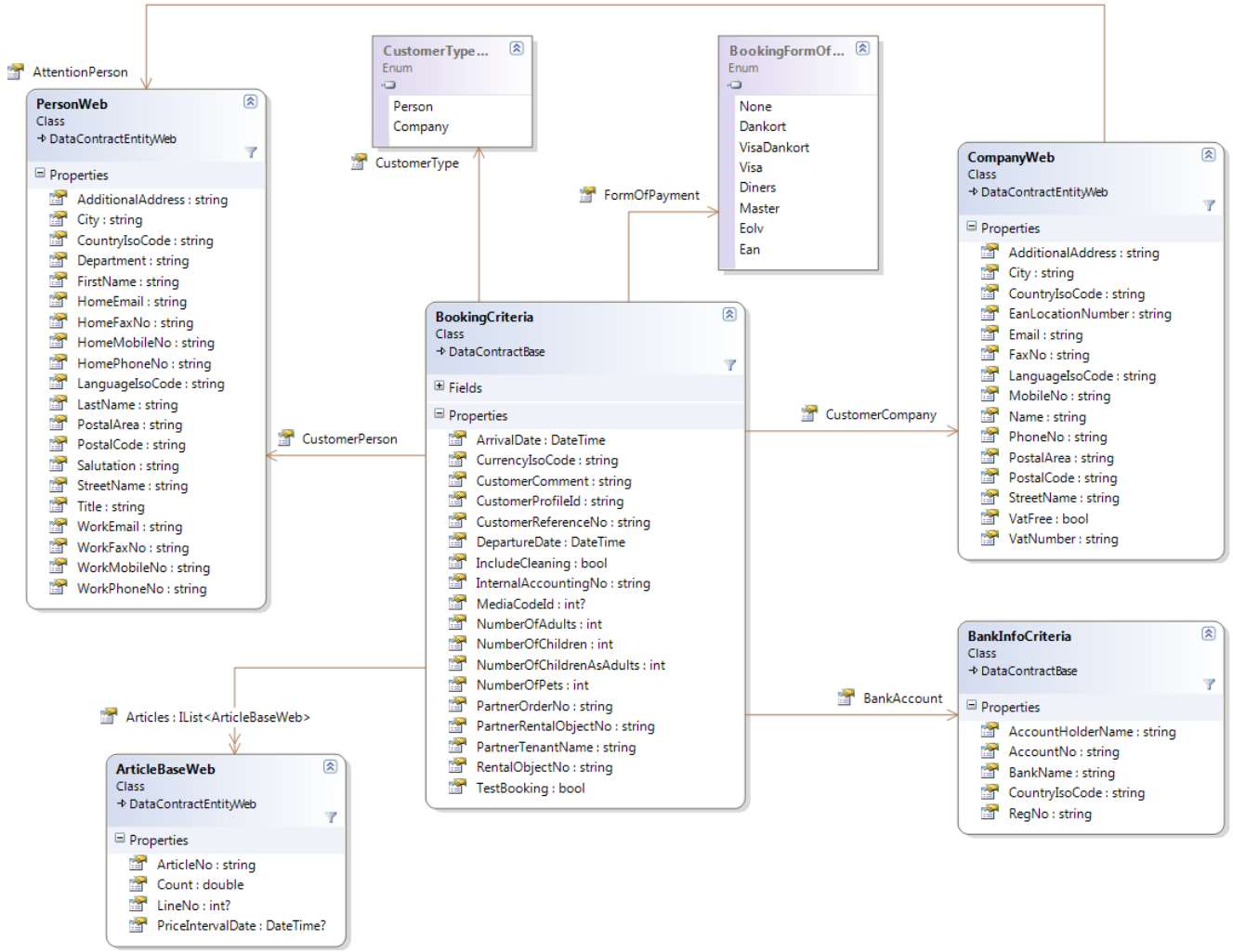

Figure 11: BookingCriteria

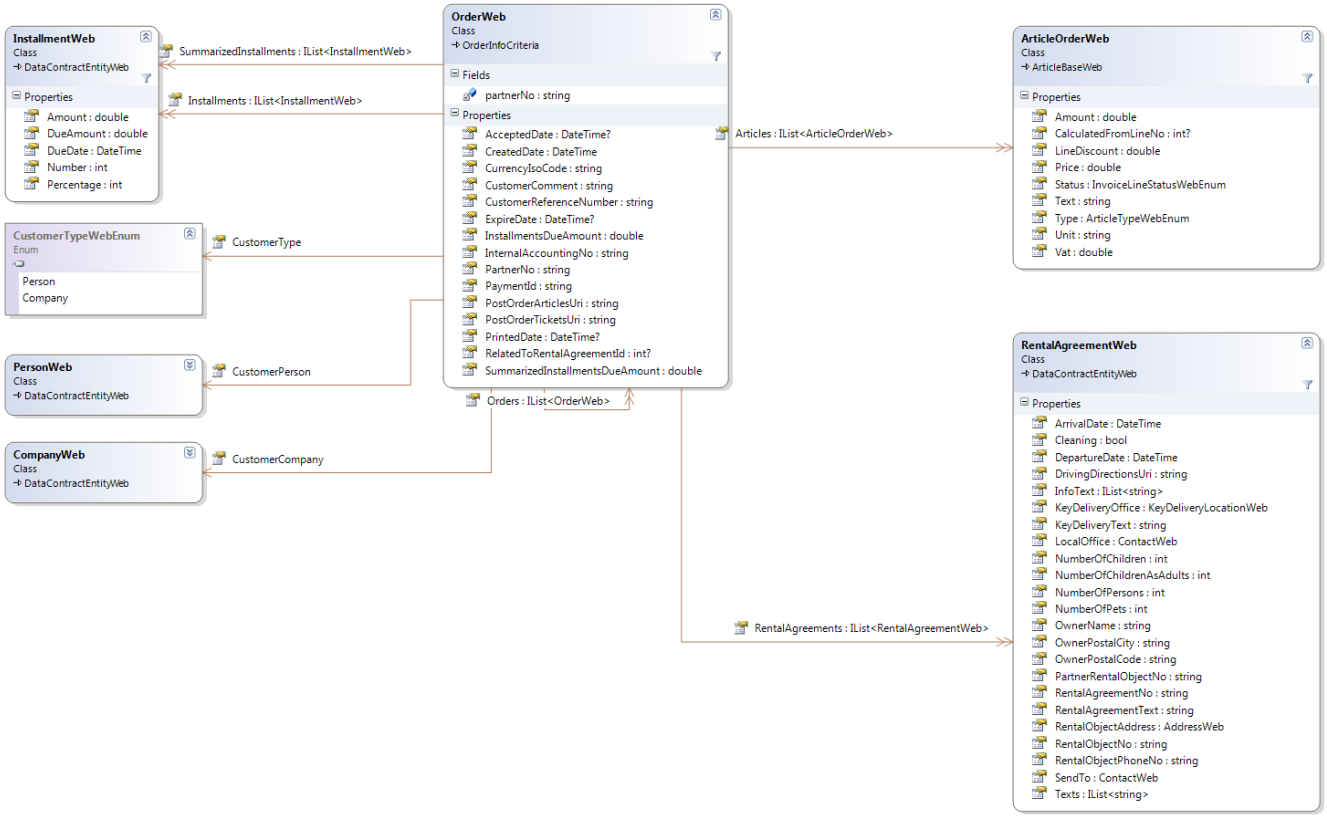

Figure 12: OrderWeb

The OrderWeb object is returned upon completing a successful booking request. The orderweb contains the following information:

- Orderweb
  - The parent orderweb contains basic data about the order. An orderweb is always immutable, meaning the order itself cannot be changed subsequently.
- Installments
  - The installments are divided into two collections:
    - SummarizedInstallments: If an orderweb has child orders, the summarized installments collection contains all installments and amounts for both the parent and child orderwebs. If 2 installments have the same date the amounts are added into a single installment.
    - Installments: The installments for a single orderweb.
- Customer
  - The customer type. If the customer is a private person then the property “CustomerPerson” contains the customer details. If the customer is a company then the property “CustomerCompany”contains the details.
- RentalAgreements
    - An orderweb can contain one or more rental agreements, which holds the detail about the booked rental object. However, in most common scenarios (including *all* internet bookings) there is at most 1 rental agreement in the rental agreement collection. Only the parent order (the original booking) can hold rental agreements.’
- Articles
  - A list of the articles sold on the order with the count, price and total amount including VAT. If an article is subsequently deleted, it will appear as a credit line with the count holding a negative value. For instance, if cleaning was sold as part of the original booking, but subsequently deleted, a new orderweb will be added to the original orderweb with cleaning having a count of -1. This is again due to the immutable nature of orderwebs.

5.2.2 BOOKING WITH REST SERVICE

When booking using the URL https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/booking, a certain number of requirements must be met in order to perform a successful booking.

For reference, please download the samples located here: http://docs.test.easybook.net/Content/Samples.zip

First, in order to call the service, basic authentication as described in earlier sections will have to be used. An example of this is part of the samples file. Second, a valid XML document representing the “BookingCriteria” as serializwed by the DataContractSerializer will have to be submitted using the POST method. Again, this can be reviewed in the samples file. The file also contains an example of a valid Booking XML (Booking.xml), which the server will accept. Use this XML document as a basis of your continued integration.

It is important to note that the booking XML document must “complete” when submittingit to the server. This means that all elements MUST be present in the document and in the exact order specified in Booking.xml. However, most of the elements can be left blank or nil as shown in the sample file. The sample file shows the **minimum** required elements (except for the articles list, which ca be left empty), that **must** contain a value in order for the booking call to go through.

Setting an element to nil is done in the following way: <CustomerCompany i:nil="true"/>. For the articles list, this can be left empty by setting it to: <Articles/>

5.3 Order Info

WCF/SOAP End Point:	Details
BookingWcfService. OrderInfo	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/order?profileid=&orderno=
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/OrderInfoJson XML: https://test.easybook.net/booking/help/operations/OrderInfoXml
Documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_OrderInfo.htm
Authentication required	Yes.
Authorization required	Yes.
Input arguments	OrderInfoCriteria object. See “Figure 13: OrderInfoCriteria”.
Output	A list of OrderWeb objects. See “Figure 12: OrderWeb”.

The OrderInfo service method returns order details for a specific order number.

In most cases, a valid customer profile ID (GUID) is required to call this service method. A customer profile ID is returned by the server upon a successful booking. If the partner has contract fulfillment obligations, then a profile Id is *not* required to access order info. The server will check however, that the partner requesting orderinfo is also associated with the order. An order number is required to retrieve order details.

Each OrderWeb contains information about the tenancy, the articles sold, the installments and any due amounts remaining on the order. Since an order may contain several tenancies, the OrderWeb object can contain multiple RentalAgreementWeb items. Each RentalAgreementWeb item represents the actual rental agreement for a specific tenancy.

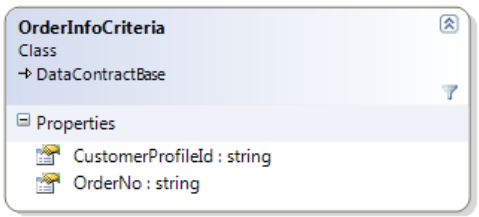

Figure 13: OrderInfoCriteria

5.4 OrderList

WCF/SOAP End Point: BookingWcfService. OrderList	Details
	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/orderlist?DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&state=&offset=&count=
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/OrderListJson XML: https://test.easybook.net/booking/help/operations/OrderListXml
Documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_OrderList.htm
Authentication required	Yes.
Authorization required	Yes.
Input arguments	OrderInfoCriteria object. See “Figure 14: OrderListCriteria”.
Output	A list of OrderWeb objects. See “Figure 15: OrderListResult”.

The OrderList service method returns a list of orders where the arrival date is within the specified input criteria. This method works for partners with and without contract fulfillment obligations. The input criteria are datefrom/to (arrival date of the tennancy) and order state. The order can have the following states:

- Created: The order is created but the order confirmation has not yet been sent to the customer.
- Printed: The order confirmation has been sent to the customer.
- Accepted: The customer has accepted the order and a payment was made.
- Deleted: The order was deleted before it has been accepted by the customer.
- Cancelled: The order was deleted after it has been accepted by the customer.

No more than 5000 orders can be returned by the service in a single call. If there are more orders than that, use the paging functionality of the service (offset and count parameters). The total amount of orders are listed in the “SearchStatisticWeb” class of the result.

The orders returned contains summarized information about the order. If additional orders are created after the tenancy, which results in provision being payed to the partner, these orders will be available in the collection “RelatedOrders” on the OrderSummaryWeb class.

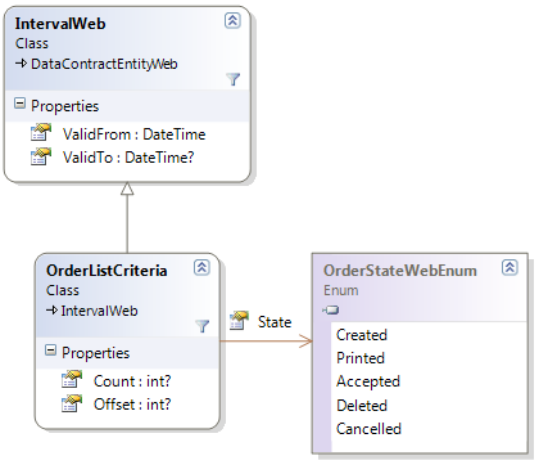

Figure 14: OrderListCriteria

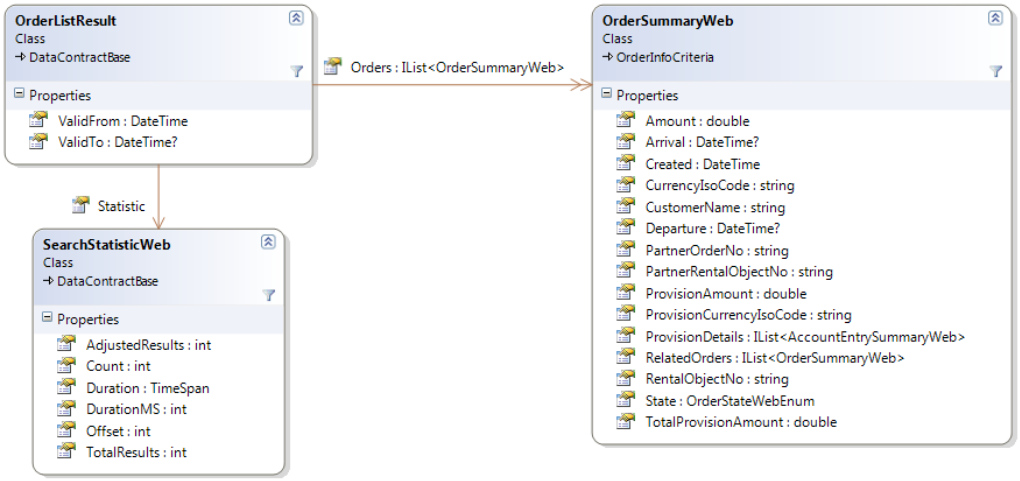

Figure 15: OrderListResult

6 PARTNERS WITH CONTRACT FULFILLMENT

In addition to the already described feeds and methods, partners with contract fulfillment have access to some additional methods, which are listed in this section.

6.1 Key Locations

WCF/SOAP End Point: BookingWcfService. RentalObjectsKeyLocations	Details
	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS GET)	Alternatively, use this URL to retrieve the data directly as XML or JSON: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/keylocation ?CountryIsoCode=&DateFrom=dd-MM-yyyy&DateTo=dd-MM-yyyy&RentalObjectNoStart=&RentalObjectNoEnd=&RegionIds=
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/KeyLocationJson XML: https://test.easybook.net/booking/help/operations/KeyLocationXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_RentalObjectsKeyLocations.htm
Authentication required	Yes.
Authorization required	Yes. Only partners with contract fulfillment obligations are allowed to call this method.
Input arguments	RentalObjectsIntervalCriteria object specifying a range of houses. See “Figure 2: RentalObjectsIntervalCriteria”.
Output	A list of RentalObjectKeyDeliveryWeb objects, one for each requested house. See “Figure 16: RentalObjectKeyDeliveryWeb”.

This method returns data for a range of houses containing key delivery information. Typical this data is only relevant for partners with fulfillment.

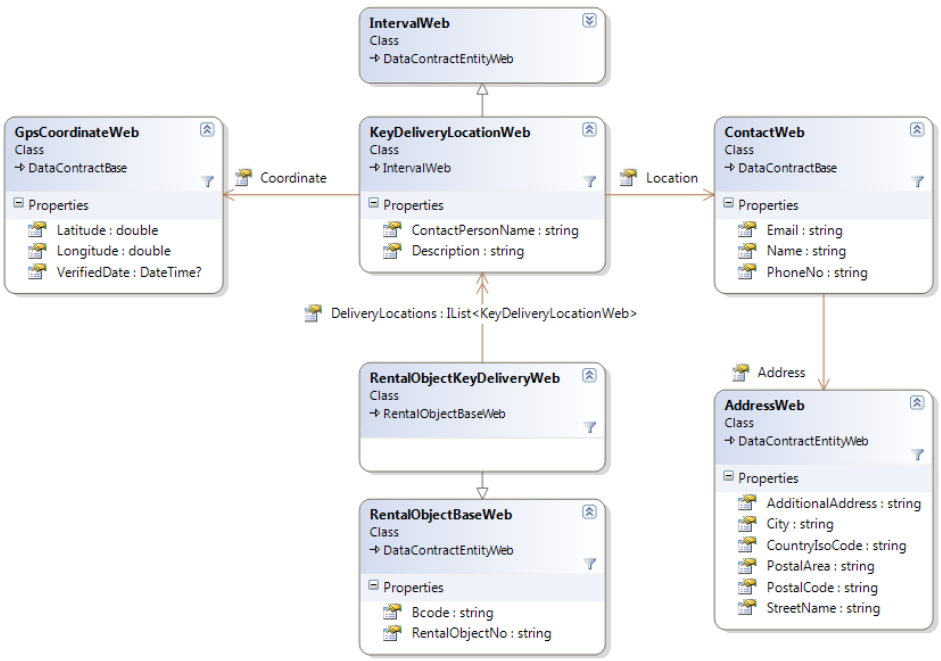

Figure 16: RentalObjectKeyDeliveryWeb

6.2 Update Order

End Point	Details
BookingWcfService. UpdateOrder	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS PUT)	An existing order can be updated by issuing a HTTPS PUT to the following URI: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/order Include the UpdateOrderCriteria as XML or JSON in the body of the HTTPS PUT.
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/UpdateOrderJson XML: https://test.easybook.net/booking/help/operations/UpdateOrderXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_UpdateOrder.htm
Authentication required	Yes.
Authorization required	Yes.
Input arguments	UpdateOrderCriteria object. See “Figure 17: UpdateOrderCriteria”.
Output	An OrderWeb object. See “Figure 12: OrderWeb”.

For partners with contract fulfillment obligations, changes to an existing order is possible using the service method “UpdateOrder”. A valid valid customer profile ID (GUID) is *not* required to call this service method. The server will check however, that the partner requesting to update an order is also associated with the order.

A full order number is required to make subsequent changes to an order. If articles have subsequently been sold or credited using the service method “UpdateOrder”, the sub-collection “Orders” on the parent OrderWeb contains all the changes. Because of the immutable nature of orderweb, each subsequent change to an order results in a new orderweb item placed in the “Orders” collection of the order that was changed. This new order item contains the details about what was changed.

The “UpdateOrder” method is primarily intended for handling additional sale in relation to already existing tenancy order. Adding articles to an existing order works exactly like placing a new booking. Add the required articles (number, count) to the list of articles. The output of this service method is identical to the output of the Booking method.

Instead of just adding new lines to an existing order it is also possible to change existing lines on any of the orderwebs related to the booking. This is done by setting the “LineNo” property on the article that should be changed to the same line number as the original orderweb line. Then just set the new desired “Count” value and the server will take care of the rest.

For desciption of PriceIntervalDate see section “5.2 Booking”.

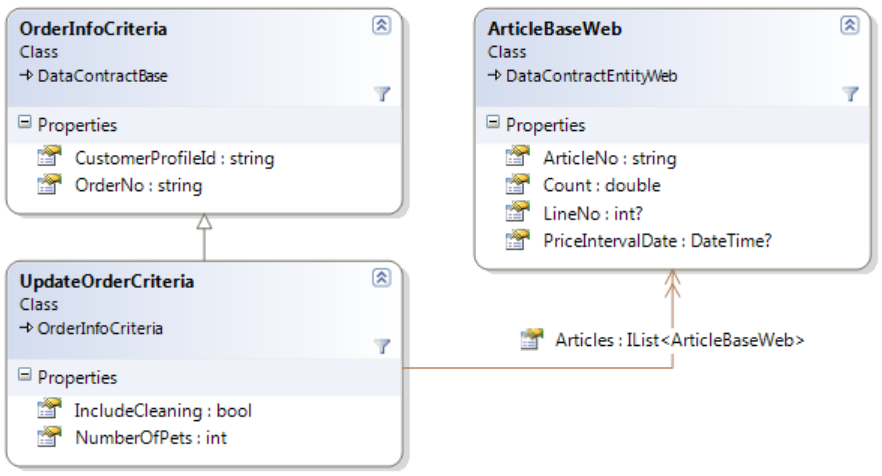

Figure 17: UpdateOrderCriteria

6.3 Delete Order

End Point	Details
BookingWcfService. DeleteOrder	See section 3.2.2 Service BookingAuthenticateWcfService for details about this service.
REST URI (HTTPS DELETE)	An existing order can be deleted by issuing a HTTPS DELETE to the following URI: https://test.easybook.net/booking/{type}/{PartnerCode}/{ConsumerId}/order Include the DeleteOrderCriteria as XML or JSON in the body of the HTTPS DELETE.
REST Documentation	JSON: https://test.easybook.net/booking/help/operations/DeleteOrderJson XML: https://test.easybook.net/booking/help/operations/DeleteOrderXml
Online documentation	http://docs.appl.easybook.net/html/M_Tse_EasyBook_Server_WcfContracts_WebClient_BookingAuthenticateWcfService_DeleteOrder.htm
Authentication required	Yes.
Authorization required	Yes. Only available for partners with contract fulfillment obligations.
Input arguments	OrderInfoCriteria object. See “Figure 13: OrderInfoCriteria”.
Output	None. Hence the order has been deleted if no exceptions are thrown.

This method deletes an existing orderweb. A valid valid customer profile ID (GUID) is *not* required to call this service method. The server will check however, that the partner requesting to delete an order is also associated with the order. A full order number is required to delete an order.

Since orderweb is ummutable a new orderweb is created which contains the credited lines of the deleted orderweb. This new orderweb is then placed in the “Orders” collection on the deleted orderweb. See also Figure 12: OrderWeb.

If the delete operation fails, an exception is thrown. Please see section “3.8 Error Handling” for more details about error handling.