# Hotel Exporter Schnittstelle - Preisberechnung

Inhalt

[Hotel Exporter Schnittstelle - Preisberechnung 1](#_Toc414446060)

[Ziel 3](#_Toc414446061)

[Hotel 4](#_Toc414446062)

[PerDayPrices 13](#_Toc414446063)

[Erwachsene 13](#_Toc414446064)

[Kinderpreise 15](#_Toc414446065)

[Additional Per Day Preise (AddPerDayPrices) 16](#_Toc414446066)

[Einmaliger Zuschlag (OneTimes) 17](#_Toc414446067)

[SpecialOffers 18](#_Toc414446068)

[Allgemein 18](#_Toc414446069)

[Regeln für Special Offers Saison übergreifend 24](#_Toc414446070)

[Early Booking 26](#_Toc414446071)

[Item Infos 27](#_Toc414446072)

[Beispiele 29](#_Toc414446073)

[Allgemeine Infos 46](#_Toc414446074)

[Erfassen der Zuschläge für Kinderpreise 46](#_Toc414446075)

[Anpassungen Webservice 46](#_Toc414446076)

[Anhang 47](#_Toc414446077)

[SpecialOffer 47](#_Toc414446078)

|  |  |  |
| --- | --- | --- |
| 1.0 | rb/mk | Original |
| 1.1 | rb | External, Itinerary und Invoice: true>1, false->0 |
| 1.2 | mk | FirstDays/LastDays korrigiert |
| 1.3 | rb | Anpassungen an der Beschreibung/Erklärung von Allotments |
| 1.4 | rb | Erläuterungen zum SpecialOffer im Anhang |
| 1.5 | rb | Ergänzende Anmerkungen zu den SpecialOffers mit komplexem Beispiel |
| 1.5.1 | rb | Ergänzende Anmerkungen zu RevolvingGroup |
| 1.5.2 | rb | Es fehlte in den Screenshots zu den SpecialOffer der neue RevolvingGroup Attribut-Beschreibung.  Tabelle mit Erläuterungen zu SpecialOffer mit Referenz zu RevolvingGroup angepasst. |
| 1.5.3 | mk | Weekdaysvalid aus SpecialOffer rausgenommen |
| 1.5.4 | rb | Es fehlten einige sinnvolle Felder in den EarlyBooking |

## Ziel

Ziel dieses Dokuments soll sein die Exportdateien von TFS Hotels aus TO Online zu beschreiben und auf Basis dieser einen Offline-Preisberechnung dieser Produkte vorzunehmen.

TO Online exportiert bei Anpassungen an den Stammdaten automatisch XML Dateien zum Hotel und lädt diese auf eine vordefinierte FTP Seite hoch. Die Datei enthält Hotel- und Zimmerbeschreibungsdaten, Preise, Kontingente, Storno- und Umbuchungsregeln.

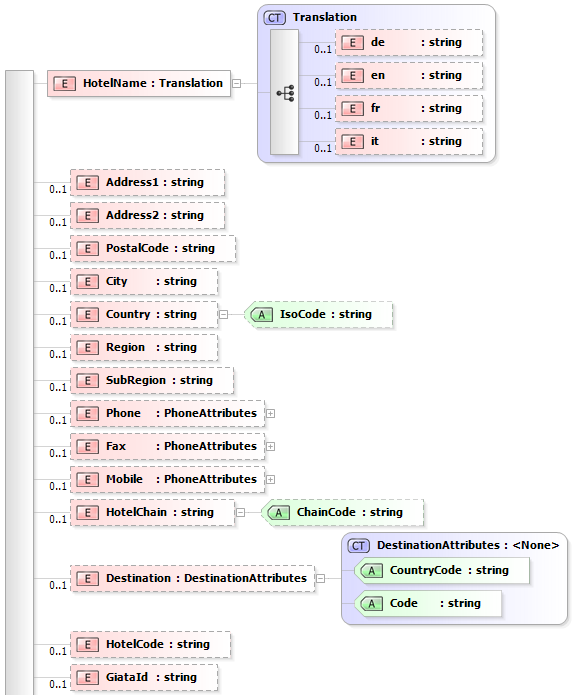
Preise werden in logische Gruppen aufgeteilt:

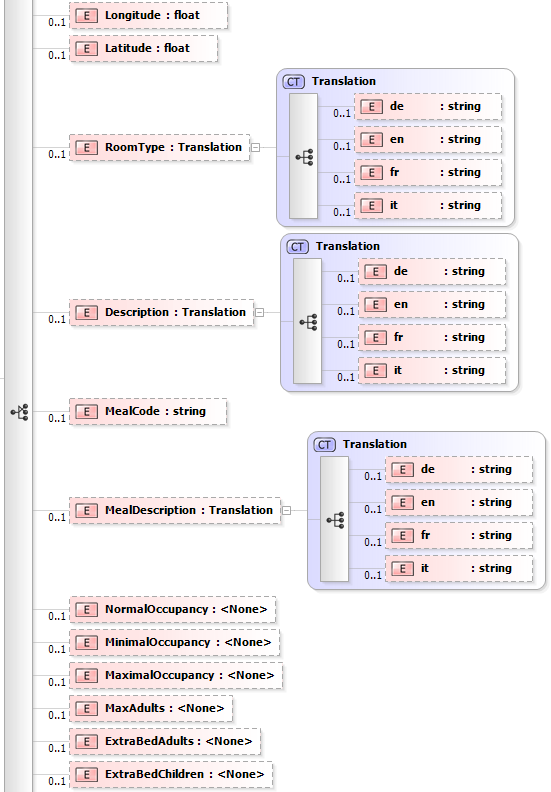
* PerDayPrices – pro Tag Preise
* AddPerDayPrices – Tageszuschläge, z.B. Hochsaisonzuschlag
* OneTimes – Einmalige Zuschläge
* SpecialOffers – z.B. Stay 7, Pax 5
* EarlyBookings – Prozentuelle oder Betrags-Rabatte bei Frühbuchungen

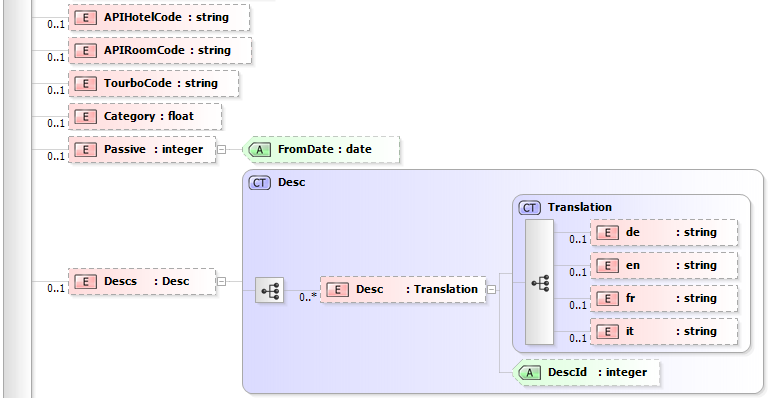
Diese Gruppen werden unten beschrieben.

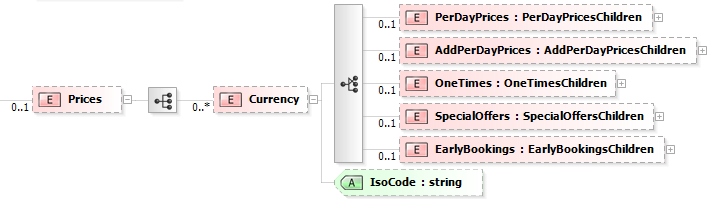
## Hotel

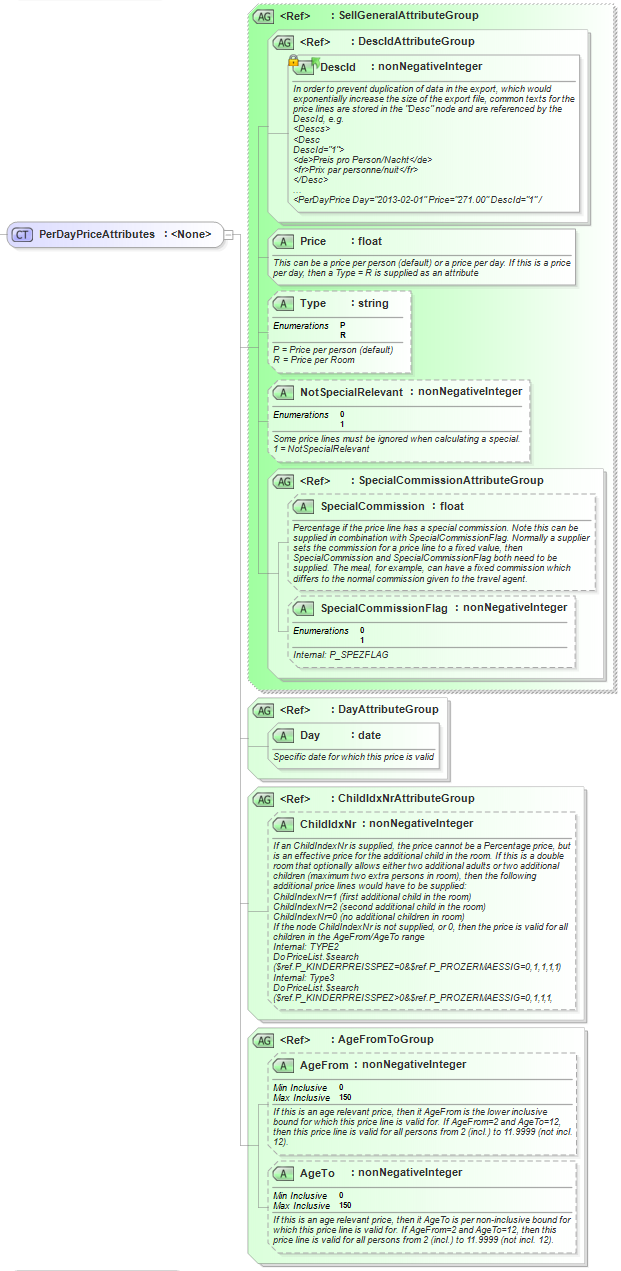
Im Hotel-Node finden sich Stammdaten zum Hotel, Zimmer, Preise, Allotments, Cancellation Fees. Beschreibungen gibt es generell in de, en, fr und it, sofern der TO Online Kunde diese Daten erfasst hat.

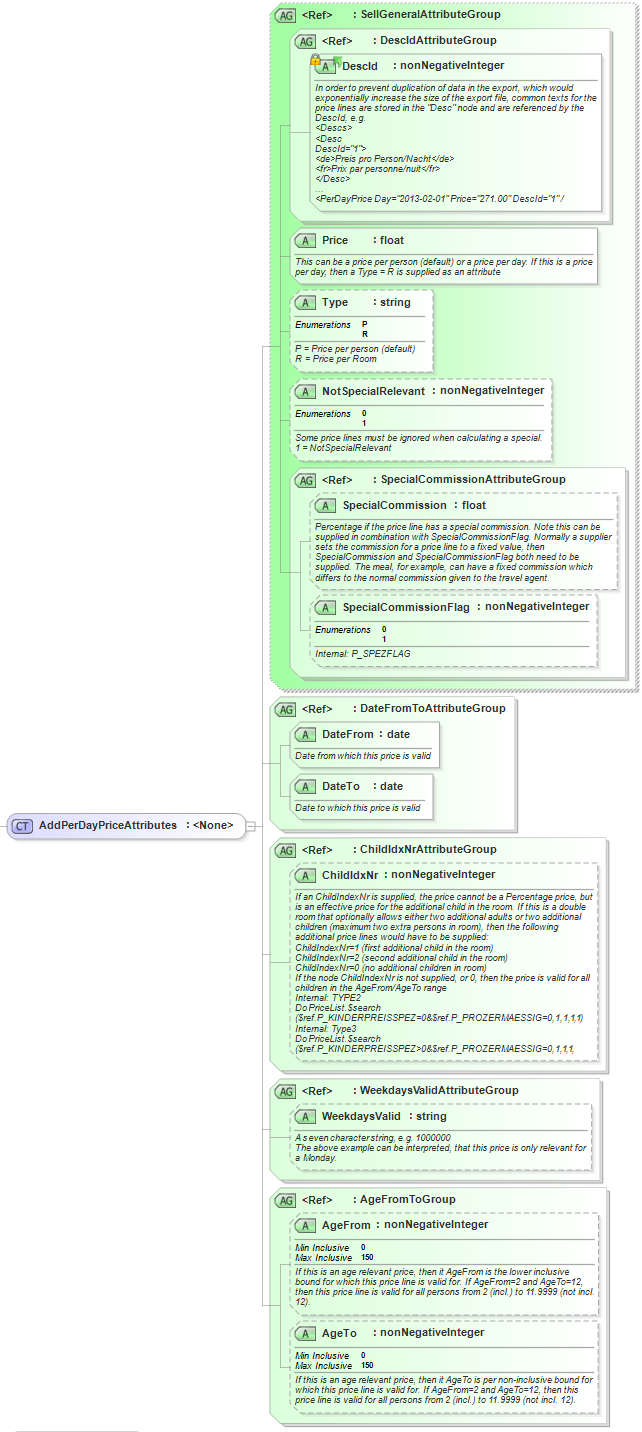


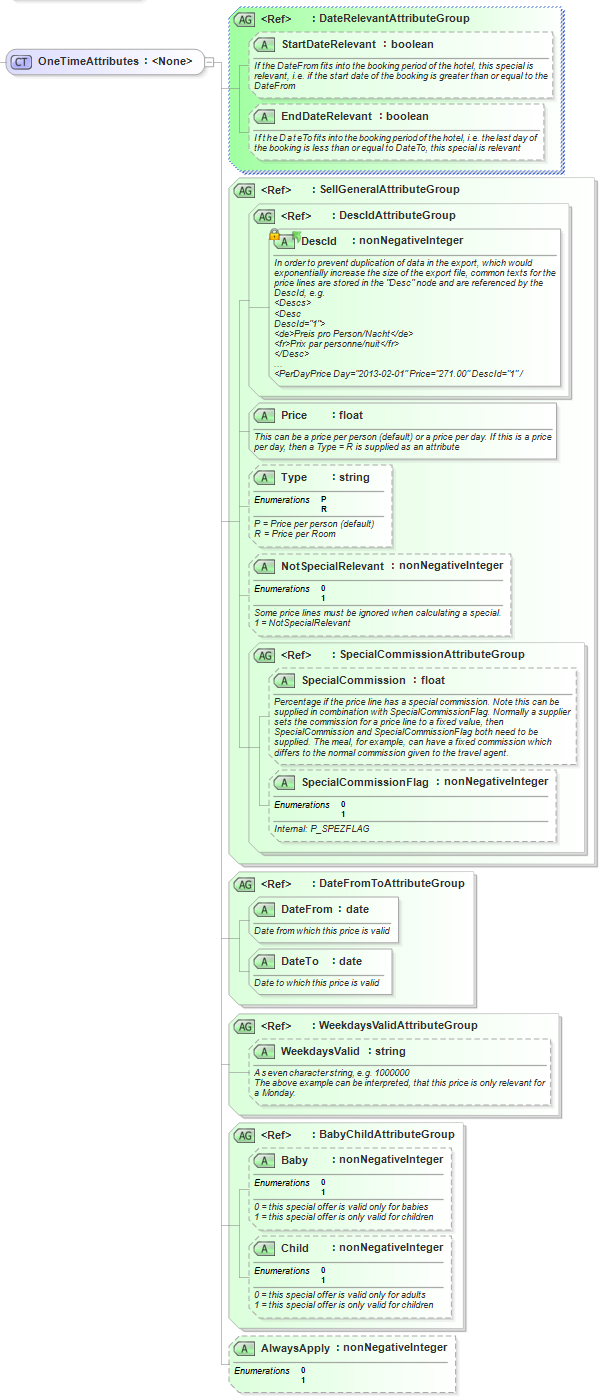


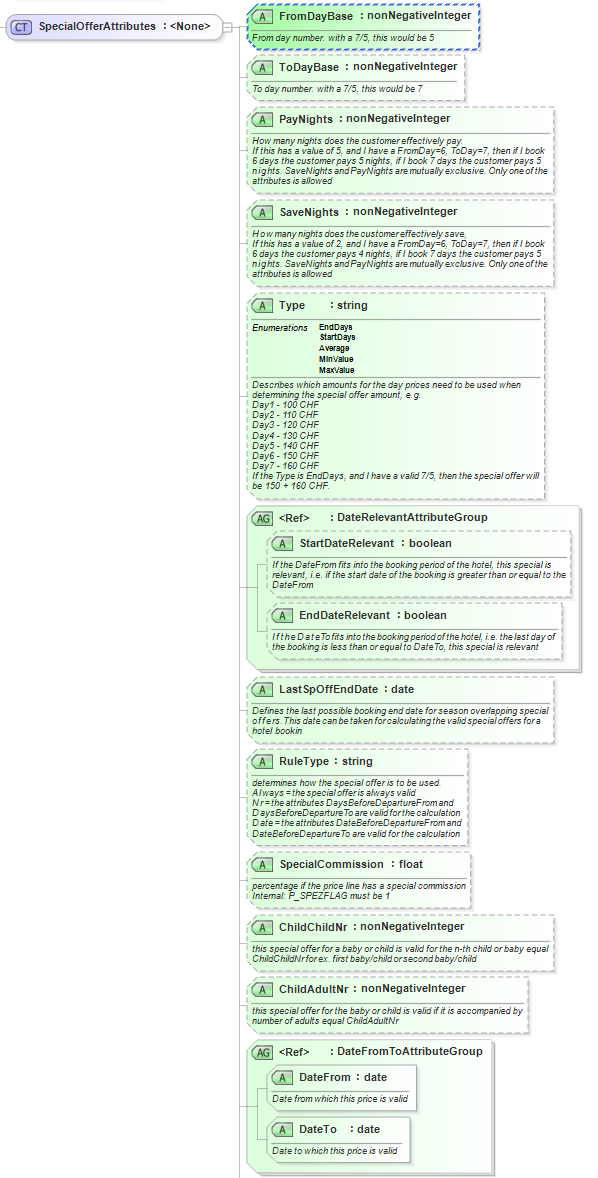




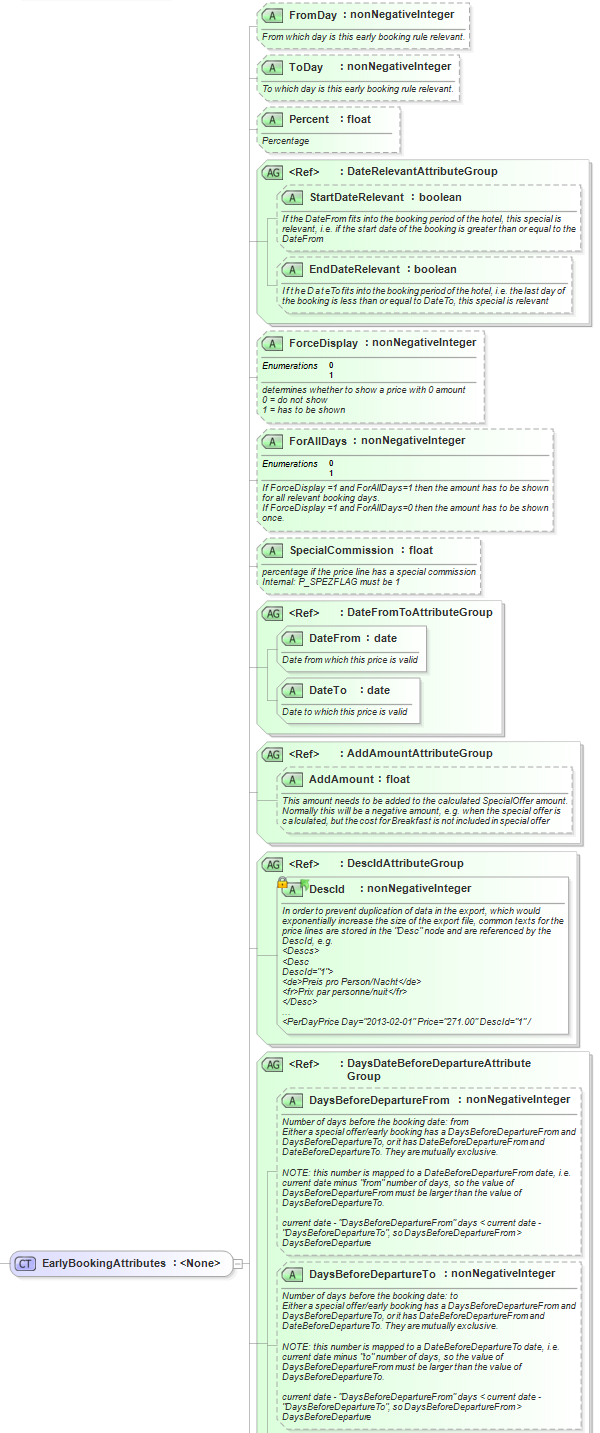


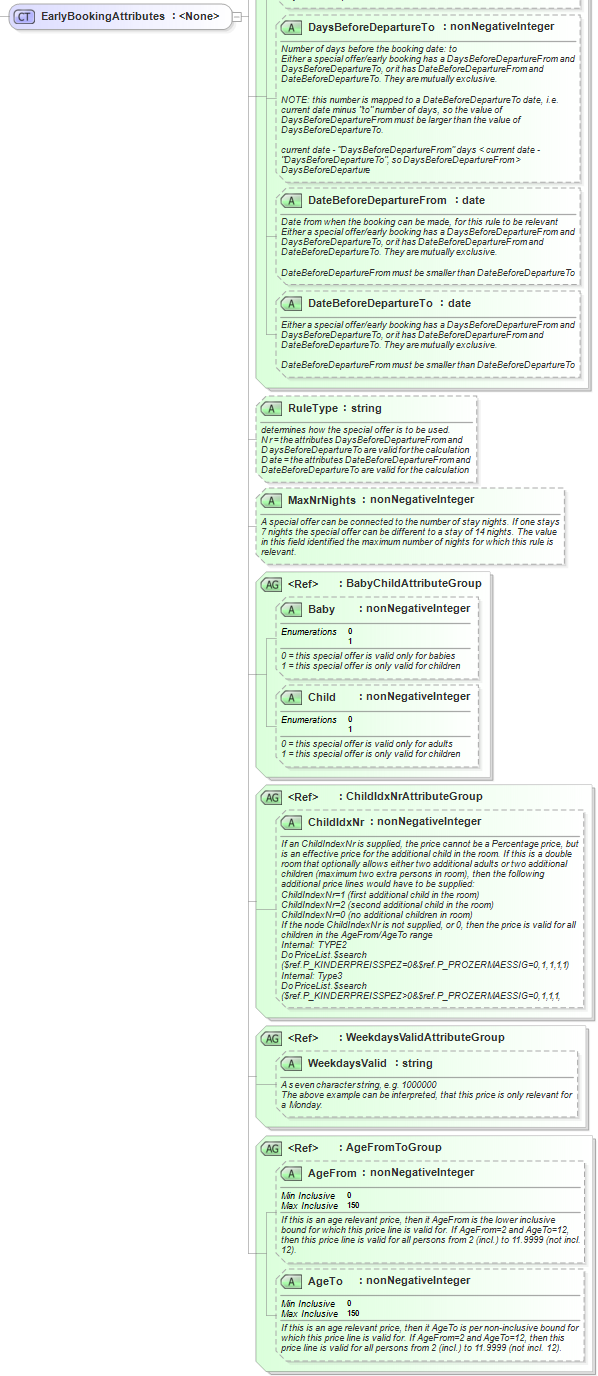


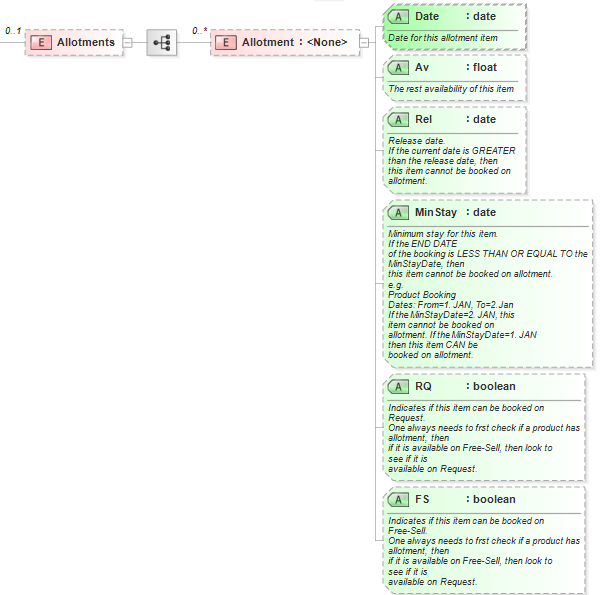












Allotments zeigen die Verfügbarkeit der Zimmer an. Es werden Allotments pro Tag (bei Hotelzimmer entspricht dies eine Übernachtung) exportiert.

* Av – der verfügbare Rest
* MinStay – Falls dieser Wert geliefert wird, muss bis mindestens dieses Datum (inklusive) im Hotel übernachtet werden, ansonsten ist eine Buchung nicht möglich.
* RQ – das Zimmer könnte, wenn kein Allotment verfügbar ist, auf Request gebucht werden können.
* FS – Bei Hotels kommt es normalerweise nicht vor, aber andere Arten von Leistungen, z.B. Autos, können auf FreeSell verkauft werden. Dies ist ein OK Status.

Ein Zimmer kann Kontingent haben. Dies bedeutet, dass ein Veranstalter vertraglich beim Hotel Zimmer reserviert. Nur der Veranstalter kann diese Zimmer dann verkaufen. Wenn der Veranstalter diese Plätze verkauft kann er diese dem Kunden direkt anbieten und dem Kunden die Zimmer sofort bestätigen, ohne Rückfrage beim Hotel. Wenn verkauft, informiert der Veranstalter einfach das Hotel. Der Prozess nennt sich Sell-and-Report.

Es gibt aber auch andere Fälle. Ein Veranstalter kann ein Zimmer verkaufen, sagt aber dem Käufer er frage noch beim Hotel nach ob es noch Plätze frei gibt. Dies kann telefonisch, per Mail oder über ein API geschehen. Preise sind aber immer noch hinterlegt. Ein Zimmer auf Anfrage zu Buchen heisst nicht dass dieses nicht buchbar ist, oder keine Preise hat, nur der Veranstalter muss nachfragen ob das Zimmer verfügbar ist.

Preise und Verfügbarkeiten haben in den meisten TO Systemen keinen Zusammenhang. In manchen TO Systemen gibt es für spezielle Allotments spezielle Preise, andere Allotments haben andere Preise und Leistungen auf Anfrage vielleicht andere Preise. Im TO Online ist dies aber nicht der Fall.

Es kann sein, dass ein Zimmer den Status RQ hat, aber diesese direkt beim Veranstalter über ein API angefragt werden kann. Der Veranstalter antwortet dann auf die Buchungsanfrage entweder mit OK, selten RQ, oder XX/CC.

Es kann aber auch Mischungen der Prozesse geben. Ein Veranstalter kann bei einem Hotel Kontingent haben, aber auch die Leistungen auf Request buchen. Zuerst wird vom Kontingent gebucht, danach wird die Leistung angefragt.

Wenn der Av>0 dann können die Leistungen als OK dargestellt werden. Jedes Zimmer wird als einzelnes Zimmer mit der Anzahl 1 verkauft.

Darf ein Zimmer verkauft werden, wenn Av=0.00 und RQ=1?

* Der Code sollte so strukturiert werden, dass dies möglich ist. Es ist immer dem TO überlassen ob dieser eine solche Buchung zulässt. Der TO lässt dann dem Reisebüro die Option offen diese Leistung zu buchen oder nicht. Meist erlauben die TOs Reisebüros Leistungen über RQ zu buchen, aber im B2C Fall kann diese Regel anders sein.

Wieso ist Av ein float-Wert?

* Der technische Grund ist, dass es sehr selten Leistungsarten gibt wo Anteile von 1 verkauft werden. In einer Jugendherberge kann ich 1 ½ Zimmer verkaufen, oder 1/3 Zimmer. Kommt aber fast nie vor, und für den produktiven Betrieb solltet der float immer als Int betrachtet werden.

Was ist, wenn Av=1.70 ist, bei der Anfrage jedoch zwei Zimmer benötigt werden?

* Es sollten nie solche Daten aus TO Online geliefert werden.

Wenn zwei Zimmer verkauft werden, dann sollte im XML auch zwei DossierItem geliefert werden. Der CMIR welcher genutzt wird um Daten an Tour Online und Cets zu übermitteln kennt nur die Anzahl 1. Daher sind zwei Einzelzimmer in TO Online zwei unabhängige Leistungen. Jede Leistung ist dann mit den entsprechenden Pax verknüpft.

Was die Berechnung angeht ob ein Zimmer gebucht werden kann oder nicht, und welcher Status das Zimmer hat, sollte folgendes berücksichtigt werden:

Zuerst muss eine Zusammenfassung der gesuchten Allotment-Tage-Daten gemacht werden. Dabei ist das AlltmtDateBegin und AlltmtDateEnd der Hotelleistung relevant welches gebucht werden soll. Bei Hotels gibt es Übernachtungen, daher muss hier besonders darauf geachtet werden das bei Hotels das End-Datum der Leistung nicht gleichzusetzen ist mit dem End-Datum der Allotment-Suche. Wenn die Zimmer-Leistung vom 1. JAN. auf den 2. JAN läuft, dann wird nur das Allotment vom 1. JAN. abgefragt.

AlltmtDateBegin = RoomDateBegin

AlltmtDateEnd = RoomDateEnd - 1 day -> aber nur bei Hotels

Eine Matrix für ein Allotment könnte wie folgt aussehen für eine Buchung vom 1 Januar Anrese bis zum 3 Januar Rückreise:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Datum** | **AVL** | **RELEASEDATE** | **MINSTAYDATE** | **RQ** | **FS** |
| 1 Januar | 5 | 25 Dezember | 2 Januar | 1 | 0 |
| 2 Januar | 6 |  | 4 Januar | 0 | 0 |
|  | **5** | **25 Januar** | **4 Januar** | **0** | **0** |

Folgende Regeln sind hier wirksam um die Tages-Liste zusammenzuführen:

* min([Avl]) as AVL
* min(coalesce([Rel],[AlltmtDateBegin])) as RELEASEDATE
* max(coalesce([Min],[AlltmtDateBegin])) as MINSTAYDATE
* min(case when [RQ] > 0 or [FS] > 0 then 1 else 0 end) as RQ
* min(case when [FS] > 0 then 1 else 0 end) as FS

Nachdem für alle Tagespreise zu einer Zeile zusammengefasst sind, gilt folgendes:

If Rel > current date

Calculate Avl as 0

End If

If Min < AlltmtDateEnd

Calculate Avl as 0

End If

If Avl > pRequired ;; pRequired wird hier normalerweise 1 sein

Calculate Status as 'OK' ;; Anzeige am Bildschirm: OK

Else If FS > 0

Calculate Status as 'FS' ;; Anzeige am Bildschirm: OK

Else If RQ > 0

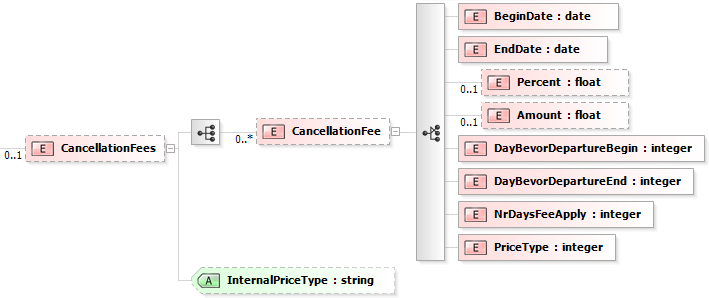
Calculate Status as 'RQ' ;; Anzeige am Bildschirm: RQ

Else

Calculate Status as 'AX' ;; Anzeige am Bildschirm: XX / nicht verfügbar

End If

Hier ist zu unterscheiden welche Stati an TO Online übermittelt werden und welche Stati am Bildschirm angezeigt werden könnten.



CancellationFees zeigen die Storno-Bedingungen des Zimmers auf.

## PerDayPrices

### Erwachsene

PerDayPrices sind Tagespreise.

Es wird ein Export vorgenommen auch wenn es in einem Zeitraum keine zusammenhängende Preise gibt. Koblenz entscheidet ob ein angefragter Zeitraum angezeigt wird oder nicht. Wen bei einer angefragten Leistungsperiode ein Preis fehlt, dann darf diese nicht angezeigt werden. Ein Export erfolgt immer dann, wenn sich an den Stammdaten etwas geändert hat.

Tagespreise sind im Allgemeinen eine Preis-Pro-Person. Es ist aber möglich, dass ein Preis als „Preis pro Zimmer“ definiert wird. Wenn nicht anders angegeben ist ein Tagespreis immer pro Person.

Hier ein Beispiel der relevanten Nodes für einen Tagespreis:

<Descs>

<Desc DescId="1">

<de>Preis pro Person/Nacht</de>

<fr>Prix par personne/nuit</fr>

</Desc>

</Descs>

<Prices>

<Currency IsoCode="CHF">

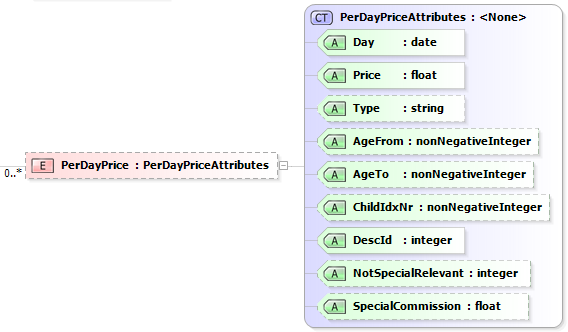
<PerDayPrices>

<PerDayPrice Day="2013-04-18"

Price="46.00"

DescId="1" />

Und hier die dazugehörige XSD:



Für die meisten Tagespreise, zumindest für Erwachsene, werden nur die drei Attribute Day, Price, DescId geliefert.

Das Attribut Day gibt an für welchen Tag der Preis relevant ist. Im Beispiel oben wäre der Preis für eine Übernachtung vom 18. April 2013 bis zum 19. April 2013 relevant.

Der Node Type ist optional. Wenn nicht geliefert hat er implizit den Wert "P", also Preis pro Person. Wird der Node geliefert und hat den Wert "R" dann gilt ein Preis pro Zimmer.

Damit die Beschreibungstexte nicht unnötigerweise immer wieder wiederholt werden müssen sind alle Preiszeilentexte eines XML ein den Node <Descs> ausgelagert. Im Beispiel oben hat der PerDayPrice die DescId="1".

Der Preis-Beschreibungs-Text

<Desc DescId="1">

<de>Preis pro Person/Nacht</de>

<fr>Prix par personne/nuit</fr>

</Desc>

muss dann aus dem Desc-Node über die <Desc DescId="1"> über die jeweilige Sprache geholt werden.

Der Node NotSpecialRelevant gibt an ob der Preis an diesem Tag für Specials berücksichtigt werden soll oder nicht.

Es kann auch vorkommen, dass ein zusätzlicher Erwachsener im Zimmer gebucht wird, z.B.

<NormalOccupancy>3</NormalOccupancy>

<MinimalOccupancy>3</MinimalOccupancy>

<MaximalOccupancy>4</MaximalOccupancy>

<MaxAdults>4</MaxAdults>

<ExtraBedAdults>1</ExtraBedAdults>

<ExtraBedChildren>1</ExtraBedChildren>

In diesem Fall zahlt der vierte Erwachsene den normalen Erwachsenen-Preis.

Es muss immer mindesten ein Erwachsener im Zimmer gebucht werden.

### Kinderpreise

Ein Zimmer hat eine Mindestbelegung, z.B.

<MinimalOccupancy>3</MinimalOccupancy>

Ist ein Kind in einem Zimmer, und wenn die Mindestbelegung nicht mit Erwachsenen voll belegt ist, gilt ein Kind als Erwachsener und zahlt den Erwachsenenpreis.

Falls nicht explizit angegeben, zahlen Babys mit einem Alter von 0 bis (aber nicht inklusive) 2 Jahren nichts. Ein Baby wird nicht als Kind oder Erwachsener gewertet, ausser es gibt für das Baby explizit einen Preis.

Ein Kinderpreis könnte wie folgt aussehen:

<PerDayPrice Day="2014-03-27"

Price="99.00"

AgeFrom="2"

AgeTo="14"

ChildIdxNr="1"

DescId="2" />

<PerDayPrice Day="2014-03-27"

Price="70.00"

AgeFrom="2"

AgeTo="14"

ChildIdxNr="2"

DescId="3" />

Alle Kinderpreise im TFS wurden beim Export so angepasst, dass das Beispiel oben wie folgt gelesen werden kann:

Das erste zusätzliche Kind im Zimmer zahlt am "2014-03-27" den Betrag "99.00" CHF, das zweite zusätzliche Kind im Zimmer zahlt am "2014-03-27" den Betrag "70.00" CHF.

Selbst wenn der Preis für das erste, zweite oder dritte Kind gleich ist werden die Kindertagespreise immer auseinandergesplittet geliefert, pro Tag und für das erste, dann zweite dann dritte Kind im Zimmer der Erwachsenen.

Das Attribut AgeTo ist immer nicht-inklusive, z.B. AgeTo="14" gilt bis zum Alter "13.9999" aber nicht für das Alter "14".

Das Alter muss immer auf Basis des ersten Leistungstags im Zimmer berechnet werden.

## Additional Per Day Preise (AddPerDayPrices)

Diese Preise müssen pro Tag dazugerechnet werden. Ein Beispiel wäre ein Halbpensions-Zuschlag. Es ist manchmal notwendig diese Zuschläge vom Tagespreis zu trennen da vielleicht der Tagespreis relevant ist für einen Special-Offer, der Zuschlag aber nicht.

<AddPerDayPrice

Day="2013-04-18"

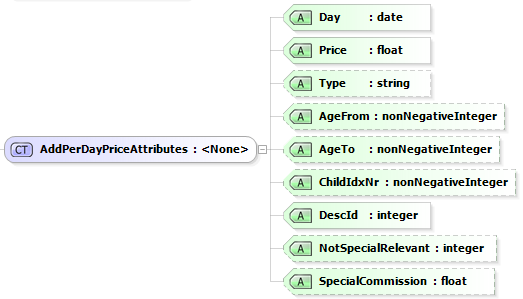
Price="42.00"

DescId="2"

NotSpecialRelevant="1" />

Der Node NotSpecialRelevant="1" bedeutet das dieses Preiselement bei Special Offers und Early Booking nicht relevant ist.

Für die DescID gilt das bereits gesagte. Der Text muss entsprechend der Sprache aus dem Desc-Node anhand der DescID geholt werden.



## Einmaliger Zuschlag (OneTimes)

Sind eimalige Zuschläge pro Pax (Room) und angegebenen Tag DateFrom="2013-12-24"

<OneTime Price="580.00"

DateTo="2013-11-10"

DescId="6"

DateFrom="2013-11-10"

WeekdaysValid="1111111"

EndDateRelevant="0"

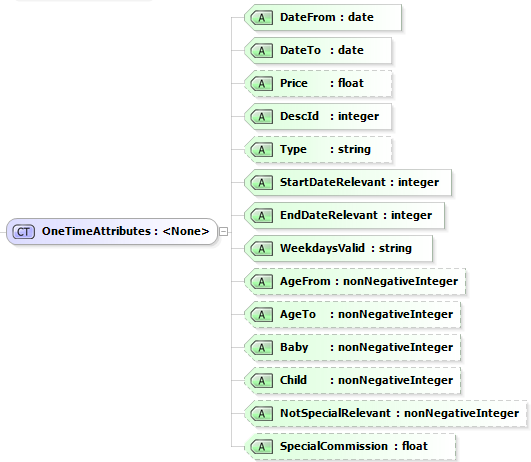
SpecialCommission="5.00"

StartDateRelevant="0"

NotSpecialRelevant="1" />

Spezielle Kommisionen!

Wenn das Attribute SpecialCommision >=0 und intern auf dem Preis das SpecialCommission Flag gesetzt ist, so muss diese Kommission für diesen Preis genommen werden. Und es gilt nur diese.



## SpecialOffers

### Allgemein

Bei dem Special gelten immer die für den Buchungszeitraum maximal gültigen Specials .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Specials | Buchungstage | | | | | | | | | | | | | Special ist gültig |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 für 6 |  |  |  |  |  |  |  |  |  |  |  |  |  | ja |
| 4 für 3 |  |  |  |  |  |  |  |  |  |  |  |  |  | nein |
| 4 für 3 |  |  |  |  |  |  |  |  |  |  |  |  |  | ja |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Das heisst man versucht am Anfang das grösstmögliche Special zu nehmen.

Für die Berechnung sind noch die folgenden Bedingungen zu berücksichtigen:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Specials | Buchungstage | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Price per Day | 100 | 110 | 105 | 99 | 102 | 102 | 99 |  |  |  |  |  |  |
| 7 für 6 Min | - 99 | | | | | | | | | | | | |
| 7 für 6 Max | - 110 | | | | | | | | | | | | |
| 7 für 6 Average | - (100 + 110 + 105 + 99 + 102 + 102 + 99) / 7 | | | | | | | | | | | | |
| 7 für 6 First Day | - 100 | | | | | | | | | | | | |
| 7 für 6 Last Day | - 99 | | | | | | | | | | | | |

<Type>:

MinValue: Es wird der Minimum Preis innerhalb des Specials genommen

MaxValue: Es wird der Maximum Preis innerhalb des Specials genommen

Average: Es wird der Durchschnitts Preis innerhalb des Specials genommen

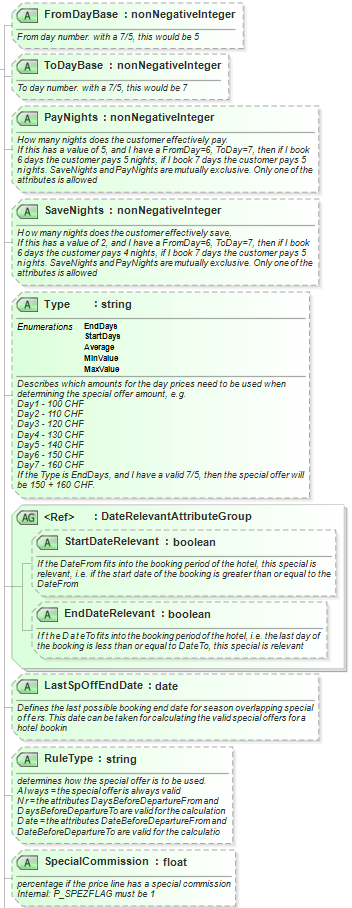
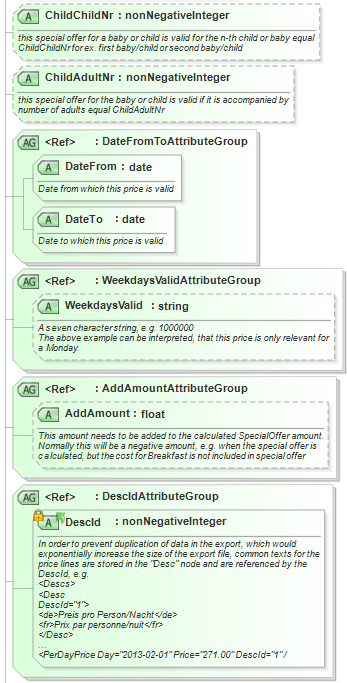
StartDays: Es gilt der Preis des ersten Tages innerhalb des Specials

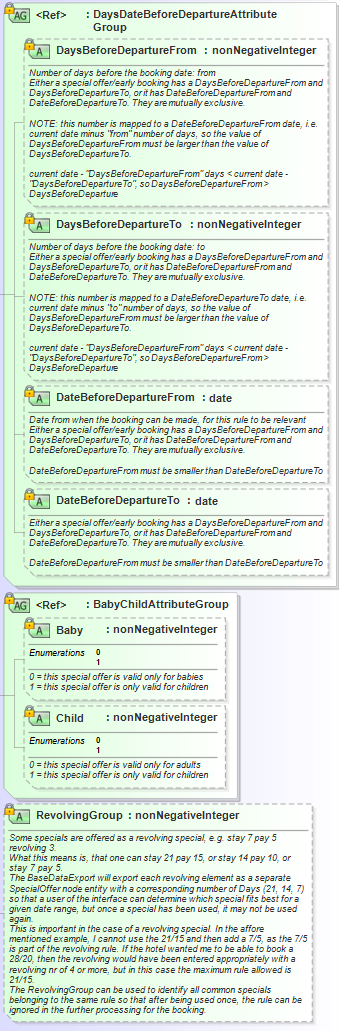
EndDays: Es gilt der Preis des letzten Tages innerhalb des Specials

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Specials | Buchungstage | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Price per Day | 100 | 110 | 105 | 99 |  |  |  |  |  |  |  |  |  |
| 4 für 2 Min | -(99+100) | | | | | | | | | | | | |
| 4 für 2 Max | -(110+105) | | | | | | | | | | | | |
| 4 für 2 Average | -(100+110+105+99)/4 | | | | | | | | | | | | |
| 4 für 2 First Day | -(100+110) | | | | | | | | | | | | |
| 4 für 2 Last Day | -(105+99) | | | | | | | | | | | | |

Es folgt eine Beschreibung der einzelnen Attribute eines SpecialOffer Nodes:







### Regeln für Special Offers Saison übergreifend

#### Grund Regel

* Bei Saisonübergreifungen wird zuerst die Kleinste genommen die noch in die neue Saison hineinreicht. Und dann die grösste aus der neuen Saison sofern das grösste mögliche Enddatum <LastSpOffEndDate> der alten Saison noch in die neue Saison übergeht.
* Ansonsten gilt immer die grösstmögliche Saison der Specials soll genommen werden.

Hotel vom 1.1.2013 bis zum 10.1.2013



Folgende Special Offer Regeln kann es geben die für die Berechnung von den effektiven Special Offers:



Um die für einen Anfragezeitraum relevanten Regeln zu finden kann folgendes SQL verwendet werden:

and

(

(

HotelStartDate between DateFrom and DateTo

and HotelEndDate between DateFrom and DateTo

)

or

(

DateFrom between HotelStartDate and HotelEndDate

and DateTo between HotelStartDate and HotelEndDate

)

or

(

HotelStartDate between DateFrom and DateTo

and HotelEndDate > DateTo

)

or

(

HotelEndDate between DateFrom and DateTo

and HotelStartDate < DateFrom

)

)

Als nächstes den Saison-Anfang der Regeln auf das Anfangsdatum des Buchungszeitraums setzen.



Jetzt müssen die Regeln sortiert werden, nach DateFrom aufsteigend, und Anzahl Tage absteigend, und DateTo absteigend.



Hier gilt die 4/3 Regel. Sie ist die kleinste in der alten Saison und das LastSpOffEndDate liegt in der neuen Saison, da wir keine Saisonunterbrüche haben.

Fang beim ersten Tag an, und suche jeden Tag ob eine der Regeln für den Buchungszeitraum passt.

1.1.2013: Die erste passende Regel ist die Regel1. Diese passt in die Saison und den Buchungszeitraum. Danach kommt die Regel 3 da diese anschliessend ist.



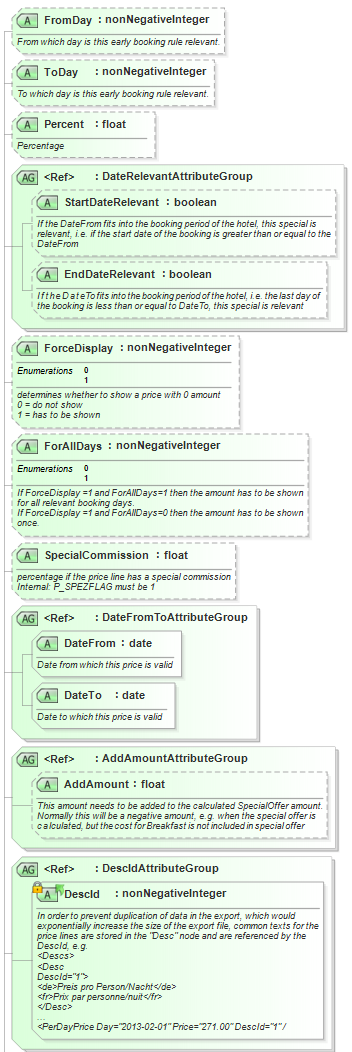
Hätten wir einen fall wie den unteren. Das heisst wir haben eine Lücke so gilt:

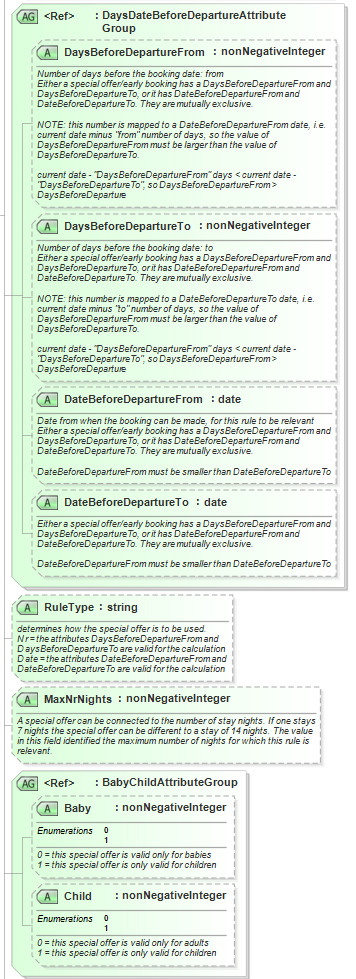


In diesem Fall gibt es eine Lücke in den Saisons das heisst die Regel5 hat nicht das gleiche LastSpOffEndDate wie Regel4. Daher gilt vom 1.1 bis zum 1.7 die 7/5 Regel und vom 17.1 bis 20.1 die Regel 4, 4/3 Regel

## Early Booking

Frühbuchherrabatte. Diese können definiert sein als fliessende Anzahl Tage vor Abreise, oder fixe Datumsangaben enthalten, entweder das Eine oder das Andere. Rabatte können prozentueller Natur sein, oder einen Betrag enthalten, oder aus beiden Elementen bestehen. Im Falle das beide Daten angegeben sind wird zuerst der End-Betrag anhand des prozentuellen Rabatts berechnet, dann der Betrag hinzuaddiert. Dies erlaubt es Situationen abzubilden wo Rabatte gewährt werden, nur das obligatorische Frühstück von X Franken ist vom Rabatt ausgeschlossen.

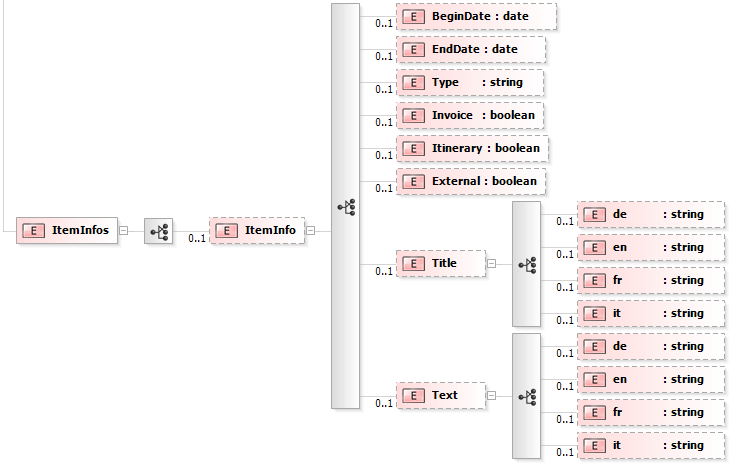






## Item Infos

Item Imfos sind Informationen zu dem Produkt welche im Buchungsablauf beim Buchen des Produktes angezeigt werden müssten, z.B. „Arbeit am Schwimmbad von X bis Y“, oder allgemeine Hinweise beim Buchen des Produkts.



Beispiel:

<ItemInfos>

<ItemInfo>

<BeginDate>2011-03-01</BeginDate>

<EndDate>2015-12-31</EndDate>

<Type>Destination</Type>

<Invoice>1</Invoice>

<Itinerary>0</Itinerary>

<External>0</External>

<Title>

<de>Hinweis:</de>

<fr />

</Title>

<Text>

<de><![CDATA[Während Feiertagen, Messezeiten oder Veranstaltungsperioden werden Zuschläge erhoben, welche nach Festlegung der Event-Daten bekannt gegeben werden und zum Zeitpunkt des Katalogdrucks tlw. noch nicht bekannt waren!]]></de>

<fr />

</Text>

</ItemInfo>

<ItemInfo>

<BeginDate>2013-11-01</BeginDate>

<EndDate>2014-10-31</EndDate>

<Type>Item</Type>

<Invoice>1</Invoice>

<Itinerary>1</Itinerary>

<External>1</External>

<Title>

<de>Hinweis:</de>

<fr />

</Title>

<Text>

<de><![CDATA[Das Zimmer kann für 24 Stunden genutzt werden. Massgeblich ist der Zeitpunkt des Checks-ins.]]></de>

<fr />

</Text>

</ItemInfo>

<ItemInfo>

<BeginDate>2013-11-01</BeginDate>

<EndDate>2014-10-31</EndDate>

<Type>Item</Type>

<Invoice>1</Invoice>

<Itinerary>1</Itinerary>

<External>1</External>

<Title>

<de>Hinweis:</de>

<fr />

</Title>

<Text>

<de><![CDATA[20% Ermässigung im Spa]]></de>

<fr />

</Text>

</ItemInfo>

</ItemInfos>

Nur ItemInfos mit Invoice=true oder External=true sollten im Buchungsablauf angezeigt werden.

## Beispiele

Beispiel 1) XML: AUH\_TESTHO\_A2.xml

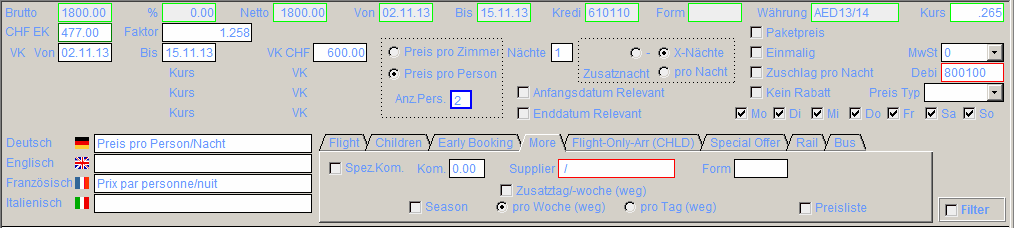
Buchungszeitraum 2.11.13 – 12.11.13

2 Erwachsene

2 Kinder 7.5 Jahre und 5 Jahre

Suche alle PerDayPreise vom 2.11.13 bis 11.11.13

Erwachsene:



<PerDayPrice Day="2013-11-02" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-03" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-04" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-05" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-06" Price="600.00" DescId="1" />

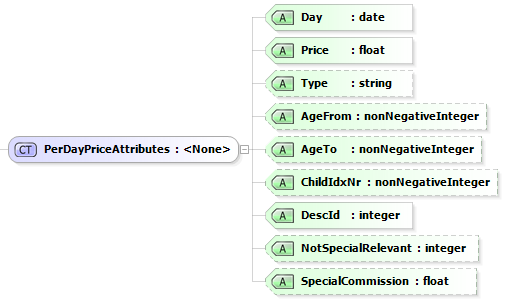
<PerDayPrice Day="2013-11-07" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-08" Price="600.00" DescId="1" />

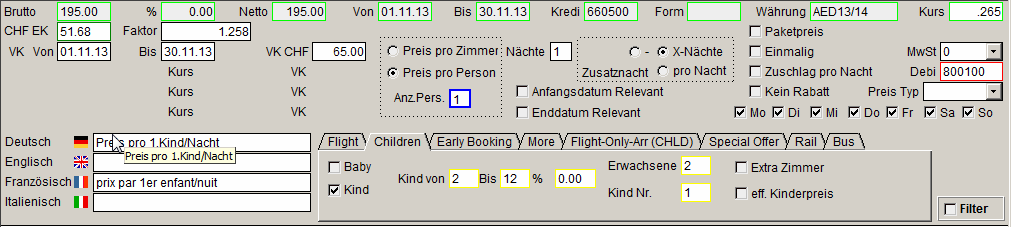
<PerDayPrice Day="2013-11-09" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-10" Price="600.00" DescId="1" />

<PerDayPrice Day="2013-11-11" Price="600.00" DescId="1" />



Kind 1:



<PerDayPrice Day="2013-11-02" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-03" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-04" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-05" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-06" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-07" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-08" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-09" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

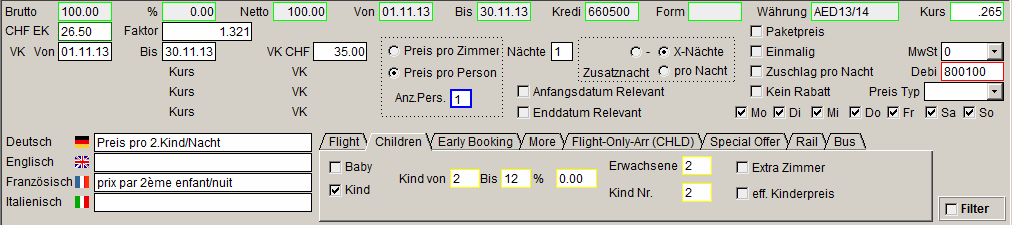
<PerDayPrice Day="2013-11-10" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

<PerDayPrice Day="2013-11-11" AgeTo="12" Price="65.00" DescId="2"

AgeFrom="2" ChildIdxNr="1" />

Kind 2:



<PerDayPrice Day="2013-11-02" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-03" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-04" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-05" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-06" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-07" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-08" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-09" AgeTo="12" Price="35.00" DescId="3"

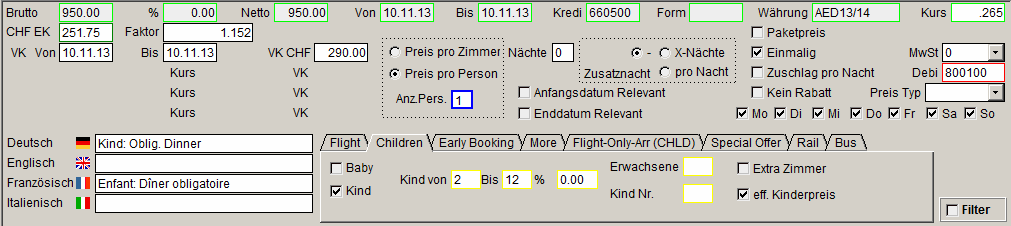
AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-10" AgeTo="12" Price="35.00" DescId="3"

AgeFrom="2" ChildIdxNr="2" />

<PerDayPrice Day="2013-11-11" AgeTo="12" Price="35.00" DescId="3" AgeFrom="2" ChildIdxNr="2" />

Suche alle OneTimes Preise für diesen Buchungszeitraum



<OneTime Price="290.00" (Kind: Oblig. Dinner)

DateTo="2013-11-10"

DescId="5"

DateFrom="2013-11-10"

AlwaysApply="0"

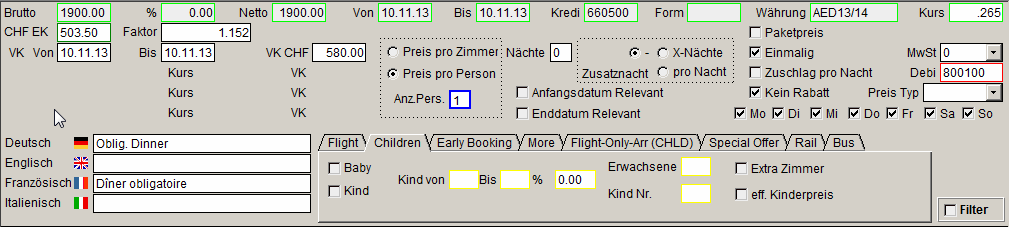
WeekdaysValid="1111111"

EndDateRelevant="0"

StartDateRelevant="0"

/>





<OneTime Price="580.00" (Oblig. Dinner)

DateTo="2013-11-10"

DescId="6"

DateFrom="2013-11-10"

AlwaysApply="0"

WeekdaysValid="1111111"

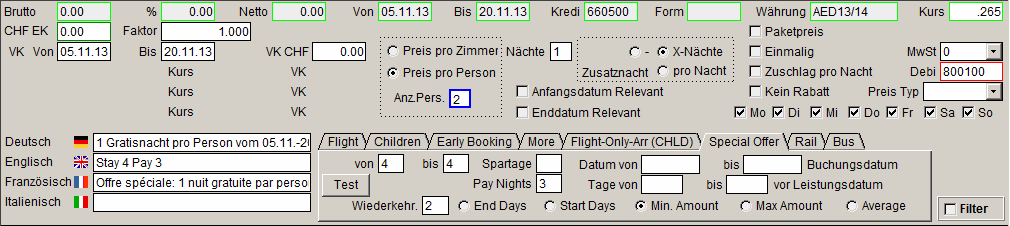
EndDateRelevant="0"

StartDateRelevant="0"

NotSpecialRelevant="1"

/>

Suche alle SpecialOffers für diesen Buchungszeitraum



<SpecialOffer Baby="0"

Days="4

Type="MinValue"

Child="0"

DateTo="2013-11-20"

DescId="8"

DateFrom="2013-11-05"

RuleType="Always"

AddAmount="0.00"

DayString="4"

PayNights="3"

ToDayBase="4"

FromDayBase="4"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2013-11-30"

StartDateRelevant="0

DaysBeforeDepartureTo=""

DaysBeforeDepartureFrom=""

/>

<SpecialOffer Baby="0"

Days="8"

Type="MinValue"

Child="0"

DateTo="2013-11-20"

DescId="8"

DateFrom="2013-11-05"

RuleType="Always"

AddAmount="0.00"

DayString="4,4"

PayNights="3"

ToDayBase="4"

FromDayBase="4"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

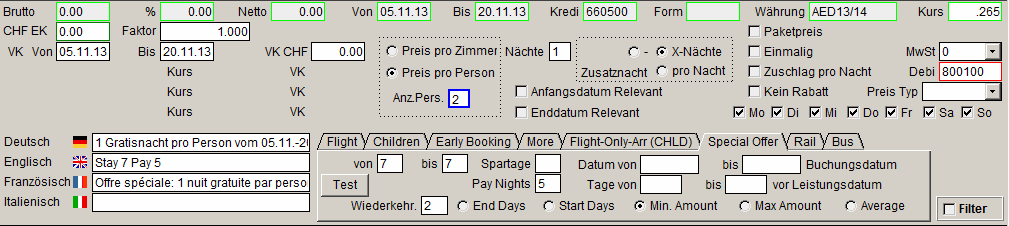
LastSpOffEndDate="2013-11-30"

StartDateRelevant="0

DaysBeforeDepartureTo=""

DaysBeforeDepartureFrom=""

/>



<SpecialOffer Baby="0"

Days="7"

Type="MinValue"

Child="0"

DateTo="2013-11-20"

DescId="8"

DateFrom="2013-11-05"

RuleType="Always"

AddAmount="0.00"

DayString="7"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2013-11-30"

StartDateRelevant="0"

DaysBeforeDepartureTo=""

DaysBeforeDepartureFrom=""

/>

<SpecialOffer Baby="0"

Days="14"

Type="MinValue"

Child="0"

DateTo="2013-11-20"

DescId="8"

DateFrom="2013-11-05"

RuleType="Always"

AddAmount="0.00"

DayString="7,7"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

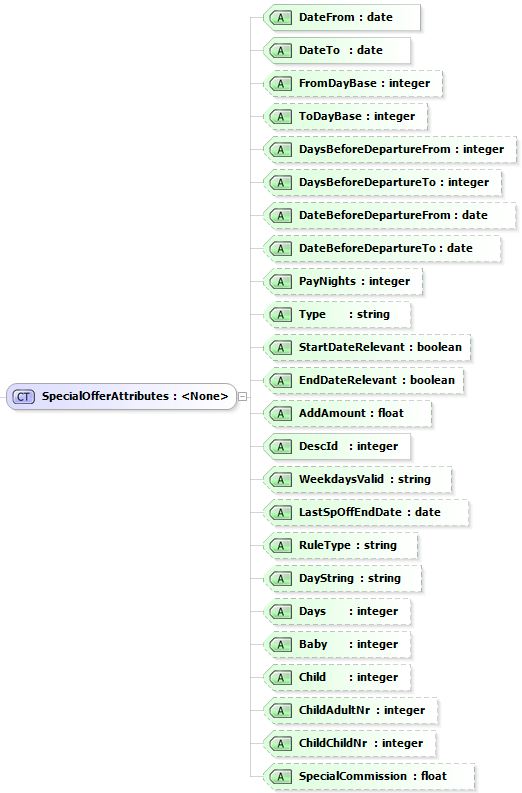
LastSpOffEndDate="2013-11-30"

StartDateRelevant="0

DaysBeforeDepartureTo=""

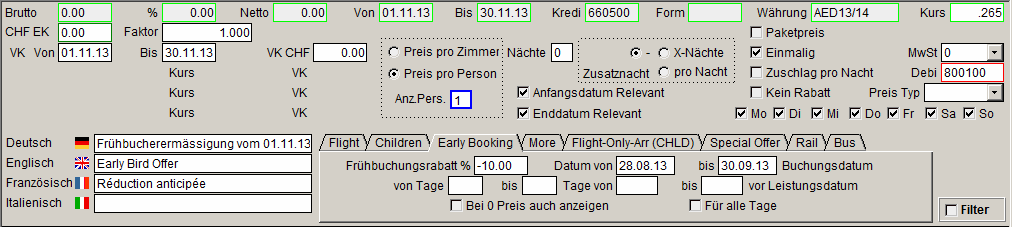
DaysBeforeDepartureFrom=""

/>





Suche EarlyBookings für den Buchungszeitraum



<EarlyBooking ToDay="0"

DateTo="2013-11-30"

DescId="9"

FromDay="0"

Percent="-10.00"

DateFrom="2013-11-01"

AddAmount="0.00"

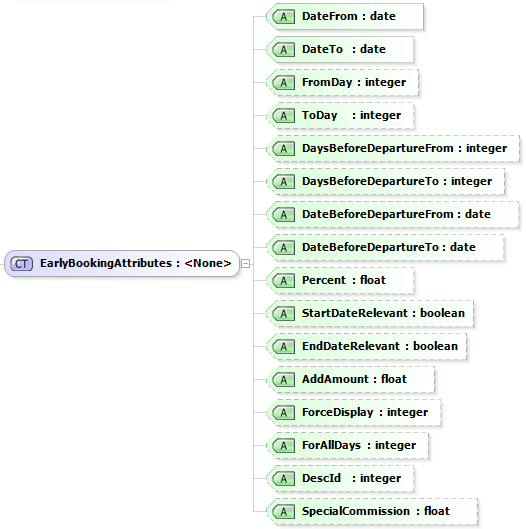
EndDateRelevant="1"

StartDateRelevant="1"

DateBeforeDepartureTo="2013-09-30"

DateBeforeDepartureFrom="2013-08-13"

/>



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Anzahl | Beschreibung | Preis | Total | Bemerkung |
| 20 | Preis pro Person | 600 | 12000 |  |
| 2 | Oblig. Dinner | 580 | 1160 |  |
| 1 | Frühbucherermässigung vom 01.11.13-30.11.13 bei Buchung bis 30.09.13 (1 x 10) | -960 | -960 |  |
| 10 | Preis pro 1.Kind/Nacht (1x 10) | 65 | 650 |  |
| 10 | Preis pro 2.Kind/Nacht(1x 10) | 35 | 350 |  |
| 2 | Kind Oblig. Dinner | 290 | 580 |  |
| 4 | 1 Gratisnacht pro Person vom 05.11.-10.11.13 | -600 | -2400 | Hier gilt MinValue es wird der kleinste Preis für diesen Zeitraum gesucht. |

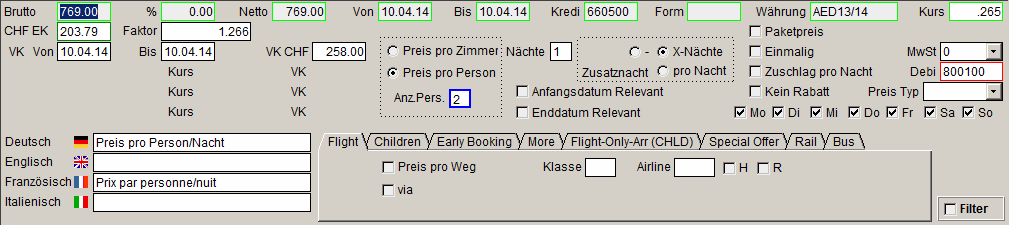
Beispiel 2) XML: AUH\_QARALB\_B2\_HB.xml

Buchungszeitraum 31.3.14 – 10.4.14

2 Erwachsene

Suche alle PerDayPreise vom 31.3.14 – 09.4.14

Erwachsene:



<PerDayPrice Day="2014-03-31" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-01" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-02" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-03" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-04" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-05" Price="258.00" DescId="1" />

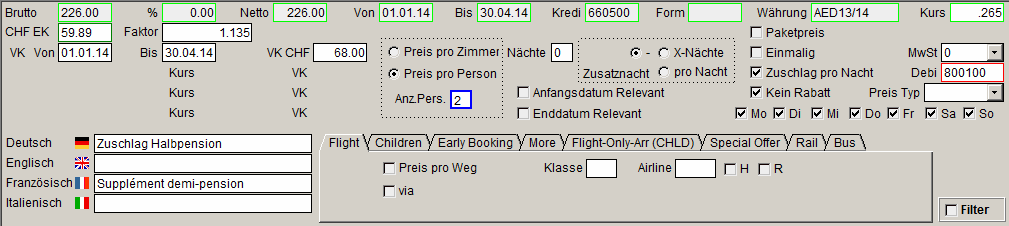
<PerDayPrice Day="2014-04-06" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-07" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-08" Price="258.00" DescId="1" />

<PerDayPrice Day="2014-04-09" Price="258.00" DescId="1" />

Suche alle AddPerDayPrice für diesen Buchungszeitraum



<AddPerDayPrice Day="2014-03-31" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-01" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-02" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-03" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-04" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-05" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-06" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-07" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

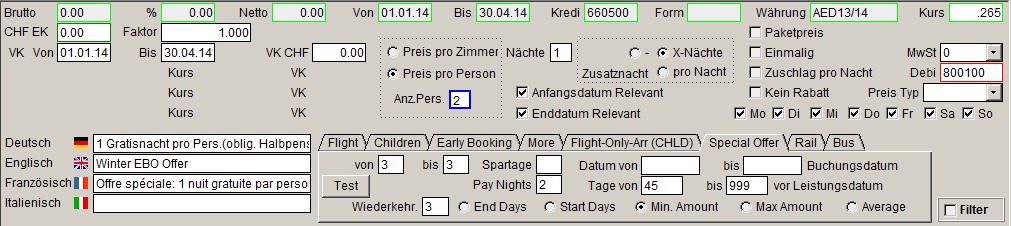
<AddPerDayPrice Day="2014-04-08" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-04-09" Price="68.00" DescId="6"

NotSpecialRelevant="1" />

Suche alle SpecialOffers für diesen Buchungszeitraum



<SpecialOffer Baby="0"

Days="3"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="3"

PayNights="2"

ToDayBase="3"

FromDayBase="3"

ChildAdultNr="0"

ChildChildNr="0" "

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-31"

StartDateRelevant="1

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>

<SpecialOffer Baby="0"

Days="6"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="3,3"

PayNights="2"

ToDayBase="3"

FromDayBase="3"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-31"

StartDateRelevant="1"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>

<SpecialOffer Baby="0"

Days="9"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="3,3,3"

PayNights="2"

ToDayBase="3"

FromDayBase="3"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

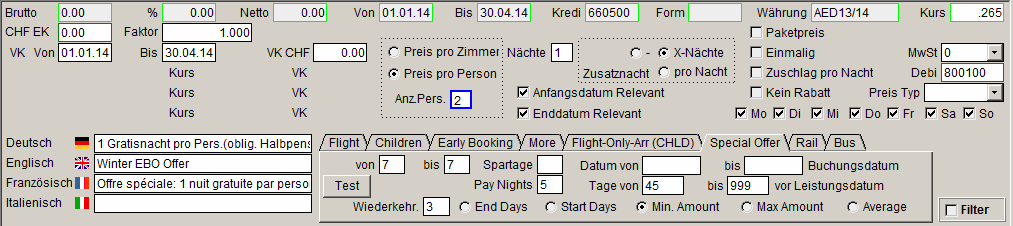
LastSpOffEndDate="2014-10-31"

StartDateRelevant="1"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>



<SpecialOffer Baby="0"

Days="7"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="7"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-31"

StartDateRelevant="1"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>

<SpecialOffer Baby="0"

Days="14"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="7,7"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-31"

StartDateRelevant="1"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>

<SpecialOffer Baby="0"

Days="21"

Type="MinValue"

Child="0"

DateTo="2014-04-30"

DescId="9"

DateFrom="2014-01-01"

RuleType="Nr"

AddAmount="0.00"

DayString="7,7,7"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-31"

StartDateRelevant="1"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45"

/>



|  |  |  |  |
| --- | --- | --- | --- |
| Anzahl | Beschreibung | Preis | Total |
| 20 | Preis pro Person/Nacht (2 x 10) | 258 | 5160 |
| 20 | Zuschlag Halbpension (2 x 10) | 68 | 1360 |
| 2 | 1 Gratisnacht pro Pers.(oblig. Halbpension berücksichtigt) vom 01.01.14-30.04.14 | -258 | -516 |
| 2 | 1 Gratisnacht pro Pers.(oblig. Halbpension berücksichtigt) vom 01.01.14-30.04.14 | -258 | -516 |
| 2 | 1 Gratisnacht pro Pers.(oblig. Halbpension berücksichtigt) vom 01.01.14-30.04.14 | -258 | -516 |

Beispiel 3) XML: DXB\_FAIJUM\_E2222\_HB.xml

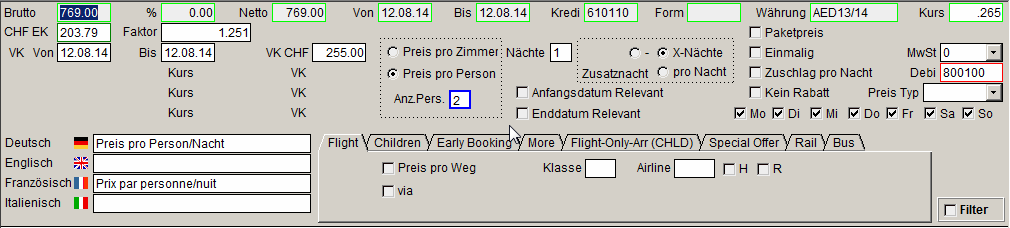
Buchungszeitraum 02.08.14 – 12.08.14

2 Erwachsene

1 Kind (7.5 Jahre)

Suche alle PerDayPreise vom 02.08.14 – 12.08.14

Erwachsene:



<PerDayPrice Day="2014-08-02" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-03" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-04" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-05" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-06" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-07" Price="255.00" DescId="1" />

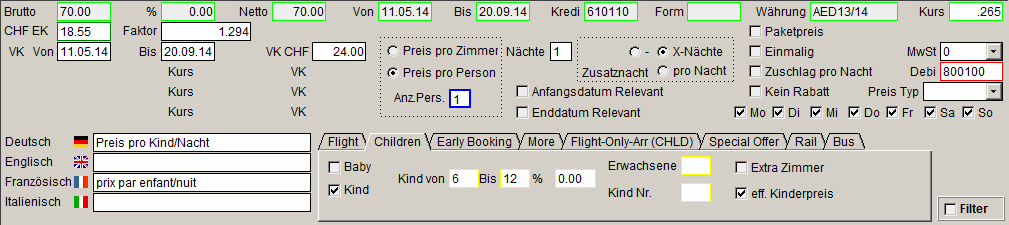
<PerDayPrice Day="2014-08-08" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-09" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-10" Price="255.00" DescId="1" />

<PerDayPrice Day="2014-08-11" Price="255.00" DescId="1" />

Kind 1:



<PerDayPrice Day="2014-08-02" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-03" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-04" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-05" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-06" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-07" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-08" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-09" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

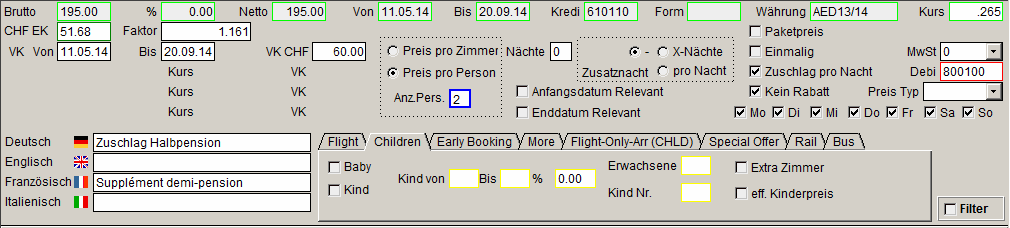
<PerDayPrice Day="2014-08-10" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

<PerDayPrice Day="2014-08-11" AgeTo="12" Price="24.00" DescId="2"

AgeFrom="6" ChildIdxNr="1" />

Suche alle AddPerDayPrice für diesen Buchungszeitraum



<AddPerDayPrice Day="2014-08-02" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-03" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-04" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-05" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-06" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-07" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-08" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-09" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

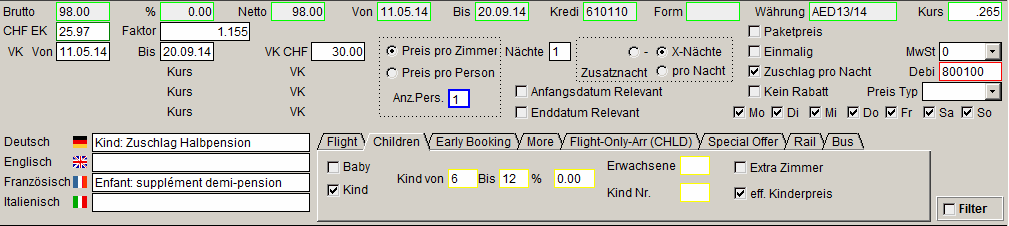
<AddPerDayPrice Day="2014-08-10" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-11" Price="60.00" DescId="7"

NotSpecialRelevant="1" />

Kind:



<AddPerDayPrice Day="2013-11-01" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-02" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-03" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-04" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-05" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-06" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-07" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-08" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-09" Type="R" AgeTo="12" Price="30.00"

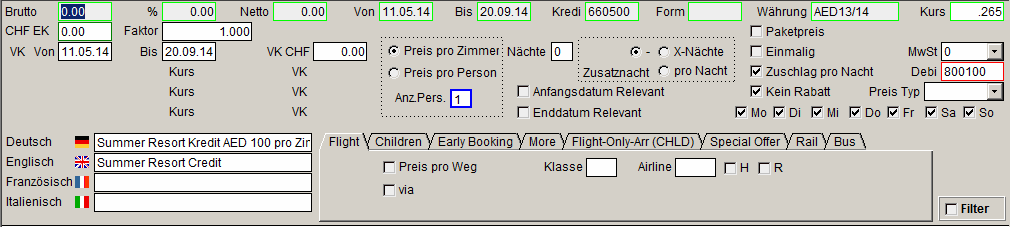
DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-10" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />

<AddPerDayPrice Day="2013-11-11" Type="R" AgeTo="12" Price="30.00"

DescId="6" AgeFrom="6" ChildIdxNr="1" />



<AddPerDayPrice Day="2014-08-02" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-03" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-04" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-05" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-06" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-07" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-08" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-09" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

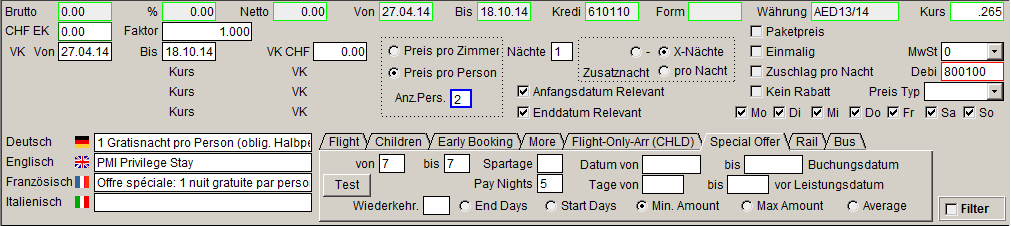
<AddPerDayPrice Day="2014-08-10" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

<AddPerDayPrice Day="2014-08-11" Type="R" Price="0.00" DescId="6"

NotSpecialRelevant="1" />

Suche alle SpecialOffers für diesen Buchungszeitraum



<SpecialOffer Baby="0"

Days="7"

Type="MinValue"

Child="0"

DateTo="2014-10-18"

DescId="8"

DateFrom="2014-04-27"

RuleType="Always"

AddAmount="0.00"

DayString="7“"

PayNights="5"

ToDayBase="7"

FromDayBase="7"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="1"

LastSpOffEndDate="2014-10-18"

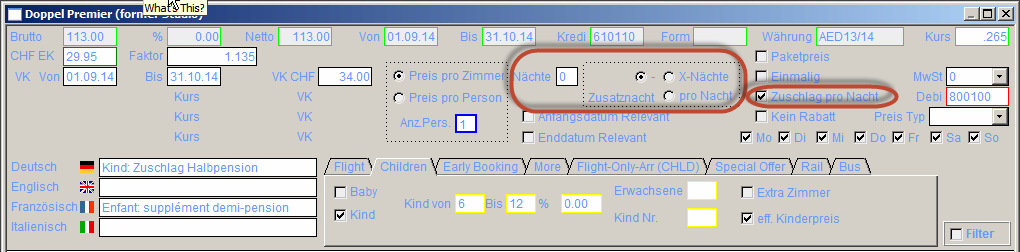
StartDateRelevant="1"



|  |  |  |  |
| --- | --- | --- | --- |
| Anzahl | Beschreibung | Preis | Total |
| 20 | Preis pro Person/Nacht | 255 | 5100 |
| 20 | Zuschlag Halbpension | 60 | 1200 |
| 10 | Summer Resort Kredit AED 100 pro Zimmer/Nacht | 0 | 0 |
| 10 | Kind: Zuschlag Halbpension | 30 | 300 |
| 10 | Preis pro Kind/Nacht | 24 | 240 |
| 4 | 1 Gratisnacht pro Pers.(oblig. Halbpension berücksichtigt) | -255 | -1020 |

## Allgemeine Infos

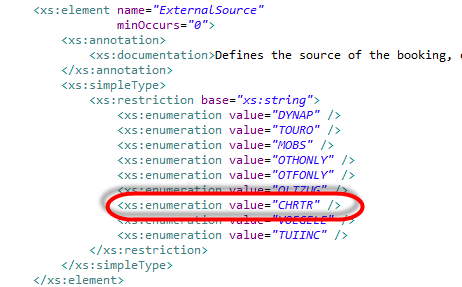
### Erfassen der Zuschläge für Kinderpreise



Für die Zuschläge müssen die Flags wie oben gesetzt werden.

### Anpassungen Webservice

Wie bei Charterflügen brauchen wir bei unseren Hotels eine eigene ExternalSource.



## Anhang

### SpecialOffer

<SpecialOffer

Baby="0"

Days="19"

Type="MinValue"

Child="0"

DateTo="2014-01-05"

DescId="5"

DateFrom="2014-01-01"

RuleType="Date"

AddAmount="0.00"

DayString="7,6,6"

PayNights="5"

ToDayBase="7"

FromDayBase="6"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2014-01-05"

StartDateRelevant="0"

Es gibt entweder DateBefore oder DaysBefore, nie beides zusammen. Folgendes soll nur beschreiben welche Werte die Felder enthalten könnten.

DateBeforeDepartureTo="2013-08-31"

DateBeforeDepartureFrom="2012-11-01"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45" />

|  |  |  |
| --- | --- | --- |
| Attribut | Beschreibung | Bemerkung |
| Baby | 1=gilt für Babys,0=alle anderen |  |
| Days | Anzahl gültige Tage | für wie viele Buchungstage gilt das Special |
| Type | MinValue, MaxValue,Average, StartDays,EndDays | Bestimmt den Basistag zur Preisberechnung |
| Child | 1=gilt für Children,0=alle anderen |  |
| DateTo | Saison End Datum | Kleiner oder gleich diesem Datum |
| DescId | Id für den Text des Specials |  |
| DateFrom | Saison Start Datum | Gleich oder grösser diesem Datum |
| RuleType | Always, Nr, Date |  |
| AddAmount | Additional Amount | Im allgemeinen Negativ und wird zum Betrag hinzuaddiert (Frühstück) |
| DayString | Gibt bei Revolving die gültigen Tage an | 7,6 = gibt bei einem 6->7 für 5, 3 Tage Abzug. Die effektiven Tage sind dann 6,7,13 |
| PayNights | Zu zahlende Tage |  |
| ToDayBase | Ab wievielt Tage (Maximum) gilt das Special | Hier steht immer die Basis |
| FromDayBase | Ab wievielt Tage (Minimum) gilt das Special | Hier steht immer die Basis |
| ChildAdultNr | Preis wenn so viele Adults im Zimmer |  |
| ChildChildNr | Kind Nummer (n) | Gilt für das n-te Kind |
| RevolvingGroup | Revolving Gruppen Bezeichnung | Wenn ein Special „Revolving“ ist, dann kann die gleiche Regel mehrfach genutzt werden. Der TO Online Preis Export exportiert alle möglichen Kombinationen eines Revolving als eigenes SpecialOffer. Damit einfach identifiziert werden kann welche SpecialOffer zueinander gehören, gibt es den Node „RevolvingGroup“. Ist eine Regel eines „RevolvingGroup“ einmal genutzt worden, darf keine weitere der Regelen der „RevolvingGroup“ genutzt werden. |
| EndDateRelevant | Das End-Datum der Leistung ist für das Special relevant | Das effektive End-Datum der Leistung muss >= Saison-Begin (DateFrom) und <= Saison-Ende (DateTo) |
| LastSpOffEndDate | Letztes mögliche Datum für das Special Offer | Mögliche Verkettungen von Special Offer haben das gleiche Datum |
| StartDateRelevant | Das Start-Datum der Leistung ist für das Special relevant | Das effektive Start-Datum der Leistung muss >= Saison-Begin (DateFrom) und <= Saison-Ende (DateTo) |
| DateBeforeDepatureTo | Gilt bis vor diesem Abreisedatum | Es gibt entweder DateBefore oder DaysBefore, nie beides |
| DateBeforeDepatureFrom | Gilt bis nach diesem Abreisedatum |
| DaysBeforeDepatureTo | Gilt bis Anzahl Tage vor Abreise | Es gibt entweder DateBefore oder DaysBefore, nie beides |
| DaysBeforeDepatureFrom | Gilt ab Anzahl Tage vor Abreise |

*Was hat nun der DayString="5,5" bei der zweiten SpecialOffer zu bedeuten?*

In der Doku haben wir dazu folgendes geschrieben:



Ein Revolving Special ist eine Special-Regel der wiederkehrend ist Wiederkehrende Specials können DayStrings haben welche mehrere Element haben.

7 für 5 Revolving 4 bedeutet ich kann folgende Regeln haben:

|  |  |  |  |
| --- | --- | --- | --- |
| Übernachtungen | DayString | Anzahl Tag Abzug | Effektive Tage |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 | 5 | 1 | 5 |
| 6 | 6 | 2 | 5,6 |
| 7 | 7 | 3 | 5,6,7 |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 | 7,5 | 4 | 5,6,7,12 |
| 13 | 7,6 | 5 | 5,6,7,12,13 |
| 14 | 7,7 | 6 | 5,6,7,12,13,14 |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 | 7,7,5 | 7 | 5,6,7,12,13,14,19 |
| 20 | 7,7,6 | 8 | 5,6,7,12,13,14,19,20 |
| 21 | 7,7,7 | 9 | 5,6,7,12,13,14,19,20,21 |
| 22 |  |  |  |
| 23 |  |  |  |
| 24 |  |  |  |
| 25 |  |  |  |
| 26 | 7,7,7,5 | 10 | 5,6,7,12,13,14,19,20,21,26 |
| 27 | 7,7,7,6 | 11 | 5,6,7,12,13,14,19,20,21,26,27 |
| 28 | 7,7,7,7 | 12 | 5,6,7,12,13,14,19,20,21,26,27,28 |

Wichtig bei Revolving-Specials ist, dass diese nur einmal genutzt werden dürfen. Wenn ich im obigen Beispiel eine 35 Tage Buchung habe, und den Special 28 für 16 anwende, kann ich danach nicht den 7 für 4 nutzen. Hätte der TO einen Special 35 für 20 erlaubt, wäre dies so auch in den Stammdaten als 7 für 4 Revolving 5 eingegeben worden. Dies ist aber oben nicht der Fall. Daher kann ich den gleichen Revolving nicht mehrfach nutzen.

Schauen wir uns mal ein Beispiel an:

<SpecialOffer

Baby="0"

Days="19"

Type="MinValue"

Child="0"

DateTo="2014-01-05"

DescId="5"

DateFrom="2014-01-01"

RuleType="Date"

AddAmount="0.00"

DayString="7,6,6"

PayNights="5"

ToDayBase="7"

FromDayBase="6"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2014-01-05"

StartDateRelevant="0"

DaysBeforeDepartureTo="999"

DaysBeforeDepartureFrom="45" />

Es kann auch anstatt DaysBeforeDepartureTo und DaysBeforeDepartureFrom folgendes geliefert werden:

DateBeforeDepartureTo="2013-08-31"

DateBeforeDepartureFrom="2012-11-01"

Zuerst geht es darum anhand von Days den grössten SpecialOffer zu finden der in dem Buchungszeitraum passt. Dazu können folgende Flags genutzt werden:

EndDateRelevant: Wenn angekreuzt, und das Enddatum der Leistung liegt innerhalb des DateFrom und DateTo , also innerhalb der Buchungssaison, dann kann der Special angewendet werden.

StartDateRelevant: Wenn angekreuzt, und das Startdatum der Leistung liegt innerhalb des DateFrom und DateTo , also innerhalb der Buchungssaison, dann kann der Special angewendet werden.

LastSpOffEndDate: wenn das letzte Datum der Leistung nach diesem Datum liegt, dann darf dieses Special **nicht** angewendet werden. Durch die Lieferung dieser Daten sparen wir euch die Programmierung wie bei einem Saisonwechsel der Specials verfahren werden soll.

Wenn hier

Ein paar Tips zur Vorbereitung auf die Verarbeitung der Specials:

1. Baue eine Liste der Specials auf, und sortiere diese Lists absteigend nach Days desc, StartDateRelevant asc, EndDateRelevant asc , LastSpOffEndDate desc.
2. Am einfachsten ist es diese Berechnung pro Pax zu machen. So können die Kinder auch, falls diese Tagespreise haben, berücksichtigt werden.

Sortiere die Pax-Liste nach den Erwachsenen, dann Kinder und Babys mit Geburtsdatum aufsteigend. Nutze den Wert für die Zimmerbelegung <NormalOccupancy>1</NormalOccupancy> um die korrekte Regel für das Kind zu ermitteln. Bei einem Doppelzimmer kann es vorkommen, dass ich 2 Erwachsene im Zimmer habe (Normalbelegung) und zwei Kinder. In diesem Fall wird es eine Regel geben mit ChildAdultNr="2" / ChildChildNr="1" und ChildAdultNr="2" / ChildChildNr="2" . Es kann aber auch Verträge geben wo nur das erste Kind ein Special bekommt, das zweite nicht, daher diese Anmerkung.

Wenn dieser ermittelt ist, geht es darum zu bestimmen welche Tage nun für den Kunden kostenlos sind. Die Grundregel in diesem Fall ist bleibe 7 Nächte, zahle 5. Aber es wird für dich schwierig sein zu bestimmen welche Tage jetzt kostenlos sind. Dafür haben wir den DayString geliefert.

Die 19 Tage können jetzt in folgende Gruppen aufgeteilt werden:

|  |  |
| --- | --- |
| 1,2,3,4,5,6,7 | 7 Nächte |
| 8,9,10,11,12,13 | 6 Nächte |
| 14,15,16,17,18,19 | 6 Nächte |

Jetzt kann auf Basis des Wertes von Type entscheiden werden welche Tage für die jeweilige Tages-Gruppe als Abzug gerechnet werden kann.

Gültige Werte von Type sind:



* Wenn MinValue, die Tagespreis-Liste nach Amount aufsteigend sortieren, und dann so viele Einträge nehmen bis in der Liste nur noch PayNights übrig sind.
* Wenn MaxValue, die Tagespreis-Liste nach Amount absteigend sortieren, und dann so viele Einträge nehmen bis in der Liste nur noch PayNights übrig sind.
* Wenn Average, dann die Summe der Tagespreise nehmen, den Average berechen, und dann ist der Abzug TotalSumme\*Average – PayNights\*Average
* Wenn StartDays, dann wären im Fall 1,2,3,4,5,6,7 oben die Tage 1,2 der SpecialOffer Abzug
* Wenn EndDays, dann wären im Fall 1,2,3,4,5,6,7 oben die Tage 6,7 der SpecialOffer Abzug

Diese Logik kann dann für jede Gruppe durchgeführt werden um für jede Gruppe die Tage zu ermitteln deren Preis dem Kunden erstattet werden.

Es kann beim Special auch ein AddAmount mitgeliefert werden. Dieser wird in Fällen genutzt wo nicht der gesamte Tages-Betrag zum Abzug kommen soll. Ein solcher Fall ist, wenn der Tagespreis zum Abzug kommt, aber exklusive einen Betrag für Frühstück.



Wenn der TagesAbzug 200 CHF ist, und AddAmount=-15, dann ist der Tages-Abzug=185 CHF.

Hier ein Beispiel mit folgenden Parametern:

Anreise: 01JUL2016

Abreise: 05AUG2016

Duration: 35 Übernachtungen

<SpecialOffers>

<SpecialOffer Baby="0"

Days="5"

Type="MinValue"

Child="0"

DateTo="2016-10-31"

DescId="5"

DateFrom="2015-11-01"

RuleType="Always"

AddAmount="0.00"

DayString="5"

PayNights="4"

ToDayBase="5"

FromDayBase="5"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2016-10-31"

StartDateRelevant="0"

DaysBeforeDepartureTo="0"

DaysBeforeDepartureFrom="0" />

<SpecialOffer Baby="0"

Days="8"

Type="MinValue"

Child="0"

DateTo="2016-10-31"

DescId="5"

DateFrom="2015-11-01"

RuleType="Always"

AddAmount="0.00"

DayString="8"

PayNights="6"

ToDayBase="8"

FromDayBase="8"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2016-10-31"

StartDateRelevant="0"

DaysBeforeDepartureTo="0"

DaysBeforeDepartureFrom="0" />

<SpecialOffer Baby="0"

Days="10"

Type="MinValue"

Child="0"

DateTo="2016-10-31"

DescId="5"

DateFrom="2015-11-01"

RuleType="Always"

AddAmount="0.00"

DayString="10"

PayNights="8"

ToDayBase="10"

FromDayBase="10"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2016-10-31"

StartDateRelevant="0"

DaysBeforeDepartureTo="0"

DaysBeforeDepartureFrom="0" />

<SpecialOffer Baby="0"

Days="16"

Type="MinValue"

Child="0"

DateTo="2016-10-31"

DescId="5"

DateFrom="2015-11-01"

RuleType="Always"

AddAmount="0.00"

DayString="16"

PayNights="12"

ToDayBase="16"

FromDayBase="16"

ChildAdultNr="0"

ChildChildNr="0"

RevolvingGroup="1"

EndDateRelevant="0"

LastSpOffEndDate="2016-10-31"

StartDateRelevant="0"

DaysBeforeDepartureTo="0"

DaysBeforeDepartureFrom="0" />

</SpecialOffers>

In diesem Fall ist die Logik wie folgt, bei einer Buchung 1JUL2016 bis 5AUG2016 (Duration: 35 Übernachtungen):

• Da dies ein Hotel ist und wir Übernachtungen betrachten, ist der erste Tag 1JUL2016, der letzte der 4 AUG2016.

• Die Special-Liste absteigend nach Days sortieren, ToDayBase absteigend, DateFrom aufsteigend, DateTo absteigend.

• Wir fangen mit Tag 1 an, also mit dem 1JUL2016.

• Die erste Regel welche ich finde ist die für Days=16.

• 1JUL2016 + 16 Days <= LastSpOffEndDate

• Der 1JUL2016 und 16JUL2016 (der Preis dieses Tages gilt) passt zwischen dem DateFrom und DateTo, auch in Bezug zu den Flags StartDateRelevant und EndDateRelevant.

• Da diese Regel passt kann diese angewendet werden.

• Als nächste folgen Tests ob der Special auch genommen werden darf:

o Datum Tests:

If StartDateRelevant=1 und EndDateRelevant=0

If DateFrom<=1JUL2016

Special darf genommen werden

Else

//Dieses Special kann ignoriert werden weil das StartDatum meiner Leistung grösser ist als das EndDatum des Specials, liegt also ausserhalb der Saison des Specials//

Special darf nicht genommen werden

End If

Else If EndDateRelevant=1 und StartDateRelevant=0

If DateTo>=16JUL2016

Special darf genommen werden

Else

//Dieses Special kann ignoriert werden weil das EndDatum meiner Leistung kleiner ist als das StartDatum des Specials, liegt also ausserhalb der Saison des Specials//

Special darf nicht genommen werden

End If

Else If StartDateRelevant=1 und EndDateRelevant=1

If DateFrom<=1JUL2016 und DateTo>=1JUL2016 und DateFrom<=16JUL2016 und DateTo>=16JUL2016

// Nur If AnfangsDatum und EndDatum der Leistung innerhalb der SpecialSaison sind, nur dann gilt dieser Special//

Special darf genommen werden

Else

// Dieses Special kann ignoriert werden weil AnfangsDatum oder EndDatum der Leistung ausserhalb der SpecialSaison liegt//

Special darf nicht genommen werden

End If

Else

If 1JUL2016<DateFrom

// Dieses Special kann ignoriert werden weil AnfangsDatum der Leistung ausserhalb der SpecialSaison liegt//

Special darf nicht genommen werden

End If

If 16JUL2016>DateTo

If 16JUL2016> LastSpOffEndDate

//Dieses Special kann ignoriert werden weil EndDatum der Leistung ausserhalb der SpecialSaison liegt//

//Dieses Special kann ignoriert werden weil das EndDatum der Leistung auch ausserhalb der SpecialOfferEndDate (bei verketteten Specials) für diesen Special liegt//

Special darf nicht genommen werden

End If

End If

End If

o Wir haben jetzt einen Spezial gefunden Wir testen ob wir jetzt aufhören dürfen mit Suchen für diesen Fall

If 16JUL2016<=DateTo

//Gibt es eine Regel welche in meine aktuelle Saison reinpasst, dann sollte ich diese verwenden.//

If (wir schon einen gefunden haben)

If 16JUL2016=DateTo

//Es gibt einen Sonderfall, wo es ein reveolving Special gibt, z.B. 7/5 und 14/10.//

//Wenn aufgrund von dem P\_LAST\_SP\_OFF\_ENDDATE ein Revolving in diese Periode reinpassen würde, dann darf//

//ich nicht einfach die grössere Regel ignorieren. Nur wenn es keine grösser passende Regel gibt,//

//und die aktuelle Regel würde nicht PERFEKT passen, nur dann kann ich eine schlechter passende Regel nutzen.//

//Wenn das berechnete EndDatum mit dem Ende der Saison übereinstimmt, dann muss ich diese den grösseren//

//saisonüberschneidenden Regeln vorziehen.//

Wir nehmen den SpecialOffer und hören auf zu suchen

End If

Else

Wir nehmen den SpecialOffer und hören auf zu suchen

End If

Else If 16JUL2016>DateTo&StartDateRelevant=1&EndDateRelevant=0

//Wenn UseStartDateOnly gesetzt ist dann ist es egal was das EndDate der Saison ist, nutze den ersten besten passenden Special.//

Wir nehmen den SpecialOffer und hören auf zu suchen

Else

If DateTo<=LastSpOffEndDate

//Bei direkt aufeinanderfolgenden Saisons ist es erlaubt dass ein Special in die nächste Saison reinreicht, selbst wenn es eigentlich nicht direkt passt.//

//Aber dann nur die Regel die am wenigsten in die neue Saison reinreicht.//

//Dafür gibt es das SpecialOfferEndDate welches das letzte Datum der direkt nachfolgenden Saisons eines Specials berechnet.//

//Es wird immer weiter gesucht bis entweder ein besser passender Special in der vorherigen Saison gefunden wird, oder es//

//gibt keine Specials in der alten Saison die passen könnten. Wegen dem Loop und der Sortiertung nach DAYS absteigend//

//bleibt immer die kleinste Alternative übrig.//

Wir merken uns den SpecialOffer und suchen weiter

End If

End If

o Suche anhand vom Type den richtigen Preis in meinem Special Range (1JUL2016 bis 16JUL2016)

MinValue = kleinster Preis

StartDay = 1JUL2016

EndDay = 16JUL2016

MaxValue = höchster Preis

Avarage = Durchschnitts Preis der Zeit vom 1JUL2016 bis 16JUL2016

• Diese Regel, und alle verbundenen Regeln, dürfen nicht mehr angewendet werden. Wenn der Kunde es wollte, dass diese Regel weiterhin genutzt werden könnte, hätte dieser einen höheren Wert für Revolving definiert, bzw. das Hotel hätte den Vertrag so definiert.

• Wir fangen jetzt als nächstes mit dem 17JUL2016 an

• Die erste Regel welche ich finde ist die für Days=10. Da der Days=16 bereits genommen wurde und dieser Revolving = 1 ist. Wenn der Vertrag es erlauben würde dass ein Buchender den Special in einer Buchung mehrfach nutzen kann, dann wäre dieser als Revolving 2 oder höher definiert worden, und dann hätte es eine Regel in den SpecialOffer gegeben mit Days=32, aber dies ist im Beispiel nicht der Fall, daher kann ich die Regel nur einmal nutzen.

• 17JUL2016 + 10 Days <= LastSpOffEndDate

• Der 17JUL2016 und 27JUL2016 passt zwischen dem DateFrom und DateTo, auch in Bezug zu den Flags StartDateRelevant und EndDateRelevant.

• Da diese Regel passt kann diese angewendet werden.

• Hier kommen wieder die Tests

• Wir fangen jetzt als nächstes mit dem 27JUL2016 an

• Die erste Regel welche ich finde ist die für Days=8.

• 27JUL2016 + 8 Days <= LastSpOffEndDate

• Der 27JUL2016 und 3AUG2016 passt zwischen dem DateFrom und DateTo, auch in Bezug zu den Flags StartDateRelevant und EndDateRelevant.

• Da diese Regel passt kann diese angewendet werden.

• Hier kommen wieder die Tests

• Es ist nur noch ein Tag übrig, und es gibt kein Special welcher grösser/gleich diese Anzahl Tage ist daher können wir abbrechen.