

ПОИСК ТУРОВ

API 1.13

ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Редакция 2.0 от «08» сентября 2014 г.





ОГЛАВЛЕНИЕ

1. ОПИСАНИЕ СЕРВИСА	3
2. ПОДКЛЮЧЕНИЕ К СЕРВИСУ	4
2.1. Старый клиент на .NET 1.0 - 2.0	5
2.2. Клиент на .NET 3.5 или выше	6
3. ВЕРСИОННОСТЬ.....	7
4. ЛОГИКА РАБОТЫ С СЕРВИСОМ.....	8
4.1. Получение полного набора туров	9
4.2. Получение данных по мере их поступления	10
4.3. Актуализация и сохранение данных	11
5. МЕТОДЫ	12
5.1. GetDepartCities	13
5.2. GetCountries	15
5.3. GetCities	17
5.4. GetHotelStars	19
5.5. GetMeals	21
5.6. GetHotels.....	23
5.7. GetTourOperators.....	26
5.8. GetTourDates.....	28
5.9. Методы загрузки туров	30
5.9.1. CreateRequest	31
5.9.2. GetRequestState	35
5.9.3. GetRequestResult	37
5.9.4. ContinueSearch.....	46
5.10. ActualizePrice	48
5.11. ActualizePriceByCode.....	53
5.12. SaveTourOrder.....	54

1. ОПИСАНИЕ СЕРВИСА

API поиска туров позволит вам создать собственную туристическую поисковую систему. Используя методы этого сервиса, вы сможете:

- ✓ Загружать справочники городов вылета, стран, курортов, отелей, категорий отелей, видов питания, туроператоров;
- ✓ Искать продукты более чем 130 крупнейших туроператоров России.
- ✓ Настроить автоматическое SMS- и email-оповещение менеджеров о заказах.

Внимание! На момент написания настоящего руководства функция онлайн бронирования сервисом не предусмотрена.

Внимание! Доступ к данным отдельных операторов может быть ограничен лицензией. По всем вопросам, пожалуйста, обращайтесь в Службу технической поддержки Слетать.ру по тел. **8(800)700-33-09** или по электронной почте support@sletat.ru.

Это стандартный веб-сервис, предоставляющий WSDL-документ с описанием всех методов и типов данных. Версия, выпущенная на момент написания настоящего руководства, работает только по протоколу HTTP. Для большей совместимости с различными клиентами сервис основан на SOAP версии 1.1.

Взаимодействие с сервисом возможно одним из двух способов:

- ✓ Как с обычным веб-сервисом посредством SOAP-запросов, WSDL доступен.
- ✓ Как с сервисом WCF (Windows Communication Foundation) – самый простой способ, если клиент под .NET Framework 3.5 или выше.

Логика обработки запросов не зависит от способа работы с сервисом.

При передаче строк и XML-пакетов используется кодировка UTF-8.

Работа сервиса не требует сессий, и методы не зависят друг от друга с точки зрения очередности вызова. Таким образом, очередность вызова методов диктуется реализуемым алгоритмом работы с поисковым сервисом.

Внимание! Сервис требует авторизации. Для работы используйте логин и пароль от личного кабинета на сайте sletat.ru.

Внимание! Работа с поисковым запросом (создание, запрос состояния и загрузка туров) должна выполняться с одного IP-адреса. При выполнении этого условия взаимодействие с сервисом возможно с нескольких IP адресов.



2. ПОДКЛЮЧЕНИЕ К СЕРВИСУ

Чтобы начать работу с сервисом, зарегистрируйтесь на сайте sletat.ru и получите тестовую лицензию в Службе технической поддержки Слетать.ру по электронной почте support@sletat.ru. После получения лицензии можно начинать работать с API.

Для работы с сервисом используйте WSDL-документ, расположенный по адресу:

```
http://module.sletat.ru/XmlGate.svc?wsdl
```

Современные среды разработки позволяют автоматически создавать все необходимые классы, используя WSDL-документ. Это избавляет разработчика от необходимости работать с SOAP-конвертами вручную.

Ниже подробно описаны алгоритмы работы с сервисом для клиентов, использующих .NET Framework.

2.1. Старый клиент на .NET 1.0 - 2.0

Добавьте в проект ссылку на WSDL документ, используя Web Reference. После добавления ссылки ваш проект готов к работе с сервисом. Ниже приведен пример вызова метода для получения списка городов вылета:

```
using System; using WebClient.Gate;

namespace WebClient
{
    class Program
    {
        static void Main()
        {
            // создаём экземпляр обёртки для работы с поисковым сервисом
            using (var gate = new Soap11Gate())
            {
                // указываем данные для авторизации на сервисе
                gate.AuthInfo = new AuthData() { Login = "*****", Password =
"*****" };

                // далее нам доступен вызов любых методов сервиса
                var cities = gate.GetDepartCities();
            }
        }
    }
}
```

Не забудьте указать логин и пароль вашей учетной записи на сайте sletat.ru. Данные авторизации передаются первым параметром для всех методов сервиса.

2.2. Клиент на .NET 3.5 или выше

Добавьте в проект ссылку на WSDL-документ, используя Service Reference. После добавления ссылки ваш проект готов к работе с сервисом, ниже приведен пример вызова метода для получения списка городов вылета:

```
using System; using WcfClient.Gate;

namespace WcfClient
{
    class Program
    {
        static void Main()
        {
            // создаём экземпляр обёртки для работы с поисковым сервисом
            using (var gate = new Soap11GateClient())
            {
                // указываем данные для авторизации на сервисе
                var authInfo = new AuthData() { Login = "*****", Password =
"*****" };

                // далее нам доступен вызов любых методов сервиса
                var cities = gate.GetDepartCities(authInfo);
            }
        }
    }
}
```

Не забудьте указать логин и пароль вашей учетной записи на сайте sletat.ru. Данные авторизации передаются первым параметром для всех методов сервиса.

3. ВЕРСИОННОСТЬ

Сервис может содержать несколько контрактов. Каждое изменение, несовместимое с предыдущей, публикацией будет вести к созданию контракта новой версии. Описание контрактов всех поддерживаемых версий включено в WSDL.

Каждый контракт веб-сервиса имеет уникальное имя, используемое для его идентификации. В имени любым удобным образом указывается его версия (например, в виде даты публикации). Для обозначения имени контракта используется URN (Uniform Resource Name).

Версия, выпущенная на момент написания настоящего руководства, состоит из одного контракта, использующего следующее имя:

```
urn:SletatRu:Soap11Gate:v1
```

Все разрабатываемые сервисы имеют префикс **SletatRu**, после которого указывается название контракта и его версия. Следующие версии контракта Soap11Gate будут иметь другое значение версии в последней части URN. Например:

```
urn:SletatRu:Soap11Gate:v2
```

Чтобы использовать нужную версию контракта, укажите её номер в SOAP конверте:

```
<soap:Envelope>
  <soap:Body>
    <GetDepartCities xmlns="urn:SletatRu:Soap11Gate:v1">
      ...
    </GetProductKeys>
  </soap:Body>
</soap:Envelope>
```

Если вы используете автоматически создаваемую обёртку, то при изменении текущего контракта или создании контракта новой версии заново создайте обёртку к поисковому сервису.

Новые версии контрактов будут создаваться только при наличии обратно несовместимых изменений. Все обратно совместимые изменения (добавление новых методов, добавление новых параметров в методы, добавление новых полей в структуры данных) будут добавляться в текущую версию контракта.

Версионность также соблюдается для всех типов данных.

Создание новых версий может быть сопряжено с иными задачами в ходе развития сервиса, например, группированием методов в контракты новых версий таким образом, чтобы в дальнейшем можно было отказаться от устаревших методов.

4. ЛОГИКА РАБОТЫ С СЕРВИСОМ

Метод поиска туров **CreateRequest** использует идентификаторы городов вылета, стран, отелей и т.д., поэтому сначала нужно загрузить все соответствующие справочники, используя методы:

- ✓ **GetDepartCities** – список городов вылета;
- ✓ **GetCountries** – список доступных стран;
- ✓ **GetCities** – список курортов;
- ✓ **GetHotelStars** – список категорий отелей;
- ✓ **GetMeals** – список типов питания;
- ✓ **GetHotels** – список отелей;
- ✓ **GetTourOperators** – список туроператоров.
- ✓ **GetTourDates** – список дат вылета для выбранного города.

После того как все справочники загружены и отображены пользователю, можно приступать к поиску туров. Для этого используются следующие методы:

- ✓ **CreateRequest** – создание поискового запроса;
- ✓ **GetRequestState** – получение данных о статусе загрузки туров для каждого туроператора;
- ✓ **GetRequestResult** – возвращает результаты поиска;
- ✓ **ContinueSearch** – продолжает поиск с использованием данных указанного запроса среди предложений с более высокой ценой.

Сервис производит **асинхронный** поиск туров. После получения ответа от метода **CreateRequest** (обычно не более секунды) нужно отслеживать состояние поискового запроса, поскольку туроператоры отдают данные за разное время – от 1 секунды до 2 минут.

В ответе **CreateRequest** приходит идентификатор созданного поискового запроса, который в дальнейшем используется для отслеживания состояния поискового запроса методом **GetRequestState** и получения результатов поиска методом **GetRequestResult**.

Логика работы с сервисом зависит от того, нужно ли вам получать данные до полного завершения поискового запроса. Возможны два варианта – единовременное получение полной поисковой выдачи и получение данных о турах по мере их поступления.

4.1. Получение полного набора туров

1. Создаётся поисковый запрос методом **CreateRequest**. Сохраняется идентификатор запроса, полученный в ответе.
2. Создаётся цикл для получения статуса поискового запроса. В цикле вызывается метод **GetRequestState** с использованием идентификатора запроса. В итерации цикла обязательно должна быть пауза (рекомендованное минимальное значение – 1,5 секунды). В результате вызова **GetRequestState** проверяется, все ли туроператоры обработали запрос (т.е. значение поля **IsProcessed** = “true”). На основе проверки принимается решение о продолжении цикла или (если все туроператоры обработали запрос) о переходе к методу получения туров.
3. Вызывается метод **GetRequestResult** с использованием идентификатора, полученного в ответе метода **CreateRequest**. Метод вернет все найденные туры в рамках поискового запроса.

Внимание! Сервер не хранит результаты поискового запроса продолжительное время. Данные должны быть получены и сохранены локально.

Внимание! В цикле получения статуса поискового запроса должен быть внутренний таймаут, чтобы ни при каких обстоятельствах не могли возникнуть бесконечные циклы.

4.2. Получение данных по мере их поступления

1. Создаётся поисковый запрос методом **CreateRequest**. Сохраняется полученный идентификатор.
2. Создаётся цикл для получения статуса поискового запроса. В цикле вызывается метод **GetRequestState** с использованием идентификатора запроса. В итерации цикла обязательно должна быть пауза (рекомендованное минимальное значение – 1,5 секунды). В результате вызова **GetRequestState** проверяется, есть ли найденные туры по поисковому запросу (т.е. как минимум один туроператор имеет значение поля **RowCount** более нуля). Если найденных туров нет и не все туроператоры обработали запрос, цикл повторяется; если хотя бы один туроператор сообщил о наличии туров, можно перейти к методу получения туров.
3. Вызывается метод **GetRequestResult** с использованием идентификатора запроса. Метод вернет все туры, найденные на момент вызова, а также состояние обработки поискового запроса (аналогично методу **GetRequestState**). Состояние обработки поискового запроса сохраняется, а найденные туры отображаются пользователю.
4. Создаётся новый цикл, который работает до тех пор, пока все туроператоры не обработают запрос. В цикле вызывается метод **GetRequestState** с использованием идентификатора запроса. В итерации цикла обязательно должна быть пауза (рекомендованное минимальное значение – 1,5 секунды). Результат вызова **GetRequestState** сравнивается с сохранённым состоянием обработки запроса. Если результат сравнения показывает наличие новых туров, то пользователь уведомляется о новых найденных турах. Если пользователь хочет обновить результат, то необходимо повторить действия, описанные в п. 3. Если для всех туроператоров значение поля **IsProcessed** = “true”, то цикл завершается, и пользователь получает уведомление об окончании поиска.

Внимание! Сервер не хранит результаты поискового запроса продолжительное время. Данные должны быть получены и сохранены локально.

Внимание! В цикле получения статуса поискового запроса должен быть внутренний таймаут, чтобы ни при каких обстоятельствах не могли возникнуть бесконечные циклы.

4.3. Актуализация и сохранение данных

После того, как пользователь выбрал понравившийся тур, необходимо актуализировать данные с помощью метода **ActualizePrice**. Метод возвращает не только наиболее актуальную информацию по туру, но и его подробное описание (на момент написания настоящего руководства функция реализована не для всех туроператоров).

Последний шаг – вызов метода **SaveTourOrder**, который сохраняет информацию о заказе в системе sletat.ru и позволяет использовать средства оповещения для информирования ваших менеджеров о новом заказе. На момент написания настоящего руководства в личном кабинете на сайте sletat.ru можно настроить оповещения по электронной почте и SMS. Также доступна утилита SletatRu Agent для Windows, которая уведомляет обо всех новых заказах.

5. МЕТОДЫ

В этом разделе описаны все методы поискового сервиса, приведены примеры запросов и ответов в SOAP.

Все используемые методы требуют авторизации. Данные авторизации (логин и пароль от личного кабинета на сайте sletat.ru) передаются в заголовке SOAP:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
>
  <soap:Header>
    <AuthInfo
      xmlns="urn:SletatRu:DataTypes:AuthData:v1"
    >
      <Login>
        *****
      </Login>
      <Password>
        *****
      </Password>
    </AuthInfo>
  </soap:Header>
  <soap:Body>
    <GetDepartCities
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    />
  </soap:Body>
</soap:Envelope>
```

В примерах SOAP- запросов и ответов, которые приводятся ниже для каждого метода, элемент **soap:Header** убран для лучшей читаемости документа.

Все идентификаторы в методах используют тип Int32 (если не указано иного), который может принимать значения от -2 147 483 648 до 2 147 483 647.

5.1. GetDepartCities

Метод возвращает список всех городов вылета, который вы можете отредактировать в личном кабинете на сайте sletat.ru.

Сигнатура:

```
CityCollection GetDepartCities()
```

Параметры отсутствуют.

Пример SOAP-запроса:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
>
  <soap:Body>
    <GetDepartCities
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    />
  </soap:Body>
</soap:Envelope>
```

Пример SOAP ответа:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
  <s:Body>
    <GetDepartCitiesResponse xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    >
      <GetDepartCitiesResult
        xmlns:a="urn:SletatRu:DataTypes:CityCollection:v1"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
      >
        <a:City>
          <Id
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"
          >
            1264
          </Id>
          <Name
            xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
          >
            Санкт-Петербург
          </Name>
        </a:City>
        ...
      </GetDepartCitiesResult>
```

```
</GetDepartCitiesResponse>  
</s:Body>  
</s:Envelope>
```

Коллекция городов вылета **CityCollection** содержит *n* элементов **City**.

Поля элемента **City**:

Поле	Тип	Описание
Id	Int32	Идентификатор города вылета.
Name	String	Город вылета.

5.2. GetCountries

Метод **GetCountries** возвращает список доступных направлений для любого данного города вылета. В личном кабинете на сайте sletat.ru вы можете поставить на этот список различные фильтры, чтобы в ответ возвращались данные только о тех странах, с которыми вы работаете. Список городов вылета получается методом **GetDepartCities**.

Сигнатура:

```
CountryCollection GetCountries(  
    Int32 townFromId)
```

Параметр **townFromId** – идентификатор города вылета.

Пример SOAP-запроса доступных направлений для Санкт-Петербурга:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetCountries  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <townFromId>1254</townFromId>  
    </GetCountries>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  <s:Body>  
    <GetCountriesResponse xmlns="urn:SletatRu:Contracts:Soap11Gate:v1">  
      <GetCountriesResult  
        xmlns:a="urn:SletatRu:DataTypes:CountryCollection:v1"  
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
        xmlns:b="urn:SletatRu:DataTypes:Country:v1"  
      >  
        <a:Country>  
          <Id  
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"  
          >  
            1  
          </Id>  
          <Name  
            xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"  
          >  
            Абхазия  
          </Name>
```

```
</a:Country>
...
</GetCountriesResult>
</GetCountriesResponse>
</s:Body>
</s:Envelope>
```

В ответ возвращается коллекция направлений **CountryCollection**, содержащая *n* элементов **Country**.

Поля элемента **Country**:

Поле	Тип	Описание
Id	Int32	Идентификатор страны.
Name	String	Страна.

5.3. GetCities

Метод **GetCities** возвращает список курортов для выбранного направления. Список направлений получается методом **GetCountries**.

Сигнатура:

```
CityCollection GetCities(  
    Int32 countryId)
```

Параметр **countryId** – идентификатор страны.

Пример SOAP-запроса списка курортов для Египта:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetCities  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <countryId>  
        40  
      </countryId>  
    </GetCities>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetCitiesResponse  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <GetCitiesResult  
        xmlns:a="urn:SletatRu:DataTypes:CityCollection:v1"  
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
      >  
        <a:City>  
          <Id  
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"  
          >  
            26  
          </Id>  
          <Name  
            xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
```

```
>
  Айн Сохна
</Name>
</a:City>
...
</GetCitiesResult>
</GetCitiesResponse>
</s:Body>
</s:Envelope>
```

В ответ возвращается коллекция курортов **CityCollection**, содержащая *n* элементов **City**.

Поля элемента **City**:

Поле	Тип	Описание
Id	Int32	Идентификатор курорта.
Name	String	Курорт.

5.4. GetHotelStars

Метод **GetHotelStars** возвращает список доступных категорий отелей в выбранных курортах.

Сигнатура:

```
HotelStarsCollection GetHotelStars(  
    Int32 countryId,  
    Int32[] towns)
```

Параметры:

Параметр	Тип	Описание
countryId	Int32	Идентификатор страны.
towns	Int32[]	Массив идентификаторов курортов. Если пуст, результаты возвращаются для всех курортов выбранной страны.

Пример SOAP-запроса списка категорий отелей для Хургады и Шарм-Эль-Шейха (идентификаторы взяты из GetCities):

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetHotelStars  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <countryId>40</countryId>  
      <towns>  
        <int  
          xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"  
        >  
          1592  
        </int>  
        <int  
          xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"  
        >  
          1642  
        </int>  
      </towns>  
    </GetHotelStars>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
  <s:Body>
    <GetHotelStarsResponse
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    >
      <GetHotelStarsResult
        xmlns:a="urn:SletatRu:DataTypes:HotelStarsCollection:v1"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:b="urn:SletatRu:DataTypes:HotelStars:v1"
      >
        <a:HotelStars>
          <Id
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"
          >
            401
          </Id>
          <Name
            xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
          >
            2*
          </Name>
        </a:HotelStars>
        ...
      </GetHotelStarsResult>
    </GetHotelStarsResponse>
  </s:Body>
</s:Envelope>
```

В ответ возвращается коллекция категорий отелей **HotelStarsCollection**, содержащая n элементов **HotelStars**.

Поля элемента **HotelStars**:

Поле	Тип	Описание
Id	Int32	Идентификатор категории отеля.
Name	String	Категория отеля.

5.5. GetMeals

Метод **GetMeals** возвращает список типов питания.

Сигнатура:

```
MealCollection GetMeals()
```

Параметры отсутствуют.

Пример SOAP-запроса:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
>
  <soap:Body>
    <GetMeals
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    />
  </soap:Body>
</soap:Envelope>
```

Пример SOAP-ответ:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
  <s:Body>
    <GetMealsResponse
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    >
      <GetMealsResult
        xmlns:a="urn:SletatRu:DataTypes:MealCollection:v1"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:b="urn:SletatRu:DataTypes:Meal:v1"
      >
        <a:Meal>
          <Id
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"
          >
            115
          </Id>
          <Name
            xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
          >
            AI
          </Name>
        </a:Meal>
        ...
      </GetMealsResult>
```

```
</GetMealsResponse>  
</s:Body>  
</s:Envelope>
```

В ответ возвращается коллекция типов питания **MealCollection**, содержащая *n* элементов **Meal**.

Поля элемента **Meal**:

Поле	Тип	Описание
Id	Int32	Идентификатор типа питания.
Name	String	Тип питания.

5.6. GetHotels

Метод **GetHotels** возвращает список доступных отелей в выбранной стране. Дополнительно, список отелей может быть отфильтрован по массиву курортов и категорий отелей, а также по названию самого отеля.

Сигнатура:

```
HotelCollection GetHotels(  
    Int32 countryId,  
    Int32[] towns,  
    Int32[] stars,  
    String filter,  
    Int32 count)
```

Параметры:

Параметр	Тип	Описание
countryId	Int32	Идентификатор страны.
towns	Int32[]	Массив идентификаторов курортов. Если пуст, возвращается результат по всем курортам выбранной страны.
stars	Int32[]	Массив идентификаторов категорий отелей. Если пуст, возвращается результат по всем категориям.
filter	String	Фильтр для поиска отелей по названию. Если пуст, параметр игнорируется.
count	Int32	Максимальное количество отелей в ответе (отели отбираются по рейтингу, потом сортируются по названию). При значении "-1" возвращаются данные по всем отелям.

Пример SOAP-запроса списка 4- и 5-звёздочных 500 отелей Хургады и Шарм-Эль-Шейха:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetHotels  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <countryId>40</countryId>  
      <towns>  
        <int  
          xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"  
        >  
          1592  
        </int>  
        <int
```

```

        xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
        1642
    </int>
</towns>
<stars>
    <int
        xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
        403
    </int>
    <int
        xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
        404
    </int>
</stars>
<filter
    xsi:nil="true"
/>
    <count>500</count>
</GetHotels>
</soap:Body>
</soap:Envelope>

```

Пример SOAP-ответа:

```

<s:Envelope
    xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
    <s:Body>
        <GetHotelsResponse
            xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
        >
            <GetHotelsResult
                xmlns:a="urn:SletatRu:DataTypes:HotelCollection:v1"
                xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
                xmlns:b="urn:SletatRu:DataTypes:Hotel:v1"
            >
                <a:Hotel>
                    <Id
                        xmlns="urn:SletatRu:DataTypes:DbObject:v1"
                    >
                        14646
                    </Id>
                    <Name
                        xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
                    >
                        Four Seasons Resort
                    </Name>
                    <b:Rate>4.94</b:Rate>
                    <b:StarName>5*</b:StarName>
                </a:Hotel>
                ...
            </GetHotelsResult>

```

```
</GetHotelsResponse>  
</s:Body>  
</s:Envelope>
```

В ответ возвращается коллекция отелей **HotelsCollection**, содержащая n элементов **Hotel**.

Поля элемента **Hotel**:

Поле	Тип	Описание
Id	Int32	Идентификатор отеля.
Name	String	Название отеля.
StarName	String	Категория отеля.
Rate	String	Рейтинг по пятибалльной шкале.

5.7. GetTourOperators

Метод **GetTourOperators** возвращает список доступных туроператоров. Последний может быть ограничен лицензией. Кроме того, вы можете настроить фильтрацию туроператоров в личном кабинете на сайте sletat.ru.

Метод всегда возвращает полный список туроператоров, отмечая тех, у кого есть туры, соответствующие параметрам города вылета и направления.

Сигнатура:

```
TourOperatorCollection GetTourOperators(  
    Int32 townFromId,  
    Int32 countryId)
```

Параметры:

Параметр	Тип	Описание
townFromId	Int32	Идентификатор города вылета.
countryId	Int32	Идентификатор страны.

Пример SOAP-запроса списка туроператоров с отметкой тех, у кого есть туры из Санкт-Петербурга в Египет:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetTourOperators  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <townFromId>1264</townFromId>  
      <countryId>40</countryId>  
    </GetTourOperators>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetTourOperatorsResponse  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <GetTourOperatorsResult
```

```

xmlns:a="urn:SletatRu:DataTypes:TourOperatorCollection:v1"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:b="urn:SletatRu:DataTypes:TourOperator:v1"
>
  <a:TourOperator>
    <Id
      xmlns="urn:SletatRu:DataTypes:DbObject:v1"
    >
      7
    </Id>
    <Name
      xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
    >
      Biblio Globus
    </Name>
    <b:Enabled>True</b:Enabled>
  </a:TourOperator>
  ...
</GetTourOperatorsResult>
</GetTourOperatorsResponse>
</s:Body>
</s:Envelope>

```

В ответ возвращается коллекция туроператоров **TourOperatorCollection**, содержащая n элементов **TourOperator**.

Поля элемента **TourOperator**:

Поле	Тип	Описание
Id	Int32	Идентификатор туроператора.
Name	String	Наименование туроператора.
Enabled	String	Отметка о наличии туров. Возможные значения: "true" – есть туры; "false" – нет туров.

5.8. GetTourDates

Метод **GetTourDates** возвращает список доступных дат вылета для выбранных города вылета, страны и курорта, используя внутреннюю статистику, собранную по ранее найденным турам.

Сигнатура:

```
XmlTourDates GetTourDates(  
    Int32 dptCityId,  
    Int32 countryId,  
    Int32[] resorts)
```

Параметры:

Параметр	Тип	Описание
dptCityId	Int32	Идентификатор города вылета.
countryId	Int32	Идентификатор страны.
resorts	Int32[]	Массив идентификаторов курортов. Если ни один курорт не выбран, значение поля равно нулю.

Пример SOAP-запроса доступных дат начала тура из Екатеринбурга в Тироль (Австрия):

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetTourDates  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <dptCityId>1265</dptCityId>  
      <countryId>3</countryId>  
      <resorts  
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"  
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
      >  
        <a:int>1442</a:int>  
      </resorts>  
    </GetTourDates>  
  </s:Body>  
</s:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetTourDatesResponse
```

```
xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
>
<GetTourDatesResult
  xmlns:a="urn:SletatRu:DataTypes:XmlTourDates:v1"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
>
  <a:Dates
    xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  >
    <b:string>09.03.2013</b:string>
    <b:string>09.02.2013</b:string>
    <b:string>23.02.2013</b:string>
    <b:string>02.03.2013</b:string>
    <b:string>16.03.2013</b:string>
    <b:string>02.02.2013</b:string>
    <b:string>16.02.2013</b:string>
    <b:string>06.04.2013</b:string>
  </a:Dates>
</GetTourDatesResult>
</GetTourDatesResponse>
</s:Body>
</s:Envelope>
```

XmlTourDates содержит элемент **Dates** – коллекцию дат начала тура.

5.9. Методы загрузки туров

Так как разные туроператоры возвращают результаты запроса в разные интервалы времени (до 90 секунд), в сервисе реализован **асинхронный** поиск туров. Для загрузки туров используются три метода:

- ✓ **CreateRequest** используется для создания поискового запроса. Метод принимает все необходимые поисковые параметры.
- ✓ **GetRequestState** отслеживает прогресс обработки поискового запроса туроператорами.
- ✓ **GetRequestResult** возвращает все данные, найденные по запросу на момент вызова.
- ✓ **ContinueSearch** позволяет продолжить поиск по параметрам исходного поискового запроса среди предложений с более высокой ценой.

В этом разделе приведено подробное описание всех трёх методов.

Внимание! Варианты взаимодействия методов CreateRequest, GetRequestResult и ContinueSearch описана в разделе «[Логика работы с сервисом](#)».

Внимание! Работа с поисковым запросом (создание, запрос состояния и загрузка туров) должна выполняться с одного IP-адреса. При выполнении указанного условия взаимодействие с сервисом возможно с нескольких IP-адресов.

5.9.1. CreateRequest

Метод **CreateRequest** используется для создания поискового запроса. Ответ состоит из единственного элемента CreateRequestResult, содержащего идентификатор созданного запроса. Присвоенный запросу идентификатор используется методами GetRequestState, GetRequestResult и Continue Search в продолжение всего процесса поиска.

Внимание! Варианты взаимодействия методов CreateRequest, GetRequestResult и Continue Search описана в разделе «[Логика работы с сервисом](#)».

Сигнатура:

```
Int32 CreateRequest(  
    Int32 countryId,  
    Int32 cityFromId,  
    Int32[] cities,  
    Int32[] meals,  
    Int32[] stars,  
    Int32[] hotels,  
    Int32 adults,  
    Int32 kids,  
    Int32[] kidsAges,  
    Int32 nightsMin,  
    Int32 nightsMax,  
    Int32? priceMin,  
    Int32? priceMax,  
    String currencyAlias,  
    DateTime departFrom,  
    DateTime departTo,  
    Boolean hotelIsNotInStop,  
    Boolean hasTickets,  
    Boolean ticketsIncluded,  
    Boolean useFilter,  
    Int32[] f_to_id,  
    Boolean includeDescriptions,  
    Int32? cacheMode)
```

Параметры (обязательные отмечены знаком *):

Параметр	Тип	Описание
countryId *	Int32	Идентификатор страны.
cityFromId *	Int32	Идентификатор города вылета.
cities	Int32[]	Список идентификаторов курортов, по которым должен быть произведён поиск.
meals	Int32[]	Список идентификаторов типов питания, по которым должен быть произведён поиск.
stars	Int32[]	Список идентификаторов категорий отелей, по которым должен быть произведён поиск.
hotels	Int32[]	Список идентификаторов отелей, по которым должен быть произведён поиск.
adults	Int32	Количество туристов (взрослых). Значение по

		умолчанию “2”.
kids	Int32	Количество туристов (детей). Значение по умолчанию “0”.
kidsAges	Int32	Возраст каждого ребёнка. Если параметр используется, количество элементов в массиве должно быть равным значению параметра kids .
nightsMin	Int32	Минимальная продолжительность тура (ночей). Значение по умолчанию “7”.
nightsMax	Int32	Максимальная продолжительность тура (ночей). Значение по умолчанию “7”.
priceMin	Int32?	Минимальная цена тура.
priceMax	Int32?	Максимальная цена тура.
currencyAlias	String	Валюта для параметров priceMin и priceMax . Поддерживаются значения USD, EUR и RUB. Указанная валюта будет использована в поисковой выдаче.
departFrom	DateTime	Начальная дата диапазона дат вылета в формате dd.MM.yyyy. Значение по умолчанию – текущая дата +1 день.
departTo	DateTime	Конечная дата диапазона дат вылета в формате dd.MM.yyyy. Значение по умолчанию – значение параметра dateFrom +7 дней.
hotelsNotInStop	Boolean	Фильтрация результатов поиска по наличию мест в отеле. Если значение равно “true”, в выдачу попадут только отели со статусом «под запрос» и «есть места»; если “false”, фильтр не применяется. Значение по умолчанию “false”.
hasTickets	Boolean	Фильтрация результатов поиска по наличию билетов (туда и обратно, без учёта класса). Если значение равно “true”, в результаты поиска попадут только туры, в которых ещё остались в наличии билеты; если “false”, фильтр не применяется. Значение по умолчанию – “false”.
ticketsIncluded	Boolean	Фильтрация результатов поиска по составу турпакета. Если значение параметра равно “true”, в выдачу попадут только туры с перелётами; если “false”, фильтр не применяется. Значение по умолчанию – “false”.
useFilter	Boolean	Фильтрация результатов поиска по туператорам. Если значение равно “true”, в выдачу не попадают продукты туператоров, перечисленных в параметре f_to_id ; если “false”, фильтр не применяется. Фильтр применяется только после создания поискового запроса. Значение по умолчанию – “false”.
f_to_id	Int32[]	Список идентификаторов туператоров для параметра useFilter .
includeDescriptions	Boolean	Если значение равно “true”, в выдаче будут подробно описаны отели, типы размещения и

		питания.
cacheMode	Int32?	Режим обращения к кешу системы поиска. Возможные значения: 0 – не использовать кеш; 1 – использовать кеш при ошибках поиска у туроператора; 2 – режим SmartCache (на основе статистики поисковых запросов и наполнения кеша будет принято решение об использовании кеша; если поиск по кешу не даёт достоверного результата, запрос направляется туроператору); 3 – запрос будет направлен только в кеш, без использования алгоритма SmartCache; 4 – запрос будет направлен только в кеш, алгоритм SmartCache используется для поиска наиболее актуального набора туров.

В приведённом ниже примере создаётся поисковый запрос со следующими параметрами:

- ✓ город вылета Санкт-Петербург;
- ✓ направление Египет;
- ✓ курорты Хургада и Шарм-Эль-Шейх;
- ✓ вид питания All Included;
- ✓ 2 взрослых туриста без детей;
- ✓ продолжительность тура 14 ночей;
- ✓ цены отобразить в рублях;
- ✓ дата начала вылета в диапазоне от 10 до 17 октября 2011 года;
- ✓ не выводить отели в стопе;
- ✓ выводить только туры с включенным перелётом;
- ✓ выводить только туры с наличием билетов туда и обратно.

Пример SOAP-запроса:

```
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
  <s:Body>
    <CreateRequest
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    >
      <countryId>40</countryId>
      <cityFromId>1264</cityFromId>
      <cities>
        <int
          xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        >
          1592
        </int>
        <int
          xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        >
          1642
        </int>
      </cities>
      <meals>
```

```

    <int
      xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
      115
    </int>
  </meals>
  <stars>
    <int
      xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
      403
    </int>
    <int
      xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    >
      404
    </int>
  </stars>
  <hotels
    xsi:nil="true"
  />
  <adults>2</adults>
  <kids>0</kids>
  <kidsAges xsi:nil="true" />
  <nightsMin>14</nightsMin>
  <nightsMax>14</nightsMax>
  <currencyAlias>RUB</currencyAlias>
  <departFrom>10.10.2011</departFrom>
  <departTo>17.10.2011</departTo>
  <hotelIsNotInStop>true</hotelIsNotInStop>
  <hasTickets>true</hasTickets>
  <ticketsIncluded>true</ticketsIncluded>
  <f_to_id xsi:nil="true" />
  <includeDescriptions>true</includeDescriptions>
</CreateRequest>
</s:Body>
</s:Envelope>

```

Пример SOAP-ответа:

```

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
>
  <s:Body>
    <CreateRequestResponse
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
    >
      <CreateRequestResult>4355852</CreateRequestResult>
    </CreateRequestResponse>
  </s:Body>
</s:Envelope>

```

Элемент **CreateRequestResult** содержит идентификатор поискового запроса (Int32).

5.9.2. GetRequestState

Метод **GetRequestState** возвращает статус обработки запроса для каждого туроператора.

Внимание! Варианты взаимодействия методов **CreateRequest**, **GetRequestResult** и **Continue Search** описаны в разделе «[Логика работы с сервисом](#)».

Сигнатура:

```
OperatorLoadStateCollection GetRequestState(  
    Int32 requestId)
```

Метод принимает единственный параметр **requestId** – идентификатор поискового запроса, выданный методом **CreateRequest** при создании.

Пример SOAP-запроса:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <GetRequestState  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <requestId>4355852</requestId>  
    </GetRequestState>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetRequestStateResponse  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <GetRequestStateResult  
        xmlns:a="urn:SamoSoft:DataTypes:OperatorLoadStateCollection:v1"  
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
        xmlns:b="urn:SamoSoft:DataTypes:OperatorLoadState:v1"  
      >  
        <a:OperatorLoadState>  
          <Id  
            xmlns="urn:SletatRu:DataTypes:DbObject:v1"  
          >  
            7  
          </Id>  
          <Name
```

```

        xmlns="urn:SletatRu:DataTypes:NamedDBObject:v1"
    >
        Biblio Globus
    </Name>
    <b:ErrorMessage i:nil="true"/>
    <b:ExecutionTimeMs>1561</b:ExecutionTimeMs>
    <b:IsCached>false</b:IsCached>
    <b:IsError>false</b:IsError>
    <b:IsProcessed>true</b:IsProcessed>
    <b:IsSkipped>false</b:IsSkipped>
    <b:IsTimeout>false</b:IsTimeout>
    <b:RowCount>150</b:RowCount>
    </a:OperatorLoadState>
    ...
    </GetRequestStateResult>
    </GetRequestStateResponse>
    </s:Body>
</s:Envelope>

```

На примере видно, что туроператор Biblio Globus вернул 150 туров за 1.5 секунды, состояния загрузки по другим операторам из примера были убраны.

В ответ возвращается коллекция состояний обработки запроса **OperatorLoadStateCollection**, содержащая *n* элементов **OperatorLoadState**, каждый из которых описывает статус для одного оператора.

Описание полей элемента **OperatorLoadState**:

Поле	Тип	Описание
Id	Int32	Идентификатор туроператора.
Name	String	Наименование туроператора.
IsProcessed	Boolean	Статус обработки запроса. Принимает значение "true", даже если запрос обработан с ошибкой.
IsCached	Boolean	Принимает значение "true", если ответ выдан из кеша.
IsError		Принимает значение "true", если. При обработке запроса произошла ошибка.
ErrorMessage		Описание ошибки при IsError = true .
IsTimeout		Принимает значение "true", если запрос завершен по таймауту. Поле IsError также принимает значение "true".
ExecutionTimeMs		Время обработки запроса в миллисекундах.
RowCount		Количество найденных туров.

Внимание! Поисковый запрос считается обработанным, когда для каждого туроператора поле **IsProcessed** принимает значение "true".

5.9.3. GetRequestResult

Метод **GetRequestResult** возвращает все туры, найденные в рамках обработки поискового запроса.

Внимание! Базовый алгоритм работы сервиса подразумевает асинхронную обработку поискового запроса. Если поисковый запрос не был обработан до конца (в результате метода **GetRequestState** не все туроператоры получили флаг **IsProcessed = true**), то метод **GetRequestResult** возвращает набор туров, доступный на момент его вызова. Итоговый набор туров постоянно обновляется по мере обработки запроса каждым туроператором. Если вы хотите, чтобы результаты поиска появлялись в выдаче одновременно, воспользуйтесь информацией, приведённой в разделе «[Логика работы с сервисом](#)».

Сигнатура:

```
TourData GetRequestResult(  
    Int32 requestId)
```

Метод принимает единственный параметр **requestId** – идентификатор запроса, возвращённый методом **CreateRequest** при создании запроса.

Пример SOAP-запроса:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetRequestResult  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <requestId>4355852</requestId>  
    </GetRequestResult>  
  </s:Body>  
</s:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <GetRequestResultResponse  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <GetRequestResultResult  
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
        xmlns:a="urn:SletatRu:DataTypes:XmlTourData:v1"  
      >  
        <a:HotelCount>374</a:HotelCount>  
        <a:LoadState  
          xmlns:c="urn:SamoSoft:DataTypes:OperatorLoadState:v1"
```

```

xmlns:b="urn:SamoSoft:DataTypes:OperatorLoadStateCollection:v1"
>
<b:OperatorLoadState>
  <Id
    xmlns="urn:SletatRu:DataTypes:DbObject:v1"
  >
    1819010031
  </Id>
  <Name
    xmlns="urn:SletatRu:DataTypes:NamedDbObject:v1"
  >
    Типоператор #1819010031
  </Name>
  <c:ErrorMessage i:nil="true"/>
  <c:ExecutionTimeMs>2812</c:ExecutionTimeMs>
  <c:IsCached>false</c:IsCached>
  <c:IsError>false</c:IsError>
  <c:IsPersistentCacheUsed>false</c:IsPersistentCacheUsed>
  <c:IsProcessed>true</c:IsProcessed>
  <c:IsSkipped>false</c:IsSkipped>
  <c:IsTimeout>false</c:IsTimeout>
  <c:MaxPrice>30952</c:MaxPrice>
  <c:MinPrice>19424</c:MinPrice>
  <c:RowCount>40</c:RowCount>
</b:OperatorLoadState>
</a:LoadState>
<a:OilTaxes xmlns:b="urn:SletatRu:DataTypes:XmlTourOilTax:v1">
  <b:XmlTourOilTax>
    <b:AirCompanyName i:nil="true"/>
    <b:AirportName i:nil="true"/>
    <b:CurrencyName i:nil="true"/>
    <b>Date1>2013-11-25T00:00:00</b>Date1>
    <b>Date2>2014-12-31T00:00:00</b>Date2>
    <b:FlightNumber i:nil="true"/>
    <b:PartnerName i:nil="true"/>
    <b:ResortId i:nil="true"/>
    <b:ResortName i:nil="true"/>
    <b:SourceId>1819010041</b:SourceId>
    <b:Tax>0</b:Tax>
  </b:XmlTourOilTax>
</a:OilTaxes>
<a:RequestId>539037314</a:RequestId>
<a:Rows xmlns:b="urn:SletatRu:DataTypes:XmlTourRecord:v1">
  <b:XmlTourRecord>
    <b:Adults>2</b:Adults>
    <b:BusinessTicketsDpt>Stop</b:BusinessTicketsDpt>
    <b:BusinessTicketsRtn>Stop</b:BusinessTicketsRtn>
    <b:CheckInDate>07.05.2014</b:CheckInDate>
    <b:CityFromId>832</b:CityFromId>
    <b:CityFromName>Москва</b:CityFromName>
    <b:CountryId>35</b:CountryId>
    <b:CountryName>Греция</b:CountryName>
    <b:CreditMonthlyPay i:nil="true"/>
    <b:CreditMonths i:nil="true"/>
    <b:Currency>RUB</b:Currency>
    <b>Description>
      1, Касторья, Хлои-Фундукли, Esperos Palace 4*+, Double Classical, __, 2Ad,
      Half Board 1, Халкидики-Афон, Урануполи, Dioni Apartments Apartments, Apartment 1
      Broom, __, 2Ad, Room Rate
    </b>Description>
    <b:DiscountCaption/>

```

```

<b:DiscountExpires i:nil="true"/>
<b:DiscountPercent i:nil="true"/>
<b:EarlyBookingValidTill i:nil="true"/>
<b:EconomTicketsDpt>Available</b:EconomTicketsDpt>
<b:EconomTicketsRtn>Available</b:EconomTicketsRtn>
<b:FewBusinessTicketsDpt>0</b:FewBusinessTicketsDpt>
<b:FewBusinessTicketsRtn>0</b:FewBusinessTicketsRtn>
<b:FewEconomTicketsDpt>-1</b:FewEconomTicketsDpt>
<b:FewEconomTicketsRtn>0</b:FewEconomTicketsRtn>
<b:FewPlacesInHotel>0</b:FewPlacesInHotel>
<b:Flags>3072</b:Flags>
<b:Found i:nil="true"/>
<b:GiftCaption/>
<b:HotelDescription/>
<b:HotelDescriptionUrl>
  http://hotels.sletat.ru/?id=13298
</b:HotelDescriptionUrl>
<b:HotelId>13298</b:HotelId>
<b:HotelIsInStop>Available</b:HotelIsInStop>
<b:HotelName>Esperos Palace</b:HotelName>
<b:HotelPhotosCount>45</b:HotelPhotosCount>
<b:HotelRating>9.62</b:HotelRating>
<b:HotelTitleImageUrl>
  http://hotels.sletat.ru/i/p/13298_0.jpg
</b:HotelTitleImageUrl>
<b:HtPlaceDescription/>
<b:HtPlaceId>13103</b:HtPlaceId>
<b:HtPlaceName>DBL</b:HtPlaceName>
<b:IsDemoMode>false</b:IsDemoMode>
<b:Kids>0</b:Kids>
<b:MealDescription/>
<b:MealId>113</b:MealId>
<b:MealName>HB</b:MealName>
<b:Nights>2</b:Nights>
<b:OfferId>985863881</b:OfferId>
<b:OfferIdStr>985863881</b:OfferIdStr>
<b:OriginalHotelName>Esperos Palace</b:OriginalHotelName>
<b:OriginalHtPlaceName>2Ad</b:OriginalHtPlaceName>
<b:OriginalMealName>Half Board</b:OriginalMealName>
<b:OriginalRoomName>Double Classical, _</b:OriginalRoomName>
<b:OriginalStarName>4*</b:OriginalStarName>
<b:OriginalTownName>Касторья, Хлои-Фундукли</b:OriginalTownName>
<b:Price>50</b:Price>
<b:PriceType>0</b:PriceType>
<b:ReceivingParty/>
<b:ResortDescriptionUrl/>
<b:ResortId>546</b:ResortId>
<b:ResortName>Кастория</b:ResortName>
<b:RoomId>0</b:RoomId>
<b:RoomName>Double Classical, _</b:RoomName>
<b:SourceId>1819010039</b:SourceId>
<b:SourceImageUrl/>
<b:SourceName/>
<b:SourceSearchFormUrl/>
<b:StarId>403</b:StarId>
<b:StarName>4*</b:StarName>
<b:TicketsIncluded>Included</b:TicketsIncluded>
<b:TourArchiveHash>AsBcr96RHxUv1810cTI28A==</b:TourArchiveHash>
<b:TourHash>ZdVTLJ1FPyDJIFpekzdUXg==</b:TourHash>
<b:TourName>_MCK_Шоп-тур с обяз._2014_Отд.</b:TourName>
<b:TourUrl/>

```

```

        </b:XmlTourRecord>
    </a:Rows>
    <a:RowCount>2318</a:RowCount>
    <a:SourcesVisas
        xmlns:b="urn:SletatRu:DataTypes:XmlVisaRange:v1"
    >
        <b:CurrencyName>EUR</b:CurrencyName>
        <b:Items
            xmlns:c="urn:SletatRu:DataTypes:XmlVisaRangeItem:v1"
        >
            <c:XmlVisaRangeItem>
                <c:MaxPrice>70</c:MaxPrice>
                <c:MinPrice>30</c:MinPrice>
                <c:SourceId>1819010049</c:SourceId>
            </c:XmlVisaRangeItem>
        </b:Items>
    </a:SourcesVisas>
    <a:Visa
        xmlns:b="urn:SletatRu:DataTypes:XmlVisa:v1"
        i:nil="true"
    />
</GetRequestResultResult>
</GetRequestResultResponse>
</s:Body>
</s:Envelope>

```

В ответ возвращается структура **GetRequestResultResult**, состоящая из следующих полей:

- ✓ **HotelCount** – общее количество отелей в поисковой выдаче;
- ✓ **LoadState** – массив данных о состоянии обработки поискового запроса, в котором каждый элемент **OperatorLoadState** описывает состояние обработки запроса одним оператором (структура **OperatorLoadState** описана в разделе [“GetRequestState”](#));
- ✓ **OilTaxes** – массив данных о топливных сборах, в котором каждый элемент информация о топливных сборах.
- ✓ **RequestId** – идентификатор поискового запроса;
- ✓ **Rows** – состоит из структур **XmlTourRecord**, которые описывают найденные туры;
- ✓ **RowCount** – количество туров, найденных в рамках поискового запроса на момент формирования ответа (в примере по заданному запросу найдено 410 цен);
- ✓ **SourcesVisas** – необходимо описание;
- ✓ **Visa** – информация о стоимости визы;

Поля структуры **XmlTourRecord**:

Поле	Тип	Описание
Adults	Int32	Количество туристов (взрослых).
BusinessTicketsDpt	Enum	Наличие билетов бизнес-класса (туда). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
BusinessTicketsRtn	Enum	Наличие билетов бизнес-класса (обратно). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
CheckInDate	String	Дата вылета в формате dd.MM.yyyy.

CityFromId	Int32	Идентификатор города вылета.
CityFromName	String	Город вылета.
CountryId	Int32	Идентификатор направления.
CountryName	String	Направление.
CreditMonthlyPay		
CreditMonths		
Currency	String	Валюта, в которой отображены цены. Возможные значения: RUB, USD и EUR.
Description		
DiscountCaption		
DiscountExpires		
DiscountPercent		
EarlyBookingValidTill		
EconomTicketsDpt	Enum	Наличие билетов эконом-класса (туда). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
EconomTicketsRtn	Enum	Наличие билетов эконом-класса (обратно). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
FewBusinessTicketsDpt	Int32	Если есть билеты бизнес-класса (поле BusinessTicketsDpt имеет значение “Available”) и значение этого поля отлично от нуля, то билетов МАЛО. Если туроператор сообщает о том, что билетов мало, но не указывает точного количества, поле принимает значение “-1”. Если передано любое иное число, отличное от нуля, то это точное количество оставшихся билетов.
FewBusinessTicketsRtn	Int32	Если есть билеты бизнес-класса обратно (поле BusinessTicketsRtn имеет значение “Available”) и значение этого поля отлично от нуля, то билетов МАЛО. Если туроператор сообщает о том, что билетов мало, но не указывает точного количества, поле принимает значение “-1”. Если передано любое иное число, отличное от нуля, то это точное количество оставшихся билетов.
FewEconomTicketsDpt	Int32	Если есть билеты эконом-класса (поле EconomTicketsDpt имеет значение “Available”) и значение этого поля отлично от нуля, то билетов МАЛО. Если туроператор сообщает о том, что билетов мало, но не указывает точного количества, поле принимает значение “-1”. Если передано любое иное число, отличное от нуля, то это точное количество оставшихся билетов.
FewEconomTicketsRtn	Int32	Если есть места на перелёт эконом-классом обратно (поле EconomTicketsRtn имеет значение “Available”) и значение этого поля отлично от нуля, то билетов МАЛО. Если туроператор сообщает о том, что билетов мало, но не указывает точного количества, поле принимает значение “-1”. Если передано любое иное число, отличное от нуля, то это точное количество оставшихся билетов.

FewPlacesInHotel	Int32	Если есть места в отеле (поле HotellsInStop имеет значение "Available") и значение этого поля отлично от нуля, то мест в отеле МАЛО. Если туроператор сообщает о том, что мест мало, но не указывает точного количества, поле принимает значение "-1". Если передано любое иное число, отличное от нуля, то это точное количество оставшихся мест.
Flags		
Found		
GiftCaption		
HotelDescription	String	Краткое описание отеля. Выдаётся только если параметр includeDescriptions имеет значение "true". Может быть пустым, если отель не слинкован или не имеет описания.
HotelDescriptionUrl	String	Ссылка на описание отеля.
HotelId	Int32	Идентификатор отеля. Если отель не слинкован, принимает значение "0".
HotellsInStop	Enum	Наличие мест в отеле. Возможные значения: Stop – нет мест, Available – есть места, Request – под запрос, и Unknown – нет данных.
HotelName	String	Название отеля.
HotelPhotosCount	Int32	Количество фотографий отеля.
HotelRating	Single	Рейтинг отеля по десятибалльной шкале. Если рейтинг отсутствует, поле принимает значение "0".
HotelTitleImageUrl	String	Ссылка на титульное изображение отеля в уменьшенном виде (115x80). Размеры изображения могут быть указаны самостоятельно с использованием следующего шаблона ссылки: http://hotelcard.dyndns.org/i/p/{hotel_id}_{image_index}_{h}_{w}.jpg – где: {hotel_id} – идентификатор отеля; {image_index} – индекс изображения отеля, начинается с 0; {h} – высота изображения (максимальное значение 300); {w} – требуемая ширина изображения (максимальное значение 300); Изображения не предоставляются для неслинкованных отелей, а также для отелей с отсутствующим описанием.
HtPlaceDescription	String	Описание размещения, выдаётся только если параметр includeDescriptions имеет значение "true". Может быть пустым, если размещение не слинковано или не имеет описания.
HtPlaceId	Int32	Идентификатор типа размещения. Если данные не слинкованы, поле принимает значение "0".
HtPlaceName	String	Тип размещения.

IsDemoMode	Boolean	Если выдача производится в демо-режиме, поле принимает значение “true”. В демо-режиме не отображаются цены, а также все результаты поиска заполнены демонстрационными данными.
Kids	Int32	Количество туристов (детей).
MealDescription	String	Описание типа питания. Выдаётся только если параметр includeDescriptions имеет значение “true”. Может быть пустым, если данные не слинкованы или тип питания не описан.
MealId	Int32	Идентификатор типа питания. Если данные не слинкованы, поле принимает значение “0”.
MealName	String	Тип питания.
Nights	Int32	Продолжительность тура (ночей).
OfferId	Int64	Идентификатор ценового предложения. Используется вместе с SourceId для актуализации тура и оформлении заказа.
OfferIdStr		
OriginalHotelName	String	Название отеля по данным туроператора.
OriginalStarName	String	Категории отеля по данным туроператора.
OriginalTownName	String	Название курорта по данным туроператора.
OriginalMealName	String	Тип питания по данным туроператора.
OriginalHtPlaceName	String	Тип размещения по данным туроператора.
OriginalRoomName	String	Типа номера по данным туроператора.
Price	Int32	Цена тура в запрошенной валюте.
PriceType	Int32	Тип цены. Возможные значения: 0 – полная цена тура, 1- цена в пересчёте на человека. Независимо от типа цены стоимость тура выдаётся конечной.
ReceivingParty		
ResortDescriptionUrl	String	Ссылка на описание курорта. Строка может быть пустой, если курорт не слинкован или не имеет ссылки на описание.
ResortId	Int32	Идентификатор курорта. Если курорт не слинкован, поле принимает значение “0”.
ResortName	String	Курорт.
RoomId	Int32	Идентификатор типа номера. Если тип номера не слинкован, поле принимает значение “0”.
RoomName	String	Тип номера.
SourceId	Int32	Зашифрованный идентификатор туроператора, значение используется вместе с OfferId при актуализации и заказе туров. В расширенной версии сервиса предоставляется оригинальный идентификатор туроператора (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).

SourceImageUrl	String	Ссылка на логотип туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
SourceName	String	Наименование туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
SourceSearchFormUrl	String	Ссылка на форму поиска туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
StarId	Int32	Идентификатор категории отеля. Если данные не слинкованы, поле принимает значение "0".
StarName	String	Категория отеля.
TicketsIncluded	Enum	Состав турпакета. Возможные значения: NotIncluded – авиаперелёт не включён в стоимость тура, Included – авиаперелёт включён, и Unknown – нет данных.
TourArchiveHash		
TourHash		
TourName	String	Название тура. Поле может быть пустым, если туроператор не предоставляет таких данных.
TourUrl	String	Ссылка на описание / бронирование тура на сайте туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru). На момент написания настоящего руководства некоторые туроператоры не имеют ссылок в сервисе Слетать.ру.

Элемент **Visa** содержит информацию о стоимости визы.

Поля элемента **Visa**:

Поле	Тип	Описание
Price	Int32	Размер визового сбора
CurrencyName	String	Валюта поля Price . Возможные значения: RUB, USD или EUR.

Элемент **Visa** может быть пустым (NULL), если в системе нет информации о стоимости визы. Если элемент **Visa** не пустой, то все вложенные элементы обязательно имеют значение.

Структура **OilTaxes** заполняется информацией о топливных сборах (элементы **XmlTourOilTax**) для туров в результате поиска.

Поля элемента **XmlTourOilTax** (знаком * отмечены обязательные поля):

Поле	Тип	Описание
------	-----	----------

AirCompanyName	String	Название авиакомпании, к которой относится информация по топливному сбору. Если информация относится ко всем авиакомпаниям, поле остаётся пустым. Может быть несколько элементов OilTax с одинаковыми значениями всех полей, кроме AirCompanyName и Tax. Таким образом сообщается информация о размерах топливных сборов у разных авиакомпаний. Если у одного из элементов OilTax поле AirCompanyName оставлено пустым, то информация этого элемента относится ко всем авиакомпаниям, а остальные элементы OilTax описывают исключения.
AirportName	String	Название аэропорта.
CurrencyName *	String	Валюта поля Tax. Возможные значения: RUB, USD, или EUR.
Date1 *	DateTime	Дата начала действия топливного сбора.
Date2 *	DateTime	Дата окончания действия топливного сбора.
FlightNumber	Int32	Номер рейса, к которому применяется сбор. Если информация относится ко всем авиакомпаниям, поле остаётся пустым.
PartnerName	String	Принимающая сторона. Если поле заполнено, топливный сбор действителен только для туров с указанной принимающей стороной. Если топливный сбор действителен для всех принимающих сторон, поле остаётся пустым. Может быть несколько элементов OilTax с одинаковыми значениями всех полей, кроме PartnerName и Tax.
ResortId	Int32	Идентификатор курорта.
ResortName		Название курорта.
Sourceld *	Int32	Идентификатор туроператора в зашифрованном виде. В расширенной версии сервиса поле содержат оригинальные идентификаторы туроператоров (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
Tax *	Int32	Размер топливного сбора. Если топливный сбор отсутствует, поле принимает значение "0".

Если в системе нет информации по топливным сборам для тура, то множество элементов **OilTax** будет пустым (NULL).

Чтобы сопоставить туры из выдачи и элементы Xml**TOilTax**, нужно отфильтровать элементы **OilTax** по параметру **Sourceld**, и отобрать туры, даты вылета которых попадают в диапазон, заданный полями **Date1** и **Date2**.

5.9.4. ContinueSearch

Метод **ContinueSearch** создаёт новый поисковый запрос с использованием данных предыдущего. Логика работы метода:

1. Проверяется статус обработки исходного поискового запроса (значение **requestId** последнего указывается в запросе метода **ContinueSearch**). Если поиск по проверяемому запросу ещё не завершён (т. е. если хотя бы один оператор ещё не выставил значение поля **IsProcessed** "true"), возвращается текст с описанием ошибки.
2. Если проверяемый запрос обработан, то максимальная цена в его выдаче увеличивается на единицу и становится минимальной для нового запроса.
3. Метод **ContinueSearch** создаёт новый поисковый запрос (в статистике такие запросы также учитываются как новые) и возвращает его идентификатор. Все параметры нового поискового запроса, кроме цены, остаются теми же, что и параметры исходного.
4. Полученный запрос обрабатывается методами **GetRequestState** и **GetRequestResult**.

Важно! Продолжение поиска возможно в течение 20 минут с начала предыдущего поиска.

Сигнатура:

```
XmlContinueSearchData ContinueSearch(  
    Int32 requestId)
```

Метод принимает единственный параметр **requestId** – идентификатор завершённого поискового запроса.

Пример SOAP-запроса:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <ContinueSearch  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <requestId>69587384</requestId>  
    </ContinueSearch>  
  </s:Body>  
</s:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>
```

```
<s:Body>
  <ContinueSearchResponse
    xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
  >
    <ContinueSearchResult
      xmlns:a="urn:SletatRu:DataTypes:XmlContinueSearchData:v1"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
    >
      <a:RequestId>69587436</a:RequestId>
    </ContinueSearchResult>
  </ContinueSearchResponse>
</s:Body>
</s:Envelope>
```

Структура **XmlContinueSearchData** содержит поле **RequestId** с идентификатором нового поискового запроса.

5.10. ActualizePrice

Метод **ActualizePrice** необходим для актуализации предложений туроператоров.

Сигнатура:

```
XmlActualPriceData ActualizePrice(  
    Int32 sourceId,  
    Int64 offerId,  
    Int32 requestId)
```

Параметры:

Параметр	Тип	Описание
requestId	Int32	Идентификатор поискового запроса. Выдаётся методами CreateRequest или ContinueSearch .
offerId	Int64	Идентификатор ценового предложения. Выдаётся каждому туру методом GetRequestResult .
sourceId	Int32	Зашифрованный идентификатор туроператора. Выдаётся каждому туру методом GetRequestResult .

Пример SOAP-запроса:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <ActualizePrice  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <sourceId>14173204</sourceId>  
      <offerId>1287318711</offerId>  
      <requestId>2862432</requestId>  
    </ActualizePrice>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope  
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <s:Body>  
    <ActualizePriceResponse  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"
```

```

>
<ActualizePriceResult
  xmlns:a="urn:SletatRu:DataTypes:XmlActualPriceData:v1"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
>
  <a:IsError>false</a:IsError>
  <a:IsFound>true</a:IsFound>
  <a:RandomNumber>869</a:RandomNumber>
  <a:TourInfo
    xmlns:b="urn:SletatRu:DataTypes:XmlActualTourInfo:v1"
  >
    <b:BusinessTicketsDpt>Stop</b:BusinessTicketsDpt>
    <b:BusinessTicketsRtn>Stop</b:BusinessTicketsRtn>
    <b:CheckIn>16.10.2011</b:CheckIn>
    <b:CityFromName>S.Petersburg</b:CityFromName>
    <b:CountryName>Египет</b:CountryName>
    <b:EconomTicketsDpt>Available</b:EconomTicketsDpt>
    <b:EconomTicketsRtn>Stop</b:EconomTicketsRtn>
    <b:HotelDescriptionUrl>
      http://hotelcard.dyndns.org/?id=2703
    </b:HotelDescriptionUrl>
    <b:HotelIsInStop>Available</b:HotelIsInStop>
    <b:HotelName>ELYSEES HOTEL HURGHADA 4*</b:HotelName>
    <b:HotelPhotosCount>12</b:HotelPhotosCount>
    <b:HotelTitleImageUrl>
      http://hotelcard.dyndns.org/i/p/2703_0.jpg
    </b:HotelTitleImageUrl>
    <b:HtPlaceName>DBL</b:HtPlaceName>
    <b:MealName>ALL INCLUSIVE</b:MealName>
    <b:Nights>14</b:Nights>
    <b:OriginalPrice>1516</b:OriginalPrice>
    <b:OriginalPriceCurrency>y.e</b:OriginalPriceCurrency>
    <b:Price>48816</b:Price>
    <b:PriceCurrency>RUB</b:PriceCurrency>
    <b:ResortName>Hurghada</b:ResortName>
    <b:RoomName>STANDARD ROOM</b:RoomName>
    <b:SourceId>0</b:SourceId>
    <b:SourceName/>
    <b:SourceSearchFormUrl/>
    <b:StarName>4*</b:StarName>
    <b:SysCityFromId>1264</b:SysCityFromId>
    <b:SysCityFromName>Санкт-Петербург</b:SysCityFromName>
    <b:SysCountryId>40</b:SysCountryId>
    <b:SysCountryName>Египет</b:SysCountryName>
    <b:SysHotelId>2703</b:SysHotelId>
    <b:SysHotelName>Apartment Hotel Elysees</b:SysHotelName>
    <b:SysHtPlaceId>13103</b:SysHtPlaceId>
    <b:SysHtPlaceName>DBL</b:SysHtPlaceName>
    <b:SysMealId>115</b:SysMealId>
    <b:SysMealName>AI</b:SysMealName>
    <b:SysResortId>1592</b:SysResortId>
    <b:SysResortName>Хургада</b:SysResortName>
    <b:SysRoomId>0</b:SysRoomId>
    <b:SysRoomName/>
    <b:SysStarId>403</b:SysStarId>
    <b:SysStarName>4*</b:SysStarName>
    <b:TicketsIncluded>Included</b:TicketsIncluded>

```

```

        <b:TourName>Hurghada (Saint-Petersburg)</b:TourName>
    </a:TourInfo>
</ActualizePriceResult>
</ActualizePriceResponse>
</s:Body>
</s:Envelope>

```

В ответ возвращается структура **ActualizePriceResult**, состоящая из следующих полей:

- ✓ **TourInfo** – структура, содержащая основную информацию о туре;
- ✓ **RandomNumber** – случайное трёхзначное число, по которому можно просмотреть информацию о туре в личном кабинете на сайте sletat.ru или в программе SletatRu Agent, используется для того, чтобы ваши менеджеры могли точно идентифицировать тур, который клиент нашёл самостоятельно;
- ✓ **IsFound** – поле принимает значение “false”, если тур не был найден в базе туроператора;
- ✓ **IsError** – принимает значение “true”, если при актуализации тура возникла ошибка.

Поля структуры **TourInfo**:

Поле	Тип	Описание
CountryName	String	Название страна по данным туроператора.
CityFromName	String	Название города вылета по данным туроператора.
ResortName	String	Название курорта по данным туроператора.
TourName	String	Название тура по данным туроператора.
TourDate	String	Дата вылета в формате dd.MM.yyyy.
Nights	Int32	Продолжительность тура (ночей).
HotelName	String	Название отеля по данным туроператора.
StarName	String	Категория отеля по данным туроператора.
RoomName	String	Тип номера по данным туроператора.
MealName	String	Тип питания по данным туроператора.
TicketsIncluded	Enum	Состав турпакета. Возможные значения: NotIncluded – авиаперелёт в стоимость тура не включён, Included – авиаперелёт включён, и Unknown – нет данных.
HotelsInStop	Enum	Наличие мест в отеле. Возможные значения: Stop – нет мест, Available – есть места, Request – под запрос, и Unknown – нет данных.
EconomTicketsDpt	Enum	Наличие билетов эконом-класса (туда). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
EconomTicketsRtn	Enum	Наличие билетов эконом-класса (обратно). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
BusinessTicketsDpt	Enum	Наличие билетов бизнес-класса (туда). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.
BusinessTicketsRtn	Enum	Наличие билетов бизнес-класса (обратно). Возможные значения: Stop – нет билетов, Available – есть билеты, Request – под запрос, и Unknown – нет данных.

HtPlaceName	String	Тип размещения по данным туроператора.
Price	Int32	Цена тура в запрошенной валюте.
PriceCurrency	String	Обозначение валюты. Возможные значения: RUB, EUR или USD.
OriginalPrice	Int32	Оригинальная цена тура в исходной валюте. Поскольку параметр доступен не всегда, значение этого поля может дублировать значение поля Price .
OriginalPriceCurrency	String	Обозначение валюты (по данным туроператора) поля OriginalPrice .
SourceId	Int32	Идентификатор туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
SourceName	String	Наименование туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в службу поддержки support@sletat.ru).
SourceSearchFormUrl	String	Ссылка на форму поиска туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru).
SysCountryId	Int32	Идентификатор страны по справочнику стран.
SysCountryName	String	Название страны по справочнику стран.
SysCityFromId	Int32	Идентификатор города вылета по справочнику городов вылета.
SysCityFromName	String	Название города вылета по справочнику городов вылета.
SysResortId	Int32	Идентификатор курорта по справочнику курортов. Если курорт туроператора не слинкован, поле принимает значение "0".
SysResortName	String	Название курорта по справочнику курортов. Если курорт туроператора не слинкован, поле принимает пустое значение.
SysHotelId	Int32	Идентификатор отеля по справочнику отелей. Если отель туроператора не слинкован, поле принимает значение "0".
SysHotelName	String	Название отеля по справочнику отелей. Если отель туроператора не слинкован, поле принимает пустое значение.
SysStarId	Int32	Идентификатор категории отеля по справочнику. Если категория отеля туроператора не слинкована, поле принимает значение "0".
SysStarName	String	Категория отеля по справочнику. Если категория отеля туроператора не слинкована, поле принимает пустое значение.
SysRoomId	Int32	Идентификатор типа номера по справочнику. Если тип номера туроператора не слинкован, поле принимает значение "0".
SysRoomName	String	Тип номера по справочнику. Если тип номера туроператора не слинкован, поле принимает пустое значение.
SysMealId	Int32	Идентификатор типа питания по справочнику. Если тип питания туроператора не слинкован, поле принимает значение "0".
SysMealName	String	Тип питания по справочнику. Если тип питания туроператора не слинкован, поле принимает пустое значение.
SysHtPlaceId	Int32	Идентификатор типа размещения по справочнику. Если тип размещения туроператора не слинкован, поле принимает значение "0".
SysHtPlaceName	String	Тип размещения. Если размещение туроператора не слинковано, поле принимает пустое значение.

HotelDescriptionUrl	String	Ссылка на описание отеля.
HotelTitleImageUrl	String	Ссылка на титульное изображение отеля в уменьшенном виде (115x80). Вы можете изменить размеры изображения, используя шаблон ссылки: <code>http://hotelcard.dyndns.org/i/p/{hotel_id}_{image_index}_{h}_{w}.jpg</code> где: {hotel_id} – идентификатор отеля; {image_index} – индекс изображения отеля, начинается с 0; {h} – высота изображения (максимальное значение 300); {w} – ширина изображения (максимальное значение 300). Изображения не предоставляются, если отель не слинкован или отсутствует его описание.
HotelPhotosCount	Int32	Количество фотографий отеля.
TourUrl	String	Ссылка на описание / бронирование тура на сайте туроператора. Поле заполняется только при покупке расширенной версии сервиса (по всем вопросам обращайтесь в Службу технической поддержки Слетать.ру по тел. 8(800)700-33-09 или по электронной почте support@sletat.ru). На момент написания настоящего руководства некоторые туроператоры не имеют ссылок в сервисе Слетать.ру.
OriginalCurrencyId	Int32	Идентификатор валюты тура по данным туроператора. Возможные значения: 2 – USD, 3 – EUR, 5 – RUR.

5.11. ActualizePriceByCode

Метод **ActualizePriceByCode** позволяет получать информацию о туре по четырёхзначному коду с сайта sletat.ru или по трёхзначному коду, полученному с помощью метода актуализации расширенного сервиса поиска XML/JSON. Четырёхзначные коды отображаются на сайте sletat.ru в поиске, а также в разделах «Горящие туры» и «Раннее бронирование».

Сигнатура:

```
XmlActualPriceData ActualizePriceByCode(  
    Int32 code)
```

Метод принимает единственный параметр **code** – четырёхзначный код тура с сайта Sletat.Ru (целое число 1000-9999) или трёхзначный код (целое число 100-999).

Пример SOAP-запроса:

```
<soap:Envelope  
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
>  
  <soap:Body>  
    <ActualizePriceByCode  
      xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"  
    >  
      <code>9999</code>  
    </ActualizePriceByCode>  
  </soap:Body>  
</soap:Envelope>
```

Ответ идентичен выдаче метода **ActualizePrice**.

Выполнение метода может быть завершено с ошибкой по следующим причинам:

- ✓ код тура устарел (действителен сутки с момента получения);
- ✓ код тура указан неверно (принимаются только четырёхзначные коды с сайта sletat.ru);
- ✓ нет подходящей лицензии для выполнения запроса (был указан трёхзначный код тура, но у клиента нет действительной лицензии на сервис актуализации или на расширенный сервис XML/JSON).

Описание ошибок всегда возвращается.

5.12. SaveTourOrder

Метод **SaveTourOrder** добавляет заказ тура в систему Слетать.ру. В вашем личном кабинете на сайте sletat.ru вы можете настроить SMS- и email-уведомления о новых заказах. Приложение **SletatRu Agent** для Windows также отслеживает новые заказы и присылает вам оповещения.

Сигнатура:

```
void SaveTourOrder(  
    Int32 requestId,  
    Int64 offerId,  
    Int32 sourceId,  
    String user,  
    String email,  
    String phone,  
    String info)
```

Параметры:

Параметр	Тип	Описание
requestId	Int32	Идентификатор поискового запроса. Выдаётся методами CreateRequest или ContinueSearch при создании нового поискового запроса;
offerId	Int64	Идентификатор ценового предложения. Выдаётся каждому туру методом GetRequestResult .
sourceId	Int32	Идентификатор туроператора. Выдаётся каждому туру методом GetRequestResult .
user	String	Имя туриста.
email	String	Контактный e-mail.
phone	String	Контактный телефон.
info	String	Комментарий туриста (необязательное поле).

Пример SOAP-запроса:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <SaveTourOrder xmlns="urn:SletatRu:Contracts:Soap11Gate:v1">  
      <requestId>2857817</requestId>  
      <offerId>225355464</offerId>  
      <sourceId>14146146</sourceId>  
      <user>John Doe</user>  
      <email>johndoe@sample.com</email>  
      <phone>+7 905 555 55 55</phone>  
      <info>any additional information here</info>  
    </SaveTourOrder>  
  </soap:Body>  
</soap:Envelope>
```

Пример SOAP-ответа:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SaveTourOrderResponse xmlns="urn:SletatRu:Contracts:Soap11Gate:v1"/>
  </s:Body>
</s:Envelope>
```

В случае успешного добавления заказа никакой информации метод не возвращает.