HotSchedules Web Services

API Documentation v3.0

Table of Contents

Introduction

HotSchedules provides a Web Services API for client integration. This API can be used to exchange Employee, Labor and Sales information with HotSchedules directly, rather than via file exchange or HSConnect integration.

This API consists of a set of SOAP web services, implemented using the CXF framework. Calls to these services should be done via serialized data objects. Services will use a UsernameToken security header, which must be explicitly added by the third party before we will accept their web service calls. We will provide each company with a username and password for secure access to the API services.

Getting started

How to request a username/password

Please contact your HotSchedules account manager for a username/password.

SSL certificates

Our web services connect through an SSL encrypted connection, requiring a certificate to be present on the client side. HotSchedules uses DigiCert for our services.hotschedules.com SSL certificate. Because GoDaddy will not generate SSL certificates directly off of the root cert, we also require their intermediate certificate to be installed on the client. The certificates required can be downloaded directly from GoDaddy.

Download location: https://certs.godaddy.com/anonymous/repository.seam

The two certificates required are:

Go Daddy Class 2 Certification Authority Root Certificate

- gd-class2-root.crt
- Certificate File Hash (sha1) : F0 EC B0 51 00 08 A3 1A D3 60 9A C8 C5 54 10 C3 9B F4 BA A2
- Certificate Thumbprint (sha1) : 27 96 ba e6 3f 18 01 e2 77 26 1b a0 d7 77 70 02 8f 20 ee e4

Go Daddy Secure Server Certificate (Intermediate Certificate)

- gd_intermediate.crt
- Certificate File Hash (sha1) : 17 93 21 10 65 FA 24 D9 46 53 18 83 DE 62 CA 0F AE 4D 61 7D
- Certificate Thumbprint (sha1) : 7C 46 56 C3 06 1F 7F 4C 0D 67 B3 19 A8 55 F6 0E BC 11 FC 44

Once you have the certificates downloaded, you must install them in your keystore. Here is an Oracle article explaining keystores as well as how to install a certificate into your keystore:

Oracle Certificate Installation HOW-TO

Generally, the keystore file is created in your installed JRE's directory (/jre/lib/security). The default name is cacerts, and the default password is changeit. If you need to generate a new keystore file, see the following link for directions.

Apache SSL Configuration HOW-TO

Once the keystore is located in the /lib/security directory of your Java install, and the certificates are installed in the keystore, your code should be able to correctly navigate the SSL connection.

Testing

In order to test the web services, the client must have a test site set up in HotSchedules. The testing will be done on our production servers, but on a site that is not tied to any real client. HotSchedules will set up a test site for the client to use. After the setup is complete, there are a few things that the client needs.

- **Concept number:** from HotSchedules this corresponds to the external ref of the concept to which the test site belongs. Must be set by HotSchedules and must be unique within a company. This is parameter number 1 in the web service operation calls (concept) and must match exactly.

- **Store number:** from HotSchedules, this corresponds to the site's store number/external ref. This is parameter number 2 in the web service operation calls (storeNum) and must match exactly.

- **Username and password** to log in and view the test site. This is needed so the tester can verify employee inserts/terminations, sales data, etc.

Once this is all set up and the client has this data, he/she can make web services calls (such as *EmpService.setEmps()*) with the concept and storeNum IDs provided from HotSchedules.

NOTE: in order to test sales data, the client will have to use the HotSchedules Proforma II report. This means that it must be set up by HotSchedules to be viewable. Included in this setup is adding day parts, adding sales categories and revenue centers, as well as adding sales summary items.

Testing employee data

The most basic data to be tested in the web services API is employee data. This is where I would suggest all clients begin their testing. There are two operations in the EmpService web service.

setEmps (int concept, int storeNum, WSEmployee[] emps)

This is the main operation to add/remove/update employes in the HotSchedules system. See the Testing section above for information on the first two parameters (concept and storeNum). The third parameter will be an array of WSEmployee objects, each object representing one employee to be inserted/terminated/updated.

Steps for testing:

Try creating a single WSEmployee object and setting only the first name, last name, emp number (POS ID) and status (1). This will insert a single active employee that you should be able to see in the Staff tab of the test site.

Next, try adding two employees at once, each with different names and emp numbers.

NOTE: the emp number MUST be unique for employee, as this is how we verify the uniqueness of each employee within a store.

Once you can add multiple employees, try setting the other non-vital information on one of these already adding employees and updating them. Add a phone number (10 digit) or an address.

Next, change an already-inserted employee's status to -1, to test terminating an employee. Once you run this sync, you should no longer be able to see the employee in HotSchedules, as our system will treat this employee as terminated. If your POS provides fire dates, this would be a good place to test that as well, though you cannot check this particular piece of data in the UI.

setEmpJobs(int concept, int storeNum, WSEmpJob[] jobs)

This is the operation that allows you to add jobs to employees in HotSchedules. The employee must already exist in HotSchedules, so this should be tested after the setEmps operation. It works much like setEmps, except with WSEmpJob objects representing jobs, instead of the WSEmployee objects.

NOTE: for employee jobs, the job must already exist on the site. to view existing jobs, look under settings->company under the jobs section. If there are none, please ask HotSchedules to add a few jobs with corresponding external (POS) IDs.

Choose a single employee who is already inserted into HotSchedules (from the tests above). Create a single WSEmpJob object for him/her, corresponding to a job from the HotSchedules test site. Make sure you set the posEmpId and posJobId fields, as these are required. Once you run this test, you should see the job next tot he employee's name in HotSchedules on the staff tab.

Next try adding multiple jobs to the same employee.

After this is working, try adding multiple jobs to multiple employees through one web service call. This is pretty much all that needs to be tested for employee jobs.

HotSchedules Available Web Services

Name	Description
EmpService	This service provides operations for a third party to push or request employee and employee job data into HotSchedules. The WSDL is available at: https://services.hotschedules.com/api/services/EmpService?wsdl
ScheduleService	This service is intended for third parties to be able to grab scheduled shifts from HotSchedules and import them into their POS/data warehouse/enterprise/etc. system. The WSDL is available at: https://services.hotschedules.com/api/services/ScheduleService?wsdl
TimeCardService	This service is intended for third parties to be able to import their time cards into the HotSchedules system or get time cards from HotSchedules in a straightforward fashion. The WSDL is available at: https://services.hotschedules.com/api/services/TimeCardService?wsdl
SalesService	This service is intended for third parties to be able to import their sales items into the HotSchedules system in a straightforward fashion. Currently there is only one method for setting sales items. The WSDL is available at: http://services.hotschedules.com/api/services/SalesService?wsdl
ProjectedSalesService	This service is intended for third parties to be able to grab projected sales data from HotSchedules and import it into their POS/data warehouse/enterprise/etc. system. The WSDL is available at: https://services.hotschedules.com/api/services/ProjectedSalesService?wsdl
LaborService	This service is intended for third parties to be able to request labor data from HotSchedules and import it into their POS/data warehouse/enterprise/etc. system. The WSDL is available at: https://services.hotschedules.com/api/services/LaborService?wsdl
VolumeService	This service is intended for third parties to be able to request and send volume driver related data from/to HotSchedules. Driver items include examples such as guests, tables, entrees….etc. The WSDL is available at: https://services.hotschdules.com/api/services/VolumeService?wsdl
TimeOffService	This service is intended for third parties to be able to request approved timeoff requests from HotSchedules The WSDL is available at: http://services.hotschedules.com/api/services/TimeOffService?wsdl
CertificationService	This service is intended for third parties to be able to request and set certification requests from HotSchedules The WSDL is available at: http://services.hotschedules.com/api/services/CertificationService?wsdl

EmpService Web Service

These services provides operations for a third party to push or request employee and employee job data into HotSchedules. The WSDL is available at:

https://services.hotschedules.com/api/services/EmpService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods

Name	Description
getEmpAvailability	This method takes in a concept ID and a store ID and returns an array of wsEmpAvailability objects. It is meant to get a list of all availability for that store.
getEmpInfo	This method takes in a concept ID and a store ID and returns an array of wsEmpInfo objects. It is meant to get a list of all employees for that store.
getEmpJobs	This method takes in a concept ID and a store ID and returns an array of wsEmpJob objects. It is meant to get a list of all jobs assigned to all employees for that store.
getStoreEmployees	This method takes in a concept ID, store ID, a flag to determine if only active employees are returned, and returns an array of WSEmployee objects.
getStoreJobs	This method takes in a concept ID and a store ID, and returns an array of all jobs currently defined in HotSchedules for the given concept/store.
sayHello	Missing description.
setEmpJobs	This method takes in a concept ID, store ID, and an array of WSEmpJob objects to assign jobs to individual employees. This method returns a WSReturn object.
setEmps	This method takes in a concept ID, store ID and an array of WSEmployee objects. Using the authentication from the username token and the conecpt and store IDs, the server will resolve which HotSchedules client this sync is for. The array of employees will be parsed on the server side to employees who need to be inserted or updated in the HS database. This method returns a WSReturn object.

Method: getEmpAvailability

This method takes in a concept ID and a store ID and returns an array of wsEmpAvailability objects. It is meant to get a list of all employee availability for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
activeOnly	boolean	1..1	Boolean that defines whether or not to include terminated employees in response.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsEmpAvailability Array	1..1	Array of wsEmpAvailability objects, which describes all jobs assigned to all active employees for that store

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/EmpService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262747932814">
        <wsse:Username>laura1234!</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
        <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">9iOMGex3ZDEpdw4xsAweWA==</wsse:Nonce>
        <wsu:Created>2016-01-12T19:37:59.328Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <emp:getEmpAvailability>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <activeOnly>true</activeOnly>
    </emp:getEmpAvailability>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getEmpAvailabilityResponse
xmlns:ns1="http://services.hotschedules.com/api/services/EmpService">
      <return>
        <item>
          <availabilities>
            <dayName>Sunday</dayName>
            <dayNum>1</dayNum>
            <parHoursMax>-1</parHoursMax>
            <parHoursMin>-1</parHoursMin>
            <parShiftsMax>-1</parShiftsMax>
            <parShiftsMin>-1</parShiftsMin>
            <partialBeforeAfter>After</partialBeforeAfter>
            <partialTime>11:45</partialTime>
            <shiftId>964362624</shiftId>
            <shiftName>Breakfast</shiftName>
            <statusName>Partially Available</statusName>
            <statusNum>1</statusNum>
          </availabilities>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
        </item>
      </return>
    </ns1:getEmpAvailabilityResponse>
  </soap:Body>
</soap:Envelope>
```

Method: getEmpInfo

This method takes in a concept ID and a store ID and returns an array of wsEmpInfo objects. It is meant to get a list of all employees assigned to a schedule for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
activeOnly	boolean	1..1	Boolean that defines whether or not to include terminated employees in response.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsEmpInfoArray	1..1	Array of wsEmpInfo objects, which describes all jobs assigned to all active employees for that store

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/EmpService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262475127713">
          <wsse:Username>laura1234!</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">deQn1F/GUBV/fMNJ7pdatA==</wsse:Nonce>
          <wsu:Created>2016-01-12T18:52:31.276Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <emp:getEmpInfo>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <activeOnly>true</activeOnly>
    </emp:getEmpInfo>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
   <soap:Body>
      <ns1:getEmpInfoResponse xmlns:ns1="http://services.hotschedules.com/api/services/EmpService">
         <return>
            <item>
               <accountCreated>2015-11-04T10:54:22.807-06:00</accountCreated>
               <assignedSchedules>
                  <extId>0</extId>
                  <hsId>964330043</hsId>
                  <name>Kitchen</name>
               </assignedSchedules>
               <assignedSchedules>
                  <extId>-1</extId>
                  <hsId>964318722</hsId>
                  <name>Meetings</name>
               </assignedSchedules>
               <assignedSchedules>
                  <extId>0</extId>
                  <hsId>964330041</hsId>
                  <name>Team Member</name>
               </assignedSchedules>
               <empHrId>-1</empHrId>
               <empNum>1</empNum>
               <lastUpdated>2015-11-09T10:29:31.123-06:00</lastUpdated>
               <permissionSetName>Employee</permissionSetName>
            </item>
            <item>
               <accountCreated>2015-11-04T08:33:41.090-06:00</accountCreated>
               <empHrId>-1</empHrId>
               <empNum>-1</empNum>
               <permissionSetName>HS ASC Support</permissionSetName>
            </item>
            <item>
               <accountCreated>2015-11-04T08:42:02.483-06:00</accountCreated>
               <empHrId>-1</empHrId>
               <empNum>-1</empNum>
               <lastUpdated>2015-11-05T18:29:55.857-06:00</lastUpdated>
               <permissionSetName>Admin ALL</permissionSetName>
            </item>
```

```
        <item>
          <accountCreated>2015-11-04T08:49:32.623-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2015-11-04T08:49:33.120-06:00</lastUpdated>
          <permissionSetName>Default HS Support User</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-04T08:49:33.887-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2015-11-04T08:49:34.093-06:00</lastUpdated>
          <permissionSetName>Default HotSchedules Employee</permissionSetName>
        </item>
      </return>
    </ns1:getEmpInfoResponse>
  </soap:Body>
</soap:Envelope>
```

Method: getEmpJobs

This method takes in a concept ID and a store ID and returns an array of wsEmpJob objects. It is meant to get a list of all jobs assigned to all employees for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsEmpJobArray	1..1	Array of wsEmpJob objects, which describes all jobs assigned to all active employees for that store

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/EmpService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262410123010">
        <wsse:Username>laura1234!</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
        <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">cZvq5MrVptRUWgLIIqmYRg==</wsse:Nonce>
        <wsu:Created>2016-01-12T18:41:41.230Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <emp:getEmpJobs>
      <concept>1</concept>
      <storeNum>1</storeNum>
    </emp:getEmpJobs>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
   <soap:Body>
      <ns1:getEmpJobsResponse xmlns:ns1="http://services.hotschedules.com/api/services/EmpService">
         <return>
            <item>
               <clientId>19054117</clientId>
               <hsEmpId>17896647</hsEmpId>
               <hsJobId>27229283</hsJobId>
               <ovtWage>14.25</ovtWage>
               <posEmpId>1</posEmpId>
               <posJobId>-1</posJobId>
               <primary>true</primary>
               <regWage>9.5</regWage>
               <storeNum>1</storeNum>
            </item>
               <item>
               <clientId>19054117</clientId>
               <hsEmpId>17896647</hsEmpId>
               <hsJobId>27229285</hsJobId>
               <ovtWage>11.25</ovtWage>
               <posEmpId>1</posEmpId>
               <posJobId>-1</posJobId>
               <primary>false</primary>
               <regWage>7.5</regWage>
               <storeNum>1</storeNum>
            </item>
         </return>
      </ns1:getEmpJobsResponse>
   </soap:Body>
</soap:Envelope>
```

Method: getStoreEmployees

This method takes in a concept ID, store ID, a flag to determine if only active employees are returned, and returns an array of WSEmployee objects.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
activeOnly	boolean	1..1	Boolean that defines whether or not to include terminated employees in response.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsEmployeeArray	1..1	Array of WSEmployee objects

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/EmpService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
         <wsse:UsernameToken>
            <wsse:Username>REDACTED</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </SOAP-ENV:Header>
   <ns1:Body>
      <ns2:getStoreEmployees>
         <concept>1</concept>
        <storeNum>37</storeNum>
         <activeOnly>true</activeOnly>
      </ns2:getStoreEmployees>
   </ns1:Body>
</SOAP-ENV:Envelope> SAMPLE
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
   <soap:Body>
      <ns1:getStoreEmployeesResponse>
       <return>
          <item>
           <address/>
           <address2/>
           <altId>0</altId>
           <city/>
           <clientId>REDACTED</clientId>
           <email> </email>
           <empNum>-1</empNum>
           <FName>REDACTED</FName>
           <hsId>REDACTED</hsId>
           <LName>REDACTED</LName>
           <mobile> </mobile>
           <NName/>
           <phone>no number</phone>
           <state/>
           <status>1</status>
           <storeNum>37</storeNum>
           <zipCode/>
           <dob>1971-12-07T00:00:00-06:00</dob>
           <hireDate>2010-03-02T18:34:24.803-06:00</hireDate>
           <termDate>2014-08-31T00:00:00-05:00</termDate>
          </item>
          <item>
           <address>6504 Bridge Point Parkway</address>
           <address2>Suite 425</address2>
           <altId>0</altId>
           <city>Austin</city>
           <clientId>REDACTED</clientId>
           <email/>
           <empNum>-1</empNum>
           <FName>HotSchedules</FName>
           <hsId>REDACTED</hsId>
           <LName>Support</LName>
           <mobile/>
           <NName>Support</NName>
           <phone>(512) 219-1492</phone>
           <state>TX</state>
           <status>1</status>
           <storeNum>37</storeNum>
```

```
                <zipCode>78730</zipCode>
                <dob>2000-02-01T00:00:00-06:00</dob>
                <hireDate>2010-07-07T09:59:00.907-05:00</hireDate>
                <termDate>2014-08-31T00:00:00-05:00</termDate>
            </item>
        </return>
    </ns1:getStoreEmployeesResponse>
  </soap:Body>
</soap:Envelope>
```

Method: getStoreJobs

This method takes in a concept ID and a store ID, and returns an array of all jobs currently defined in HotSchedules for the given concept/store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsJobArray	1..1	Array of WSJobs.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/EmpService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
         <wsse:UsernameToken>
            <wsse:Username>REDACTED</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
      </wsse:UsernameToken>
     </wsse:Security>
  </SOAP-ENV:Header>
  <ns1:Body>
     <ns2:getStoreJobs>
      <concept>1</concept>
      <storeNum>37</storeNum>
     </ns2:getStoreJobs>
  </ns1:Body>
</SOAP-ENV:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
      <soap:Body>
       <ns1:getStoreJobsResponse>
          <return>
                <item>
                 <clientId>REDACTED</clientId>
                 <defRate>0.0</defRate>
                 <hsId>753820797</hsId>
                 <jobName>A BAR</jobName>
                 <posId>746000</posId>
                 <storeNum>37</storeNum>
                </item>
                <item>
                 <clientId>REDACTED</clientId>
                 <defRate>0.0</defRate>
                 <hsId>753820798</hsId>
                 <jobName>A BOH TRAINING</jobName>
                 <posId>741900</posId>
                 <storeNum>37</storeNum>
                </item>
                <item>
                 <clientId>REDACTED</clientId>
                 <defRate>0.0</defRate>
                 <hsId>753827134</hsId>
                 <jobName>A CAP</jobName>
                 <posId>748100</posId>
                 <storeNum>37</storeNum>
                </item>
          </return>
       </ns1:getStoreJobsResponse>
    </soap:Body>
   </soap:Envelope>
```

Method: setEmpJobs

This method takes in a concept ID, store ID, and an array of WSEmpJob objects to assign jobs to individual employees. This method returns a WSReturn object.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
jobs	wsEmpJobArray	1..1	Array of WSEmpJob objects. Each object represents one employee job, so a single employee can have one or more employee jobs (EmpJob) assigned to him/her. Each job that the employee works will be a separate object in this array.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description
Exception	Exception	Missing description.

Method: setEmps

This method takes in a concept ID, store ID and an array of WSEmployee objects. Using the authentication from the username token and the conecpt and store IDs, the server will resolve which HotSchedules client this sync is for. The array of employees will be parsed on the server side to employees who need to be inserted or updated in the HS database. This method returns a WSReturn object.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
emps	wsEmployeeArray	1..1	Array of WSEmployee objects. Each object represents an employee at this store.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description
Exception	Exception	Missing description.

Name	Description
dataEmpJob	Employee job object. Contains job information (job code, regular and OT rates, primary flag) for a given employee job.
dataEmployee	Employee information object. Contains employee information (name, address, phone info, active/terminated status)
dataJob	General job information object. Contains job information (job code, job name, default rate).
wsEmpJob	Extends dataEmpJob.
wsEmployee	Extends dataEmployee (contains DOB, hire date, termination date)
wsJob	Extends dataJob.
wsReturn	Contains array of error strings, fail count, success flag, success count.

Array EmpJobs

Employee job object. Contains job information (job code, regular and OT rates, primary flag) for a given employee job.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
hsEmpId	int	1..1	No	HotSchedules internal employee account ID
hsJobId	int	1..1	No	HotSchedules internal job code ID
ovtWage	float	1..1	No	Overtime hourly wage rate for employee
posEmpId	int	1..1	No	POS numeric ID for employee
posJobId	int	1..1	No	POS numeric ID for the job code
primary	boolean	1..1	No	Boolean flag to designate if the job code is the primary job for the employee
regWage	float	1..1	No	Regular hourly wage rate for employee
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
date	dateTime	0..1	No	Date job was added for the employee

Referenced By

- Complex Type wsEmpJob

Emp Array

Employee information object. Contains employee information (name, address, phone info, active/terminated status)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
address	string	0..1	No	Optional-Employee Address line 1
address2	string	0..1	No	Optional-Employee Address line 2
altId	int	1..1	No	Alternative ID for employee generally reserved for the HR ID.
city	string	0..1	No	City field for Address
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
email	string	0..1	No	Employee's email address
empNum	int	1..1	No	Employee POS ID
FName	string	0..1	No	Employee First Name
hsId	int	1..1	No	Optional-HotSchedule unique employee account ID
LName	string	0..1	No	Employee Last Name
mobile	string	0..1	No	Optional-Employees Mobile Phone Number
NName	string	0..1	No	Optional-Employee Nick Name
phone	string	0..1	No	Optional-Employee phone number
state	string	0..1	No	State field for Address
status	int	1..1	No	Active = 1, Inactive = 0, Terminated = -1
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
zipCode	string	0..1	No	Zip Code from Address

Referenced By

- Complex Type wsEmployee

WSJob Array

General job information object. Contains job information (job code, job name, default rate).

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
defRate	float	1..1	No	Default Pay Rate for a Job Code
hsId	int	1..1	No	Optional-HotSchedules numeric ID for Job Code
jobName	string	0..1	No	Name for Job Code
posId	int	1..1	No	Numeric POS ID for Job Code
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.

Referenced By

- Complex Type wsJob

wsEmpJobArray

Extends dataEmpJob.

Derived By

Extending dataEmpJob

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
hsEmpId	int	1..1	No	Optional-HotSchedule unique employee account ID
hsJobId	int	1..1	No	Optional-HotSchedules internal ID for the Job Code
ovtWage	float	1..1	No	Overtime hourly wage for job code for the employee
posEmpId	int	1..1	No	POS numeric identifier for the employee
posJobId	int	1..1	No	POS numeric identifier for the job
primary	boolean	1..1	No	Flag to indicate if a job is the primary job for the employee
regWage	float	1..1	No	Regular hourly wage for the job for the employee
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
date	dateTime	0..1	No	Date job was added for the employee

Referenced By

- Element item [type wsEmpJobArray]

Extends dataEmployee (contains DOB, hire date, termination date)

Derived By

Extending dataEmployee

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
address	string	0..1	No	Optional-Employee Address line 1
address2	string	0..1	No	Optional-Employee Address line 2
altId	int	1..1	No	Alternative ID for employee generally reserved for the HR ID.
city	string	0..1	No	City field for Address
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
email	string	0..1	No	Employee's email address
empNum	int	1..1	No	Employee POS ID
FName	string	0..1	No	Employee First Name
hsId	int	1..1	No	Optional-HotSchedule unique employee account ID
LName	string	0..1	No	Employee Last Name
mobile	string	0..1	No	Optional-Employees Mobile Phone Number
NName	string	0..1	No	Optional-Employee Nick Name
phone	string	0..1	No	Optional-Employee phone number
state	string	0..1	No	State field for Address
status	int	1..1	No	Active = 1, Inactive = 0, Terminated = -1
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
zipCode	string	0..1	No	Zip Code from Address
dob	dateTime	0..1	No	Employee Date of Birth
hireDate	dateTime	0..1	No	Employee Hire Date

termDate	dateTime	0..1	No	Employee Termination date. Only required for employees with a status of -1

Referenced By

- Element item [type wsEmployeeArray]

Extends dataJob.

Derived By

Extending dataJob

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
defRate	float	1..1	No	Default Pay Rate for a Job Code
hsId	int	1..1	Yes	Optional-HotSchedules numeric ID for Job Code
jobName	string	0..1	No	Name for Job Code
posId	int	1..1	No	Numeric POS ID for Job Code
storeNum	int	1..1	No	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.

Referenced By

- Element item [type wsJobArray]

Contains array of error strings, fail count, success flag, success count.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
errors	string	0..*	Yes	Description of the error code
failCount	int	1..1	No	Total number of records that failed to meet basic import specifications
success	boolean	1..1	No	Indicator of the success or failure
successCount	int	1..1	No	Total number of records that successfully met basic import specifications

Name	Description
address [type dataEmployee]	Address Line 1
address2 [type dataEmployee]	Address Line 2
altId [type dataEmployee]	Alternative ID for employee generally reserved for the HR ID.
city [type dataEmployee]	City field for Address
clientId [type dataEmpJob]	Unique client ID provided via HotSchedules
clientId [type dataEmployee]	Unique client ID provided via HotSchedules
defRate [type dataJob]	Default Pay Rate for job code
dob [type wsEmployee]	Employee Date of Birth
email [type dataEmployee]	Employee email address
empNum [type dataEmployee]	Employee POS ID
errors [type wsReturn]	Error Codes
Exception	Exception Codes
failCount [type wsReturn]	Total number of records that failed to meet basic import specifications
FName [type dataEmployee]	Employee First Name
hireDate [type wsEmployee]	Employee Hire Date
hsEmpId [type dataEmpJob]	HotSchedules Internal employee account ID
hsId [type dataEmployee]	HotSchedules Internal employee account ID
hsId [type dataJob]	HotSchedules Internal employee account ID
hsJobId [type dataEmpJob]	HotSchedules Internal job code ID
item [type wsEmpJobArray]	All jobs assigned to all active employees for that store
item [type wsEmployeeArray]	All employees at this store.
item [type wsJobArray]	All jobs at this store.
jobName [type dataJob]	Name of Job Code
LName [type dataEmployee]	Employee Last Name
message [type Exception]	Missing description.
mobile [type dataEmployee]	Employee mobile phone number
NName [type dataEmployee]	Employee Nick Name
ovtWage [type dataEmpJob]	Overtime hourly rate for employee for job
phone [type dataEmployee]	Employee phone number
posEmpId [type dataEmpJob]	Employee POS ID
posId [type dataJob]	Job Code POS ID

posJobId [type dataEmpJob]	Job Code POS ID
primary [type dataEmpJob]	Flag to indicate if job is the primary job for the employee
regWage [type dataEmpJob]	Regular Hourly wage for employee for job
state [type dataEmployee]	State field from Address

Name	Description
status [type dataEmployee]	Active = 1, Inactive = 0, Terminated = -1
storeNum [type dataEmpJob]	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
storeNum [type dataEmployee]	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
storeNum [type dataJob]	Unique numeric store ID within HotSchedules. Generally set up to mirror the client internal store IDs.
success [type wsReturn]	Missing description.
successCount [type wsReturn]	Number of records that successfully met the basic import specification
termDate [type wsEmployee]	Employee termination date
zipCode [type dataEmployee]	Zip code from Address

ScheduleService Web Service

Description

This service is intended for third parties to be able to grab scheduled shifts from HotSchedules and import them into their POS/data warehouse/enterprise/etc. system. The WSDL is available at:

https://services.hotschedules.com/api/services/ScheduleService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods: ScheduleService

Methods

Name	Description
getLocations	This method takes in a concept ID and store ID. It returns an array of dataLocation objects, each of which represent one HotSchedules schedule location.
getSchedule	This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem objects, which represent one scheduled shift each, for import into the POS. This method returns a limited amount of data per shift: employee HS ID, employee POS ID, internal HS Job ID, POS Job ID, shift start date/time, shift end date/time, work week start date/time, work week end date/time.
getSchedule2	This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem2 objects, which represent one scheduled shift each, for import into the POS. This method returns the same data as getSchedule, plus extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS.
getScheduleV3	This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem3 objects, which represent one scheduled shift each, for import into the POS. This method returns the same data as getSchedule, plus extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS. This method uses hsSimpleDate objects for dates.
getShiftsV3	This method takes in a concept ID, store ID, start and end dates and three flags (isHouse, isScheduled and isPosted). It returns an array of WSScheduleItem3 objects, which represent one scheduled shift each. What shifts are returned depends on the flags set: isHouse - includes scheduled or posted shifts that are not assigned to an employee (called 'house shifts' in HotSchedules). isScheduled - includes shifts that are in schedules that have been saved in HotSchedules, but might not have been posted isPosted - includes shifts that are in schedules that have been set to the 'posted' status within HotSchedules This method returns extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS.

	This method uses hsSimpleDate objects for dates.
sayHello	Missing description.

Method: getLocations

Description

This method takes in a concept ID and store ID. It returns an array of dataLocation objects, each of which represent one HotSchedules schedule location.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	dataLocationArray	1..1	Array of dataLocation objects.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
         <wsse:UsernameToken>
            <wsse:Username>REDACTED</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </SOAP-ENV:Header>
   <ns1:Body>
      <ns2:getLocations>
       <concept>1</concept>
       <storeNum>103</storeNum>
      </ns2:getLocations>
   </ns1:Body>
</SOAP-ENV:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
      <soap:Body>
         <ns1:getLocationsResponse>
          <return>
                <item>
                 <disabled>false</disabled>
                 <id>677337753</id>
                 <name>Location Name 1</name>
                </item>
                <item>
                 <disabled>false</disabled>
                 <id>517879406</id>
                 <name>Location Name 2</name>
                </item>
          </return>
         </ns1:getLocationsResponse>
      </soap:Body>
</soap:Envelope>
```

Method: getSchedule

Description

This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem objects, which represent one scheduled shift each, for import into the POS. This method returns a limited amount of data per shift: employee HS ID, employee POS ID, internal HS Job ID, POS Job ID, shift start date/time, shift end date/time, work week start date/time, work week end date/time.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
start	dateTime	1..1	First day of scheduled shifts you are requesting.
end	dateTime	1..1	Last day of scheduled shifts you are requesting.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsScheduleItemArray	1..1	Array of WSScheduleItem objects, which represent one scheduled shift each.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
  <wsse:Security mustUnderstand="true">
     <wsse:UsernameToken>
        <wsse:Username>REDACTED</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
  >REDACTED</wsse:Password>
     </wsse:UsernameToken>
  </wsse:Security>
 </SOAP-ENV:Header>
  <ns1:Body>
     <ns2:getSchedule>
      <concept>1</concept>
      <storeNum>37</storeNum>
      <start>2014-04-18T00:00:00</start>
      <end>2014-04-18T00:00:00</end>
     </ns2:getSchedule>
  </ns1:Body>
</SOAP-ENV:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
      <soap:Body>
       <ns1:getScheduleResponse>
          <return>
                <item>
                 <empHsId>REDACTED</empHSId>
                 <empPosId>REDACTED</empPosId>
                 <jobHsId>2129482342</jobHsId>
                 <jobPosId>453</jobPosId>
                 <in>2014-04-18T05:00:00-05:00</in>
                 <out>2014-04-18T16:00:00-05:00</out>
                 <weekEnd>2014-04-24T00:00:00-05:00</weekEnd>
                 <weekStart>2014-04-18T00:00:00-05:00</weekStart>
                </item>
                <item>
                 <empHSId>REDACTED</empHSId>
                 <empPosId>REDACTED</empPosId>
                 <jobHsId>837906321</jobHsId>
                 <jobPosId>600</jobPosId>
                 <in>2014-04-18T06:00:00-05:00</in>
                 <out>2014-04-18T14:00:00-05:00</out>
                 <weekEnd>2014-04-24T00:00:00-05:00</weekEnd>
                 <weekStart>2014-04-18T00:00:00-05:00</weekStart>
                </item>
          </return>
       </ns1:getScheduleResponse>
    </soap:Body>
</soap:Envelope>
```

Method: getSchedule2

Description

This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem2 objects, which represent one scheduled shift each, for import into the POS. This method returns the same data as getSchedule, plus extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
start	dateTime	1..1	First day of scheduled shifts you are requesting.
end	dateTime	1..1	Last day of scheduled shifts you are requesting.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsScheduleItem2Array	1..1	Array of WSScheduleItem2 objects, which represent one scheduled shift each.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
         <wsse:UsernameToken>
            <wsse:Username>REDACTED</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </SOAP-ENV:Header>
   <ns1:Body>
        <ns2:getSchedule2>
         <concept>1</concept>
         <storeNum>37</storeNum>
         <start>2014-04-18T00:00:00</start>
         <end>2014-04-18T00:00:00</end>
      </ns2:getSchedule2>
   </ns1:Body>
</SOAP-ENV:Envelope>
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
   <soap:Body>
         <ns1:getSchedule2Response>
          <return>
                <item>
                 <empHSId>REDACTED</empHSId>
                 <empPosId>REDACTED</empPosId>
                 <jobHsId>2129482342</jobHsId>
                 <jobPosId>453</jobPosId>
                 <in>2014-04-18T05:00:00-05:00</in>
                 <out>2014-04-18T16:00:00-05:00</out>
                 <weekEnd>2014-04-24T00:00:00-05:00</weekEnd>
                 <weekStart>2014-04-18T00:00:00-05:00</weekStart>
                 <locationId>578176916</locationId>
                 <ovtMinutes>0</ovtMinutes>
                 <ovtRate>0.0</ovtRate>
                 <payRate>0.0</payRate>
                 <regMinutes>660</regMinutes>
                 <scheduleId>2129477321</scheduleId>
                 <specialPay>0.0</specialPay>
                </item>
                <item>
                 <empHSId>REDACTED</empHSId>
                 <empPosId>REDACTED</empPosId>
                 <jobHsId>837906321</jobHsId>
                 <jobPosId>600</jobPosId>
                 <in>2014-04-18T06:00:00-05:00</in>
                 <out>2014-04-18T14:00:00-05:00</out>
                 <weekEnd>2014-04-24T00:00:00-05:00</weekEnd>
                 <weekStart>2014-04-18T00:00:00-05:00</weekStart>
                 <locationId>-1</locationId>
                 <ovtMinutes>0</ovtMinutes>
                 <ovtRate>0.0</ovtRate>
                 <payRate>24.0</payRate>
                 <regMinutes>480</regMinutes>
                 <scheduleId>2129477321</scheduleId>
                 <specialPay>0.0</specialPay>
              </item>
           </return>
       </ns1:getSchedule2Response>
      </soap:Body>
  </soap:Envelope>
```

Method: getScheduleV3

Description

This method takes in a concept ID, store ID, start and end dates. It returns an array of WSScheduleItem3 objects, which represent one scheduled shift each, for import into the POS. This method returns the same data as getSchedule, plus extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS.

This method uses hsSimpleDate objects for dates.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
start	hsSimpleDate	1..1	First day of scheduled shifts you are requesting. This method uses hsSimpleDate objects for dates.
end	hsSimpleDate	1..1	Last day of scheduled shifts you are requesting. This method uses hsSimpleDate objects for dates.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsScheduleItem3Array	1..1	Array of WSScheduleItem3 objects, which represent one scheduled shift each. Dates are hsSimpleDate objects and times are hsSimpleTime objects.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns2="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
            <wsse:UsernameToken>
             <wsse:Username>REDACTED</wsse:Username>
               <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </SOAP-ENV:Header>
   <ns1:Body>
      <ns2:getScheduleV3>
            <concept>1</concept>
         <storeNum>37</storeNum>
         <start>
               <day>18</day>
            <month>4</month>
            <year>2014</year>
         </start>
         <end>
               <day>18</day>
            <month>4</month>
            <year>2014</year>
         </end>
      </ns2:getScheduleV3>
   </ns1:Body>
</SOAP-ENV:Envelope> SAMPLE
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
    <soap:Body>lOCA
        <ns1:getScheduleV3Response>
         <return>
            <item>
                <empHSId>1712094</empHSId>
                <empPosId>190</empPosId>
                <jobHsId>489527461</jobHsId>
                <jobPosId>120000</jobPosId>
                <inDate>
                    <day>18</day>
                    <month>4</month>
                    <year>2014</year>
                </inDate>
                <inTime>
                    <hours>6</hours>
                    <militaryTime>true</militaryTime>
                    <minutes>30</minutes>
                    <seconds>0</seconds>
                </inTime>
                <locationId>-1</locationId>
                <outDate>
                    <day>18</day>
                    <month>4</month>
                    <year>2014</year>
                </outDate>
                <outTime>
                    <hours>15</hours>
                    <militaryTime>true</militaryTime>
                    <minutes>0</minutes>
                    <seconds>0</seconds>
                </outTime>
                <ovtMinutes>0</ovtMinutes>
                <ovtRate>0.0</ovtRate>
                <payRate>0.0</payRate>
                <regMinutes>0</regMinutes>
                <scheduleId>-1</scheduleId>
                <specialPay>0.0</specialPay>
                <weekEnd>
                    <day>24</day>
                    <month>4</month>
                    <year>2014</year>
                </weekEnd>
                <weekStart>
                    <day>18</day>
                    <month>4</month>
                    <year>2014</year>
                </weekStart>
```

```
            </item>
          </return>
        </ns1:getScheduleV3Response>
    </soap:Body>
</soap:Envelope>
```

Method: getShiftsV3

Description

This method takes in a concept ID, store ID, start and end dates and three flags (isHouse, isScheduled and isPosted). It returns an array of WSScheduleItem3 objects, which represent one scheduled shift each. What shifts are returned depends on the flags set:

isHouse - includes scheduled or posted shifts that are not assigned to an employee (called 'house shifts' in HotSchedules).

isScheduled - includes shifts that are in schedules that have been saved in HotSchedules, but might not have been posted

isPosted - includes shifts that are in schedules that have been set to the 'posted' status within HotSchedules

This method returns extended scheduled shift data, including location ID, regular pay rate, OT pay rate, scheduled regular minutes, scheduled OT minutes (if any), scheduled special pay (if any) and the unique schedule ID, internal to HS.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
day	int	1..1	Day formatted dd
month	int	1..1	Month formatted mm
year	int	1..1	Year formatted yyyy
isHouse	boolean	1..1	Include house shifts? These are shifts which were never assigned to someone on a schedule, or were once assigned but removed from an employee when their status, job, etc. changed.

isScheduled	boolean	1..1	Include scheduled shifts? These are shifts which are on a schedule that has been saved in HotSchedules, but might not have been posted
isPosted	boolean	1..1	These are shifts that are in schedules that have been set to the 'posted' status within HotSchedules.
jobCodes	intArray	1..1	Integer array of job codes to be included in this method's return. if this parameter is null or empty, all jobs will be included.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsScheduleItem3Array	1..1	Array of wsScheduleItem3 objects, which represent all shifts that match the criteria in the method call

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sch="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
      <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
         <wsse:UsernameToken wsu:Id="UsernameToken-5A4A0145444B759EC8145271260874466">
            <wsse:Username>laura1234!</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
            <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">m2/v7Yt2bmHeTDmYvu85CA==</wsse:Nonce>
            <wsu:Created>2016-01-13T19:16:48.744Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
  <soapenv:Body>
     <sch:getShiftsV3>
        <concept>1</concept>
        <storeNum>1</storeNum>
        <start>
           <day>1</day>
           <month>1</month>
           <year>2016</year>
        </start>
        <end>
           <day>9</day>
           <month>2</month>
           <year>2016</year>
        </end>
        <isHouse>false</isHouse>
        <isScheduled>true</isScheduled>
        <isPosted>true</isPosted>
        <jobCodes>
<!--Zero or more repetitions:
  <item>871507284</item>-->
        </jobCodes>
```

```
    </sch:getShiftsV3>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getShiftsV3Response xmlns:ns1="http://services.hotschedules.com/api/services/ScheduleService">
      <return>
        <item>
          <empHSId>17896647</empHSId>
          <empPosId>1</empPosId>
          <jobHsId>964330045</jobHsId>
          <jobPosId>-1</jobPosId>
          <inDate>
            <day>11</day>
            <month>1</month>
            <year>2016</year>
          </inDate>
          <inTime>
            <hours>7</hours>
            <militaryTime>true</militaryTime>
            <minutes>0</minutes>
            <seconds>0</seconds>
          </inTime>
          <locationId>-1</locationId>
          <outDate>
            <day>11</day>
            <month>1</month>
            <year>2016</year>
          </outDate>
          <outTime>
            <hours>15</hours>
            <militaryTime>true</militaryTime>
            <minutes>0</minutes>
            <seconds>0</seconds>
          </outTime>
          <ovtMinutes>0</ovtMinutes>
          <ovtRate>0.0</ovtRate>
          <payRate>9.5</payRate>
          <regMinutes>480</regMinutes>
          <scheduleId>964330041</scheduleId>
          <specialPay>0.0</specialPay>
          <weekEnd>
```

```
            <day>16</day>
            <month>1</month>
            <year>2016</year>
          </weekEnd>
          <weekStart>
            <day>10</day>
            <month>1</month>
            <year>2016</year>
          </weekStart>
        </item>
                  </return>
    </ns1:getShiftsV3Response>
  </soap:Body>
</soap:Envelope>
```

Method: sayHello

Description

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	string	1..1	Missing description.

Method: getTemplates

Description

This method takes in a concept ID, store ID, start and end dates. This method returns template information.
Action

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
day	int	1..1	Day formatted dd
month	int	1..1	Month formatted mm
year	int	1..1	Year formatted yyyy

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return			

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sch="http://services.hotschedules.com/api/services/ScheduleService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262988637423">
        <wsse:Username>laura1234!</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
        <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">WjFLOOT1CCJWLffHSRdCnQ==</wsse:Nonce>
        <wsu:Created>2016-01-12T20:18:06.374Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <sch:getTemplates>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <start>
        <day>1</day>
        <month>1</month>
        <year>2016</year>
      </start>
      <end>
        <day>20</day>
        <month>1</month>
        <year>2016</year>
      </end>
    </sch:getTemplates>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

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

```
  <soap:Body>
    <ns1:getTemplatesResponse xmlns:ns1="http://services.hotschedules.com/api/services/ScheduleService">
      <return>
        <item>
          <durationMinutes>360</durationMinutes>
          <endTime>
            <hours>16</hours>
            <militaryTime>true</militaryTime>
            <minutes>0</minutes>
            <seconds>0</seconds>
          </endTime>
          <jobCode>-1</jobCode>
          <jobName>Baker</jobName>
          <locationId>-1</locationId>
          <locationName>NO_LOC</locationName>
          <scheduleId>964330043</scheduleId>
          <scheduleName>Kitchen</scheduleName>
          <shiftName>Kitchen</shiftName>
          <startTime>
            <hours>10</hours>
            <militaryTime>true</militaryTime>
            <minutes>0</minutes>
            <seconds>0</seconds>
          </startTime>
          <templateDesc>Low Volume</templateDesc>
          <templateGroupId>-1</templateGroupId>
          <templateGroupName>NOT_GROUPED</templateGroupName>
          <templateId>964318045</templateId>
          <templateName>Low Volume</templateName>
          <weekday>6</weekday>
          <weekdayName>Friday</weekdayName>
        </item>
 </return>
    </ns1:getTemplatesResponse>
  </soap:Body>
</soap:Envelope>
```

Complex Types: ScheduleService

Complex Types

Name	Description
dataLocation	Contains information for a single HotSchedules schedule location, including name, internal HS ID and a 'disabled' flag.
dataScheduleItem	Contains individual shift information (store ID, employee ID, job code)
hsSimpleDate	Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.
hsSimpleTime	Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.
intArray	Missing description.
wsScheduleItem	Extends dataScheduleItem (contains clock in and clock out date/times)
wsScheduleItem2	extends wsScheduleItem (includes location ID, regular and overtime minutes, and rates, special pay and schedule ID)
wsScheduleItem3	Extends dataScheduleItem (contains clock in and clock out date/times as hsSimpleDate and hsSimpleTime objects, regular and overtime minutes and rates, special pay internal HS ID for the schedule and work week start and end dates as hsSimpleDate objects)

Contains information for a single HotSchedules schedule location, including name, internal HS ID and a 'disabled' flag.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
disabled	boolean	1..1	No	Missing description.
id	int	1..1	No	Missing description.
name	string	0..1	No	Missing description.

Referenced By

- Element item [type dataLocationArray]

Contains individual shift information (store ID, employee ID, job code)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
empHSId	int	1..1	No	Optional-Internal HotSchedules Employee Account ID
empPosId	int	1..1	No	POS numeric employee ID
jobHsId	int	1..1	No	Optional-Internal HotSchedules Job Code ID
jobPosId	int	1..1	No	POS numeric job code ID

Referenced By

- Complex Type wsScheduleItem
- Complex Type wsScheduleItem3

Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Day formatted dd
month	int	1..1	No	Month formatted mm
year	int	1..1	No	Year formatted yyyy

Referenced By

- Element inDate [type wsScheduleItem3]
- Element outDate [type wsScheduleItem3]
- Element weekEnd [type wsScheduleItem3]
- Element weekStart [type wsScheduleItem3]

Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Element inTime [type wsScheduleItem3]
- Element outTime [type wsScheduleItem3]

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
item	int	0..*	Yes	Missing description.

Extends dataScheduleItem (contains clock in and clock out date/times)

Derived By

Extending dataScheduleItem

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
empHSId	int	1..1	No	Optional-HotSchedules Internal employee account ID
empPosId	int	1..1	No	POS numeric employee ID
jobHsId	int	1..1	No	Optional-HotSchedules Internal job code ID
jobPosId	int	1..1	No	POS numeric job code ID
in	dateTime	0..1	No	Scheduled in time and date
out	dateTime	0..1	No	Scheduled out time and date
weekEnd	dateTime	0..1	No	Week End Date
weekStart	dateTime	0..1	No	Week Start Date

Referenced By

- Element item [type wsScheduleItemArray]
- Complex Type wsScheduleItem2

extends wsScheduleItem (includes location ID, regular and overtime minutes, and rates, special pay and schedule ID)

Derived By

Extending wsScheduleItem

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
empHSId	int	1..1	No	HotSchedules internal employee account ID
empPosId	int	1..1	No	POS numeric employee ID
jobHsId	int	1..1	No	HotSchedules internal job code ID
jobPosId	int	1..1	No	POS numeric job code ID
in	dateTime	0..1	No	Scheduled in time and date
out	dateTime	0..1	No	Schedules out time and date
weekEnd	dateTime	0..1	No	Schedule Week end date
weekStart	dateTime	0..1	No	Schedule Week start date
locationId	int	1..1	No	Numeric ID for job location. Location represents more granular break down of job codes into sections or positions.
ovtMinutes	int	1..1	No	Total Overtime Minutes for employee
ovtRate	float	1..1	No	Overtime hourly rate for employee
payRate	float	1..1	No	Regular hourly pay rate for employee
regMinutes	int	1..1	No	Regular minutes for shift
scheduleId	int	1..1	No	HotSchedules unique ID to identify schedule
specialPay	float	1..1	No	Special Pay amounts

Referenced By

- Element item [type wsScheduleItem2Array]

Extends dataScheduleItem (contains clock in and clock out date/times as hsSimpleDate and hsSimpleTime objects, regular and overtime minutes and rates, special pay internal HS ID for the schedule and work week start and end dates as hsSimpleDate objects)

Derived By

Extending dataScheduleItem

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
empHSId	int	1..1	No	HotSchedules internal employee account ID
empPosId	int	1..1	No	POS numeric employee ID
jobHsId	int	1..1	No	HotSchedules internal job code ID
jobPosId	int	1..1	No	POS numeric job code ID
inDate	hsSimpleDate	0..1	No	Scheduled in date
inTime	hsSimpleTime	0..1	No	Scheduled in time
locationId	int	1..1	No	Numeric ID for job location. Location represents more granular break down of job codes into sections or positions.
outDate	hsSimpleDate	0..1	No	Schedules out date
outTime	hsSimpleTime	0..1	No	Schedules out time
ovtMinutes	int	1..1	No	Total Overtime Minutes for employee
ovtRate	float	1..1	No	Overtime hourly rate for employee
payRate	float	1..1	No	Regular hourly pay rate for employee
regMinutes	int	1..1	No	Total Regular Minutes for employee
scheduleId	int	1..1	No	HotSchedules unique ID to identify schedule
specialPay	float	1..1	No	Special Pay amounts
weekEnd	hsSimpleDate	0..1	No	Schedule Week end date
weekStart	hsSimpleDate	0..1	No	Schedule Week start date

Referenced By

- Element item [type wsScheduleItem3Array]

TimeCardService Web Service

Description

This service is intended for third parties to be able to import their time cards into the HotSchedules system or get time cards from HotSchedules in a straightforward fashion. The WSDL is available at:

https://services.hotschedules.com/api/services/TimeCardService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods: TimeCardService

Methods

Name	Description
getTimeCards	This method takes in a concept ID, store ID, start and end dates. It returns an array of wsTimeCard3 objects, which represent one employee time card each. Each time card has information for one employee punch record, including business date, regular and OT minutes and wages, clock in and clock out times. If this store is using HotSchedules' web-based timeclock for employee clock-in, any open punches in the date range are also included in the response.
getTimecardsDeclaredTips	This method takes in a concept ID, store ID, start and end dates. It returns an array of wsTimeCardsDeclaredTips objects, which represent one employee time card each including Declared Tips. Each time card has information for one employee punch record, including business date, regular and OT minutes and wages, clock in, clock out times and declared tips. If this store is using HotSchedules' web-based timeclock for employee clock-in, any open punches in the date range are also included in the response.
sayHello	Missing description.
setTimeCards	This method takes in a concept ID, store ID, a start and end date and an array of WSTimeCard objects. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains time cards for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the time cards are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.
setTimeCardsDeclaredTips	This method takes in a concept ID, store ID, a start and end date and an array of WSTimeCardsDeclaredTips objects. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains time cards for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the time cards are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.

setTimeCardsV3	This method takes in a concept ID, store ID, a start and end date and an array of WSTimeCard objects. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains time cards for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the time cards are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object. This method uses hsSimpleDate objects for dates.

Method: getTimeCards

Description

This method takes in a concept ID, store ID, start and end dates. It returns an array of wsTimeCard3 objects, which represent one employee time card each. Each time card has information for one employee punch record, including business date, regular and OT minutes and wages, clock in and clock out times. If this store is using HotSchedules' web-based timeclock for employee clock-in, any open punches in the date range are also included in the response.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
day	int	1..1	Day formatted dd
Month	int	1..1	Month formatted mm
year	int	1..1	Year formated yyyy

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsTimeCard3Array	1..1	Array of wsTimeCard3 objects. Each object has data for one employee punch record.

Faults

Name	Content	Description

Exception	Exception	Missing description.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://services.hotschedules.com/api/services/TimeCardService"
xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
      <wsse:Security mustUnderstand="true">
        <wsse:UsernameToken>
          <wsse:Username>REDACTED</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
        </wsse:UsernameToken>
      </wsse:Security>
  </SOAP-ENV:Header>
  <ns2:Body>
    <ns1:getTimeCards>
      <concept>1</concept>
      <storeNum>103</storeNum>
      <start>
        <day>25</day>
        <month>2</month>
        <year>2014</year>
      </start>
      <end>
        <day>25</day>
        <month>2</month>
        <year>2014</year>
      </end>
    </ns1:getTimeCards>
  </ns2:Body>
</SOAP-ENV:Envelope> SAMPLE
```

RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
   <soap:Body>
      <ns1:getTimeCardsResponse>
       <return>
          <item>
             <breakMinutes>0</breakMinutes>
             <empPosId>210</empPosId>
             <extId>156393867196</extId>
             <hsId>156393867197</hsId>
             <jobExtId>12</jobExtId>
             <jobName>Expediter</jobName>
             <ovtHrs>0.0</ovtHrs>
             <ovtMins>0</ovtMins>
             <ovtTtl>0.0</ovtTtl>
             <ovtWage>0.0</ovtWage>
             <regHrs>4.6833334</regHrs>
             <regTtl>0.0</regTtl>
             <regWage>8.0</regWage>
             <spcHrs>0.0</spcHrs>
             <spcTtl>0.0</spcTtl>
             <storeNum>103</storeNum>
             <businessDate>
                <day>25</day>
                <month>2</month>
                <year>2014</year>
             </businessDate>
             <clockIn>2014-02-25T10:26:00-06:00</clockIn>
             <clockOut>2014-02-25T15:07:00-06:00</clockOut>
          </item>
          <item>
             <breakMinutes>0</breakMinutes>
             <empPosId>207</empPosId>
             <extId>156393867200</extId>
             <hsId>156393867201</hsId>
             <jobExtId>12</jobExtId>
             <jobName>Expediter</jobName>
             <ovtHrs>0.0</ovtHrs>
             <ovtMins>0</ovtMins>
             <ovtTtl>0.0</ovtTtl>
             <ovtWage>0.0</ovtWage>
             <regHrs>4.983333</regHrs>
             <regTtl>0.0</regTtl>
             <regWage>8.0</regWage>
             <spcHrs>0.0</spcHrs>
             <spcTtl>0.0</spcTtl>
             <storeNum>103</storeNum>
             <businessDate>
                <day>25</day>
```

```
                    <month>2</month>
                    <year>2014</year>
                </businessDate>
                <clockIn>2014-02-25T16:28:00-06:00</clockIn>
                <clockOut>2014-02-25T21:27:00-06:00</clockOut>
            </item>
        </return>
    </ns1:getTimeCardsResponse>
  </soap:Body>
</soap:Envelope>
```

Method: getTimeCardsDeclaredTips

Description

This method takes in a concept ID, store ID, start and end dates. It returns an array of wsTimeCard3 objects, which represent one employee time card each. Each time card has information for one employee punch record, including business date, regular and OT minutes and wages, clock in, clock out times and declared tips. If this store is using HotSchedules' web-based timeclock for employee clock-in, any open punches in the date range are also included in the response.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
day	int	1..1	Day formatted dd
Month	int	1..1	Month formatted mm
year	int	1..1	Year formated yyyy

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsTimeCard3Array	1..1	Array of wsTimeCard3 objects. Each object has data for one employee punch record.

Faults

Name	Content	Description

Exception	Exception	Missing description.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tim="http://services.hotschedules.com/api/services/TimeCardService">
  <soapenv:Header>
      <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
         <wsse:UsernameToken wsu:Id="UsernameToken-5A4A0145444B759EC8145271596284975">
            <wsse:Username>laura1234!</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laur
a1234!</wsse:Password>
            <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bin
ary">mafwHuh4YzpETuPwHxlC2g==</wsse:Nonce>
            <wsu:Created>2016-01-13T20:12:42.849Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
  <soapenv:Body>
    <tim:getTimeCardsDeclaredTips>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <start>
        <day>1</day>
        <month>1</month>
        <year>2016</year>
      </start>
      <end>
        <day>10</day>
        <month>1</month>
        <year>2016</year>
      </end>
    </tim:getTimeCardsDeclaredTips>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getTimeCardsDeclaredTipsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/TimeCardService">
      <return>
        <item>
          <breakMinutes>0</breakMinutes>
          <empPosId>2</empPosId>
          <extId>844268158324</extId>
          <hsId>844268158325</hsId>
          <jobExtId>-1</jobExtId>
          <jobId>-1</jobId>
          <jobName>Baker</jobName>
          <ovtHrs>1.0</ovtHrs>
          <ovtMins>0</ovtMins>
          <ovtTtl>0.0</ovtTtl>
          <ovtWage>0.0</ovtWage>
          <regHrs>8.0</regHrs>
          <regTtl>0.0</regTtl>
          <regWage>0.0</regWage>
          <spcHrs>0.0</spcHrs>
          <spcTtl>0.0</spcTtl>
          <storeNum>1</storeNum>
          <businessDate>
            <day>10</day>
            <month>1</month>
            <year>2016</year>
          </businessDate>
          <clockIn>2016-01-10T15:00:00-06:00</clockIn>
          <clockOut>2016-01-11T00:00:00-06:00</clockOut>
          <declaredTips>0.0</declaredTips>
        </item>
       </return>
    </ns1:getTimeCardsDeclaredTipsResponse>
```

```
  </soap:Body>
</soap:Envelope>
```

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
cards	wsTimeCardArray	1..1	Array of WSTimeCard objects. Each object represents one time card/punch record at this store.
start	dateTime	1..1	Business date of the first sales item in the array. This is a basic dateTime object.
end	dateTime	1..1	Business date of the last sales item in the array. This is a basic dateTime object.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description

Exception	Exception	Missing description.

This method uses hsSimpleDate objects for dates.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
cards	wsTimeCard3Array	1..1	Array of WSTimeCard objects. Each object represents one time card/punch record at this store.
start	hsSimpleDate	1..1	Business date of the first sales item in the array. This method uses hsSimpleDate objects for dates.
end	hsSimpleDate	1..1	Business date of the last sales item in the array. This method uses hsSimpleDate objects for dates.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description
Exception	Exception	Missing description.

Name	Description
dataTimeCard	Contains time card data (employee ID, store ID, job ID and name, regular hours or minutes, OT hours or minutes, total regular wages, total OT wages, break minutes, special pay data)
hsSimpleDate	Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.
wsReturn	Contains array of error strings, fail count, success flag, success count.
wsTimeCard	Extends dataTimeCard (includes business date and clock in and clockout date/times)
wsTimeCardDeclaredTips	Extends dataTimeCard (includes business date and hsSimpleDate object and clock in and clockout date/times as hsSimpleDate and hsSimpleTime objects)
wsTimeCard3	Extends dataTimeCard (includes business date and hsSimpleDate object and clock in and clockout date/times as hsSimpleDate and hsSimpleTime objects)

wsTimeCard and wsTimeCard3 Array

Contains time card data (employee ID, store ID, job ID and name, regular hours or minutes, OT hours or minutes, total regular wages, total OT wages, break minutes, special pay data)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
breakMinutes	long	1..1	No	Number of non-paid break minutes in shift
empPosId	int	1..1	No	POS numeric Employee ID
extId	long	1..1	No	Unique Transaction ID for the time card record
hsId	long	1..1	No	Optional-Internal HotSchedules employee Account ID
jobExtId	int	1..1	No	POS numeric Job Code ID
jobName	string	0..1	No	POS job code name
ovtHrs	float	1..1	No	Overtime hours in shift
ovtMins	int	1..1	No	Overtime minutes in shift
ovtTtl	float	1..1	No	Overtime total pay amount
ovtWage	float	1..1	No	Overtime hourly pay rate
regHrs	float	1..1	No	Regular hours represented in shift
regTtl	float	1..1	No	Regular total pay amount
regWage	float	1..1	No	Regular hourly pay rate
spcHrs	float	1..1	No	Optional-Special Pay Hours
spcTtl	float	1..1	No	Optional-Special Pay total pay amount
storeNum	int	1..1	No	Unique numeric store identifier. Generally set up to mirror the client internal store ID.

Referenced By

- Complex Type wsTimeCard

- Complex Type wsTimeCard3

Remarks

Time amounts can be expressed in hours or minutes, but both not are not needed. Use the one that best matches your source data.

Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Formatted dd
month	int	1..1	No	Formatted mm
year	int	1..1	No	Formatted yyyy

Referenced By

- Element businessDate [type wsTimeCard3]

Contains array of error strings, fail count, success flag, success count.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
errors	string	0..*	Yes	Missing description.
failCount	int	1..1	No	Number of records that failed to meet basic import specifications
success	boolean	1..1	No	Indicator of the success or failure
successCount	int	1..1	No	Total number of records that successfully met basic import specifications

Method: SetTimecardsV3

Extends dataTimeCard (includes business date and clock in and clockout date/times)

Derived By

Extending dataTimeCard

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
breakMinutes	long	1..1	No	Number of non-paid break minutes in shift
empPosId	int	1..1	No	POS numeric Employee ID
extId	long	1..1	No	Optional-Unique transaction ID for the time card record
hsId	long	1..1	No	Optional-Internal HotSchedules employee Account ID. Not required, will be set by the service.
jobExtId	int	1..1	No	POS numeric Job Code ID
jobName	string	0..1	No	POS job code name
ovtHrs	float	1..1	No	Optional-Overtime hours in shift
ovtMins	int	1..1	No	Optional-Overtime minutes in shift
ovtTtl	float	1..1	No	Optional-Overtime total pay amount
ovtWage	float	1..1	No	Optional-Overtime hourly pay rate
regHrs	float	1..1	No	Regular hours represented in shift
regTtl	float	1..1	No	Regular total pay amount
regWage	float	1..1	No	Regular hourly pay rate
spcHrs	float	1..1	No	Optional-Special Pay Hours
spcTtl	float	1..1	No	Optional-Special Pay total pay amount
storeNum	int	1..1	No	Unique numeric store identifier. Generally set up to mirror the client internal store ID.
businessDate	dateTime	0..1	No	Date of business the timecard applied to

clockIn	dateTime	0..1	No	Clock in timestamp for the shift
clockOut	dateTime	0..1	No	Clock out timestamp for the shift

Referenced By

- Element item [type wsTimeCardArray]

Description

Extends dataTimeCard (includes business date and hsSimpleDate object and clock in and clockout date/times as hsSimpleDate and hsSimpleTime objects)

Derived By

Extending dataTimeCard

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
breakMinutes	long	1..1	No	Number of non-paid break minutes in shift
empPosId	int	1..1	No	POS numeric Employee ID
extId	long	1..1	No	Optional-Unique transaction ID for time card record
hsId	long	1..1	No	Optional-Internal HotSchedules employee Account ID
jobExtId	int	1..1	No	POS numeric Job Code ID
jobName	string	0..1	No	POS job code name
ovtHrs	float	1..1	No	Optional-Overtime hours in shift
ovtMins	int	1..1	No	Optional-Overtime minutes in shift
ovtTtl	float	1..1	No	Optional-Overtime total pay amount
ovtWage	float	1..1	No	Optional-Overtime hourly pay rate
regHrs	float	1..1	No	Regular hours represented in shift
regTtl	float	1..1	No	Regular total pay amount
regWage	float	1..1	No	Regular hourly pay rate
spcHrs	float	1..1	No	Optional-Special Pay Hours
spcTtl	float	1..1	No	Optional-Special Pay total pay amount
storeNum	int	1..1	No	Unique numeric store identifier. Generally set up to mirror the client internal store ID.
businessDate	hsSimpleDate	0..1	No	Date of business the timecard applied to
clockIn	dateTime	0..1	No	Clock in timestamp for the shift
clockOut	dateTime	0..1	No	Clock out timestamp for the shift

Referenced By

Element item [type wsTimeCard3Array]

SalesItemService Web Service

Description

This service is intended for third parties to be able to import their sales items into the HotSchedules system in a straightforward fashion. Currently there is only one method for setting sales items. The WSDL is available at:

http://services.hotschedules.com/api/services/SalesService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods: SalesItemService

Methods

Name	Description
getRVCs	This method takes in a concept ID and a store ID and returns an array of revenue centers for that store. Revenue centers will typically establish where to attribute a sale: Bar, Dining, To go, etc.Revenue centers can be local to a particular store, or defined in HotSchedules as belonging to an entire group (called group-level RVCs)
getSalesCats	This method takes in a concept ID and a store ID and returns an array of sales categories for that store. Sales categories will typically establish what kind of item was sold: Food, Beverage, Alcohol, Merchandise.Sales categories can be local to a particular store, or defined in HotSchedules as belonging to an entire group (called group-level sales categories)
getSalesItemsV3	This method takes in a concept ID, store ID and returns an array of sales items for that store. Sales items indicated which item was sold.
sayHello	Missing description.
setSalesItems	This method takes in a concept ID, store ID, a start and end date and an array of WSSalesItem objects. Using the authentication from the username token and the conecpt and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains sales items for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of sales, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the sales items are already in the HS database and do not need to be updated, then nothing will change. The method returns a WSReturn object.
setSalesItemsV3	This method takes in a concept ID, store ID, a start and end date and an array of WSSalesItem objects. Using the authentication from the username token and the conecpt and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains sales items for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of sales, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the sales items are already in the HS database and do not need to be updated, then nothing will change. The method returns a WSReturn object. This method uses hsSimpleDate objects for dates.

Method: getRVCs

Description

This method takes in a concept ID and a store ID and returns an array of revenue centers for that store.

Revenue centers will typically establish where to attribute a sale: Bar, Dining, To go, etc.Revenue centers can be local to a particular store, or defined in HotSchedules as belonging to an entire group (called group-level RVCs)

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsRevenueCenterArray	1..1	Array of wsRevenueCenter objects, which represents all revenue centers defined for that store. This will include store and group level revenue centers, and will be flagged appropriately.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Sample Calls

REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sal="http://services.hotschedules.com/api/services/SalesService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
      <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
         <wsse:UsernameToken wsu:Id="UsernameToken-5A4A0145444B759EC8145272034993393">
            <wsse:Username>HSAPIUser</wsse:Username>
            <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">ao24
wO2n8gkh7lp</wsse:Password>
            <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">eup88j1y/qnxGKg2Rn6J4A==</wsse:Nonce>
            <wsu:Created>2016-01-13T21:25:49.933Z</wsu:Created>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
  <soapenv:Body>
    <sal:getRVCs>
      <concept>101</concept>
      <storeNum>111</storeNum>
    </sal:getRVCs>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getRVCsResponse xmlns:ns1="http://services.hotschedules.com/api/services/SalesService">
      <return>
        <item>
          <extId>10</extId>
          <groupLevel>true</groupLevel>
          <revenueCenterName>Dine in</revenueCenterName>
        </item>
        <item>
          <extId>17</extId>
          <groupLevel>true</groupLevel>
          <revenueCenterName>Bar</revenueCenterName>
```

```
          </item>
          <item>
            <extId>15</extId>
            <groupLevel>true</groupLevel>
            <revenueCenterName>To go</revenueCenterName>
          </item>
        </return>
      </ns1:getRVCsResponse>
    </soap:Body>
</soap:Envelope>
```

Method: getSalesCats

Description

This method takes in a concept ID and a store ID and returns an array of sales categories for that store. Sales categories will typically establish what kind of item was sold: Food, Beverage, Alcohol, Merchandise.Sales categories can be local to a particular store, or defined in HotSchedules as belonging to an entire group (called group-level sales categories)

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsSalesCategoryArray	1..1	Array of wsSalesCategory objects, which represents all sales categories defined for that store. This will include store and group level sales categories, and will be flagged appropriately.

Faults

Name	Content	Description
Exception	Exception	Missing description.

REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sal="http://services.hotschedules.com/api/services/SalesService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-5A4A0145444B759EC8145272075572695">
          <wsse:Username>HSAPIUser</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">ao24
wO2n8gkh7lp</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">HWXdmwP4l8qNEDjt4a0G9Q==</wsse:Nonce>
          <wsu:Created>2016-01-13T21:32:35.726Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <sal:getSalesCats>
      <concept>101</concept>
      <storeNum>111</storeNum>
    </sal:getSalesCats>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getSalesCatsResponse xmlns:ns1="http://services.hotschedules.com/api/services/SalesService">
      <return>
        <item>
          <extId>10</extId>
          <groupLevel>true</groupLevel>
          <salesCategoryName>Food</salesCategoryName>
        </item>
        <item>
          <extId>20</extId>
          <groupLevel>true</groupLevel>
          <salesCategoryName>Beverages</salesCategoryName>
        </item>
```

```
          <item>
            <extId>30</extId>
            <groupLevel>true</groupLevel>
            <salesCategoryName>Beer</salesCategoryName>
          </item>
          <item>
            <extId>40</extId>
            <groupLevel>true</groupLevel>
            <salesCategoryName>Wine</salesCategoryName>
          </item>
        </return>
      </ns1:getSalesCatsResponse>
    </soap:Body>
</soap:Envelope>
```

Method: getSalesItemsV3

Description

This method takes in a concept ID, store ID, start and end dates.It returns an array of sales for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
day	int	1..1	Formatted dd
month	int	1..1	Formatted mm
year	int	1..1	Formatted yyyy

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsSalesItem3Array	1..1	Missing description.

Faults

Name	Content	Description
Exception	Exception	Missing description.

Example

REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sal="http://services.hotschedules.com/api/services/SalesService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
```

```
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
          <wsse:UsernameToken wsu:Id="UsernameToken-5A4A0145444B759EC81452722705669100">
             <wsse:Username>HSAPIUser</wsse:Username>
             <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">ao24
wO2n8gkh7lp</wsse:Password>
             <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">WHQ2rdJcHsI92WDsEaBDpA==</wsse:Nonce>
             <wsu:Created>2016-01-13T22:05:05.669Z</wsu:Created>
          </wsse:UsernameToken>
       </wsse:Security>
    </soapenv:Header>
   <soapenv:Body>
      <sal:getSalesItemsV3>
        <concept>101</concept>
        <storeNum>111</storeNum>
        <start>
          <day>1</day>
          <month>1</month>
          <year>2015</year>
        </start>
        <end>
          <day>1</day>
          <month>2</month>
          <year>2015</year>
        </end>
      </sal:getSalesItemsV3>
   </soapenv:Body>
</soapenv:Envelope>

RESPONSE
          <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:getSalesItemsV3Response xmlns:ns1="http://services.hotschedules.com/api/services/SalesService">
        <return>
<item>
             <clientId>16477839</clientId>
             <empId>-1</empId>
             <extId>-1</extId>
             <rvc>-1</rvc>
             <salesCat>-1</salesCat>
             <storeNum>111</storeNum>
             <ttl>13.45</ttl>
```

```
            <businessDate>
              <day>3</day>
              <month>1</month>
              <year>2015</year>
            </businessDate>
            <transDate>
              <day>4</day>
              <month>1</month>
              <year>2015</year>
            </transDate>
            <transTime>
              <hours>2</hours>
              <militaryTime>true</militaryTime>
              <minutes>28</minutes>
              <seconds>35</seconds>
            </transTime>
          </item>
          <item>
            <clientId>16477839</clientId>
            <empId>-1</empId>
            <extId>-1</extId>
            <rvc>-1</rvc>
            <salesCat>-1</salesCat>
            <storeNum>111</storeNum>
            <ttl>0.0</ttl>
            <businessDate>
              <day>3</day>
              <month>1</month>
              <year>2015</year>
            </businessDate>
            <transDate>
              <day>4</day>
              <month>1</month>
              <year>2015</year>
            </transDate>
            <transTime>
              <hours>2</hours>
              <militaryTime>true</militaryTime>
              <minutes>33</minutes>
              <seconds>19</seconds>
            </transTime>
          </item>
        </return>
      </ns1:getSalesItemsV3Response>
    </soap:Body>
</soap:Envelope>
```

Method: SetSalesItems

Description

This method takes in a concept ID, store ID, a start and end date and an array of WSSalesItem objects. If the sales items are already in the HS database and do not need to be updated, then nothing will change. The method returns a WSReturn object.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
sales	wsSalesItemArray	1..1	Array of WSSalesItem objects. Each object represents one sales item at this store.
start	dateTime	1..1	Business date of the first sales item in the array.
end	dateTime	1..1	Business date of the last sales item in the array.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description
Exception	Exception	Missing description.

Method: SetSalesItemsV3

Description

This method takes in a concept ID, store ID, a start and end date and an array of WSSalesItem objects. If the sales items are already in the HS database and do not need to be updated, then nothing will change. The method returns a WSReturn object.

This method uses hsSimpleDate objects for dates.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
sales	wsSalesItem3Array	1..1	Array of WSSalesItem objects. Each object represents one sales item at this store.
start	hsSimpleDate	1..1	Business date of the first sales item in the array. This method uses hsSimpleDate objects for dates.
end	hsSimpleDate	1..1	Business date of the last sales item in the array. This method uses hsSimpleDate objects for dates.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Faults

Name	Content	Description
Exception	Exception	Missing description.

Name	Description
dataSalesItem	Contains sales item data (sales category, revenue center, sales amount, ID of employee that made the sale, store ID where the sale ocurred)
hsSimpleDate	Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.
hsSimpleTime	Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.
wsReturn	Contains array of error strings, fail count, success flag, success count.
wsRevenueCenter	Missing description.
wsRevenueCenterArray	An array of wsRevenueCenterArray objects. Each wsRevenueCenter object describes the total revenue center sales for a given revenue center. Each wsRevenueCenter object contains - a RevenueCenter (which corresponds to the numeric ID that identifies this revenue center)
wsSalesCategory	Missing description.
wsSalesCategoryArray	Missing description.
wsSalesItem	Extends dataSalesItem (includes business date and calendar date of the sale)
wsSalesItem3	Extends dataTimeCard (includes business date and hsSimpleDate object and clock in and clockout date/times as hsSimpleDate and hsSimpleTime objects)

Contains sales item data (sales category, revenue center, sales amount, ID of employee that made the sale, store ID where the sale occurred)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
catName	string	0..1	No	Name of the category
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
empId	int	1..1	No	HotSchedules internal employee account ID
extId	int	1..1	No	Numeric ID from the external system associated with the transaction
rvc	int	1..1	No	Revenue Center ID
rvcName	string	0..1	No	Name of the revenue center
salesCat	int	1..1	No	Numeric (integer) identifier for the sales category
storeNum	int	1..1	No	Numeric (integer) identifier for the location. Must be unique within the concept.
ttl	float	1..1	No	Total of the sales category

Referenced By

- Complex Type wsSalesItem
- Complex Type wsSalesItem3

Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Formatted dd
month	int	1..1	No	Formatted mm
year	int	1..1	No	Formatted yyyy

Referenced By

- Element businessDate [type wsSalesItem3]
- Element transDate [type wsSalesItem3]

Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Element transTime [type wsSalesItem3]

Contains array of error strings, fail count, success flag, success count.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
errors	string	0..*	Yes	Missing description.
failCount	int	1..1	No	Number of records that failed to meet basic import specifications
success	boolean	1..1	No	Indicator of the success or failure
successCount	int	1..1	No	Total number of records that successfully met basic import specifications

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
extId	int	1..1	No	Numeric (integer) identifier for the location. Must be unique within the concept.
groupLevel	boolean	1..1	No	Missing description.
revenueCenterName	string	0..1	No	Missing description.

Referenced By

- Element item [type wsRevenueCenterArray]

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
item	wsRevenueCenter	0..*	Yes	Missing description.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
extId	int	1..1	No	Numeric ID from the external system associated with the transaction
groupLevel	boolean	1..1	No	Missing description.
salesCategoryName	string	0..1	No	Missing description.

Referenced By

- Element item [type wsSalesCategoryArray]

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
item	wsSalesCategory	0..*	Yes	Missing description.

Extends dataSalesItem (includes business date and calendar date of the sale)

Derived By

Extending dataSalesItem

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
catName	string	0..1	No	Missing description.
clientId	int	1..1	No	Missing description.
empId	int	1..1	No	Missing description.
extId	int	1..1	No	Missing description.
rvc	int	1..1	No	Missing description.
rvcName	string	0..1	No	Missing description.
salesCat	int	1..1	No	Missing description.
storeNum	int	1..1	No	Missing description.
ttl	float	1..1	No	Missing description.
businessDate	dateTime	0..1	No	Missing description.
dateTime	dateTime	0..1	No	Missing description.

Referenced By

- Element item [type wsSalesItemArray]

Description

Extends dataTimeCard (includes business date and hsSimpleDate object and clock in and clockout date/times as hsSimpleDate and hsSimpleTime objects)

Derived By

Extending dataSalesItem

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
catName	string	0..1	No	Name of the category
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
empId	int	1..1	No	HotSchedules internal employee account ID
extId	int	1..1	No	Numeric ID from the external system associated with the transaction
rvc	int	1..1	No	Revenue Center ID
rvcName	string	0..1	No	Name of the revenue center
salesCat	int	1..1	No	Numeric (integer) identifier for the sales category
storeNum	int	1..1	No	Numeric (integer) identifier for the location. Must be unique within the concept.
ttl	float	1..1	No	Total of the sales category
businessDate	hsSimpleDate	0..1	No	Business date of transaction
transDate	hsSimpleDate	0..1	No	Date of the transaction
transTime	hsSimpleTime	0..1	No	Time of the transaction

Referenced By

- Element item [type wsSalesItem3Array]

Elements: SalesItemService

Elements

Name	Description
amPm [type hsSimpleTime]	Missing description.
businessDate [type wsSalesItem]	Missing description.
businessDate [type wsSalesItem3]	Missing description.
catName [type dataSalesItem]	Missing description.
clientId [type dataSalesItem]	Missing description.
dateTime [type wsSalesItem]	Missing description.
day [type hsSimpleDate]	Missing description.
empId [type dataSalesItem]	Missing description.
errors [type wsReturn]	Missing description.
Exception	Missing description.
extId [type dataSalesItem]	Missing description.
extId [type wsRevenueCenter]	Missing description.
extId [type wsSalesCategory]	Missing description.
failCount [type wsReturn]	Missing description.
groupLevel [type wsRevenueCenter]	Missing description.
groupLevel [type wsSalesCategory]	Missing description.
hours [type hsSimpleTime]	Missing description.
item [type wsRevenueCenterArray]	Missing description.
item [type wsSalesCategoryArray]	Missing description.
item [type wsSalesItem3Array]	Missing description.
item [type wsSalesItemArray]	Missing description.
message [type Exception]	Missing description.
militaryTime [type hsSimpleTime]	Missing description.
minutes [type hsSimpleTime]	Missing description.
month [type hsSimpleDate]	Missing description.
revenueCenterName [type wsRevenueCenter]	Missing description.
rvc [type dataSalesItem]	Missing description.

rvcName [type dataSalesItem]	Missing description.
salesCat [type dataSalesItem]	Missing description.
salesCategoryName [type wsSalesCategory]	Missing description.
seconds [type hsSimpleTime]	Missing description.
storeNum [type dataSalesItem]	Missing description.
success [type wsReturn]	Missing description.
successCount [type wsReturn]	Missing description.
transDate [type wsSalesItem3]	Missing description.

Name	Description
transTime [type wsSalesItem3]	Time of the transaction
ttl [type dataSalesItem]	Total of the sales category
year [type hsSimpleDate]	year yyyy format

Type string

Referenced By

- Complex Type hsSimpleTime

Type dateTime

Referenced By

- Complex Type wsSalesItem

Type hsSimpleDate

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Formatted dd
month	int	1..1	No	Formatted mm
year	int	1..1	No	Formatted yyyy

Referenced By

- Complex Type wsSalesItem3

Type string

Referenced By

- Complex Type dataSalesItem
- Complex Type wsSalesItem
- Complex Type wsSalesItem3

Type int

Referenced By

- Complex Type dataSalesItem
- Complex Type wsSalesItem
- Complex Type wsSalesItem3

Type dateTime

Referenced By

- Complex Type wsSalesItem
- Complex Type hsSimpleDate

Complex Type dataSalesItem

- Complex Type wsSalesItem
- Complex Type wsSalesItem3

- Complex Type wsReturn

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
message	string	0..1	No	Missing description.

- Complex Type dataSalesItem
- Complex Type wsSalesItem
- Complex Type wsSalesItem3

- Complex Type wsRevenueCenter

- Complex Type wsSalesCategory

- Complex Type wsReturn

- Complex Type wsRevenueCenter

- Complex Type wsSalesCategory

- Complex Type hsSimpleTime

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
extId	int	1..1	No	Missing description.
groupLevel	boolean	1..1	No	Missing description.
revenueCenterName	string	0..1	No	Missing description.

Referenced By

- Complex Type wsRevenueCenterArray

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
extId	int	1..1	No	Missing description.
groupLevel	boolean	1..1	No	Missing description.
salesCategoryName	string	0..1	No	Missing description.

Referenced By

- Complex Type wsSalesCategoryArray

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
catName	string	0..1	No	Name of the category
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
empId	int	1..1	No	HotSchedules internal employee account ID
extId	int	1..1	No	Numeric ID from the external system associated with the transaction
rvc	int	1..1	No	Revenue Center ID
rvcName	string	0..1	No	Name of the revenue center
salesCat	int	1..1	No	Numeric (integer) identifier for the sales category
storeNum	int	1..1	No	Numeric (integer) identifier for the location. Must be unique within the concept.
ttl	float	1..1	No	Total of the sales category
businessDate	hsSimpleDate	0..1	No	Business date of transaction
transDate	hsSimpleDate	0..1	No	Date of the transaction
transTime	hsSimpleTime	0..1	No	Time of the transaction

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
catName	string	0..1	No	Name of the category
clientId	int	1..1	No	Unique identifier for client provided via HotSchedules.
empId	int	1..1	No	HotSchedules internal employee account ID
extId	int	1..1	No	Numeric ID from the external system associated with the transaction
rvc	int	1..1	No	Revenue Center ID
rvcName	string	0..1	No	Name of the revenue center
salesCat	int	1..1	No	Numeric (integer) identifier for the sales category
storeNum	int	1..1	No	Numeric (integer) identifier for the location. Must be unique within the concept.
ttl	float	1..1	No	Total of the sales category
businessDate	dateTime	0..1	No	Missing description.
dateTime	dateTime	0..1	No	Missing description.

- Complex Type wsSalesItem
- Complex Type wsSalesItem3

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Day formatted dd
month	int	1..1	No	Month formatted mm
year	int	1..1	No	Year formatted yyyy

Referenced By

- Complex Type wsSalesItem3

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Complex Type wsSalesItem3

- Complex Type dataSalesItem
- Complex Type wsSalesItem
- Complex Type wsSalesItem3

- Complex Type hsSimpleDate

ProjectedSalesService Web Service

Description

This service is intended for third parties to be able to grab projected sales data from HotSchedules and import it into their POS/data warehouse/enterprise/etc. system. The WSDL is available at:

https://services.hotschedules.com/api/services/ProjectedSalesService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods: ProjectedSalesService

Methods

Name	Description
getProjectedSales	This method takes in a concept ID, store ID, start and end dates. It returns an array of WSProjectedSales objects, which represent the projected sales totals for a particular piece of time, for import into the POS.
getProjectedSalesV3	This method takes in a concept ID, store ID, start and end dates. It returns an array of WSProjectedSales objects, which represent the projected sales totals for a particular piece of time, for import into the POS. This method uses hsSimpleDate objects for dates and hsSimpleTime objects for times.
sayHello	Missing description.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
start	dateTime	1..1	First day of projected sales.
end	dateTime	1..1	Last day of projected sales.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsProjectedSalesArray	1..1	Array of WSProjectedSales objects, which represent the projected sales totals for a particular period.

This method uses hsSimpleDate objects for dates and hsSimpleTime objects for times.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	int	1..1	The identifier for the location's concept/group. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	1..1	Numeric (integer) identifier for the location. Must be unique within the concept.
start	hsSimpleDate	1..1	First day of projected sales requested. This is an hsSimpleDate object.
end	hsSimpleDate	1..1	Last day of projected sales requested. This is an hsSimpleDate object.

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsProjectedSales3Array	1..1	Array of WSProjectedSales3 objects, which represent the projected sales totals for a particular period. hsSimpleDate objects are used for dates and hsSimpleTime objects for times.

Remarks

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://services.hotschedules.com/api/services/ProjectedSalesService"
xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
```

<wsse:Security mustUnderstand="true">

```
            <wsse:UsernameToken>
                <wsse:Username>REDACTED</wsse:Username>
                <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText"
>REDACTED</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <ns2:Body>
          <ns1:getProjectedSalesV3>
            <concept>1</concept>
            <storeNum>37</storeNum>
            <start>
                <day>18</day>
                <month>4</month>
                <year>2012</year>
            </start>
            <end>
                <day>18</day>
                <month>4</month>
                <year>2012</year>
            </end>
          </ns1:getProjectedSalesV3>
    </ns2:Body>
</SOAP-ENV:Envelope> SAMPLE
RESPONSE:
  <?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
    <soap:Body>
          <ns1:getProjectedSalesV3Response>
            <return>
                <item>
                    <businessDate>
                        <day>18</day>
                        <month>4</month>
                        <year>2012</year>
                    </businessDate>
                    <dateTotal>27500.0</dateTotal>
                    <dayPartTotals>
                        <dayPartEndTime>
                            <amPm/>
                            <hours>16</hours>
                            <militaryTime>true</militaryTime>
                            <minutes>0</minutes>
                            <seconds>0</seconds>
                        </dayPartEndTime>
                        <dayPartName>Lunch</dayPartName>
                        <dayPartStartTime>
                            <amPm/>
                            <hours>4</hours>
```

```
            <militaryTime>true</militaryTime>
            <minutes>0</minutes>
            <seconds>0</seconds>
        </dayPartStartTime>
        <dayPartTotal>6258.0</dayPartTotal>
    </dayPartTotals>
    <dayPartTotals>
        <dayPartEndTime>
            <amPm/>
```

```
                        <hours>4</hours>
                        <militaryTime>true</militaryTime>
                        <minutes>0</minutes>
                        <seconds>0</seconds>
                     </dayPartEndTime>
                     <dayPartName>Dinner</dayPartName>
                     <dayPartStartTime>
                        <amPm/>
                        <hours>16</hours>
                        <militaryTime>true</militaryTime>
                        <minutes>0</minutes>
                        <seconds>0</seconds>
                     </dayPartStartTime>
                     <dayPartTotal>21242.0</dayPartTotal>
                  </dayPartTotals>
               </item>
            </return>
         </ns1:getProjectedSalesV3Response>
      </soap:Body>
   </soap:Envelope>
```

Method: sayHello

Description

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	string	1..1	Missing description.

Complex Types: ProjectedSalesService

Complex Types

Name	Description
hsSimpleDate	Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.
hsSimpleTime	Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.
wsDayPartTotal	Contains daypart start and end times, daypart name and daypart total projected sales, as well as an array of wsSummaryItemTotal objects.
wsDayPartTotal3	Contains daypart start and end times as hsSimpleDate objects, daypart name and daypart total projected sales, as well as an array of wsSummaryItemTotal objects.
wsProjectedSales	Contains a business date and a total of projected sales for that business date, as well as an array of wsDayPartTotal objects.
wsProjectedSales3	Contains a business date as an hsSimpleDate object and a total of projected sales for that business date, as well as an array of wsDayPartTotal objects.
wsSummaryItemTotal	Contains a summary item name and total projected sales amount.

Description

Simple date object that excludes any time zone or locale data. Consists of day, month and year integer values.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Formatted dd
month	int	1..1	No	Formatted mm
year	int	1..1	No	Formatted yyyy

Referenced By

- Element businessDate [type wsProjectedSales3]

Simple time object that excludes any time zone or locale data. Consists of hour, minute and second integer values, am/pm string indicator and militaryTime flag.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Element dayPartEndTime [type wsDayPartTotal3]
- Element dayPartStartTime [type wsDayPartTotal3]

Contains daypart start and end times, daypart name and daypart total projected sales, as well as an array of wsSummaryItemTotal objects.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
dayPartEndTime	dateTime	0..1	No	Day part end time
dayPartName	string	0..1	No	Day part name
dayPartStartTime	dateTime	0..1	No	Day part start time
dayPartTotal	float	1..1	No	Day part total
summaryItemTotals	wsSummaryItemTotal	0..*	Yes	Missing description.

Referenced By

- Element dayPartTotals [type wsProjectedSales]

Remarks

Dayparts are a HotSchedules scheduling concept. Summary items are a HotSchedules grouping concept. Further definition is beyond the scope of this document. Please refer to HotSchedules User Training.

Contains daypart start and end times as hsSimpleDate objects, daypart name and daypart total projected sales, as well as an array of wsSummaryItemTotal objects.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
dayPartEndTime	hsSimpleTime	0..1	No	Day part end time
dayPartName	string	0..1	No	Day part name
dayPartStartTime	hsSimpleTime	0..1	No	Day part start time
dayPartTotal	float	1..1	No	Day part total
summaryItemTotals	wsSummaryItemTotal	0..*	Yes	Missing description.

Referenced By

- Element dayPartTotals [type wsProjectedSales3]

Remarks

Dayparts are a HotSchedules scheduling concept. Summary items are a HotSchedules grouping concept. Further definition is beyond the scope of this document. Please refer to HotSchedules User Training.

Contains a business date and a total of projected sales for that business date, as well as an array of wsDayPartTotal objects.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
businessDate	dateTime	0..1	No	Business date of transaction
dateTotal	float	1..1	No	Total for the business date
dayPartTotals	wsDayPartTotal	0..*	Yes	Total for the day part

Referenced By

- Element item [type wsProjectedSalesArray]

Contains a business date as an hsSimpleDate object and a total of projected sales for that business date, as well as an array of wsDayPartTotal objects.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
businessDate	hsSimpleDate	0..1	No	Business date of transaction
dateTotal	float	1..1	No	Total for the business date
dayPartTotals	wsDayPartTotal3	0..*	Yes	Total for the day part

Referenced By

- Element item [type wsProjectedSales3Array]

Contains a summary item name and total projected sales amount.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
summaryItemName	string	0..1	No	Missing description.
summaryItemTotal	float	1..1	No	Missing description.

Referenced By

- Element summaryItemTotals [type wsDayPartTotal]
- Element summaryItemTotals [type wsDayPartTotal3]

Remarks

Summary items are a HotSchedules grouping concept. Further definition is beyond the scope of this document. Please refer to HotSchedules User Training.

Name	Description
amPm [type hsSimpleTime]	Missing description.
businessDate [type wsProjectedSales]	Missing description.
businessDate [type wsProjectedSales3]	Missing description.
dateTotal [type wsProjectedSales]	Missing description.
dateTotal [type wsProjectedSales3]	Missing description.
day [type hsSimpleDate]	Missing description.
dayPartEndTime [type wsDayPartTotal]	Missing description.
dayPartEndTime [type wsDayPartTotal3]	Missing description.
dayPartName [type wsDayPartTotal]	Missing description.
dayPartName [type wsDayPartTotal3]	Missing description.
dayPartStartTime [type wsDayPartTotal]	Missing description.
dayPartStartTime [type wsDayPartTotal3]	Missing description.
dayPartTotal [type wsDayPartTotal]	Missing description.
dayPartTotal [type wsDayPartTotal3]	Missing description.
dayPartTotals [type wsProjectedSales]	Missing description.

dayPartTotals [type wsProjectedSales3]	Missing description.
hours [type hsSimpleTime]	Missing description.
item [type wsProjectedSales3Array]	Missing description.
item [type wsProjectedSalesArray]	Missing description.
militaryTime [type hsSimpleTime]	Missing description.
minutes [type hsSimpleTime]	Missing description.
month [type hsSimpleDate]	Missing description.
seconds [type hsSimpleTime]	Missing description.
summaryItemName [type wsSummaryItemTotal]	Missing description.
summaryItemTotal [type wsSummaryItemTotal]	Missing description.
summaryItemTotals [type wsDayPartTotal]	Missing description.
summaryItemTotals [type wsDayPartTotal3]	Missing description.

Name	Description
year [type hsSimpleDate]	Missing description.

Type string

Referenced By

- Complex Type hsSimpleTime

Type dateTime

Referenced By

- Complex Type wsProjectedSales

Type hsSimpleDate

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
day	int	1..1	No	Day formatted dd
month	int	1..1	No	Month formatted mm
year	int	1..1	No	Year formatted yyyy

Referenced By

- Complex Type wsProjectedSales3

- Complex Type wsProjectedSales

- Complex Type wsProjectedSales3

- Complex Type hsSimpleDate

- Complex Type wsDayPartTotal

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Complex Type wsDayPartTotal3

- Complex Type wsDayPartTotal

- Complex Type wsDayPartTotal3

- Complex Type wsDayPartTotal

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
amPm	string	0..1	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	1..1	No	Hour value
militaryTime	boolean	1..1	No	Military Time indicator
minutes	int	1..1	No	Minutes value
seconds	int	1..1	No	Seconds value

Referenced By

- Complex Type wsDayPartTotal3

- Complex Type wsDayPartTotal

- Complex Type wsDayPartTotal3

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
dayPartEndTime	dateTime	0..1	No	Day part end time
dayPartName	string	0..1	No	Day part name
dayPartStartTime	dateTime	0..1	No	Day part start time
dayPartTotal	float	1..1	No	Day part total
summaryItemTotals	wsSummaryItemTotal	0..*	Yes	Missing description.

Referenced By

- Complex Type wsProjectedSales

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
dayPartEndTime	hsSimpleTime	0..1	No	Day part end time
dayPartName	string	0..1	No	Day part name
dayPartStartTime	hsSimpleTime	0..1	No	Day part start time
dayPartTotal	float	1..1	No	Day part total
summaryItemTotals	wsSummaryItemTotal	0..*	Yes	Missing description.

Referenced By

- Complex Type wsProjectedSales3

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
businessDate	hsSimpleDate	0..1	No	Business date of transaction
dateTotal	float	1..1	No	Total for the business date
dayPartTotals	wsDayPartTotal3	0..*	Yes	Total for the day part

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
businessDate	dateTime	0..1	No	Business date of transaction
dateTotal	float	1..1	No	Total for the business date
dayPartTotals	wsDayPartTotal	0..*	Yes	Total for the day part

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
summaryItemName	string	0..1	No	Missing description.
summaryItemTotal	float	1..1	No	Missing description.

Referenced By

- Complex Type wsDayPartTotal

Component	Type	Occurs	Nillable ?	Description
SEQUENCE		1..1		
summaryItemName	string	0..1	No	Missing description.
summaryItemTotal	float	1..1	No	Missing description.

Referenced By

- Complex Type wsDayPartTotal3

LaborService Web Service

Description

This service is intended for third parties to be able to request labor data from HotSchedules and import it into their POS/data warehouse/enterprise/etc. system. The WSDL is available at: https://services.hotschedules.com/api/services/LaborService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Simple Types
- Elements

Methods: LaborService

Methods

Name	Description
getLaborByJobAndInterval	This method will take a concept ID, store number, start and end dates and a labor type and return a list of total labor by job code for each interval in the date range requested for that concept, store and labor type. Intervals are configured during initial setup for the customer and are typically 30 minutes or 15 minutes.
GetLaborByBusDate	This method will take a concept ID, store number, start and end dates and a labor type and return a list of total labor by job code for each interval for the business date requested for that concept, store and labor type. Intervals are configured during initial setup for the customer and are typically 30 minutes or 15 minutes.
sayHello	sayHello is a test method intended to be used to validate access to the service. It returns a text message indicating success.

Method: getLaborByJobAndInterval

Description

This method will take a concept ID, store number, start and end dates and a labor type and return a list of total labor by job code for each interval in the date range requested for that concept, store and labor type.

Intervals are configured during initial setup for the customer and are typically 30 minutes.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
start	hsSimpleDate	Start date for the range of data requested
end	hsSimpleDate	End date for the range of data requested
laborType	laborType	Type of labor requested. Allowed types are "optimal", "forecasted" and "scheduled".

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	wsLaborJobArray	Returns a wsLaborJobArray object, which is an array of wsLaborJob objects. Each wsLaborJob object contains - a jobCode (which corresponds to the numeric ID that identifies this job) - a jobName (the name for this job) - a wsLaborInterval object, which can contain + an intervalDate object (an hsSimpleDate that determines the date for this particular interval) + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects) + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled" + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime) ** Note about labor types. "Scheduled" labor refers to shifts that have been scheduled and posted and are assigned to an employee. This will not include house shifts or shifts not assigned to an employee. "forecasted" labor refers to labor shifts generated after forecast labor drivers and applying labor shift generation rules within the Activity-based forecasting module in HotSchedules "optimal" labor refers to labor shifts generated by using actual labor driver values and applying labor shift generation rules specified in the Activity-based forecasting module in HotSchedules. It represents the labor shifts that would have been generated using an exact forecast of labor drivers

Example

*****Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://services.hotschedules.com/api/services/LaborService"
xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <wsse:Security mustUnderstand="true">
      <wsse:UsernameToken>
        <wsse:Username>REDACTED</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordT
ext">REDACTED</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <ns2:Body>
    <ns1:getLaborByJobAndInterval>
      <concept>1</concept>
      <storeNum>1101645</storeNum>
      <start>
        <day>8</day>
        <month>7</month>
        <year>2014</year>
      </start>
      <end>
        <day>14</day>
        <month>7</month>
        <year>2014</year>
      </end>
      <laborType>forecasted</laborType>
    </ns1:getLaborByJobAndInterval>
  </ns2:Body>
</SOAP-ENV:Envelope>
```

Response:

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope>
  <soap:Body>
    <ns1:getLaborByJobAndIntervalResponse>
      <return>
        <item>
          <interval>
            <intervalDate>
              <day>8</day>
              <month>7</month>
              <year>2014</year>
            </intervalDate>
            <intervalTime>
              <hours>0</hours>
              <militaryTime>true</militaryTime>
              <minutes>0</minutes>
              <seconds>0</seconds>
            </intervalTime>
            <laborType>forecasted</laborType>
            <volume>0.0</volume>
          </interval>
          <interval>
            <intervalDate>
              <day>8</day>
              <month>7</month>
              <year>2014</year>
            </intervalDate>
            <intervalTime>
              <hours>0</hours>
              <militaryTime>true</militaryTime>
              <minutes>30</minutes>
              <seconds>0</seconds>
            </intervalTime>
            <laborType>forecasted</laborType>
            <volume>0.0</volume>
          </interval>
          <jobCode>16000</jobCode>
          <jobName>Crew Person</jobName>
        </item>
      </return>
    </ns1:getLaborByJobAndIntervalResponse>
  </soap:Body>
</soap:Envelope>

Method: getLaborByJobBusDate

Description

This method will take a concept ID, store number, start and end dates and a labor type and return a list of total labor by job code for each interval in the date range requested for that concept, store and labor type.

Intervals are configured during initial setup for the customer and are typically 30 minutes.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
start	hsSimpleDate	Start date for the range of data requested
end	hsSimpleDate	End date for the range of data requested
laborType	laborType	Type of labor requested. Allowed types are "optimal", "forecasted" and "scheduled".

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	wsLaborJobArray	Returns a wsLaborJobArray object, which is an array of wsLaborJob objects. Each wsLaborJob object contains - a jobCode (which corresponds to the numeric ID that identifies this job) - a jobName (the name for this job) - a wsLaborInterval object, which can contain + an intervalDate object (an hsSimpleDate that determines the date for this particular interval) + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects) + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled" + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime) ** Note about labor types. "Scheduled" labor refers to shifts that have been scheduled and posted and are assigned to an employee. This will not include house shifts or shifts not assigned to an employee. "forecasted" labor refers to labor shifts generated after forecast labor drivers and applying labor shift generation rules within the Activity-based forecasting module in HotSchedules "optimal" labor refers to labor shifts generated by using actual labor driver values and applying labor shift generation rules specified in the Activity-based forecasting module in HotSchedules. It represents the labor shifts that would have been generated using an exact forecast of labor drivers

Example

*****Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ns0="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://services.hotschedules.com/api/services/LaborService"
xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <wsse:Security mustUnderstand="true">
      <wsse:UsernameToken>
        <wsse:Username>REDACTED</wsse:Username>
        <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordT
ext">REDACTED</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <ns2:Body>
    <ns1:getLaborByJobAndInterval>
      <concept>1</concept>
      <storeNum>1101645</storeNum>
      <start>
        <day>8</day>
        <month>7</month>
        <year>2014</year>
      </start>
      <end>
        <day>14</day>
        <month>7</month>
        <year>2014</year>
      </end>
      <laborType>forecasted</laborType>
    </ns1:getLaborByJobAndInterval>
  </ns2:Body>
</SOAP-ENV:Envelope>
```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getEmpInfoResponse xmlns:ns1="http://services.hotschedules.com/api/services/EmpService">
      <return>
        <item>
          <accountCreated>2015-11-04T10:54:22.807-06:00</accountCreated>
          <assignedSchedules>
            <extId>0</extId>
            <hsId>964330043</hsId>
            <name>Kitchen</name>
          </assignedSchedules>
          <assignedSchedules>
            <extId>-1</extId>
            <hsId>964318722</hsId>
            <name>Meetings</name>
          </assignedSchedules>
          <assignedSchedules>
            <extId>0</extId>
            <hsId>964330041</hsId>
            <name>Team Member</name>
          </assignedSchedules>
          <empHrId>-1</empHrId>
          <empNum>1</empNum>
          <lastUpdated>2015-11-09T10:29:31.123-06:00</lastUpdated>
          <permissionSetName>Employee</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-04T08:33:41.090-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <permissionSetName>HS ASC Support</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-04T08:42:02.483-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2015-11-05T18:29:55.857-06:00</lastUpdated>
          <permissionSetName>Admin ALL</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-04T08:49:32.623-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2015-11-04T08:49:33.120-06:00</lastUpdated>
          <permissionSetName>Default HS Support User</permissionSetName>

```
        </item>
        <item>
          <accountCreated>2015-11-04T08:49:33.887-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2015-11-04T08:49:34.093-06:00</lastUpdated>
          <permissionSetName>Default HotSchedules Employee</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-05T18:02:12.477-06:00</accountCreated>
          <assignedSchedules>
            <extId>0</extId>
            <hsId>964330043</hsId>
            <name>Kitchen</name>
          </assignedSchedules>
          <empHrId>-1</empHrId>
          <empNum>2</empNum>
          <lastUpdated>2015-11-09T10:30:32.927-06:00</lastUpdated>
          <permissionSetName>Employee</permissionSetName>
        </item>
        <item>
          <accountCreated>2015-11-09T09:38:47.617-06:00</accountCreated>
          <assignedSchedules>
            <extId>0</extId>
            <hsId>964330043</hsId>
            <name>Kitchen</name>
          </assignedSchedules>
          <empHrId>-1</empHrId>
          <empNum>3</empNum>
          <lastUpdated>2015-11-09T10:29:31.170-06:00</lastUpdated>
          <permissionSetName>Employee</permissionSetName>
        </item>
        <item>
          <accountCreated>2016-01-07T11:25:55.020-06:00</accountCreated>
          <empHrId>-1</empHrId>
          <empNum>-1</empNum>
          <lastUpdated>2016-01-07T11:28:13.373-06:00</lastUpdated>
          <permissionSetName>Admin</permissionSetName>
        </item>
      </return>
    </ns1:getEmpInfoResponse>
  </soap:Body>
</soap:Envelope>
```

Method: sayHello

Description

sayHello is a test method intended to be used to validate access to the service. It returns a text message indicating success.

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	string	

Complex Types: LaborService

Complex Types

Name	Description
Exception	
hsSimpleDate	Simplified date object used by most HS API methods. Includes day, month and year int values.
hsSimpleTime	Simplified time object used by most HS API methods. Includes hours, minutes and seconds int values, string for amPm enum and militaryTime boolean. If militaryTime is set to true, amPm is ignored.
wsLaborInterval	wsLaborInterval objects describe the total labor of a given labor type during a given interval. A wsLaborInterval object contains + an intervalDate object (an hsSimpleDate that determines the date for this particular interval) + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects) + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled" + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)
wsLaborJob	Each wsLaborJob object contains - a jobCode (which corresponds to the numeric ID that identifies this job) - a jobName (the name for this job) - a wsLaborInterval object, which can contain + an intervalDate object (an hsSimpleDate that determines the date for this particular interval) + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects) + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled" + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)
wsLaborJobArray	An array of wsLaborJob objects. Each wsLaborJob object describes the total labor for a given labor type for a given job during a given interval. Each wsLaborJob object contains - a jobCode (which corresponds to the numeric ID that identifies this job) - a jobName (the name for this job)

	- a wsLaborInterval object, which can contain + an intervalDate object (an hsSimpleDate that determines the date for this particular interval) + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects) + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled" + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)

Complex Type: Exception

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
message	string	No	

Referenced By

- Element Exception

Complex Type: hsSimpleDate

Description

Simplified date object used by most HS API methods. Includes day, month and year int values.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
day	int	No	Day formatted dd
month	int	No	Month formatted mm
year	int	No	Year formatted yyyy

Referenced By

- Element intervalDate [type wsLaborInterval]

Complex Type: hsSimpleTime

Description

Simplified time object used by most HS API methods. Includes hours, minutes and seconds int values, string for amPm enum and militaryTime boolean. If militaryTime is set to true, amPm is ignored.

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
amPm	string	No	AM or PM indicator. amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	No	Hour value
militaryTime	boolean	No	Military Time indicator
minutes	int	No	Minutes value
seconds	int	No	Seconds value

Referenced By

- Element intervalTime [type wsLaborInterval]

Complex Type: wsLaborInterval

Description

wsLaborInterval objects describe the total labor of a given labor type during a given interval.
A wsLaborInterval object contains
+ an intervalDate object (an hsSimpleDate that determines the date for this particular interval)
+ an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects)
+ a laborDate object (an enum which can be "optimal", "forecasted", "scheduled"
+ an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
intervalDate	hsSimpleDate	No	
intervalTime	hsSimpleTime	No	
laborType	laborType	No	
volume	double	No	

Referenced By

- Element interval [type wsLaborJob]

Complex Type: wsLaborJob

Description

Each wsLaborJob object contains
- a jobCode (which corresponds to the numeric ID that identifies this job)
- a jobName (the name for this job)
- a wsLaborInterval object, which can contain
    + an intervalDate object (an hsSimpleDate that determines the date for this particular interval)
    + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects)
    + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled"
    + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
interval	wsLaborInterval	Yes	
jobCode	int	No	
jobName	string	No	

Referenced By

- Element item [type wsLaborJobArray]

Complex Type: wsLaborJobArray

Description

An array of wsLaborJob objects. Each wsLaborJob object describes the total labor for a given labor type for a given job during a given interval.
Each wsLaborJob object contains
- a jobCode (which corresponds to the numeric ID that identifies this job)
- a jobName (the name for this job)
- a wsLaborInterval object, which can contain
    + an intervalDate object (an hsSimpleDate that determines the date for this particular interval)
    + an intervalTime object (an hsSimpleTime that determines the start time for this particular interval. Interval length in implied by the span between two intervalTime objects)
    + a laborDate object (an enum which can be "optimal", "forecasted", "scheduled"
    + an volume value (which represents the value for the laborDate type for this particular interval, determined by intervalDate and intervalTime)

Derived By

Restricting anyType

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
item	wsLaborJob	Yes	

Simple Types: LaborService

Simple Types

Name	Description
laborType	Valid labor types for getLaborByJobAndInterval

Simple Type: laborType

Description

Valid labor types for getLaborByJobAndInterval

Derived By

Restricting string

Enumeration

Value	Description
optimal	This is the result of applying the shift generation rules to the actual labor drivers for the span requested. Also called optimals from actuals.
forecasted	This is the result of applying the shift generation rules to forecasted labor drivers for the span requested. Also called optimals from forecast.
scheduled	These are the labor hours actually scheduled for the span requested. This does not include house shifts or shifts not assigned to any employee.

Referenced By

- Element laborType [type wsLaborInterval]

Elements: LaborService

Elements

Name	Description
amPm [type hsSimpleTime]	
day [type hsSimpleDate]	
Exception	
hours [type hsSimpleTime]	
interval [type wsLaborJob]	
intervalDate [type wsLaborInterval]	
intervalTime [type wsLaborInterval]	
item [type wsLaborJobArray]	
jobCode [type wsLaborJob]	
jobName [type wsLaborJob]	
laborType [type wsLaborInterval]	
message [type Exception]	
militaryTime [type hsSimpleTime]	
minutes [type hsSimpleTime]	
month [type hsSimpleDate]	
seconds [type hsSimpleTime]	
volume [type wsLaborInterval]	
year [type hsSimpleDate]	

Element: amPm [type hsSimpleTime]

Derived By

Type string

Referenced By

- Complex Type hsSimpleTime

Element: day [type hsSimpleDate]

Derived By

Type int

Referenced By

- Complex Type hsSimpleDate

Element: Exception

Derived By

Type Exception

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
message	string	No	

Element: hours [type hsSimpleTime]

Derived By

Type int

Referenced By

- Complex Type hsSimpleTime

Element: interval [type wsLaborJob]

Derived By

Type wsLaborInterval

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
intervalDate	hsSimpleDate	No	
intervalTime	hsSimpleTime	No	
laborType	laborType	No	
volume	double	No	

Referenced By

- Complex Type wsLaborJob

Element: intervalDate [type wsLaborInterval]

Derived By

Type hsSimpleDate

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
day	int	No	Day formatted dd
month	int	No	Month formatted mm
year	int	No	Year formatted yyyy

Referenced By

- Complex Type wsLaborInterval

Element: intervalTime [type wsLaborInterval]

Derived By

Type hsSimpleTime

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
amPm	string	No	amPm enum. If militaryTime is set to true, amPm is ignored.
hours	int	No	
militaryTime	boolean	No	
minutes	int	No	
seconds	int	No	

Referenced By

- Complex Type wsLaborInterval

Element: item [type wsLaborJobArray]

Derived By

Type wsLaborJob

Content Model

Contains elements as defined in the following table.

Component	Type	Nillable?	Description
interval	wsLaborInterval	Yes	
jobCode	int	No	
jobName	string	No	

Referenced By

- Complex Type wsLaborJobArray

Element: jobCode [type wsLaborJob]

Derived By

Type int

Referenced By

- Complex Type wsLaborJob

Element: jobName [type wsLaborJob]

Derived By

Type string

Referenced By

- Complex Type wsLaborJob

Element: laborType [type wsLaborInterval]

Derived By

Type laborType

Enumeration

Value	Description
optimal	
forecasted	
scheduled	

Referenced By

- Complex Type wsLaborInterval

Element: message [type Exception]

Derived By

Type string

Referenced By

- Complex Type Exception

Element: militaryTime [type hsSimpleTime]

Derived By

Type boolean

Referenced By

- Complex Type hsSimpleTime

Element: minutes [type hsSimpleTime]

Derived By

Type int

Referenced By

- Complex Type hsSimpleTime

Element: month [type hsSimpleDate]

Derived By

Type int

Referenced By

- Complex Type hsSimpleDate

Element: seconds [type hsSimpleTime]

Derived By

Type int

Referenced By

- Complex Type hsSimpleTime

Element: volume [type wsLaborInterval]

Derived By

Type double

Referenced By

- Complex Type wsLaborInterval

Element: year [type hsSimpleDate]

Derived By

Type int

Referenced By

- Complex Type hsSimpleDate

Methods: VolumeService

Methods

Name	Description
getGuestCounts	This method will take a concept ID, store number, start and end dates and return a list of guest counts for the date range requested.
getVolumeCounts	This method will take a concept ID, store number, start and end dates and a volume type and return a list of volume counts for the date range requested. Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”, “DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS”
sayHello	sayHello is a test method intented to be used to validate access to the service. It returns a text message indicating success.
setGuestCounts	This method takes in a concept ID, store ID, business date, date time, guest count and a revenue center for the purpose of submitting actual guest count drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains guest counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.
setVolumeCounts	This method takes in a concept ID, store ID, business date, date time, volume amount, volume type, and a revenue center for the purpose of submitting actual volume drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains volume driver counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.
getDriversbyInterval	This method takes in a concept ID, store ID, start and end dates, a volume type, and a data type and returns a list of volume counts for each interval in the date range requested. Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”,

	“DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS” Supported Data Types are: “ACTUAL”, “ADJ_FORECASTED”, and “PRE_ADJ_FORECASTED”
setForecastDrivers	This method takes in a concept ID, store ID, workweek startdate and enddate , starttime and endtime, volume amount, volume type, and a revenue center for the purpose of submitting forecasted volume drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains volume driver counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change.

Method: getGuestCounts

Description

This method will take a concept ID, store number, start and end dates and return a list of guest counts for the date range requested.

Action

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
start	dateTime	Start date for the range of data requested. This is a basic dateTime object.
end	dateTime	End date for the range of data requested. This is a basic dateTime object.

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	wsGuestCountsArray	Returns a wsGuestCountsArray object, which is an array of wsGuestCountsArray objects. Each wsGuestCounts object contains - a businessDate - a dateTime - a guestCount - a rvcExtID which represents the numeric revenue center ID associated with the guest

Example

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:getGuestCounts>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <startDate>2014-09-24T00:00:00</startDate>
      <endDate>2014-09-25T00:00:00</endDate>
    </vol:getGuestCounts>
  </soapenv:Body>
</soapenv:Envelope>
```

*****Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getGuestCountsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/VolumeService">
      <return>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:35:00-05:00</dateTime>
          <guestCount>1.0</guestCount>
          <rvcExtId>4</rvcExtId>
        </item>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:38:00-05:00</dateTime>
          <guestCount>1.0</guestCount>
          <rvcExtId>4</rvcExtId>
        </item>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:41:00-05:00</dateTime>
          <guestCount>1.0</guestCount>
          <rvcExtId>4</rvcExtId>
        </item>
```

Method: getVolumeCounts

Description

This method will take a concept ID, store number, start and end dates and a volume type and return a list of volume counts for the date range requested.

Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”, “DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS”

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
volumeType	volumeType	Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”, “DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS”
start	hsSimpleDate	Start date for the range of data requested
end	hsSimpleDate	End date for the range of data requested

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	wsVolumeCountsArray	Returns a wsVolumeCounts object, which is an array of wsVolumeCounts objects. Each wsLaborJob object contains - a businessDate - a dateTime - a volumeType - a volumeCount - a rvcExtID which represents the numeric revenue center ID associated with the volume count

Example

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:getVolumeCounts>
      <concept>1</concept>
      <storeNum>101</storeNum>
      <volumeType>GUESTS</volumeType>
      <startDate>2014-09-24T00:00:00</startDate>
      <endDate>2014-09-25T00:00:00</endDate>
    </vol:getVolumeCounts>
  </soapenv:Body>
</soapenv:Envelope>
```

*****Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getGuestCountsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/VolumeService">
      <return>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:35:00-05:00</dateTime>
         <volumeType>GUESTS</volumeType>
       <volumeCount>1.0</volumeCount>
          <rvcExtId>4</rvcExtId>
        </item>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:38:00-05:00</dateTime>
        <volumeType>GUESTS</volumeType>
          <volumeCount>1.0</volumeCount>
          <rvcExtId>4</rvcExtId>
        </item>
        <item>
          <businessDate>2014-09-24T00:00:00-05:00</businessDate>
          <dateTime>2014-09-24T06:41:00-05:00</dateTime>
        <volumeType>GUESTS</volumeType>
          <volumeCount>1.0</volumeCount>
          <rvcExtId>4</rvcExtId>
        </item>
```

Method: sayHello

Description

sayHello is a test method intented to be used to validate access to the service. It returns a text message indicating success.

Action

Output

The outputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
return	string	

Method: setGuestCounts

Description

This method takes in a concept ID, store ID, business date, date time, guest count and a revenue center for the purpose of submitting actual guest count drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains guest counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
businessDate	hsSimpleDate	Business date of transaction
dateTime	dateTime	Date Time of the transaction
guestCount	Int	Number of guests for the transaction
rvcExtID	int	Numeric ID for the revenue center associated with the transaction

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Example

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:setGuestCounts>
      <concept>?</concept>
      <storeNum>?</storeNum>
      <guests>
        <item>
         <businessDate>2014-10-05T00:00:00</businessDate>
          <dateTime>2014-10-05T06:41:00-05:00</dateTime>
          <guestCount>1</guestCount>
          <rvcExtId>4</rvcExtId>
        </item>
      </guests>
    </vol:setGuestCounts>
  </soapenv:Body>
</soapenv:Envelope>
```

Method: setVolumeCounts

Description

This method takes in a concept ID, store ID, business date, date time, volume amount, volume type, and a revenue center for the purpose of submitting actual volume drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains volume driver counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change. This method returns a WSReturn object.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
businessDate	hsSimpleDate	Business date of transaction
dateTime	dateTime	Date Time of the transaction
rvcExtID	int	Numeric ID for the revenue center associated with the transaction
volumeAmount	Int	Value of the volume count for the transaction
volumeType	volumeType	Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”, “DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS”

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Example

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:setVolumeCounts>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <volumeData>
        <item>
          <businessDate>2014-10-05T00:00:00</businessDate>
          <dateTime>2014-10-05T06:41:00-05:00</dateTime>
          <rvcExtId>4</rvcExtId>
          <volumeAmount>1</volumeAmount>
          <volumeType>GUESTS</volumeType>
        </item>
      </volumeData>
    </vol:setVolumeCounts>
  </soapenv:Body>
</soapenv:Envelope>
```

Method: getDriversbyInterval

Description

This method will take a concept ID, store number, start and end dates, volume type, and data type and return a list of total driver amount for each interval in the date range requested for that concept, store and labor type.

Intervals are configured during initial setup for the customer and are typically 30 minutes or 15 minutes.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
startDate	hsSimpleDate	Start date for the range of data requested
endDate	hsSimpleDate	End date for the range of data requested
volumeType	driverClass	Classification of driver requested. Allowed types would be all of the classifications supported from API, HSC, or FTP integration. “Guests”, “Tables”, “Entrees”, “Deliveries”, and “Products”.
dataType	driverType	Type of driver requested. Allowed types are “ACTUAL”, “ADJ_FORECASTED”, and “PRE_ADJ_FORECASTED”.

Example:

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:getDriversByInterval>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <startDate>
        <day>3</day>
        <month>10</month>
        <year>2014</year>
      </startDate>
      <endDate>
        <day>9</day>
        <month>10</month>
        <year>2014</year>
      </endDate>
      <volumeType>TABLE</volumeType>
      <dataType>ACTUAL</dataType>
    </vol:getDriversByInterval>
  </soapenv:Body>
</soapenv:Envelope>
```

*****Return:

Method: setForecastDrivers

Description

This method takes in a concept ID, store ID, workweek startdate and enddate , starttime and endtime, volume amount, volume type, and a revenue center for the purpose of submitting forecasted volume drivers to HotSchedules from a third party system or point of sale. Using the authentication from the username token and the concept and store IDs, the server will resolve which HotSchedules client this sync is for. The array contains volume driver counts for a range of dates, corresponding to the start and end dates. The server-side logic can handle overlapping data (i.e. if you sync 7 days worth of time cards, every day, 6 days of it will be "overlapping" data) and will insert and update data as needed. If the guest are already in the HS database and do not need to be updated, then nothing will change.

Input

The inputs of this method are the arguments defined by the following table.

Argument	Type	Description
ALL		
concept	int	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	int	Numeric (integer) identifier for the store. Must be unique within the concept.
WorkWeekStartDate	hsSimpleDate	Day, Month and Year
rvcExtID	int	Numeric ID for the revenue center associated with the transaction
DriverAmount	Int	Value of the driver amount for the transaction
volumeType	volumeType	Supported Volume Types are: “TABLE”, “ENTRÉE”, “GUESTS”, “DELIVERIES”, “PRODUCTS”, and “TRANSACTIONS”

Output (Literal)

The outputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
return	wsReturn	1..1	WSReturn object

Example

*****Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:vol="http://services.hotschedules.com/api/services/VolumeService">
  <soapenv:Header/>
  <soapenv:Body>
    <vol:setForecastDrivers>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <workweekStartDate>
        <day>1</day>
        <month>1</month>
        <year>2016</year>
      </workweekStartDate>
      <driverData>
        <!--Zero or more repetitions:-->
        <item>
          <driverAmount>10</driverAmount>
          <!--Optional:-->
          <intervalEndDate>
            <day>?</day>
            <month>?</month>
            <year>?</year>
          </intervalEndDate>
          <!--Optional:-->
          <intervalEndTime>
            <!--Optional:-->
            <amPm>?</amPm>
            <hours>?</hours>
            <militaryTime>?</militaryTime>
            <minutes>?</minutes>
            <seconds>?</seconds>
          </intervalEndTime>
          <!--Optional:-->
          <intervalStartDate>
            <day>?</day>
            <month>?</month>
            <year>?</year>
          </intervalStartDate>
          <!--Optional:-->
          <intervalStartTime>
            <!--Optional:-->
            <amPm>?</amPm>
            <hours>?</hours>
```

```
            <militaryTime>?</militaryTime>
            <minutes>?</minutes>
            <seconds>?</seconds>
          </intervalStartTime>
          <rvcId>1</rvcId>
          <!--Optional:-->
          <volumeType>1</volumeType>
        </item>
      </driverData>
    </vol:setForecastDrivers>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:setForecastDriversResponse xmlns:ns1="http://services.hotschedules.com/api/services/VolumeService">
      <return>
        <failCount>0</failCount>
        <success>true</success>
        <successCount>1</successCount>
      </return>
    </ns1:setForecastDriversResponse>
  </soap:Body>
</soap:Envelope>
```

CertificationService Web Service

These services provides operations for a third party to push or request employee certification data into HotSchedules. The WSDL is available at:

http://services.hotschedules.com/api/services/CertificationService?wsdl

Type

SOAP

Style

RPC

See Also

- Methods
- Complex Types
- Elements

Methods

Name	Description
getClientCertifications	This method takes in a concept ID and a store ID and returns an array of certification objects. It is meant to get a list of all certifications for that store.
getEmployeeCertifications	This method takes in a concept ID, store ID and employee POS ID and returns an array of employee certificate objects. It is meant to get a list of all employee certification information for an employee for that store.
removeEmployeeCertifications	This method takes in a concept ID and a store ID. It is meant to get allow you to remove certifications from a specified employee for that store.
setEmployeeCertifications	This method takes in a concept ID, store ID. It is meant to set certifications from a specified employee for that store.

Method: getClientCertifications

This method takes in a concept ID and a store ID and returns client certification objects. It is meant to get a list of all certifications for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/CertificationService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262747932814">
          <wsse:Username>laura1234!</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">9iOMGex3ZDEpdw4xsAweWA==</wsse:Nonce>
          <wsu:Created>2016-01-12T19:37:59.328Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <cer:getClientCertifications>
      <concept>1</concept>
      <storeNum>1</storeNum>
    </cer:getClientCertifications>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getClientCertificationsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/CertificationService">
      <return>
        <item>
          <certExtRef>-1</certExtRef>
          <certName>Airport Badge</certName>
          <certType>2</certType>
          <certTypeDescription>Expires Without AutoInactivation</certTypeDescription>
          <groupLevel>false</groupLevel>
        </item>
        <item>
          <certExtRef>-1</certExtRef>
          <certName>21 Day Certification</certName>
          <certType>2</certType>
          <certTypeDescription>Expires Without AutoInactivation</certTypeDescription>
          <groupLevel>true</groupLevel>
        </item>
      </return>
    </ns1:getClientCertificationsResponse>
  </soap:Body>
</soap:Envelope>
```

Method: getEmployeeCertifications

This method takes in a concept ID and a store ID and returns client certification objects. It is meant to get a list of all certifications assigned to an employee for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.
empPOSId	int	1..1	POS numeric employee ID

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/CertificationService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262747932814">
          <wsse:Username>laura1234!</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">9iOMGex3ZDEpdw4xsAweWA==</wsse:Nonce>
          <wsu:Created>2016-01-12T19:37:59.328Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <cer:getEmployeeCertifications>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <empPosId>100</empPosId>
    </cer:getEmployeeCertifications>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<ns1:getEmployeeCertificationsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/CertificationService">
<return>
<item>
<certExtRef>-1</certExtRef>
<certName>21 Day Certification</certName>
<certType>2</certType>
<certTypeDescription>Expires Without AutoInactivation</certTypeDescription>
<groupLevel>true</groupLevel>
<empPosId>100</empPosId>
<expirationDate>2016-11-30T00:00:00-06:00</expirationDate>
</item>
</return>
</ns1:getEmployeeCertificationsResponse>
</soap:Body>
</soap:Envelope>
```

Method: removeEmployeeCertifications

This method takes in a concept ID and a store ID. It is meant to remove certifications assigned to an employee for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/CertificationService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262747932814">
          <wsse:Username>laura1234!</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">9iOMGex3ZDEpdw4xsAweWA==</wsse:Nonce>
          <wsu:Created>2016-01-12T19:37:59.328Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
    <soapenv:Body>
    <cer:removeEmployeeCertifications>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <certs>
       <item>
          <certExtRef>200</certExtRef>
          <certName>21 Day Certification</certName>
          <certType>2</certType>
           <certTypeDescription>Expires Without AutoInactivation</certTypeDescription>
          <groupLevel>true</groupLevel>
          <empPosId>100</empPosId>
          <expirationDate>2016-11-30T00:00:00-06:00</expirationDate>
        </item>
      </certs>
    </cer:removeEmployeeCertifications>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:setEmployeeCertificationsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/CertificationService">
      <return>
        <failCount>0</failCount>
        <success>true</success>
        <successCount>1</successCount>
      </return>
    </ns1:setEmployeeCertificationsResponse>
  </soap:Body>
</soap:Envelope>
```

Method: setEmployeeCertifications

This method takes in a concept ID and a store ID. It is meant to add certifications assigned to an employee for that store.

Input (Literal)

The inputs of this method are the arguments defined by the following table.

Argument	Type	Occurs	Description
ALL		1..1	
concept	Int	1..1	The identifier for the location's concept. Must be unique within the company. Contact HotSchedules if you're not sure about this value.
storeNum	Int	1..1	Numeric (integer) identifier for the store. Must be unique within the concept.

Example

SAMPLE CALL:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:emp="http://services.hotschedules.com/api/services/CertificationService"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
     <wsse:Security
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
soapenv:mustUnderstand="1">
        <wsse:UsernameToken wsu:Id="UsernameToken-63A716DE011BE2D649145262747932814">
          <wsse:Username>laura1234!</wsse:Username>
          <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">laura
1234!</wsse:Password>
          <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Bina
ry">9iOMGex3ZDEpdw4xsAweWA==</wsse:Nonce>
          <wsu:Created>2016-01-12T19:37:59.328Z</wsu:Created>
        </wsse:UsernameToken>
     </wsse:Security>
  </soapenv:Header>
     <soapenv:Body>
    <cer:setEmployeeCertifications>
      <concept>1</concept>
      <storeNum>1</storeNum>
      <certs>
          <item>
          <certExtRef>500</certExtRef>
          <certName>Laura Test</certName>
          <certType>0</certType>
          <certTypeDescription>Testingnewcert</certTypeDescription>
          <groupLevel>1</groupLevel>
          <empPosId>100</empPosId>
           <expirationDate>?</expirationDate>
        </item>
      </certs>
    </cer:setEmployeeCertifications>
  </soapenv:Body>
</soapenv:Envelope>
```

RESPONSE:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:removeEmployeeCertificationsResponse
xmlns:ns1="http://services.hotschedules.com/api/services/CertificationService">
      <return>
        <failCount>0</failCount>
        <success>true</success>
        <successCount>1</successCount>
      </return>
    </ns1:removeEmployeeCertificationsResponse>
  </soap:Body>
</soap:Envelope>
```