

# **myallocator.com**

## **API Specification 1.5**

MX2 Solutions Ltd.

December 1, 2011

# Contents

<b>1</b>	<b>Revision History</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
2.0.1	Terminology . . . . .	3
2.0.2	Test account . . . . .	3
2.0.3	Online XML examples . . . . .	4
<b>3</b>	<b>API methods</b>	<b>4</b>
3.1	GetProperties . . . . .	4
3.1.1	Request . . . . .	4
3.1.2	Example Response . . . . .	4
3.2	GetRoomTypes . . . . .	5
3.2.1	Request . . . . .	5
3.2.2	Example Response . . . . .	5
3.3	SetRoomTypes . . . . .	6
3.3.1	Request - Creating a room . . . . .	6
3.3.2	Response - Creating a room . . . . .	7
3.3.3	Request - Removing a room . . . . .	7
3.3.4	Response - Removing a room . . . . .	8
3.4	SetAllocation . . . . .	8
3.4.1	Request . . . . .	8
3.4.2	Example Response . . . . .	9
3.4.3	Submitting to all channels . . . . .	10
3.4.4	Setting allocation in background . . . . .	11
3.5	GetUpdateStatus . . . . .	11
3.6	SetLogin . . . . .	12
3.6.1	Request - Creating a login and property . . . . .	12
3.6.2	Response - Creating a login and property . . . . .	13
3.6.3	Request - Updating a login and property . . . . .	13
3.6.4	Response - Updating a login and property . . . . .	14
3.6.5	Passive login . . . . .	14
3.7	GetBookings . . . . .	14
3.7.1	Request - Querying for bookings . . . . .	14
3.7.2	Response - Querying for bookings . . . . .	15
<b>4</b>	<b>Codes</b>	<b>18</b>
4.1	Channel codes . . . . .	18
4.2	Error codes . . . . .	18
4.2.1	Global errors . . . . .	18
4.2.2	API method specific errors and warnings . . . . .	19
4.2.3	Channel specific errors . . . . .	19

# 1 Revision History

Date	Version	Changes
2011-09-17	1.5	New method <code>SetRoomTypes</code> to add/update/remove rooms.
2011-09-17	1.4.1	<code>GetBookings</code> : Minor correction regarding the end date. It's not the departure date but rather the departure date - 1.
2011-01-15	1.4	New methods <code>SetAllocation</code> (non-blocking), <code>SetLogin</code> , <code>GetUpdateStatus</code> , <code>GetBookings</code> . Support for <code>MinStay</code> and <code>MaxStay</code> .
2010-11-09	1.3.1	Updated channel list. <code>GetRoomTypes</code> : Obsoleted "Ensuite", "DoubleBed" and "Beds" (replaced by new property "Occupancy"). <code>GetProperties</code> : shows which days are configure for weekends.
2010-05-30	1.3	<code>GetRoomTypes</code> includes a room description (Label). Removed need to list channels to update to and ability to exclude channels. Skipped channels now warnings rather than errors.
2010-05-05	1.2	Added links to XML samples. New channel: hb
2010-04-30	1.1	Changed <i>Room</i> to <i>RoomType</i> to clarify matters
2010-04-27	1.0	Initial release

## 2 Introduction

If you would like to use the myallocator.com API you need to request a vendor ID and password first. It will need to be submitted with every availability upload.

Requests to our server need to be send using the POST method. The XML string should be stored in a parameter called 'xmlRequestString'. Send requests to:

`http://api.myallocator.com`

Your customers need to create an account on myallocator.com before they can use the API connection of your software. The details they create there will be stored in your application and are also required for each request.

There are three API methods you can use. The first one is *GetProperties*, which will return with a list of properties configured on myallocator.com. You'll need the property ID for the other two methods.

The second method is *GetRoomTypes*, which returns a list of rooms configured on myallocator.com. These need to be mapped to the rooms in your software.

Finally, and most importantly, *SetAllocation* uploads availability to the channels. The customer is required to add all their channel credentials on myallocator.com before being able to update availability.

### 2.0.1 Terminology

*Vendor*: You as the owner and developer of the front desk software.

*User/Customer*: Someone using your product and who is also registered on myallocator.com.

*Allocation*: A number indicating how many available beds or rooms there are.

### 2.0.2 Test account

**Customer ID:** *Please contact us for login details*

**Customer Password:**

**Property IDs:** 63, 64, 65

Only the property with ID 63 has rooms setup with the following configuration:

Units	Beds	IsPrivate	Gender	Channels
3	1	true	n/a	hc, iw, adv
3	2	true	n/a	hc, iw, adv
3	5	false	males	hc, iw
3	10	false	males	hc, iw
3	10	false	females	hc, iw
3	30	false	mixed	hc, iw

*Note:* Channels are not specific to rooms. The only reason 'adv' is not showing on all rooms is because they don't support shared accommodation (dormitories).

### 2.0.3 Online XML examples

As copy&pasting from a PDF file can introduce some unwanted whitespace, all XML examples mentioned in this spec can also be found at this address:

[http://myallocator.com/n/api\\_examples.xml](http://myallocator.com/n/api_examples.xml)

## 3 API methods

### 3.1 GetProperties

Users on myallocator.com might have more than one property attached to their username. This request lists which properties are associated with the customer and provides a property ID, needed to update the availability of this property.

#### 3.1.1 Request

```
<?xml version="1.0" encoding="UTF-8"?>
<GetProperties>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
</GetProperties>
```

Listing 1: Example of a *GetProperties* request

#### 3.1.2 Example Response

```
<?xml version="1.0" encoding="utf-8"?>
<GetPropertiesResponse>
  <Properties>
    <Property>
      <Id>19</Id>
      <Name>Property 1</Name>
      <Breakfast></Breakfast>
      <Currency>GBP</Currency>
      <PaidUntil>2011-11-11</PaidUntil>
      <Weekend>
        <Day name="Friday">5</Day>
        <Day name="Saturday">6</Day>
      </Weekend>
    </Property>
    <Property>
      <Id>13</Id>
      <Name>Property 2</Name>
      <Breakfast>IN</Breakfast>
      <Currency>EUR</Currency>
```

```

    <PaidUntil>2011-11-11</PaidUntil>
    <Weekend>
    </Weekend>
  </Property>

  <Property>
    <Id>15</Id>
    <Name>Property 3</Name>
    <Breakfast>EX</Breakfast>
    <Currency>GBP</Currency>
    <PaidUntil>2011-11-11</PaidUntil>
    <Weekend>
      <Day name="Friday">5</Day>
      <Day name="Saturday">6</Day>
      <Day name="Sunday">7</Day>
    </Weekend>
  </Property>
</Properties>
</GetPropertiesResponse>

```

Listing 2: Example of a *GetProperties* response

Tag	Description
Id	The property ID which will reference the property.
Name	The name of this property as set by the customer.
Breakfast	Can be <i>IN</i> (included), <i>EX</i> (excluded) or empty.
Currency	The default currency set by the customer. Informational.
PaidUntil	Usage of myallocator.com paid for until this date. Informational. Format: YYYY-MM-DD.
Weekend / Day	Shows which days are marked as weekend days by the customer on Myallocator. Monday = 1, Sunday = 7. The name of the weekday is also given in the <i>name</i> property. See <i>SetAllocation</i> for information on when this is used.

## 3.2 GetRoomTypes

### 3.2.1 Request

```

<?xml version="1.0" encoding="UTF-8"?>
<GetRoomTypes>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
</GetRoomTypes>

```

Listing 3: Example of *GetRoomTypes* request

### 3.2.2 Example Response

```

<?xml version="1.0" encoding="utf-8"?>
<GetRoomTypesResponse>
  <RoomTypes>
    <RoomType>
      <Id>59</Id>
      <Label>1-bed private ensuite</Label>
      <Units>3</Units>
      <Occupancy>1</Occupancy>
      <Beds>1</Beds>
    </RoomType>
  </RoomTypes>
</GetRoomTypesResponse>

```

```

    <Gender>MK</Gender>
    <DoubleBed>>false</DoubleBed>
    <Ensuite>>false</Ensuite>
    <PrivateRoom>>true</PrivateRoom>
  </RoomType>

  <RoomType>
    <Id>63</Id>
    <Label>2-bed private</Label>
    <Units>4</Units>
    <Occupancy>2</Occupancy>
    <Beds>2</Beds>
    <Gender>MK</Gender>
    <DoubleBed>>false</DoubleBed>
    <Ensuite>>false</Ensuite>
    <PrivateRoom>>true</PrivateRoom>
  </RoomType>

  <RoomType>
    <Id>49</Id>
    <Label>5-bed female shared</Label>
    <Units>3</Units>
    <Occupancy>5</Occupancy>
    <Beds>5</Beds>
    <Gender>FE</Gender>
    <DoubleBed>>false</DoubleBed>
    <Ensuite>>false</Ensuite>
    <PrivateRoom>>false</PrivateRoom>
  </RoomType>
</RoomTypes>
</GetRoomTypesResponse>

```

Listing 4: Example of a *GetRoomTypes* response

Tag	Description
Id	The room type ID which will reference the room type.
Units	Number of rooms of this type.
Occupancy	Number of persons that can stay in this room.
Beds	[ <b>Obsolete</b> ] Replaced with <i>Occupancy</i> (see above).
Gender	Gender restriction for shared rooms. <i>MA</i> for males, <i>FE</i> for females, <i>MI</i> if mixed. Always set to 'MI' for private rooms.
DoubleBed	[ <b>Obsolete</b> ] Option removed from Myallocator. Now always defaults to <i>false</i> .
Ensuite	[ <b>Obsolete</b> ] Option removed from Myallocator. Now always defaults to <i>false</i> .
PrivateRooms	Whether this room is private or shared/dorm ( <i>true</i> or <i>false</i> ).

### 3.3 SetRoomTypes

This method allows you to create, update or remove rooms on myallocator.com. Please note that you can only send a single CreateRooms OR UpdateRooms OR RemoveRooms.

To create a room you would send the following request.

#### 3.3.1 Request - Creating a room

```

<?xml version="1.0" encoding="UTF-8"?>
<SetRoomTypes>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
  </Auth>

```

```

    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
</CreateRooms>
<RoomTypes>
  <RoomType>
    <Label>2-bed private with sea-view</Label>
    <Units>4</Units>
    <Occupancy>2</Occupancy>
    <PrivateRoom>true</PrivateRoom>
  </RoomType>
  <RoomType>
    <Label>6-bed female dorm</Label>
    <Units>3</Units>
    <Occupancy>6</Occupancy>
    <Gender>FE</Gender>
    <PrivateRoom>>false</PrivateRoom>
  </RoomType>
</RoomTypes>
</CreateRooms>
</SetRoomTypes>

```

Listing 5: Example of creating a room

Tag	Description
Label	String that describes the room, usually provided by the customer. If omitted, a label will automatically be created from the other properties. (optional)
Units	How many rooms of this type there are. This option doesn't actually limit how many rooms can be set as available, it's only informational.
Occupancy	Number of many people that can stay in this room.
PrivateRoom	<i>true</i> if it's a private room, <i>false</i> for dormitories.
Gender	Only needed for dormitories. <i>MA</i> for males, <i>FE</i> for females, <i>MI</i> if mixed.

### 3.3.2 Response - Creating a room

New RoomTypeId tags will appear in the same order as sent.

```

<?xml version="1.0" encoding="utf-8"?>
<SetRoomTypesResponse>
  <Success>true</Success>
  <RoomTypeIds>
    <RoomTypeId>35</RoomTypeId>
    <RoomTypeId>36</RoomTypeId>
  </RoomTypeIds>
</SetRoomTypesResponse>

```

Listing 6: Example of a successful *SetRooms* response

To remove a room you would send the following request.

### 3.3.3 Request - Removing a room

```

<?xml version="1.0" encoding="UTF-8"?>
<SetRoomTypes>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
  <RemoveRooms>
    <RoomTypeIds>

```

```

    <RoomTypeId>35</RoomTypeId>
    <RoomTypeId>36</RoomTypeId>
  </RoomTypeIds>
</RemoveRooms>
</SetRoomTypes>

```

Listing 7: Example of removing a room

Tag	Description
RoomTypeId	Id of room to be removed.

### 3.3.4 Response - Removing a room

```

<?xml version="1.0" encoding="utf-8"?>
<SetRoomTypesResponse>
  <Success>>true</Success>
</SetRoomTypesResponse>

```

Listing 8: Example of a successful *SetRooms* response

## 3.4 SetAllocation

An allocation sets the number of rooms or beds (depending on whether the room type is a private room or shared/dorm) **available** during any specific time frame. For more details see below.

### 3.4.1 Request

```

<?xml version="1.0" encoding="UTF-8" ?>
<SetAllocation>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>

  <Channels>
    <Channel>hc</Channel>
    <Channel>iwb</Channel>
  </Channels>

  <Allocations>
    <Allocation>
      <RoomTypeId>59</RoomTypeId>
      <StartDate>2010-06-01</StartDate>
      <EndDate>2010-08-12</EndDate>
      <Units>3</Units>
      <Prices>
        <Price>20.00</Price>
        <Price weekend="true">25.00</Price>
      </Prices>
    </Allocation>
  </Allocations>
</SetAllocation>

```

Listing 9: Example of a *SetAllocation* request

Tag	Description
Channels/Channel	A list of channels that the customer wants to update. See below for channel codes.
Allocations/Allocation	Individual allocations. There can be as many as you like, but a the date ranges should never overlap (within one room id).
RoomTypeId	Room type id as returned by GetRoomTypes.
StartDate/EndDate	Format: YYYY-MM-DD.
Units	How many beds or rooms should be available on this day. Whether it's beds or rooms depends on whether the room is a dorm or a private room. To remove allocation set Units to 0.
Prices/Price	Price per person for shared/dorm rooms or per room for private rooms. You can submit two Price tags. The default one and one with the attribute <i>weekend="true"</i> . If a Price tag with this attribute has been submitted the rate only applies to weekdays set by the customer as weekends (see <i>GetProperties</i> for which days are set as weekend days).

### 3.4.2 Example Response

The response will always include the *Success*, *Errors* and *Warnings* tags. If *Success* is set to "true" the *Errors* tag will be empty. If *Success* is set to "partial" or "false" at least one Error tag is included.

```
<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
  <Success>true</Success>
  <Errors></Errors>
  <Warnings>
    <Warning channel="hc">
      <WarningId>207</WarningId>
      <WarningMsg>
        Skipped room type (not setup with channel).
      </WarningMsg>
    </Warning>
  </Warnings>
</SetAllocationResponse>
```

Listing 10: Example of a successful *SetAllocation* response

```
<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
  <Success>partial</Success>
  <Errors>
    <Error channel="adv">
      <ErrorId>15</ErrorId>
      <ErrorMsg>
        Missing or wrong channel credentials on myallocator.com
      </ErrorMsg>
    </Error>
  </Errors>
  <Warnings></Warnings>
</SetAllocationResponse>
```

Listing 11: Partial success (at least one channel succeeded)

```
<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
  <Success>>false</Success>
  <Errors>
    <Error channel="adv">
```

```

    <ErrorId>15</ErrorId>
    <ErrorMsg>
      Missing or wrong channel credentials on myallocator.com
    </ErrorMsg>
  </Error>
</Errors>
<Warnings></Warnings>
</SetAllocationResponse>

```

Listing 12: Failed update (all channels came back with errors)

```

<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
  <Success>>false</Success>
  <Errors>
    <Error">
      <ErrorId>17</ErrorId>
      <ErrorMsg>
        Start date too far in the future (>2 years)
      </ErrorMsg>
    </Error>
  </Errors>
  <Warnings></Warnings>
</SetAllocationResponse>

```

Listing 13: Failed update (error before submitting to any channel)

### 3.4.3 Submitting to all channels

You can also submit to all available channels without specifying explicitly which channels to update. Use the channel code "all" to do this. Channels that are not set up by the user will be skipped as indicated by a warning.

It is also possible to submit to all channels while excluding certain channels. Add the attribute `exclude="true"` to skip a channel. See example below. Use the channel code "all" to do this.

```

<?xml version="1.0" encoding="UTF-8" ?>
<SetAllocation>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>

  <Channels>
    <Channel>all</Channel>
    <Channel exclude="true">gom</Channel>
    <Channel exclude="true">hc</Channel>
  </Channels>

  <Allocations>
    ...
  </Allocations>
</SetAllocation>

```

Listing 14: Submit to all channels excluding Gomio and Hostelsclub

Tag	Description
Success	Can be <i>true</i> (all channel updates succeeded), <i>partial</i> (not all channels succeeded) or <i>false</i> (all channel updates failed).
Errors/Error	Check with the list of errors below. Might contain the attribute <i>channel=".."</i> to indicate whether the error only applies to a specific channel.
Warnings/Warning	Check with the list of errors below. Might contain the attribute <i>channel=".."</i> to indicate whether the warning only applies to a specific channel.

### 3.4.4 Setting allocation in background

You can also run the SetAllocation update in the background and query in short intervals for the success for the updates. This enables you to show the update progress to the user while it's still running. To enable this feature you need to add the option QueryForStatus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SetAllocation>
  <Auth>
    ...
  </Auth>

  <Options>
    <QueryForStatus>true</QueryForStatus>
  </Options>
  ...
</SetAllocation>
```

Listing 15: Running the allocation update in background

The SetAllocation response will show the additional parameter UpdateId, which is needed to query for the update result. See the next section on how to query for the update results.

```
<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
  <Success>true</Success>
  <Errors></Errors>
  <Warnings></Warnings>
  <UpdateId>832522158</UpdateId>
</SetAllocationResponse>
```

Listing 16: SetAllocation response for background update

## 3.5 GetUpdateStatus

Use this method to query for the status of a SetAllocation update. It will list the results by channel. Big updates are split into several parts which is reflected in the Parts and ActivePart parameter.

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetUpdateStatus>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>

  <UpdateId>832522158</UpdateId>
</GetUpdateStatus>
```

Listing 17: Example of querying for a background update

```

<?xml version="1.0" encoding="UTF-8" ?>
<GetUpdateStatusResponse>
  <Channels>
    <Channel code="gom">
      <ActivePart>3</ActivePart>
      <Parts>3</Parts>
      <Warnings></Warnings>
      <Errors></Errors>
      <Success>>false</Success>
    </Channel>
    <Channel code="hc">
      <ActivePart>3</ActivePart>
      <Parts>0</Parts>
      <Errors></Errors>
      <Success>>true</Success>
      <Warnings>
        <Warning>
          <WarningId>207</WarningId>
          <WarningMsg>Skipped room type (not setup with channel).</WarningMsg>
        </Warning>
      </Warnings>
    </Channel>
    <Channel code="iwb">
      <ActivePart>0</ActivePart>
      <Errors>
        <ErrorId>15</ErrorId>
        <ErrorMsg>Missing or wrong channel credentials on myallocator.com</ErrorMsg>
      </Errors>
      <Parts>0</Parts>
      <Success>>false</Success>
      <Warnings>
      </Warnings>
    </Channel>
  </GetUpdateStatusResponse>

```

Listing 18: GetUpdateStatus response

Tag	Description
Parts	A big update can be split into smaller parts. This can be used as a progress indicator. 0 parts mean that the update is completed (or aborted due to an error).
ActivePart	The current part number being submitted to the channel.

### 3.6 SetLogin

This method allows you to create and update customer accounts on myallocator.com. Before you can use this method we'll have to explicitly enable you for this functionality, as some aspects with regards to customer payment will need to be discussed.

To create a login and property you would send the following request.

#### 3.6.1 Request - Creating a login and property

```

<?xml version="1.0" encoding="UTF-8"?>
<SetLogin>
  <Auth>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
  <CreateLogin>
    <UserId>New Customer Id</UserId>
    <UserPassword>New Customer Password</UserPassword>
    <CustomerFirstName>Customer first name</CustomerFirstName>
  </CreateLogin>
</SetLogin>

```

```

    <CustomerLastName>Customer family name</CustomerLastName>
    <CustomerEmail>Customer email address</CustomerEmail>
    <PropertyName>Name of property</PropertyName>
    <ExpiryDate>2012-05-05</ExpiryDate>
    <Currency>EUR</Currency>
    <Breakfast>IN</Breakfast>
  </CreateLogin>
</SetLogin>

```

Listing 19: Example of creating a customer account

Tag	Description
UserId	The new customer id to log into myallocator.com. Valid characters: uppercase/lowercase letters, digits, underscore, dash, period, @
UserPassword	The new customer password. Should contain a minimum of 8 characters.
CustomerFirstName	Customer's first name (optional).
CustomerLastName	Customer's family name (optional).
CustomerEmail	Customer's email address.
PropertyName	Name of hotel/hostel/B&B/...
ExpiryDate	Day on which the login to myallocator.com expires. No availability updates (even through the API) can be made after this date. Format: YYYY-MM-DD.
Currency	3-letter ISO 4217 currency code. This is the default currency of the property.
Breakfast	Can be <i>IN</i> (included), <i>EX</i> (excluded) or empty.

### 3.6.2 Response - Creating a login and property

```

<?xml version="1.0" encoding="utf-8"?>
<SetLoginResponse>
  <Success>true</Success>
  <PropertyId>70</PropertyId>
</SetAllocationResponse>

```

Listing 20: Example of a successful *SetLogin* response

To update a login you would send the following request:

### 3.6.3 Request - Updating a login and property

```

<?xml version="1.0" encoding="UTF-8"?>
<SetLogin>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>
  <UpdateLogin>
    <UserPassword>Changed Customer Password</UserPassword>
    <CustomerFirstName>Changed first name</CustomerFirstName>
    <CustomerLastName>Changed family name</CustomerLastName>
    <CustomerEmail>Changed email address</CustomerEmail>
    <PropertyName>Changed name of property</PropertyName>
    <ExpiryDate>2013-05-05</ExpiryDate>
    <Currency>USD</Currency>
    <Breakfast></Breakfast>
  </UpdateLogin>

```

```
</SetLogin>
```

Listing 21: Example of updating a customer account

All tags within *UpdateLogin* are optional.

Tag	Description
UserPassword	Specify the old password within the <i>Auth</i> tag and new the password within the <i>UpdateLogin</i> tag.

### 3.6.4 Response - Updating a login and property

```
<?xml version="1.0" encoding="utf-8"?>
<SetLoginResponse>
  <Success>true</Success>
</SetLoginResponse>
```

Listing 22: Example of a successful *SetLogin* response

### 3.6.5 Passive login

You can provide direct links to different parts of our website without having the customer log in. This is useful to directly link to the room and channel setup on myallocator.com.

To do this, send a POST request to this URL:

<https://myallocator.com/callbacks/plogin.xt>

POST parameter	Value
UserId	The customer id on myallocator.com
UserPassword	The customer password on myallocator.com.
PropertyId	The property ID on myallocator.com.
Destination	<i>room_setup</i> - Setup page for rooms and property settings <i>channel_setup</i> - Setup page to map channel rooms to Myallocator rooms

Please make sure that you send this request using HTTPS. If you do not store the plain password in your database you can also let us know which hash algorithm you use.

## 3.7 GetBookings

This method allows you to query for bookings made to a specific property by booking date, modification date or arrival date.

It is important to know that not every booking that is returned through the API necessarily resulted in an adjustment of the other channels. If the booking is not mapped to any rooms on our system, or if the channel has just been setup, the adjustment will not be carried out.

### 3.7.1 Request - Querying for bookings

```
<?xml version="1.0" encoding="UTF-8"?>
<GetBookings>
  <Auth>
    <UserId>Customer User ID</UserId>
    <UserPassword>Customer Password</UserPassword>
    <PropertyId>Property ID on myallocator.com</PropertyId>
```

```

    <VendorId>Your Vendor ID</VendorId>
    <VendorPassword>Your Vendor Password</VendorPassword>
  </Auth>

  <ArrivalStartDate>2010-01-01</ArrivalStartDate>
  <ArrivalEndDate>2013-01-01</ArrivalEndDate>

  <CreationStartDate>2010-01-01</CreationStartDate>
  <CreationEndDate>2013-01-01</CreationEndDate>

  <ModificationStartDate>2010-01-01</ModificationStartDate>
  <ModificationEndDate>2013-01-01</ModificationEndDate>
</GetBookings>

```

Listing 23: Example of querying for bookings

The requests consists of search criteria by date. Only specify the StartDate/EndDate of one criteria.

Tag	Description
ArrivalStartDate/ArrivalEndDate	Query for date of arrival (first day of staying).
CreationStartDate/CreationStartDate	Query for date of the booking creation on myallocator.com.
ModificationStartDate/ModificationEndDate	Query for date of the booking modification on myallocator.com. A modification can occur if the booking has been cancelled.

### 3.7.2 Response - Querying for bookings

Different channels return a differen amount of information about a booking, therefore many fields are optional.

```

<?xml version="1.0" encoding="utf-8"?>
<GetBookingsResponse>
  <Bookings>
    <Booking>
      <Channel>hb</Channel>
      <StartDate>2012-01-01</StartDate>
      <EndDate>2012-01-03</EndDate>
      <IsCancellation>>false</IsCancellation>
      <MyallocatorId>4d7e47e467458b927c000000</MyallocatorId>
      <MyallocatorCreationDate>2011-03-14</MyallocatorCreationDate>
      <MyallocatorCreationTime>16:52:52</MyallocatorCreationTime>
      <MyallocatorModificationDate>2011-03-14</MyallocatorModificationDate>
      <MyallocatorModificationTime>16:52:52</MyallocatorModificationTime>
      <OrderId>1234</OrderId>
      <OrderDate>2011-11-11</OrderDate>
      <OrderTime>11:11:00</OrderTime>
      <OrderSource>Hostelbookers.com</OrderSource>
      <OrderAdults>2</OrderAdults>
      <OrderChildren>2</OrderChildren>
      <Deposit>12.00</Deposit>
      <TotalPrice>82.00</TotalPrice>
      <TotalCurrency>GBP</TotalCurrency>

    <Customers>
      <Customer>
        <CustomerFName>John</CustomerFName>
        <CustomerLName>Smith</CustomerLName>
        <CustomerAddress>522 Main Rd</CustomerAddress>
        <CustomerArrivalTime>13:00</CustomerArrivalTime>
        <CustomerEmail>john@googlemail.com</CustomerEmail>
        <CustomerNationality>UK</CustomerNationality>
        <CustomerPhone>+44 1234567890</CustomerPhone>
        <CustomerCompany>Johnston Ltd.</CustomerCompany>
      </Customer>
    </Customers>
  </Booking>
</Bookings>
</GetBookingsResponse>

```

```

    <CustomerCity>Stirling</CustomerCity>
    <CustomerState>Stirlingshire</CustomerState>
    <CustomerPostCode>FK8 2HE</CustomerPostCode>
    <CustomerCountry>UK</CustomerCountry>
    <CustomerNote>Bringing a dog</CustomerNote>
  </Customer>
</Customers>

<Rooms>
  <Room>
    <StartDate>2012-01-01</StartDate>
    <EndDate>2012-01-03</EndDate>
    <Price>40.00</Price>
    <Currency>GBP</Currency>
    <RoomTypeIds>
      <RoomTypeId>117</RoomTypeId>
    </RoomTypeIds>
    <RoomDesc>2 peoples (1Double bed)</RoomDesc>
    <Units>2</Units>
  </Room>
  <Room>
    <StartDate>2012-01-02</StartDate>
    <EndDate>2012-01-02</EndDate>
    <Price>42.00</Price>
    <Currency>GBP</Currency>
    <RoomTypeIds>
      <RoomTypeId>119</RoomTypeId>
    </RoomTypeIds>
    <RoomDesc>Dormitory Room</RoomDesc>
    <Units>4</Units>
  </Room>
</Rooms>

</Booking>
</Bookings>
</GetBookingsResponse>

```

Listing 24: Example of a successful *GetBookings* response

Tag	Description
Channel	2-3 letter channel code
StartDate	Date of arrival (YYYY-MM-DD)
EndDate	Date before day of departure (YYYY-MM-DD)
IsCancellation	<i>true</i> or <i>false</i>
MyallocatorId	Booking ID on myallocator.com (hexadecimal number)
MyallocatorCreationDate	Date of booking creation on myallocator.com
MyallocatorCreationTime	Time of booking creation on myallocator.com
MyallocatorModificationDate	Date of booking modification on myallocator.com (in case of a cancellation the modification date is different from the creation date)
MyallocatorModificationTime	Time of booking modification on myallocator.com
OrderId	Booking ID on the channel
Rooms/Room	List of booked rooms
- RoomTypeId	List of booked myallocator.com room IDs. Several channel room can be mapped to one myallocator.com room. If -1 it means that we couldn't match the channel room to one of our rooms.
- Units	Number of booked rooms (for private rooms) or beds (for shared rooms)
- StartDate	Staying in this room from this date on
- EndDate	Leaving in this room on this date
<b>Optional tags:</b>	
OrderSource	Which website the booking originates from
OrderDate	The date the booking has been created on the channel
OrderTime	The time the booking has been created on the channel
OrderAdults	Number of adults
OrderChildren	Number of children
Deposit	Amount of money already taken as a deposit
TotalPrice	Total cost of booking
TotalCurrency	Currency for total cost of booking
Customers/Customer	Customer details (can be more than one)
- CustomerFName	First name
- CustomerLName	Family name
- CustomerEmail	Email address
- CustomerPhone	Phone number
- CustomerAddress	Address
- CustomerCompany	Company name
- CustomerCity	City
- CustomerState	State
- CustomerPostCode	Post code
- CustomerCountry	Country name (not necessarily a 3-letter code)
- CustomerArrivalTime	Time of arrival
- CustomerNote	Note to the property owner
Rooms/Room	Room details
- Price	Price for this room and stay
- Currency	Currency for the price
- RoomDesc	Room description on the channel
- Occupancy	Number of persons this room is booked for
- Breakfast	Wheter breakfast was ordered along with the booking. Can be <i>true</i> or <i>false</i>

## 4 Codes

### 4.1 Channel codes

Code	Full name	Updates up to	Notes
all	All live channels	see below	
hc	Hostelsclub	2 years	
hb	Hostelbookers	1 year	
gom	Gomio	2 years	
iwb	InstantWorldBooking	2 years	
rtg	RatesToGo	variable	
adv	HotelAdvisor	2 years	
hde	Hotel.de	2 years	
esc	Escapio	2 years	
hi	HIHostels	2 years	
rec	Reconline (GDS)	2 years	
bp	BudgetPlaces	2 years	
ct	Cultuzz	2 years	
lmg	LetMeGo	2 years	
(go2)	Go2Hostels	1 year	Channel closed down Jan 1 <sup>st</sup> 2011

### 4.2 Error codes

#### 4.2.1 Global errors

Errors are divided into three categories. A global error will have a single *Errors* tag with no further encapsulation and only a single *Error* tag. An API method specific error will be included in the method name. Again, there will only be a single *Error* tag. The third type of error is channel specific. The *Errors* tag is included in the method name and there may be multiple *Error* tags.

Code	Description
1	Could not parse XML

```
<?xml version="1.0" encoding="utf-8"?>
<Errors>
  <Error>
    <ErrorId>1</ErrorId>
    <ErrorMsg>Could not parse XML</ErrorMsg>
  </Error>
</Errors>
```

Listing 25: Global error example

#### 4.2.2 API method specific errors and warnings

Code	Description
2	Missing authentication tags
3	Invalid user or user password
4	Invalid vendor or vendor password
5	Vendor disabled
6	User has no credit left
7	User has no permission to change availability for this property
8	No such API method
9	Unsupported channel
10	No channels selected
11	No allocations submitted
12	Invalid room type id (does not exist or not assigned to this property)
13	Missing allocation info (price, dates, units)
14	Internal error. Support has been notified!
15	Missing or wrong channel credentials on myallocator.com
16	End date before start date
17	Start date too far in the future (>2 years)
18	Invalid property id
19	Vendor not enabled to use this method
20	Missing required XML fields
21	No applicable dates submitted
22	Invalid update id
301	Invalid characters in new username
302	Username exists already
303	Invalid values (check Breakfast, Currency, ExpiryDate)
401	Invalid date format
402	Missing or invalid search criteria

```
<?xml version="1.0" encoding="utf-8"?>
<GetRoomTypes>
  <Errors>
    <Error>
      <ErrorId>3</ErrorId>
      <ErrorMsg>Invalid user or user password</ErrorMsg>
    </Error>
  </Errors>
</GetRoomTypes>
```

Listing 26: API method specific error example

#### 4.2.3 Channel specific errors

Code	Description
202	Channel skipped (not setup)
203	Channel did not respond
204	Channel skipped (no applicable rooms)
205	Incorrect room setup. The room type mapping needs to be updated on myallocator.com.
206	Channel only updates up to a certain period in the future. Some dates were skipped.
207	Skipped room type (not setup with channel)
208	Partial success: ...
99	<i>Check/Display ErrMsg or WarningMsg content</i>

```
<?xml version="1.0" encoding="utf-8"?>
<SetAllocationResponse>
```

```
<Errors>
  <Error channel="hc">
    <ErrorMsg>
      Provider not responding. Please try again later.
    </ErrorMsg>
  </Error>

  <Error channel="iwb">
    <ErrorMsg>
      Provider not responding. Please try again later.
    </ErrorMsg>
  </Error>
</Errors>
<Success>false</Success>
<Warnings></Warnings>
</SetAllocationResponse>
```

Listing 27: Channel specific error example