

# Application Programming Interface (API) Documentation

07/09/2020

## Copyright Information

© 2020 by Vermont Systems

This document is the property of Vermont Systems and is provided in conjunction with an agreement between the customer and Vermont Systems for Licensed Software or Software as a Service (SaaS). The document(s) and software referred to in this publication may not be copied, distributed, electronically transmitted, posted on the web or altered in any way without the express written consent of Vermont Systems. The information contained in this document is subject to change without notice.

Vermont Systems  
12 Market Place  
Essex Junction, VT 05452

[www.vermontsystems.com](http://www.vermontsystems.com)

## Contents

|   |    |
|---|----|
| <i>comAPI</i> Configuration .....                                   | 5  |
| Web Server Configuration .....                                      | 5  |
| Transaction Server Configuration .....                              | 5  |
| RecTrac Configuration .....   | 5  |
| API Table Search Flow .....   | 7  |
| Authenticate to API .....   | 7  |
| Determine the table to search (Optional) .....                      | 7  |
| Determine the fields to search in a specific table (Optional) ..... | 7  |
| Fetch Table Records.....  | 7  |
| Close Session .....   | 7  |
| API Tee Time Flow .....   | 8  |
| Authenticate to API .....   | 8  |
| Search Tee Times .....  | 8  |
| Set Daily or Member Information.....                                | 8  |
| Add a Tee Time to the Cart.....                                     | 8  |
| Remove a Tee Time from the Cart.....                                | 9  |
| Cancel a Tee Time and Add to the Cart .....                         | 9  |
| Process Payment .....   | 9  |
| Close Session .....   | 9  |
| API Request References.....   | 10 |
| POST <BaseHref>/authenticate/login .....                            | 10 |
| POST <BaseHref>/authenticate/logout.....                            | 10 |
| POST <BaseHref>/authenticate/member/<option>/<data>.....            | 11 |
| GET <BaseHref>/search/<module>.....                                 | 13 |
| GET <BaseHref>/search/<table> .....                                 | 14 |
| GET <BaseHref>/cart/items .....                                     | 15 |
| PUT <BaseHref>/cart/items/<module>/<id> .....                       | 15 |

|  |    |
|--|----|
| DELETE <BaseHref>/cart/items/<id>.....                             | 16 |
| POST <BaseHref>/cart/process .....                                 | 17 |
| GET <BaseHref>/cart/receipt .....                                  | 17 |
| POST <BaseHref>/cart/cancel/<id>.....                              | 18 |
| GET <BaseHref>/search/tables .....                                 | 19 |
| GET <BaseHref>/search/fields/<table> .....                         | 19 |
| POST <BaseHref>/PassMember/Update/<PassType>/<XREF>/<Status> ..... | 20 |
| GET <BaseHref>/PassMember/Lookup/<PassType>/<XREF> .....           | 20 |
| API Table Filtering .....  | 22 |
| Filtering Limitations .....  | 22 |
| Filtering Format .....   | 22 |
| Valid Filter By Values .....                                       | 22 |
| API Standard Error Codes .....                                     | 23 |
| API Request Query References .....                                 | 24 |
| GET <BaseHref>/search/gr .....                                     | 24 |
| PUT <BaseHref>/cart/items/gr/<id> .....                            | 24 |
| API Response References .....                                      | 25 |
| GET <BaseHref>/search/gr .....                                     | 25 |
| GET <BaseHref>/cart/items .....                                    | 25 |
| PUT <BaseHref>/cart/items/<module>/<id> .....                      | 25 |
| DELETE <BaseHref>/cart/items/<id>.....                             | 25 |
| POST <BaseHref>/cart/cancel/<id>.....                              | 25 |
| API Notes .....  | 27 |
| General Notes .....  | 27 |

## comAPI Configuration

### Web Server Configuration

- 1 An API\_Live.wsc file must be created on the web server which points to the WebSpeed broker hostname and port running on the transaction server. This will most likely be the RecTrac client WebSpeed broker.

### Transaction Server Configuration

- 1 In the RecTrac programs directory there is a config.ini file which must be updated to include the Base HREF of the API. The Base HREF will be the domain name with the WSC directory and the name of the \*.wsc file created on the Web Server. Example below:
  - a. [https://www.mydomain.com/wsc/api\_live.wsc/] interface=API
  - b. If there is a specific tenant for your configuration the “tenant=<TenantName>” must also be entered under the configuration section for the API.

### RecTrac Configuration

- 1 Open up Interface Parameters under Management => System Management => Misc Management.
  - a. Click **Add** when the screen finishes loading.
  - b. Enter in a unique **Record Code** which represents this as the API.
  - c. Enter in a **Description** which describes what this interface parameter represents.
  - d. Choose the **Interface Type** of API.
  - e. Enter the **Base HREF** which will be the same as the one entered in the config.ini.
  - f. All other fields are optional and can be left as default values.
  - g. Click Save to save this interface parameter.
- 2 Open up User Management under Management => System Management => User/Menu/Profile Management.
  - a. Click **Add** when the screen finishes loading.
  - b. Enter a **User Name** that will be used for authentication to the API and for all transactions through the API.
  - c. Choose a **User Group(s)** from the picklist to assign this user to.
  - d. Fill out any other fields that may be required.
  - e. Click Save to save this user.

**Note:** The next step requires an access code from Vermont Systems to enter the License Profile in RecTrac.

- 3 Open up Profile Management under Management => System Management => User/Menu/Profile Management.
  - a. Find the **License** profile in the datagrid and selected it.
  - b. Click **Change** and on the screen enter in the **Access Code** from Vermont Systems.
  - c. Make sure the **Customer ID** field is filled in with the proper customer ID.
  - d. Under the **API Modules** choose the API modules required for your use case.
  - e. Click **Generate API Key**. In the **API Key** field copy the API key and save as this will be required for communication with the API.

## API Table Search Flow

### Authenticate to API

- 1 Send a POST to:  
`<BaseHREF>/authenticate/login?APIKey=<APIKey>&username=<Username>`
- 2 Store the **sessionID** in the response back from this process.

### Determine the table to search (Optional)

- 1 Send a GET to:  
`<BaseHREF>/search/tables?SessionID=<SessionID>`
- 2 This returns an array of tables in the database.

### Determine the fields to search in a specific table (Optional)

- 1 Send a GET to:  
`<BaseHREF>/search/fields/<Table>?SessionID=<SessionID>`
- 2 This returns an array of fields in the table specified. Note that there are both database fields and calculated fields in this array.

### Fetch Table Records

- 1 Send a GET to:  
`<BaseHREF>/search/get/<Table>?SessionID=<SessionID>&fields=<FieldList>`
- 2 This returns an array of records in the table specified. Additional parameters can be passed to filter the records returned as well as the fields to send back in the response.

### Close Session

- 1 Send a POST to:  
`<BaseHREF>/authenticate/logout?SessionID=<SessionID>`
- 2 This will close the session and cleanup any leftover session information.

## API Tee Time Flow

**Note:** Below is a standard flow basing the flow on a guest search, choosing a tee time, setting the member information, adding a tee time to the cart and finishing the payment process.

### Authenticate to API

- 3 Send a POST to:  
`<BaseHREF>/authenticate/login?APIKey=<APIKey>&username=<Username>`
- 4 Store the **sessionID** in the response back from this process.

### Search Tee Times

- 1 Send a GET to:  
`<BaseHREF>/search/get/gr?fields=*&players=1&count=5&SessionID=<SessionID>&time=9:00am&date=5/1/2016`
- 2 This request will get tee times for a single player at 9:00am on 5/1/2016 and will return 5 results.
- 3 In the response the **records** will include all the information on the tee time. Use the **cart\_link** field for the purchase of a tee time.

### Set Daily or Member Information

- 1 For a Daily Player send a POST to:  
`<BaseHREF>/authenticate/member/daily?SessionID=<SessionID>&firstname=<FirstName>&lastname=<LastName>`
- 2 For a Member who has their Pass Number send a POST to:  
`<BaseHREF>/authenticate/member/passnumber/<PassNumber>?SessionID=<SessionID>`
- 3 Use the **success** property of the response to determine if the daily player or member were set properly.

### Add a Tee Time to the Cart

- 1 Send a PUT to:  
`<BaseHREF>/cart/items/gr/<ID>?players=1&holes=18&SessionID=<SessionID>`
- 2 Use the **success** property of the response to determine if the tee time was purchased properly or not. Also returned will be **records** which are all of the items added to the cart. If more than 1 player



or other tee times have been purchased there will be multiple records in the array. Included in the record is also the **remove\_link** field which can be used to remove the item later from the cart.

## Remove a Tee Time from the Cart

- 1 Send a DELETE to:  
`<BaseHref>/cart/items/<ID>?SessionID=<SessionID>`
- 2 Use the **success** property of the response to determine if the tee time was removed properly or not. Also returned will be **records** which are all of the items added to the cart. If more than 1 player or other tee times have been purchased there will be multiple records in the array.

## Cancel a Tee Time and Add to the Cart

- 1 Send a DELETE to:  
`<BaseHref>/cart/cancel/<ID>?SessionID=<SessionID>`
- 2 The `<ID>` in the URL will be the Confirmation Number which can be found on the receipt and is also the `sadetail_id` field when viewing an item in the cart prior to the payment process.
- 3 Use the **success** property of the response to determine if the cancel process was successful or not. Also returned will be **records** which are all of the items added to the cart. If more than 1 player or other tee times have been purchased there will be multiple records in the array.

## Process Payment

- 1 Send a POST to:  
`<BaseHref>/cart/process?SessionID=<SessionID>`
- 2 Use the **success** property of the response to determine if the payment process was successful or not. Also returned will be the **receipt\_link** URL which can be used to get a copy of the receipt generated.

## Close Session

- 3 Send a POST to:  
`<BaseHref>/authenticate/logout?SessionID=<SessionID>`
- 4 This will close the session and anything left in the cart will be removed.

## API Request References

### POST <BaseHREF>/authenticate/login

**Purpose:**

This is the first process for all requests to the API which authenticates the API Key, sets the RecTrac User and returns a proper Session ID for user for all other requests.

**Request Path:**

N/A

**Request Query:**

*APIKey [String] (Required):* The API Key from the License Profile must be sent on this request.

*Username [String] (Required):* The user that was setup in RecTrac for all API transactions must be sent.

**Response:**

*success [Boolean]:* True/False depending on if the process succeeded or not.

*sessionID [String]:* The Session ID which should be used for all subsequent requests after this one.

**Error Codes:**

1005: "Unable to write to table %1".

99970: "The username must be sent."

99971: ""The user is inactive, locked out or not allowed to login at this time."

**Example:**

[https://www.mydomain.com/wsc/api\\_live.wsc/authenticate/login?APIKey=1234567890&Username=API](https://www.mydomain.com/wsc/api_live.wsc/authenticate/login?APIKey=1234567890&Username=API)

### POST <BaseHREF>/authenticate/logout

**Purpose:**

This request will logout an API Session already created. This should be done at the end of all processes to keep sessions from lingering longer than they have to.

**Request Path:**

N/A

**Request Query:**

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

**Response:**

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

**Error Codes:**

N/A

**Example:**

[https://www.mydomain.com/wsc/api\\_live.wsc/authenticate/logout?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/authenticate/logout?SessionID=[SESSIONID])

## POST <BaseHref>/authenticate/member/<option>/<data>

**Purpose:**

Authenticate a member and set it for the duration of a session. The member can be found by a pass number, person ID, Pass ID or simply be a daily player who does not have a household in the RecTrac database.

**Request Path:**

*option (Required)*: This must be one of the follow: PassNumber, SAPersonID, PassID or Daily.

*data (Optional)*: If using PassNumber, SAPersonID or PassID options should be the ID.

**Request Query:**

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

*FirstName [String] (Optional)*: The first name of the person when set to the "Daily" option.

*LastName [String] (Optional)*: The last name of the person when set to the "Daily" option.

*Address1 [String] (Optional)*: The address line 1 of the person when set to "Daily" option.

*Address2 [String] (Optional)*: The address line 2 of the person when set to "Daily" option.

*City [String] (Optional)*: The city of the person when set to "Daily" option.

*State [String] (Optional)*: The state of the person when set to "Daily" option.

*ZipCode [String] (Optional)*: The zip code of the person when set to "Daily" option.

*Phone [String] (Optional)*: The phone number of the person. Can be set on any option.

*Email [String] (Optional)*: The email address of the person. Can be set on any option.

### **Response:**

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

*firstName [String]*: The first name of the person.

*lastName [String]*: The last name of the person.

*address1 [String]*: The address line 1 of the person.

*address2 [String]*: The address line 2 of the person.

*city [String]*: The city of the person.

*state [String]*: The state of the person.

*zipCode [String]*: The zip code of the person.

*phoneNumber [String]*: The phone number of the person.

*emailAddress [String]*: The email address of the person.

*householdNumber [String]*: The RecTrac household number of the household used. 999999999 = Daily

*memberID [String]*: The RecTrac member ID of the member used.

### **Error Codes:**

99990: "Only PassNumber, PersonID, PassID and Daily options are valid."

99991: "A Daily first and last name are required."

99992: "A Daily Household was not found."

99993: "A Daily Member was not found."

99994: "A valid Pass Number, Member ID or Pass ID must be set."

99995: "A household has already been set. All transactions must be completed in order to change households."

99996: "Household not found."

99997: "Member not found."

**Example:**

`https://www.mydomain.com/wsc/api_live.wsc/authenticate/member/daily?SessionID=[SESSIONID]&FirstName=John&LastName=Doe`

**GET <BaseHref>/search/<module>****Purpose:**

Search through a specific module and return results that can be used for purchase. Currently Tee Times (GR Module) are the only results available for querying.

**Request Path:**

*module (Required):* A valid RecTrac module. Currently **GR** is the only module available.

**Request Query:**

*SessionID [String] (Required):* The Session ID from the response of the Authenticate/Login request.

*Fields [String] (Required):* Use \* to show all fields or make this a comma delimited list of fields. See **API Response References** section for possible output fields.

*Count [Number] (Optional):* Optional max count of records to fetch. Defaults to 100 if not set.

*Per [Number] (Optional):* Optional count per page if using paging. Defaults to 10 if not set.

*Page [Number] (Optional):* Optional page to view of results. Defaults to 1 if not set.

**Response:**

*success [Boolean]:* True/False depending on if the process succeeded or not.

*sessionID [String]:* The Session ID used for this request.

*records [Array]:* See **API Response References** section.

*total [Number]:* This is the total number of records found.

*page [Number]:* This is the page currently viewed if using paging. If not using paging this will be 1.

*start [Number]:* This is the start record number in the records received.

*end [Number]:* This is the end record number in the records received.

*pages [Number]:* This is the total number of pages based off per page total and total results found.

**Error Codes:**

1006: "The '%1' parameter was not set."

1009: "No filters were added cannot process all records in a table."

1011: "No columns were added cannot process without any output."

1012: "The module '%1' is not licensed."

#### Example:

```
https://www.mydomain.com/wsc/api_live.wsc/search/gr?SessionID=[SESSIONID]&fields=*&date=12/31/2016&players=1&holes=18&count=5
```

## GET <BaseHref>/search/<table>

#### Purpose:

Search through a specific table for records in the database.

#### Request Path:

*table* (Required): A valid RecTrac database table.

#### Request Query:

*SessionID* [String] (Required): The Session ID from the response of the Authenticate/Login request.

*Fields* [String] (Required): Use \* to show all fields or make this a comma delimited list of fields. See **API Response References** section for possible output fields. If using \* no calculated fields will be used.

*Count* [Number] (Optional): Optional max count of records to fetch. Defaults to 100 if not set.

*Filters* [Numerous] (Optional): See **API Table Filtering** section.

#### Response:

*success* [Boolean]: True/False depending on if the process succeeded or not.

*sessionID* [String]: The Session ID used for this request.

*records* [Array]: See **API Response References** section.

#### Error Codes:

1006: "The '%1' parameter was not set."

1009: "No filters were added cannot process all records in a table."

1011: "No columns were added cannot process without any output."

#### Example:

```
https://www.mydomain.com/wsc/api_live.wsc/search/arsection?SessionID=[SESSIONID]&fields=*
```

## GET <BaseHref>/cart/items

### Purpose:

Fetch all the items in the cart for the session passed in.

### Request Path:

N/A

### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

### Response:

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

*records [Array]*: See **API Response References** section.

### Example:

`https://www.mydomain.com/wsc/api_live.wsc/cart/items?SessionID=[SESSIONID]`

## PUT <BaseHref>/cart/items/<module>/<id>

### Purpose:

Add an item to your cart. Currently only tee times can be added to the cart.

### Request Path:

*module [String] (Required)*: The module to purchase an item for. Currently GR is the only module available.

*id [Number] (Required)*: The ID of the item you are purchasing. This is the ID in the RecTrac database of the item being purchased. This can be found in a number of ways but is best used in conjunction with the search results.

### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

See **API Request Query References** section for other possible parameters.

### Response:

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

*records [Array]*: See **API Response References** section.

#### **Error Codes:**

1006: "The '%1' parameter was not set."

1012: "The module '%1' is not licensed."

76000: "The household and family member must be set prior to adding to cart."

76001: "Guest household is not allowed for this particular transaction."

76002: *Various errors depending on response from add to cart process.*

76005: "Unable to process action due to processing errors."

#### **Example**

*[https://www.mydomain.com/wsc/api\\_live.wsc/cart/items/gr/12345?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/cart/items/gr/12345?SessionID=[SESSIONID])*

## **DELETE <BaseHref>/cart/items/<id>**

#### **Purpose:**

Remove an item from the cart.

#### **Request Path:**

*id [Number] (Optional)*: If the ID is not sent it will empty the cart. If the ID is sent it will remove this one item from the cart. See **API Response References** section for more information on the `remove_link` value.

#### **Request Query:**

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

#### **Response:**

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

*records [Array]*: See **API Response References** section.

#### **Example**

*[https://www.mydomain.com/wsc/api\\_live.wsc/cart/items/09876?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/cart/items/09876?SessionID=[SESSIONID])*



## POST <BaseHref>/cart/process

### Purpose:

Process all of the items in the cart creating a receipt and finalizing the items in the RecTrac database.

### Request Path:

N/A

### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

*Email [String] (Optional)*: Optional email address to send the receipt to.

### Response:

*success [Boolean]*: True/False depending on if the process succeeded or not.

*sessionID [String]*: The Session ID used for this request.

*receipt\_link [String]*: A full URL to fetch the receipt for this transaction.

### Error Codes:

77000: "Nothing in the cart to process."

77001: "The household has a balance that must be paid off in RecTrac first."

77002: Various payment errors that come from the payment process.

### Example

[https://www.mydomain.com/wsc/api\\_live.wsc/cart/process?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/cart/process?SessionID=[SESSIONID])

## GET <BaseHref>/cart/receipt

### Purpose:

Fetch a receipt from a previous payment process.

### Request Path:

N/A

### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

**Response:**

Raw PDF receipt is sent back in this request.

**Error Codes:**

78000: "A Receipt Number is required in order to fetch a receipt."

78001: "Unable to find a receipt with receipt number '%1'."

**Example**

`https://www.mydomain.com/wsc/api_live.wsc/cart/receipt/1000?SessionID=[SESSIONID]`

**POST <BaseHref>/cart/cancel/<id>****Purpose:**

Cancel an item that has been processed and finished.

**Request Path:**

*id [Number] (Required):* The ID of the record to cancel. For tee times this is the confirmation ID which can be viewed on the receipt.

**Request Query:**

*SessionID [String] (Required):* The Session ID from the response of the Authenticate/Login request.

**Response:**

*success [Boolean]:* True/False depending on if the process succeeded or not.

*sessionID [String]:* The Session ID used for this request.

*records [Array]:* See **API Response References** section.

**Error Codes:**

1007: "Unable to find a record for table %1."

1012: "The module '%1' is not licensed."

79000: "A valid ID must be entered."

79001: "You cannot Cancel, Change, or Renew an Item with a %1 status."

79002: Various cancellation errors returned from cancellation process.

79003: "The household set is different from the household on the cancelled record. Finish the previous process before cancelling."

79004: "Unable to process action due to processing errors."

#### Example

`https://www.mydomain.com/wsc/api_live.wsc/cart/cancel/123456?SessionID=[SESSIONID]`

## GET <BaseHref>/search/tables

#### Purpose:

Fetch all tables available in the database to fetch records from.

#### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

#### Response:

*success [Boolean]*: True/False depending on if the process succeeded or not.

*tables [Array]*: See **API Response References** section.

#### Example

`https://www.mydomain.com/wsc/api_live.wsc/search/tables?SessionID=[SESSIONID]`

## GET <BaseHref>/search/fields/<table>

#### Purpose:

Fetch all tables available in the database to fetch records from.

#### Request Path:

*table [String] (Required)*: The table to fetch fields from

#### Request Query:

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

#### Response:

*success [Boolean]*: True/False depending on if the process succeeded or not.

*fields [Array]*: See **API Response References** section.

**Example**

[https://www.mydomain.com/wsc/api\\_live.wsc/search/fields/arsection?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/search/fields/arsection?SessionID=[SESSIONID])

**POST**

**<BaseHref>/PassMember/Update/<PassType>/<XREF>/<Status>**

**Purpose:**

This request will update someone's pass status based on Member X-Ref for Active, Revoked and/or Expired passes.

**Request Path:**

*Pass Type (Required) The Pass Code*

*X-Ref (Required) The member's XRef value*

*Status (Required) The Pass Status*

**Request Query:**

*SessionID [String] (Required): The Session ID from the response of the Authenticate/Login request.*

**Response:**

*success [Boolean]: True/False depending on if the process succeeded or not.*

**Error Codes:**

N/A

**Example:**

[https://www.mydomain.com/wsc/api\\_live.wsc/passmember/update/SWIM/123456/Active?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/passmember/update/SWIM/123456/Active?SessionID=[SESSIONID])

**GET <BaseHref>/PassMember/Lookup/<PassType>/<XREF>**

**Purpose:**

This request will look up Active Members by X-Ref.

**Request Path:**

*Pass Type (Required) The Pass Code*

*X-Ref (Required)* The member's XRef value

**Request Query:**

*SessionID [String] (Required)*: The Session ID from the response of the Authenticate/Login request.

**Response:**

*success [Boolean]*: True/False depending on if the process succeeded or not.

*success [Boolean]*: True/False depending on if the Member's Pass is Active status.

**Error Codes:**

N/A

**Example:**

[https://www.mydomain.com/wsc/api\\_live.wsc/passmember/lookup/SWIM/123456?SessionID=\[SESSIONID\]](https://www.mydomain.com/wsc/api_live.wsc/passmember/lookup/SWIM/123456?SessionID=[SESSIONID])

## API Table Filtering

### Filtering Limitations

- Filtering can only be done on database fields.
- Filtering cannot be done on calculated fields.
- Filtering can only be done once per database field.

### Filtering Format

Each filter is formatted with the full field name which is both the table name and the database field name with an underscore in between. Example: *arsection\_activitynumber* is the name of the database table ARSection field ActivityNumber.

Each field has to parts of a filter, the value and the filter by option. By default when the filter by option is left out the filter is considered an *equals*. The format for each filter is full field name underscore *filter* or full field name underscore *filterby*. Example: to get all activities that begin with the number 1 the filter would be *arsection\_activitynumber\_filter=1&arsection\_activitynumber\_filterby=begins*.

### Valid Filter By Values

- eq: Equals a specific value.
- ne: Not equals a specific value.
- begins: Begins with a specific value. Can only be used with character based value fields.
- lt: Less than a specific value
- le: Less than or equal to a specific value
- gt: Greater than a specific value
- ge: Greater than or equal to a specific value
- contains: Contains a specific value. Can use asterisks for wildcarding. Can only be used with character based value fields.

## API Standard Error Codes

**Note:** Below are standard error codes which may show up in the course of a request. Anytime a %n is in the error code this represents a dynamic value at run time that will change according to the outcome.

- 1000: "Authorization required."
- 1001: "HTTPS Required."
- 1002: "API Key invalid or not available."
- 1003: "Invalid location"
- 1004: "The method used is not allowed."
- 1005: "Unable to write to table %1."
- 1006: "The '%1' parameter was not set."
- 1007: "Unable to find a record for table %1."
- 1008: "The value of '%2' for field '%1' is invalid."
- 1009: "No filters were added cannot process all records in a table."
- 1010: "The value '%1' is not a valid value for the '%2' parameter."
- 1011: "No columns were added cannot process without any output."
- 1012: "The module '%1' is not licensed."
- 1013: "No API Parameter found."

## API Request Query References

### GET <BaseHref>/search/gr

**Note:** Below are the additional parameters that can be used during searching tee times.

*course [Number]:* The course number. This can be found in RecTrac when creating courses.

*players [Number]:* The number of players to search and eventually purchase a tee time.

*date [String]:* The date (mm/dd/yyyy) to search tee times on.

*time [String]:* The time (hh:mm am) to search tee times on.

*holes [String]:* The number of holes to search for. Valid options are 9, 18 or 9/18.

### PUT <BaseHref>/cart/items/gr/<id>

**Note:** Below are the additional parameters that can be used when purchasing a tee time.

*players [Number]:* The number of players that will be purchasing the tee time.

*holes [String]:* The number of holes the player(s) will be purchasing.

*notes [String]:* Any notes to attach to this tee time.



## API Response References

### GET <BaseHref>/search/gr

**Note:** Below are the fields in the **records** array which make up a single record.

*cart\_link [String]:* Full URL to add this tee time to the cart. Includes the method as the first part of the value.  
Example: "PUT http://www.mydomain.com/..."

*description [String]:* The course description setup on the course in RecTrac.

*tee\_time [Number]:* The time in seconds from midnight for this tee time.

*course [Number]:* The course number for the course setup in RecTrac.

*tee\_date [String]:* The date (yyyy-mm-dd) of the tee time.

*holes [String]:* The number of holes for the tee time.

*tee [String]:* Starting tee for the tee time (Front or Back).

*total\_slots [Number]:* Total slots for this tee time. Max of 5.

*open\_slots [Number]:* Total open slots for this tee time. Max of 5.

*available [Boolean]:* If this tee time is available or not. This includes rule checking that may cause a tee time to be unavailable due to various rules setup on the course.

### GET <BaseHref>/cart/items

### PUT <BaseHref>/cart/items/<module>/<id>

### DELETE <BaseHref>/cart/items/<id>

### POST <BaseHref>/cart/cancel/<id>

**Note:** Below are the fields in the **records** array which make up a single record.

*sadetail\_id [Number]*: This is the ID of the record in the cart. This ID is used for removing it from the cart.

*sadetail\_description [String]*: The description of what is in the cart.

*sadetail\_firstname [String]*: The first name of the person purchasing this item.

*sadetail\_lastname [String]*: The last name of the person purchasing this item.

*sadetail\_hhaddress1 [String]*: The address line 1 of the person purchasing this item (Daily or Household)

*sadetail\_hhaddress2 [String]*: The address line 2 of the person purchasing this item (Daily or Household)

*sadetail\_hhcitystatezip [String]*: The city/state/zip of the person purchasing this item (Daily or Household)

*sadetail\_hhemail [String]*: The email of the person purchasing this item (Daily or Household)

*sadetail\_hhphone [String]*: The phone number of the person purchasing this item (Daily or Household)

*sadetail\_householdnumber [Number]*: The household number of the household attached to this purchase.  
999999999 = Daily.

*sadetail\_totalamount [Decimal]*: The total amount of the purchased item. Not used currently.

*sadetail\_previouspaid [Decimal]*: The previously paid amount on this item. Not used currently.

*sadetail\_totaldue2 [Decimal]*: The total due of the purchased item. Not used currently.

*sadetail\_recordstatus [String]*: The status of this item which varies depending on the module. Reserved for Tee Times.

*sadetail\_cartstatus [String]*: The cart status of this item which varies based off what process this item is in.

*remove\_link [String]*: This is a full URL that can be used to remove this item from the cart.

## API Notes

### General Notes

- All communication to the API must be over an HTTPS connection unless in development mode and this should only be set for development purposes only.
- The API Key is only required on the first request to the Authenticate/Login process, after this the API Key is no longer needed.
- The Session ID which is returned on the first Authenticate/Login request must be sent to the API for each subsequent request.
- All requests return JSON response unless otherwise noted above.
- All parts of the URI are case insensitive meaning any case can be used.
- The WebTrac timeout values in Static Parameter are used for the API to determine when a session can be cleaned up.
- No fees are currently calculated during the purchase process. Fees are completely skipped in the API.
- No questions are currently used during the purchase process. Questions are completely skipped in the API.