Gimmal

REST Services

Gimmal Physical

Contents

Version History

Version	Approved By	Effective Date	Product Version	Description of Change
1	Terry Butler	02/15/2022	3.11	Created for Gimmal Version 3.11
2	Marta Farensbach	12/1/2022	3.12	Formatting updates

Gimmal

Before You Begin

The code snippets contained in this document call the following SetHeaders method. To use it, replace [USERNAME] and [PASSWORD] with the username and password of the account you are using to access the web service.

Sample Code

```
void SetHeaders(HttpWebRequest webRequest, string method, string body)
{
	webRequest.Method = method;
	webRequest.Credentials = new NetworkCredential("[USERNAME] ", "[PASSWORD]");
	webRequest.PreAuthenticate = true;
	webRequest.AllowWriteStreamBuffering = true;
	if (method == "POST" || method == "PUT")
	{
	webRequest.ContentType = "application/json";
	webRequest.ContentLength = body.Length;
	try
	{
		StreamWriter sw = new StreamWriter(webRequest.GetRequestStream());
		sw.Write(body);
		sw.Close();
	}
		catch (Exception ex){}
	}
}
```

Action

To define Actions available for a given Tab ID

Request Type

Get

URL

http://[ServerName]:[Port ID]/api/Action/[ID]

Implementation Notes

Get a list of possible Actions that can be performed for a Tab

Accepts Inputs

- Integer with the value of the Tab's ID

Input Parameter Type

- URL

Returns

- An IEnumerable<IRAction> object with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRAction
{
	Action (string, optional): name of the action,
```

```
        Caption (string, optional): human readable caption for this action,
        UrlCommand (string, optional): used to specify the update command from the
        phone,
        PostObject (string, optional): Object to include as post thing,
        AssociatedTabId (integer, optional): Tab id if needed for this action,
        AssociatedTabName (string, optional): Tab name if needed for this action,
        AssociatedColumnName (string, optional): Column name if needed for this action
}
```

Sample Code

```
public partial class IRAction
{
        public virtual String Action { get; set; }
        public virtual String Caption { get; set; }
        public virtual String UrlCommand { get; set; }
        public virtual String PostObject { get; set; }
        public virtual Int32 AssociatedTabId { get; set; }
        public virtual String AssociatedTabName { get; set; }
        public virtual String AssociatedColumnName { get; set; }
}
public IEnumerable<IRAction> GetActions()
{
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Action/201");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IEnumerable<IRAction> objActions =
        JsonConvert.DeserializeObject<IEnumerable<IRAction>>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objActions;
}
```

Dictionary

To get Data Dictionary Fields for a Tab

Request Type

Post

URL

http://[Servername]:[Port ID]/api/Dictionary

Implementation Notes

Returns an IRDictionary object containing Captions and Column Names of the given Tab ID from Infolinx's Data Dictionary.

Accepts Inputs

- Integer with the value of the applicable Tab ID
- String containing a SpecialSearch query. This must currently always be set to "quicksearch"

Input Parameter Type

- Application/json body

Input Parameter Class

```
IRDictionarySearch {
```

```
TabId (integer, required): item type to grab dictionary fields for,
SpecialSearch (string, required): special queries; must be set to “quicksearch”
}
```

Returns

- An IEnumerable<IRDictionary> object with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRDictionary {
	Caption (string, optional): Caption for field,
	ColumnName (string, optional): Column name for field
}
```

Sample Code

```
public partial class IRDictionarySearch
{
	public virtual Int32 TabId { get; set; }
	public virtual String SpecialSearch { get; set; }
}
public partial class IRDictionary
{
	public virtual String Caption { get; set; }
	public virtual String ColumnName { get; set; }
}
public IEnumerable<IRDictionary> GetIRDictionary()
{
	// w is a string containing the http://[Servername]:[Port ID] part of the URL
	IRDictionarySearch Srch = new IRDictionarySearch();
	Srch.TabId = 201;
	Srch.SpecialSearch = "quicksearch";
	string body = JsonConvert.SerializeObject(Srch);
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Dictionary");
	SetHeaders(webRequest, "POST", body);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IEnumerable<IRDictionary> objDictionary =
	JsonConvert.DeserializeObject<IEnumerable<IRDictionary>>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	webResponse = null;
	return objDictionary;
}
```

Item

Create/Update an Item

Request Type

Post

URL

http://[Servername]:[Port ID]/api/Item

Implementation Notes

Update/Create an Item.
When creating, you must provide an IRColumn array entry for each field that is required in the Item table for that Tab, with at least IRColumn.ColumnName and IRColumn.Value provided for each.
For updating you must provide an Item ID as a new entry in the IRColumn array input (not in the IRItem.id) or else a new Item will be created. **Note:** user must have rights within Gimmal Physial to modify the Item or the update will fail.

Accepts Inputs

- An IRItem with an array of IRColumns embedded in it; see Input Parameter Format below

Input Parameter Type

- Application/json body

Input Parameter Format

```
IRItem {
	columns (array[IRColumn], optional): columns, corresponds to view,
	TabSingularName (string, optional): singular name for item's tab,
	EdocFile (string, optional): Base64 encoded string of the edoc file. If
	creating, must provide a column value for I_E_USER_FILE_NAME,
	ContentType (string, optional): content type for uploading edoc file,
	Id (integer, optional): item id of item,
	QuickDescription (string, optional): Quick Description,
	Barcode (string, optional): Barcode,
	TabId (integer, optional): Tab Id
}
IRColumn
{
	ColumnName (string, optional): Name of column,
	Caption (string, optional): caption of column,
	Value (string, optional): value of column,
	Type (string, optional): data type of column.
}
```

Returns

• An IRItemResultList object with a Message, Result, and array of IRItemResult fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRItemResultList
{
	Message (string, optional): overall result string for the list,
	Result (boolean, optional): true if all succeeded, false if any failed,
	ItemResults (array[IRItemResult], optional): the list of individual item
	results
}
IRItemResult
{
	LocationBarcode (string, optional): location barcode if applicable for the
action,
	LocationDescription (string, optional): location QD if applicable for the
action,
	ItemId (integer, optional): Item Id,
	ItemBarcode (string, optional): item barcode,
```

```
        ItemDescription (string, optional): item QD,
        RequestId (integer, optional): request id if this was a request,
        Result (boolean, optional): successful or not,
        Reason (integer, optional): corresponds to some enum for harcoded results,
        Message (string, optional): may contain some extra info about why this failed
}
```

Sample Code

```
// ****************************************
// * Item - Create/Update an Item
// ****************************************
public partial class IRItem
{
        public virtual List<IRColumn> columns { get; set; }
        public virtual String TabSingularName { get; set; }
        public virtual String EdocFile { get; set; }
        public virtual String ContentType { get; set; }
        public virtual Int32 Id { get; set; }
        public virtual String QuickDescription { get; set; }
        public virtual String Barcode { get; set; }
        public virtual Int32 TabId { get; set; }
}
public partial class IRColumn
{
        public virtual String ColumnName { get; set; }
        public virtual String Caption { get; set; }
        public virtual String Value { get; set; }
        public virtual String Type { get; set; }
}
public partial class IRItemResultList
{
        public virtual String Message { get; set; }
        public virtual bool Result { get; set; }
        public virtual IRItemResult[] ItemResults { get; set; }
}
public partial class IRItemResult
{
        public virtual String LocationBarcode { get; set; }
        public virtual String LocationDescription { get; set; }
        public virtual Int32 ItemId { get; set; }
        public virtual String ItemBarcode { get; set; }
        public virtual String ItemDescription { get; set; }
        public virtual Int32 RequestId { get; set; }
        public virtual bool Result { get; set; }
        public virtual Int32 Reason { get; set; }
        public virtual String Message { get; set; }
}
public IRItemResultList CreateUpdateItem()
{
        // w is a string containing the http://[Servername]:[Port ID] part of the URL
        // IRColumns -- To create a new Item, you must provide at a minimum an IRColumn
        object
        // for each of that Item's requred fields for that Infolinx implementation,
        // though additional Item fields are allowed. The following three fields were
        required for
        // the Infolinx implementation that was used to create the sample code;
        // the fields required for your implementation will be different
        IRColumn IRC0 = new IRColumn();
```

```
        IRC0.ColumnName = "FK_ITEM_ORGANIZATION_ID";
        // IRC.Caption = "";
        IRC0.Value = "501";
        // IRC.Type = "";
        IRColumn IRC1 = new IRColumn();
        IRC1.ColumnName = "CONTENTS_RANGE___DATE";
        IRC1.Value = "1/1/2014-2/2/2015";
        IRColumn IRC2 = new IRColumn();
        IRC2.ColumnName = "I_IT_ID";
        IRC2.Value = "202";
        // Updating an existing Item requires adding the following 3 lines substituting
        the
        Item's ID for the "303" value
        // Failure to specify an Item ID will result in the creation of a new Item
        // IRColumn IRC3 = new IRColumn();
        // IRC3.ColumnName = "I_ID";
        // IRC3.Value = "303";
        IRItem IRI = new IRItem();
        IRI.columns = new List<IRColumn> { };
        IRI.columns.Add(IRC0);
        IRI.columns.Add(IRC1);
        IRI.columns.Add(IRC2);
        // IRI.columns.Add(IRC3);
        // Alternative shorthand method:
        // IRI.columns = new List<IRColumn> {
        // new IRColumn { ColumnName="FK_ITEM_ORGANIZATION_ID", Value="302"},
        // new IRColumn { ColumnName="CONTENTS_RANGE___DATE", Value="1/1/2014-
        1/1/2015"},
        // new IRColumn { ColumnName="I_IT_ID", Value="202"}
        // };
        List<IRItem> iList = new List<IRItem>();
        iList.Add(IRI);
        string body = JsonConvert.SerializeObject(iList);
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Item");
        SetHeaders(webRequest, "POST", body);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRItemResultList objNewItem =
        JsonConvert.DeserializeObject<IRItemResultList>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        webResponse = null;
        return objNewItem;
}
```

Delete an Item by its ID

Request Type

Delete

URL

http://[Servername]:[Port ID]/api/Item/[ID]

Implementation Notes

Deletes the Item that corresponds to the supplied Item ID

Accepts Inputs

- Integer Item ID

Input Parameter Type
URL
Returns
- An IRItemResult object with fields described in Response Class below

Response Content Type
Application/json
Response Class

```
IRItemResult
{
	LocationBarcode (string, optional): location barcode if applicable for the
action,
	LocationDescription (string, optional): location QD if applicable for the
action,
	ItemId (integer, optional): Item Id,
	ItemBarcode (string, optional): item barcode,
	ItemDescription (string, optional): item QD,
	RequestId (integer, optional): request id if this was a request,
	Result (boolean, optional): successful or not,
	Reason (integer, optional): corresponds to some enum for harcoded results,
	Message (string, optional): may contain some extra info about why this failed
}
```

Sample Code

```
// ***************************************************************
// * Item - Delete an Item
// * Sets column I_Deleted to true in the Item's database table
// ***************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 1203 is the ID of the Box to be deleted
// objItemResult.Result will be true if successful
public partial class IRItemResult
{
	public virtual String LocationBarcode { get; set; }
	public virtual String LocationDescription { get; set; }
	public virtual Int32 ItemId { get; set; }
	public virtual String ItemBarcode { get; set; }
	public virtual String ItemDescription { get; set; }
	public virtual Int32 RequestId { get; set; }
	public virtual bool Result { get; set; }
	public virtual Int32 Reason { get; set; }
	public virtual String Message { get; set; }
}
public IRItemResult DeleteItem()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Item/1203");
	SetHeaders(webRequest, "DELETE", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IRItemResult objItemResult =
	JsonConvert.DeserializeObject<IRItemResult>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	return objItemResult;
```

}

Get an Item by its ID

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Item/[ID]

Implementation Notes

Get an Item by its Item ID.

Accepts Inputs

- Integer Item ID

Input Parameter Type

URL

Returns

- An IRItem object containing an array of IRColumns with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRItem {
	columns (array[IRColumn], optional): columns, corresponds to view,
	TabSingularName (string, optional): singular name for item's tab,
	EdocFile (string, optional): Base64 encoded string of the edoc file. If
	creating, must provide a column value for I_E_USER_FILE_NAME,
	ContentType (string, optional): content type for uploading edoc file,
	Id (integer, optional): item id of item,
	QuickDescription (string, optional): Quick Description,
	Barcode (string, optional): Barcode,
	TabId (integer, optional): Tab Id
}
IRColumn {
	ColumnName (string, optional): Name of column,
	Caption (string, optional): caption of column,
	Value (string, optional): value of column,
	Type (string, optional): data type of column.
}
```

Sample Code

```
// ********************************************************************************
// * Item - Get a single Item by its ID
// ********************************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 303 is the ID of the Box being retrieved
// objItem will contain the Item if the get is successful
public partial class IRItem
{
	public virtual List<IRColumn> columns { get; set; }
	public virtual String TabSingularName { get; set; }
	public virtual String EdocFile { get; set; }
	public virtual String ContentType { get; set; }
	public virtual Int32 Id { get; set; }
	public virtual String QuickDescription { get; set; }
```

```
        public virtual String Barcode { get; set; }
        public virtual Int32 TabId { get; set; }
}
public partial class IRColumn
{
        public virtual String ColumnName { get; set; }
        public virtual String Caption { get; set; }
        public virtual String Value { get; set; }
        public virtual String Type { get; set; }
}
public IRItem GetItemByID()
{
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Item/303");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRItem objItem =
        JsonConvert.DeserializeObject<IRItem>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objItem;
}
```

Perform an Advanced Search for an Item

Request Type

Post

URL

http://[Servername]:[Port ID]/api/ItemSearch

Implementation Notes

Uses an IRItemSearch object to do a more advanced search for an Infolinx Item.
TabID is a required field.
The search types are limited; you cannot use this to search using a Range Field.
Special Search allowed values:

- Not set or "*none*" – default value; Special Search not set
- "*missing*" – return Items of the given TabID that have been marked as missing
- "*topshelfuser*" – return Items of the given TabID that are top level items, that can't be contained in any other item
- "*keyword*" – returns all Items of the given TabID that contain the supplied keyword in any of their database columns
- "*mycontents*" – not currently documented
- "*crudxml*" – not currently documented

Accepts Inputs

An IRItemSearch with an array of KeyValuePairs embedded in it. See Input Parameter Format below

Input Parameter Type

Application/json body

Input Parameter Format

```
IRItemSearch
{
        SpecialSearch (string, optional): special searches,
```

```
        LastSyncDate (string, optional): used for topshelfuser search to only get
        items created after this date,
        TabId (integer, required): tab id,
        Barcode (string, optional): item barcode,
        SearchTerms (Dictionary<String, object>, optional): if nothing else is
        provided, Infolinx will use this for an advanced search,
        IncludeEdocFile (boolean, optional): Whether the actual edoc file should be
        returned with the metadata. not used right now, file is always included.,
        ViewDisplay (boolean, optional): Just return the fields that should be shown
        on the view page,
        GetForeignKeyDesc (boolean, optional): For foreign keys, get the text or quick
        description instead of the id,
        CrudXML (string, optional)
}
```

Returns

- • An IEnumerable<IRItem> object containing a list of IRColumn fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRItem {
        columns (array[IRColumn], optional): columns, corresponds to view,
        TabSingularName (string, optional): singular name for item's tab,
        EdocFile (string, optional): Base64 encoded string of the edoc file. If
        creating, must provide a column value for I_E_USER_FILE_NAME,
        ContentType (string, optional): content type for uploading edoc file,
        Id (integer, optional): item id of item,
        QuickDescription (string, optional): Quick Description,
        Barcode (string, optional): Barcode,
        TabId (integer, optional): Tab Id
}
IRColumn {
        ColumnName (string, optional): Name of column,
        Caption (string, optional): caption of column,
        Value (string, optional): value of column,
        Type (string, optional): data type of column.
}
```

Sample Code

```
// ********************************************************************************
// * Item - Advance Search for Items
// ********************************************************************************
public partial class IRItem
{
        public virtual List<IRColumn> columns { get; set; }
        public virtual String TabSingularName { get; set; }
        public virtual String EdocFile { get; set; }
        public virtual String ContentType { get; set; }
        public virtual Int32 Id { get; set; }
        public virtual String QuickDescription { get; set; }
        public virtual String Barcode { get; set; }
        public virtual Int32 TabId { get; set; }
}
```

```
public partial class IRColumn
{
	public virtual String ColumnName { get; set; }
	public virtual String Caption { get; set; }
	public virtual String Value { get; set; }
	public virtual String Type { get; set; }
}
public partial class IRItemSearch
{
	public virtual String SpecialSearch { get; set; }
	public virtual String LastSyncDate { get; set; }
	public virtual int TabId { get; set; }
	public virtual String Barcode { get; set; }
	public virtual Dictionary<String, object> SearchTerms { get; set; }
	public virtual bool IncludeEdocFile { get; set; }
	public virtual bool ViewDisplay { get; set; }
	public virtual bool GetForeignKeyDesc { get; set; }
	public virtual String CrudXML { get; set; }
}
public IEnumerable<IRItem> ItemSearch()
{
	// w is a string containing the http://[Servername]:[Port ID] part of the URL
	IRItemSearch IRIS = new IRItemSearch();
	IRIS.TabId = 202; // Box Item Type
	IRIS.SearchTerms = new Dictionary<String, object>();
	// Search the "Box_Description" database column for rows with value "Little
	Box":
	IRIS.SearchTerms.Add("BOX_DESCRIPTION", "Little Box");
	// Search for Missing Boxes
	//IRIS.SpecialSearch = "missing";
	string body = JsonConvert.SerializeObject(IRIS);
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/ItemSearch");
	SetHeaders(webRequest, "POST", body);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IEnumerable<IRItem> objFoundItems =
	JsonConvert.DeserializeObject<IEnumerable<IRItem>>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	webResponse = null;
	return objFoundItems;
}
```

Get the view URL for the given Barcode of an Item

Request Type

Get

URL

http://[Servername]:[Port ID]/api/itemurlforbarcode?barcode=[Item Barcode]

Implementation Notes

Get the URL to view an Item by its Item Barcode.

Accepts Inputs

- String Item Barcode

Input Parameter Type

URL Query String

Returns

- A string containing the URL to the Item represented by the given Barcode

Response Content Type

Application/json

Response Class

string

Sample Code

```
// ********************************************************************************
// * Item - Get the URL for a single Item by its Barcode
// ********************************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 0000000202 is the Barcode of the Box being retrieved
public String GetURLByBarcode()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w +
	"/api/itemurlforbarcode?barcode=0000000202");
	SetHeaders(webRequest, "GET", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	String strURL =
	JsonConvert.DeserializeObject<String>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	return strURL;
}
```

Get the view URL for the given Item ID of an Item

Request Type

Get

URL

http://[Servername]:[Port ID]/api/itemurlforid?id=[Item ID]

Implementation Notes

Get the URL to view an Item by its Item ID.

Accepts Inputs

- Integer Item ID

Input Parameter Type

URL Query String

Returns

- A string containing the URL to the Item represented by the given Item ID

Response Content Type

Application/json

Response Class

string

Sample Code

```
// ********************************************************************************
// * Item - Get the URL for a single Item by its ID
// ********************************************************************************
```

```
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 303 is the ID of the Box whose URL we want
public String GetURLByID()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w +
	"/api/itemurlforid?id=303");
	SetHeaders(webRequest, "GET", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	String strURL =
	JsonConvert.DeserializeObject<String>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	return strURL;
}
```

Get the view URL for the given Item ID and Tab ID of an Item

Request Type

Get

URL

http://[Servername]:[Port ID]/api/itemurlforidanditid?id=[Item ID]&itemtypeid=[Item Type ID]

Implementation Notes

Get the URL to view an Item by its Item ID and Item Type ID.

Accepts Inputs

- Integer Item ID
- Integer Item Type ID

Input Parameter Type

URL Query String

Returns

- A string containing the URL to the Item represented by the given Item ID and Item Type ID

Response Content Type

Application/json

Response Class

string

Sample Code

```
// ********************************************************************************
// * Item - Get the URL for a single Item by its ID and Item Type ID
// ********************************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 303 is the ID of the Box whose URL we want
// 202 is the Item Type ID of the Box whose URL we want
public String GetURLByIDAndItemTypeID()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w +
	"/api/itemurlforidanditid?id=303&itemtypeid=202");
	SetHeaders(webRequest, "GET", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	String strURL =
	JsonConvert.DeserializeObject<String>(sr.ReadToEnd());
```

```
        sr.Close();
        webResponse.Close();
        return strURL;
}
```

Label

Queue a barcode label to be printed for an Item

Request Type

Post

URL

http://[Servername]:[Port ID]/api/Label

Implementation Notes

Create a new label for the given Item in the given Queue.

Accepts Inputs

- ID for the Infolinx label queue to be used
- ID of the Infolinx Item on which the barcode label will be affixed
- Barcode of the Infolix item
- Reason for the label

Input Parameter Type

Application/json body

Input Parameter Class

```
IRLabel
{
       QueueId (integer, required): Id for the Label Queue in infolinx,
       ItemId (integer, required): Id for Infolinx item. Must provide this or
       barcode.,
       ItemBarcode (string, required): Barcode for Infolinx item. Must provide this
       or Id.,
       Reason (string, optional): Reason for label
}
```

Returns Responses

An HttpWebResponse object

Sample Code

```
// ***********************************************************************
// * Label - Queue a barcode label to be printed for an Item
// ***********************************************************************
public partial class IRLabel
{
       public virtual int QueueId { get; set; }
       public virtual int ItemId { get; set; }
       public virtual String ItemBarcode { get; set; }
       public virtual String Reason { get; set; }
}
public HttpStatusCode QueueLabel()
{
       // w is a string containing the http://[Servername]:[Port ID] part of the URL
       IRLabel iLabel = new IRLabel();
       iLabel.QueueId = 1002; // Label Queue Profile ID from Label_Queue_List table in db
       iLabel.ItemId = 303; // Item ID
       string body = JsonConvert.SerializeObject(iLabel);
       var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Label");
```

```
        SetHeaders(webRequest, "POST", body);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        return webResponse.StatusCode;
}
```

Get Label Queues for a Tab

Request Type

Get

URL

http://[Servername]:[Port ID]/api/labelprofiles/[Tab ID]

Implementation Notes

Get label queues that are available for the given Tab ID

Accepts Inputs

- Integer Tab ID.

Input Parameter Type

URL

Input Parameter Class

None.

Returns

- An IEnumerable<IRLabelProfile> object with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRLabelProfile
{
        Name (string, optional): Profile Name,
        Id (integer, optional): Profile Id,
        VendorId (integer, optional): Vendor Id
}
```

Sample Code

```
// ********************************************************************************
// * Label - Get Label Queues for a given Tab (Item Type)
// ********************************************************************************
public partial class IRLabelProfile
{
        public virtual String Name { get; set; }
        public virtual int Id { get; set; }
        public virtual int VendorId { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 202 is the Item Type ID
public IEnumerable<IRLabelProfile> GetLabelQueues()
{
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/labelprofiles/202");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IEnumerable<IRLabelProfile> objQueues =
        JsonConvert.DeserializeObject<IEnumerable<IRLabelProfile>>(sr.ReadToEnd());
        sr.Close();
```

```
        webResponse.Close();
        return objQueues;
}
```

Picklist

To get the Picklist for a Tab

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Picklist/[Tab ID]

Implementation Notes

Gets Picklist rows for a given TabID

Accepts Inputs

- Integer TabID

Input Parameter Type

URL

Returns

- An IRPicklistItem object with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRPicklistItem
{
        Columns (array[KeyValuePair[String,Object]], optional): Columns on the
        picklist report.,
        QuickDescription (string, optional),
        Id (integer, optional),
        TabId (integer, optional)
        } KeyValuePair[String,Object] {
        Key (string, optional),
        Value (string, optional)
}
```

Sample Code

```
// ********************************************************************************
// * Picklist - Gets picklist rows for a given Tab (Item Type)
// ********************************************************************************
public partial class IRPicklistItem
{
        public virtual Dictionary<String, object> Columns { get; set; }
        public virtual String QuickDescription { get; set; }
        public virtual int Id { get; set; }
        public virtual int TabId { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 202 is the integer TabID (Item Type ID)
public IEnumerable<IRPicklistItem> GetPicklistByTab()
{
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/picklist/202");
        SetHeaders(webRequest, "GET", null);
```

```
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IEnumerable<IRPicklistItem> objItems =
        JsonConvert.DeserializeObject<IEnumerable<IRPicklistItem>>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objItems;
}
```

Request

Get all Item checkout Requests

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Request

Implementation Notes

Gets data describing all Infolinx Item checkout Requests

Accepts Inputs

None

Returns

- An IRRequest array with fields described in Response Class below. If there are no requests, the response body will have 0 length

Response Content Type

Application/json

Response Class

```
IRRequest
{
        Id (integer, optional): Auto-provided for new requests,
        ItemId (integer, optional): Required for new requests,
        ItemDescription (string, optional): Quick Description for item, only provided
        if GetDescription = true,
        DestinationItemId (integer, optional): The item id for the destination.,
        DestinationDescription (string, optional): QD for destination, only provided
        if GetDescription = true,
        Waitlist (boolean, optional): Auto-provided for new requests,
        CreateDate (string, optional): The create date.,
        RequestorName (string, optional): Auto-provided for new requests,
        RequestorDesc (string, optional): Auto-provided for new requests,
        Status (integer, optional): Status id,
        StatusDescription (string, optional): Text value for status,
        BatchNumber (integer, optional): Useful for tracking a batch of requests in
        Infolinx,
        FulfillmentMethod (integer, optional): Used if fulfillment method other than
        transferring is desired,
        RequestType (integer, optional): Required and only applicable for new
        requests,
        Comment (string, optional): Comment
}
```

Sample Code

```
// ************************************************************************
```

```
// * Request - Gets all checkout Requests in Infolinx
// ***********************************************************************
public partial class IRRequest
{
	public virtual int Id { get; set; }
	public virtual int ItemId { get; set; }
	public virtual String ItemDescription { get; set; }
	public virtual int DestinationItemId { get; set; }
	public virtual String DestinationDescription { get; set; }
	public virtual bool Waitlist { get; set; }
	public virtual String CreateDate { get; set; }
	public virtual String RequestorName { get; set; }
	public virtual String RequestorDesc { get; set; }
	public virtual int Status { get; set; }
	public virtual String StatusDescription { get; set; }
	public virtual int BatchNumber { get; set; }
	public virtual int FulfillmentMethod { get; set; }
	public virtual int RequestType { get; set; }
	public virtual String Comment { get; set; }
 }
// w is a string containing the http://[Servername]:[Port ID] part of the URL
public IEnumerable<IRRequest> GetAllRequests()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Request");
	SetHeaders(webRequest, "GET", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IEnumerable<IRRequest> objRequests =
	JsonConvert.DeserializeObject<IEnumerable<IRRequest>>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	return objRequests;
}
```

Create a new Item checkout Request

Request Type

Post

URL

http://[Servername]:[Port ID]/api/Request

Implementation Notes

Generates a new Item checkout Request. If destination Item is not provided, then a container is auto selected. Request Type can be 1 (Delivery), 2 (Pickup), or 4 (Renew).

Accepts Inputs

See Input Parameter Format below

Input Parameter Type

Application/json body

Input Parameter Format

```
IRRequest
{
	Id (integer, optional): Auto-provided for new requests,
	ItemId (integer): Required for new requests,
```

```
        ItemDescription (string, optional): Quick Description for item, only provided
        if GetDescription = true,
        DestinationItemId (integer, optional): The item id for the destination.,
        DestinationDescription (string, optional): QD for destination, only provided
        if GetDescription = true,
        Waitlist (boolean, optional): Auto-provided for new requests,
        CreateDate (string, optional): The create date.,
        RequestorName (string, optional): Auto-provided for new requests,
        RequestorDesc (string, optional): Auto-provided for new requests,
        Status (integer, optional): Status id,
        StatusDescription (string, optional): Text value for status,
        BatchNumber (integer, optional): Useful for tracking a batch of requests in
        Gimmal Physical,
        FulfillmentMethod (integer, optional): Used if fulfillment method other than
        transferring is desired,
        RequestType (integer, optional): Required and only applicable for new
        requests,
        Comment (string, optional): Comment
}
```

Returns

- An IRRequestResult object containing an IRRequest object, ItemBarcode, ItemDescription, Message, and Result fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRRequestResult
{
        Request (IRRequest, optional): The resulting request. Not included if you
        deleted the request.,
        ItemBarcode (string, optional): The barcode of the item that was requested.,
        ItemDescription (string, optional): The description of the item that was
        requested.,
        Message (string, optional): If failed, the reason why,
        Result (boolean, optional): success/fail
        } IRRequest {
        Id (integer, optional): Auto-provided for new requests,
        ItemId (integer, optional): Required for new requests,
        ItemDescription (string, optional): Quick Description for item, only provided
        if GetDescription = true,
        DestinationItemId (integer, optional): The item id for the destination.,
        DestinationDescription (string, optional): QD for destination, only provided
        if GetDescription = true,
        Waitlist (boolean, optional): Auto-provided for new requests,
        CreateDate (string, optional): The create date.,
        RequestorName (string, optional): Auto-provided for new requests,
        RequestorDesc (string, optional): Auto-provided for new requests,
        Status (integer, optional): Status id,
        StatusDescription (string, optional): Text value for status,
        BatchNumber (integer, optional): Useful for tracking a batch of requests in
        Infolinx,
        FulfillmentMethod (integer, optional): Used if fulfillment method other than
        transferring is desired,
```

```
        RequestType (integer, optional): Required and only applicable for new
        requests,
        Comment (string, optional): Comment
}
```

Sample Code

```
// ********************************************************************************
// * Request - Create a new Item checkout Request
// ********************************************************************************
public partial class IRRequest
{
        public virtual int Id { get; set; }
        public virtual int ItemId { get; set; }
        public virtual String ItemDescription { get; set; }
        public virtual int DestinationItemId { get; set; }
        public virtual String DestinationDescription { get; set; }
        public virtual bool Waitlist { get; set; }
        public virtual String CreateDate { get; set; }
        public virtual String RequestorName { get; set; }
        public virtual String RequestorDesc { get; set; }
        public virtual int Status { get; set; }
        public virtual String StatusDescription { get; set; }
        public virtual int BatchNumber { get; set; }
        public virtual int FulfillmentMethod { get; set; }
        public virtual int RequestType { get; set; }
        public virtual String Comment { get; set; }
}
public partial class IRRequestResult
{
        public virtual IRRequest Request { get; set; }
        public virtual String ItemBarcode { get; set; }
        public virtual String ItemDescription { get; set; }
        public virtual String Message { get; set; }
        public virtual bool Result { get; set; }
}
public IRRequestResult CreateRequest()
{
        // w is a string containing the http://[Servername]:[Port ID] part of the URL
        IRRequest Request = new IRRequest();
        Request.ItemId = 40000; // Required; ID of the Item being checked out
        Request.RequestType = 1; // Required; 1=Delivery; 2=Pickup; 4=Renew
        Request.DestinationItemId = 201; // ID of the user to check the Item out to
        string body = JsonConvert.SerializeObject(Request);
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Request");
        SetHeaders(webRequest, "POST", body);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRRequestResult objResult =
        JsonConvert.DeserializeObject<IRRequestResult>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        webResponse = null;
        return objResult;
}
```

Delete an Item checkout Request by Request ID

Request Type

Delete

URL

http://[Servername]:[Port ID]/api/Request/[ID]

Implementation Notes

Deletes the Item checkout Request that corresponds to the supplied Request ID. If successful, webResponse.StatusCode will be "OK."

Accepts Inputs

- Integer Request ID

Input Parameter Type

URL

Returns

- An HttpStatusCode object

Response Content Type

Application/json

Response Class

HttpStatusCode

Sample Code

```
// ********************************************************************
// * Request - Delete an Item checkout Request
// ********************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 217 is the ID of the Request to be deleted
// If successful, webResponse.StatusCode will be "OK"
public HttpStatusCode DeleteRequest()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Request/217");
	SetHeaders(webRequest, "DELETE", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	return webResponse.StatusCode;
}
```

Get an Item checkout Request by Request ID

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Request/[ID]

Implementation Notes

Gets the Item checkout Request that corresponds to the supplied Request ID.

Accepts Inputs

- Integer Request ID

Input Parameter Type

URL

Returns

- An IRRequest object with fields described in Response Class below.

Response Content Type
Application/json
Response Class
```
IRRequest
{
	Id (integer, optional): Auto-provided for new requests,
	ItemId (integer, optional): Required for new requests,
	ItemDescription (string, optional): Quick Description for item, only provided
	if GetDescription = true,
	DestinationItemId (integer, optional): The item id for the destination.,
	DestinationDescription (string, optional): QD for destination, only provided
	if GetDescription = true,
	Waitlist (boolean, optional): Auto-provided for new requests,
	CreateDate (string, optional): The create date.,
	RequestorName (string, optional): Auto-provided for new requests,
	RequestorDesc (string, optional): Auto-provided for new requests,
	Status (integer, optional): Status id,
	StatusDescription (string, optional): Text value for status,
	BatchNumber (integer, optional): Useful for tracking a batch of requests in
	Infolinx,
	FulfillmentMethod (integer, optional): Used if fulfillment method other than
	transferring is desired,
	RequestType (integer, optional): Required and only applicable for new
	requests,
	Comment (string, optional): Comment
}
```
Sample Code
```
// ********************************************************************************
// * Request - Gets checkout Request by Request ID
// ********************************************************************************
public partial class IRRequest
{
	public virtual int Id { get; set; }
	public virtual int ItemId { get; set; }
	public virtual String ItemDescription { get; set; }
	public virtual int DestinationItemId { get; set; }
	public virtual String DestinationDescription { get; set; }
	public virtual bool Waitlist { get; set; }
	public virtual String CreateDate { get; set; }
	public virtual String RequestorName { get; set; }
	public virtual String RequestorDesc { get; set; }
	public virtual int Status { get; set; }
	public virtual String StatusDescription { get; set; }
	public virtual int BatchNumber { get; set; }
	public virtual int FulfillmentMethod { get; set; }
	public virtual int RequestType { get; set; }
	public virtual String Comment { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 215 is the ID of the checkout Request to get
public IRRequest GetRequestByID()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Request/215");
	SetHeaders(webRequest, "GET", null);
```

```
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRRequest objRequest = JsonConvert.DeserializeObject<IRRequest>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objRequest;
}
```

Update an Item's existing checkout Request by Request ID

Request Type

Put

URL

http://[Servername]:[Port ID]/api/Request/[ID]

Implementation Notes

Update the existing Item checkout Request that corresponds to the given ID.

Accepts Inputs

- Request ID (required)
- Status (required)

- Possible Status Values:
  - 1 = Approved
  - 2 = Fulfilled
  - 3 = Rejected
  - 5 = Deleted

See Input Parameter Format below for additional fields.

Input Parameter Type

Application/json body

Input Parameter Format

```
IRRequest
{
        Id (integer, optional): Auto-provided for new requests,
        ItemId (integer, optional): Required for new requests,
        ItemDescription (string, optional): Quick Description for item, only provided
        if GetDescription = true,
        DestinationItemId (integer, optional): The item id for the destination.,
        DestinationDescription (string, optional): QD for destination, only provided
        if GetDescription = true,
        Waitlist (boolean, optional): Auto-provided for new requests,
        CreateDate (string, optional): The create date.,
        RequestorName (string, optional): Auto-provided for new requests,
        RequestorDesc (string, optional): Auto-provided for new requests,
        Status (integer, optional): Status id,
        StatusDescription (string, optional): Text value for status,
        BatchNumber (integer, optional): Useful for tracking a batch of requests in
        Infolinx,
        FulfillmentMethod (integer, optional): Used if fulfillment method other than
        transferring is desired,
        RequestType (integer, optional): Required and only applicable for new
        requests,
        Comment (string, optional): Comment
```

```
}
```

Returns

• An IRRequestResult object with an IRRequest object, ItemBarcode, ItemDescription, Message, and Result fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRRequestResult
{
	Request (IRRequest, optional): The resulting request. Not included if you
	deleted the request.,
	ItemBarcode (string, optional): The barcode of the item that was requested.,
	ItemDescription (string, optional): The description of the item that was
	requested.,
	Message (string, optional): If failed, the reason why,
	Result (boolean, optional): success/fail
}
IRRequest
{
	Id (integer, optional): Auto-provided for new requests,
	ItemId (integer, optional): Required for new requests,
	ItemDescription (string, optional): Quick Description for item, only provided
	if GetDescription = true,
	DestinationItemId (integer, optional): The item id for the destination.,
	DestinationDescription (string, optional): QD for destination, only provided
	if GetDescription = true,
	Waitlist (boolean, optional): Auto-provided for new requests,
	CreateDate (string, optional): The create date.,
	RequestorName (string, optional): Auto-provided for new requests,
	RequestorDesc (string, optional): Auto-provided for new requests,
	Status (integer, optional): Status id,
	StatusDescription (string, optional): Text value for status,
	BatchNumber (integer, optional): Useful for tracking a batch of requests in
	Gimmal Physical,
	FulfillmentMethod (integer, optional): Used if fulfillment method other than
	transferring is desired,
	RequestType (integer, optional): Required and only applicable for new
	requests,
	Comment (string, optional): Comment
}
```

Sample Code

```
// ***********************************************************************
// * Request - Update an existing checkout Request
// ***********************************************************************
public partial class IRRequest
{
	public virtual int Id { get; set; }
	public virtual int ItemId { get; set; }
	public virtual String ItemDescription { get; set; }
	public virtual int DestinationItemId { get; set; }
	public virtual String DestinationDescription { get; set; }
	public virtual bool Waitlist { get; set; }
	public virtual String CreateDate { get; set; }
```

```
        public virtual String RequestorName { get; set; }
        public virtual String RequestorDesc { get; set; }
        public virtual int Status { get; set; }
        public virtual String StatusDescription { get; set; }
        public virtual int BatchNumber { get; set; }
        public virtual int FulfillmentMethod { get; set; }
        public virtual int RequestType { get; set; }
        public virtual String Comment { get; set; }
}
public partial class IRRequestResult
{
        public virtual IRRequest Request { get; set; }
        public virtual String ItemBarcode { get; set; }
        public virtual String ItemDescription { get; set; }
        public virtual String Message { get; set; }
        public virtual bool Result { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 215 is the ID of the Request you're updating
public IRRequestResult UpdateRequest()
{
        IRRequest Request = new IRRequest();
        Request.Status = 2; //Possible Status Values: 1 = Approved; 2 = Fulfilled; 3 =
        Rejected; 5 = Deleted
        Request.DestinationItemId = 201; // ID of the user to check the Item out to
        string body = JsonConvert.SerializeObject(Request);
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Request/215");
        SetHeaders(webRequest, "PUT", body);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRRequestResult objResult =
        JsonConvert.DeserializeObject<IRRequestResult>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        webResponse = null;
        return objResult;
}
```

Perform an advanced search for Item checkout Requests

Request Type

Post

URL

http://[Servername]:[Port ID]/api/requestsearch

Implementation Notes

Perform an advanced search for Item checkout Requests.

Potential Report Values: 1 (Approved); 2 (Fulfilled).

Accepts Inputs

See Input Parameter Format below

Input Parameter Type

Application/json body

Input Parameter Format

```
IRRequestSearch
{
        OnlyMine (boolean, optional): only get my requests.,
```

```
        Report (integer, optional): 1: only requests on the picklist, 2: only requests
        on the pickup report. TabId is required.,
        TabId (integer, optional): only get requests for a certain tab,
        GetDescriptions (boolean, optional): Whether to retrieve item descriptions or
        not
}
```

Returns

- An IRRequest object with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRRequest
{
        Id (integer, optional): Auto-provided for new requests,
        ItemId (integer, optional): Required for new requests,
        ItemDescription (string, optional): Quick description for item, only provided
        if GetDescription = true,
        DestinationItemId (integer, optional): The item id for the destination.,
        DestinationDescription (string, optional): QD for destination, only provided
        if GetDescription = true,
        Waitlist (boolean, optional): Auto-provided for new requests,
        CreateDate (string, optional): The create date.,
        RequestorName (string, optional): Auto-provided for new requests,
        RequestorDesc (string, optional): Auto-provided for new requests,
        Status (integer, optional): Status id,
        StatusDescription (string, optional): Text value for status,
        BatchNumber (integer, optional): Useful for tracking a batch of requests in
        Infolinx,
        FulfillmentMethod (integer, optional): Used if fulfillment method other than
        transferring is desired,
        RequestType (integer, optional): Required and only applicable for new
        requests,
        Comment (string, optional): Comment
}
```

Sample Code

```
// ********************************************************************************
// * Request - Find Item checkout Requests
// ********************************************************************************
public partial class IRRequestSearch
{
        public virtual bool OnlyMine { get; set; }
        public virtual int Report { get; set; }
        public virtual int TabId { get; set; }
        public virtual bool GetDescriptions { get; set; }
}
public partial class IRRequest
{
        public virtual int Id { get; set; }
        public virtual int ItemId { get; set; }
        public virtual String ItemDescription { get; set; }
        public virtual int DestinationItemId { get; set; }
        public virtual String DestinationDescription { get; set; }
        public virtual bool Waitlist { get; set; }
        public virtual String CreateDate { get; set; }
```

```
        public virtual String RequestorName { get; set; }
        public virtual String RequestorDesc { get; set; }
        public virtual int Status { get; set; }
        public virtual String StatusDescription { get; set; }
        public virtual int BatchNumber { get; set; }
        public virtual int FulfillmentMethod { get; set; }
        public virtual int RequestType { get; set; }
        public virtual String Comment { get; set; }
}
public IEnumerable<IRRequest> FindRequest()
{
        // w is a string containing the http://[Servername]:[Port ID] part of the URL
        IRRequestSearch RequestSearch = new IRRequestSearch();
        string body = JsonConvert.SerializeObject(RequestSearch);
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/requestsearch");
        SetHeaders(webRequest, "POST", body);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IEnumerable<IRRequest> objResult =
        JsonConvert.DeserializeObject<IEnumerable<IRRequest>>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        webResponse = null;
        return objResult;
}
```

Server Time

To get the UTC time from the Infolinx Web Server

Request Type

Get

URL

http://[Servername]:[Port ID]/api/ServerTime

Implementation Notes

Gets the UTC time from the Infolinx Web Server

Accepts Inputs

- None

Input Parameter Type

- None

Returns

- • A string containing the Infolinx Web Server's UTC time

Response Content Type

Application/json

Response Class

string

Sample Code

```
// ********************************************************************************
// * ServerTime - Get the UTC time from the Infolinx Web Server
// ********************************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
public DateTime GetDateTime()
{
```

```
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/servertime");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        // Converting the string response to a DateTime object
        DateTime objItems = JsonConvert.DeserializeObject<DateTime>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objItems;
}
```

Tab

Get all Tabs

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Tab

Implementation Notes

Gets data describing all Infolinx Tabs

Accepts Inputs

None

Input Parameter Type

None

Returns

- An IRTab array with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRTab
{
        Id (integer, optional): Tab id,
        SingularName (string, optional): singular name for tab,
        PluralName (string, optional): plural name for tab,
        IsMoveable (boolean, optional): tab is moveable,
        IsRequestable (boolean, optional): tab can be requested,
        AutoGenerateBarcode (boolean, optional): tab has barcode auto-generated,
        IsBarcodeRequired (boolean, optional): whether barcode is required. if not,
        probably logical tab,
        BarcodePrefix (string, optional): prefix for barcode,
        BarcodeLength (integer, optional): length for barcode,
        IsEdoc (boolean, optional): tab is edoc,
        DisplayOrder (integer, optional): display order,
        IsRetentionEnabled (boolean, optional): tab uses retention,
        SpecialType (integer, optional): special type
}
```

Sample Code

```
// ********************************************************************************
// * Tab - Get all Tabs
// ********************************************************************************
public partial class IRTab
```

```
{
	public virtual Int32 Id { get; set; }
	public virtual String SingularName { get; set; }
	public virtual String PluralName { get; set; }
	public virtual Boolean IsMoveable { get; set; }
	public virtual Boolean IsRequestable { get; set; }
	public virtual Boolean AutoGenerateBarcode { get; set; }
	public virtual Boolean IsBarcodeRequired { get; set; }
	public virtual String BarcodePrefix { get; set; }
	public virtual Int32 BarcodeLength { get; set; }
	public virtual Boolean IsEdoc { get; set; }
	public virtual Int32 DisplayOrder { get; set; }
	public virtual Boolean IsRetentionEnabled { get; set; }
	public virtual Int32 SpecialType { get; set; }
}
public IEnumerable<IRTab> GetAllTabs()
{
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Tab");
	SetHeaders(webRequest, "GET", null);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IEnumerable<IRTab> objItems =
	JsonConvert.DeserializeObject<IEnumerable<IRTab>>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	return objItems;
}
```

Get a Subset of Tabs

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Tab?itemtypefilter=[picklist|topshelfuser]

Implementation Notes

Gets data describing specified Infolinx Tabs

Input Parameter Type

- URL Query String

Input Parameter Format

- The itemtypefilter parameter will be either “picklist” or “topshelfuser.” Picklist: return tabs that are requestable and may be contained by another tab topshelfuser: return tabs that are top level, user, or a shelf.

Returns

- An IRTab array with fields described in Response Class below

Response Content Type

Application/json

Response Class

```
IRTab
{
	Id (integer, optional): Tab id,
	SingularName (string, optional): singular name for tab,
	PluralName (string, optional): plural name for tab,
```

```
        IsMoveable (boolean, optional): tab is moveable,
        IsRequestable (boolean, optional): tab can be requested,
        AutoGenerateBarcode (boolean, optional): tab has barcode auto-generated,
        IsBarcodeRequired (boolean, optional): whether barcode is required. if not,
        probably logical tab,
        BarcodePrefix (string, optional): prefix for barcode,
        BarcodeLength (integer, optional): length for barcode,
        IsEdoc (boolean, optional): tab is edoc,
        DisplayOrder (integer, optional): display order,
        IsRetentionEnabled (boolean, optional): tab uses retention,
        SpecialType (integer, optional): special type
}
```

Sample Code

```
// ********************************************************************************
// * Tab - Get filtered Tabs
// ********************************************************************************
public partial class IRTab
{
        public virtual Int32 Id { get; set; }
        public virtual String SingularName { get; set; }
        public virtual String PluralName { get; set; }
        public virtual Boolean IsMoveable { get; set; }
        public virtual Boolean IsRequestable { get; set; }
        public virtual Boolean AutoGenerateBarcode { get; set; }
        public virtual Boolean IsBarcodeRequired { get; set; }
        public virtual String BarcodePrefix { get; set; }
        public virtual Int32 BarcodeLength { get; set; }
        public virtual Boolean IsEdoc { get; set; }
        public virtual Int32 DisplayOrder { get; set; }
        public virtual Boolean IsRetentionEnabled { get; set; }
        public virtual Int32 SpecialType { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
public IEnumerable<IRTab> GetFilteredTabs()
{
        var webRequest = (HttpWebRequest)WebRequest.Create(w +
        "/api/tab?itemtypefilter=topshelfuser");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IEnumerable<IRTab> objTabs =
        JsonConvert.DeserializeObject<IEnumerable<IRTab>>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objTabs;
}
```

Get a single Tab by Tab ID

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Tab/[ID]

Implementation Notes

Gets data describing a single specified Infolinx Tab.

Accepts Inputs

- Integer Tab ID.

Input Parameter Type

URL

Returns

- An IRTab object with fields described in Response Class below.

Response Content Type

Application/json

Response Class

```
IRTab
{
	Id (integer, optional): Tab id,
	SingularName (string, optional): singular name for tab,
	PluralName (string, optional): plural name for tab,
	IsMoveable (boolean, optional): tab is moveable,
	IsRequestable (boolean, optional): tab can be requested,
	AutoGenerateBarcode (boolean, optional): tab has barcode auto-generated,
	IsBarcodeRequired (boolean, optional): whether barcode is required. if not,
	probably logical tab,
	BarcodePrefix (string, optional): prefix for barcode,
	BarcodeLength (integer, optional): length for barcode,
	IsEdoc (boolean, optional): tab is edoc,
	DisplayOrder (integer, optional): display order,
	IsRetentionEnabled (boolean, optional): tab uses retention,
	SpecialType (integer, optional): special type
}
```

Sample Code

```
// ********************************************************************************
// * Tab - Get a single Tab by its ID
// ********************************************************************************
public partial class IRTab
{
	public virtual Int32 Id { get; set; }
	public virtual String SingularName { get; set; }
	public virtual String PluralName { get; set; }
	public virtual Boolean IsMoveable { get; set; }
	public virtual Boolean IsRequestable { get; set; }
	public virtual Boolean AutoGenerateBarcode { get; set; }
	public virtual Boolean IsBarcodeRequired { get; set; }
	public virtual String BarcodePrefix { get; set; }
	public virtual Int32 BarcodeLength { get; set; }
	public virtual Boolean IsEdoc { get; set; }
	public virtual Int32 DisplayOrder { get; set; }
	public virtual Boolean IsRetentionEnabled { get; set; }
	public virtual Int32 SpecialType { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 201 is the ID for a particular Tab
public IRTab GetTab()
{
```

```
        var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/tab/201");
        SetHeaders(webRequest, "GET", null);
        var webResponse = (HttpWebResponse)webRequest.GetResponse();
        StreamReader sr = new StreamReader(webResponse.GetResponseStream());
        IRTab objTab = JsonConvert.DeserializeObject<IRTab>(sr.ReadToEnd());
        sr.Close();
        webResponse.Close();
        return objTab;
}
```

Test

To test connection with Infolinx Web Services

Request Type

Get

URL

http://[Servername]:[Port ID]/api/Test

Implementation Notes

Tests connection with Gimmal Web Services. Does not require authentication.

Accepts Inputs

Nothing **Input Parameter Type**

None

Returns

- A successful test returns a string containing “Successfully connected to InfolinxRest”

Response Content Type

Application/json

Response Class

String

Sample Code

```
// ************************************************************************
// * Test - Test connection with Infolinx Web Services. Authentication not needed.
// ************************************************************************
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// A successful test returns string "Successfully connected to InfolinxRest"
public string Test()
{
var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/Test");
SetHeaders(webRequest, "GET", null);
var webResponse = (HttpWebResponse)webRequest.GetResponse();
StreamReader sr = new StreamReader(webResponse.GetResponseStream());
string testResponse = JsonConvert.DeserializeObject<string>(sr.ReadToEnd());
sr.Close();
webResponse.Close();
return testResponse;
}
```

Transfer

To Transfer a group of Items to a new containing Item (location)

Request Type

Post

URL

http://[Servername]:[Port ID]/api/Transfer

Implementation Notes

Transfers specified Infolinx items to the given location

Accepts Inputs

- String containing the Barcode of the Location to which Items should be transferred.
- String array containing the Barcodes of the Items to be transferred.

Input Parameter Type

Application/json body

Input Parameter Class

```
IRTransfer
{
	Location (string): barcode of location to transfer to,
	Items (array[string]): list of item barcodes to transfer
}
```

Returns

- An IRItemResultList with a Message, Result, and a repeating list of IRItemResults containing the actual success of each item.

Response Content Type

Application/json

Response Class

```
IRItemResultList
{
	Message (string, optional): overall result string for the list,
	Result (boolean, optional): true if all succeeded, false if any failed,
	ItemResults (array[IRItemResult], optional): the list of individual item
	results
}
IRItemResult
{
	LocationBarcode (string, optional): location barcode if applicable for the
	action,
	LocationDescription (string, optional): location Quick Description if
	applicable for the action,
	ItemId (integer, optional): Item Id,
	ItemBarcode (string, optional): item barcode,
	ItemDescription (string, optional): item Quick Description,
	RequestId (integer, optional): request id if this was a request,
	Result (boolean, optional): successful or not,
	Reason (integer, optional): corresponds to some enum for harcoded results,
	Message (string, optional): may contain some extra info about why this failed
}
```

Sample Code

```
// ********************************************************************************
// * Transfer - Transfer Items to a new containing Item (location)
// ********************************************************************************
```

```
public partial class IRTransfer
{
	public virtual String Location { get; set; }
	public virtual String[] Items { get; set; }
}
public partial class IRItemResultList
{
	public virtual String Message { get; set; }
	public virtual bool Result { get; set; }
	public virtual IRItemResult[] ItemResults { get; set; }
}
public partial class IRItemResult
{
	public virtual String LocationBarcode { get; set; }
	public virtual String LocationDescription { get; set; }
	public virtual Int32 ItemId { get; set; }
	public virtual String ItemBarcode { get; set; }
	public virtual String ItemDescription { get; set; }
	public virtual Int32 RequestId { get; set; }
	public virtual bool Result { get; set; }
	public virtual Int32 Reason { get; set; }
	public virtual String Message { get; set; }
}
// w is a string containing the http://[Servername]:[Port ID] part of the URL
// 0000010265 is the barcode of the user to whom the Items are being transferred
// 0000011667 and 0000017757 are the barcodes of the items being transferred
public IRItemResultList doTransfer()
{
	IRTransfer t = new IRTransfer();
	t.Location = "0000010265"; // Barcode of Location or User to which Items
	should be
	transfered
	t.Items = new string[] {"0000011667", "0000017757"}; // Array of Barcodes of
	Items to
	be transfered
	string body = JsonConvert.SerializeObject(t);
	var webRequest = (HttpWebRequest)WebRequest.Create(w + "/api/transfer");
	SetHeaders(webRequest, "POST", body);
	var webResponse = (HttpWebResponse)webRequest.GetResponse();
	StreamReader sr = new StreamReader(webResponse.GetResponseStream());
	IRItemResultList objItems =
	JsonConvert.DeserializeObject<IRItemResultList>(sr.ReadToEnd());
	sr.Close();
	webResponse.Close();
	webResponse = null;
	return objItems;
}
```

Get Error Log File

Get error log from import process

Request Type

Post

URL

http://[Servername]:[Port ID]/api/GetLogErrorFile

Implementation Notes

Method returns an IRIO item. Set the maximum file transfer size, file name and extension and count of bytes received before calling this method.

Response Content Type

Application/json

Response Class

```
IRIO
{
	Results (string, optional),
	FileName (string, optional),
	Extension (string, optional),
	CountOfBytesReceived (integer, optional),
	TotalFileLength (integer, optional),
	FileLocationType (string, optional) = ['Edoc' or 'Interchange' or 'Label' or
	'Report'],
	buffer (array[string], optional),
	MaxFileTransferSize (integer, optional),
	FilePath (string, optional),
	BarcodeString (string, optional)
}
```

Returns

- An IRIO object

Sample Code:

```
public class IRIO
{
	public string Results { get; set; }
	public string FileName { get; set; }
	public string Extension { get; set; }
	public int Position { get; set; }
	public int CountOfBytesReceived { get; set; }
	public long TotalFileLength { get; set; }
	public bool NewFile { get; set; }
	public FileLocation FileLocationType { get; set; }
	public enum FileLocation
	{
		Edoc = 1,
		Interchange = 2,
		Label = 3,
		Report = 4
	}
	public long MaxFileTransferSize { get; set; }
	public string FilePath { get; set; }
	public bool Complete { get; set; }
	public byte[] buffer { get; set; }
}
public static async Task<IRIO> GetFileInChunks(IRIO objIO)
{
	var handler = new HttpClientHandler
	{
		Proxy = WebRequest.GetSystemWebProxy(),
		UseProxy = true
	};
```

```
        using (var client = new IRIOClient(handler,InfolinxRestBaseAddress))
        {
                SetupHttpClient(client.HttpClient, UserName, Password);
                var response = await client.GetAsyncInChunks(objIO);
                response.EnsureSuccessStatusCode();
                IRIO results = await
                response.Content.ReadAsAsync<IRIO>().ConfigureAwait(true);
                return results;
        }
}
IO.MaxFileTransferSize = this._FileSizeToTransfer != 0 ?
Convert.ToInt64(this._FileSizeToTransfer) : 2097152; //default to 2MB if no size given
IO.FileName = strFileName;
IO.Extension = strExtension;
IO.CountOfBytesReceived = 0
IRIO Ret = await InfolinxRestProxy.GetFileInChunks(IO);
if (Ret.FileName.Length > 0)
{
        long lngReturnedLength = Convert.ToInt64(Ret.buffer.Length);
        IO.CountOfBytesReceived = Ret.CountOfBytesReceived;
        while (Ret.TotalFileLength >= lngReturnedLength)
{
Stream stream = new MemoryStream(Ret.buffer);
using (System.IO.FileStream output = new System.IO.FileStream(processedFilesDir +
Ret.FileName, FileMode.Append))
{
        stream.CopyTo(output);
        }
                if (Ret.TotalFileLength > lngReturnedLength)
                {
                        Ret = await InfolinxRestProxy.GetFileInChunks(IO);
                        IO.CountOfBytesReceived += Ret.CountOfBytesReceived;
                        lngReturnedLength += Convert.ToInt64(Ret.buffer.Length);
                }
                else
                {
                        this.LogAndErrorFileRetrieved = true;
                        break;
                }
        }
  }
```

Send Chunks

Request Type

Post

URL

http://[Servername]:[Port ID]/api/SendChunks

Implementation Notes

Method returns an IRIO item. Set the file name buffer size, file location, new file boolean and starting position before calling this method.

Response Content Type

Application/json

Response Class

```
IRIO
{
```

```
        Results (string, optional),
        FileName (string, optional),
        Extension (string, optional),
        CountOfBytesReceived (integer, optional),
        TotalFileLength (integer, optional),
        FileLocationType (string, optional) = ['Edoc' or 'Interchange' or 'Label' or
        'Report'],
        buffer (array[string], optional),
        MaxFileTransferSize (integer, optional),
        FilePath (string, optional),
        BarcodeString (string, optional)
}
```

Sample Code:

```
public async static Task<IRIO> PostAsyncInChuncks(IRIO objIO)
{
        var handler = new HttpClientHandler
        {
        Proxy = WebRequest.GetSystemWebProxy(),
        UseProxy = true
        };
        using (var client = new IRIOClient(handler,InfolinxRestBaseAddress))
        {
        SetupHttpClient(client.HttpClient,UserName,Password);
        var response = await client.PostAsyncInChuncks(objIO);
        response.EnsureSuccessStatusCode();
        return await response.Content.ReadAsAsync<IRIO>();
        }
}
public async Task<HttpResponseMessage> PostAsyncInChuncks(IRIO objIO)
{
        return await HttpClient.PostAsJsonAsync<IRIO>("api/sendChunks",
        objIO).ConfigureAwait(false);
}
public async Task<string> TextFileUpload()
{
        FileStream fs = null;
        string uploadedFileName = "";
        try
        {
        fs = File.OpenRead(this._fileNameAndPath);
        long lngTransferSize = fs.Length;
        fs.Close();
        fs.Dispose();
        long maxTransferSize = 2097152; //2 MB
        if (this._FileSizeToTransfer > 0)
        {
        maxTransferSize = this._FileSizeToTransfer;
        }
        if (maxTransferSize > lngTransferSize)
        {
        maxTransferSize = lngTransferSize;
        }
        byte[] buffer = new byte[maxTransferSize];
        int len;
        InfolinxRestProxy.InfolinxRestBaseAddress = this._InfolinxRestBaseAddress;
```

```
        InfolinxRestProxy.UserName = this._UserName;
        InfolinxRestProxy.Password = this._Password;
        IRIO IO = new IRIO();
        IO.FileName = this._fileName;
        IO.buffer = buffer;
        IO.FileLocationType = IRIO.FileLocation.Interchange;
        IO.Position = 0;
        IO.NewFile = true;
        int Offset = 0;
using (FileStream f = new FileStream(this._fileNameAndPath, FileMode.Open,
FileAccess.Read))
{
        try
        {
                f.Position = Offset;
                int BytesRead = 0;
                while (Offset != f.Length)
                {
                        BytesRead = f.Read(buffer, 0, buffer.Length);
                        if (BytesRead != buffer.Length)
                        {
                        maxTransferSize = BytesRead;
                        byte[] TrimmedBuffer = new byte[BytesRead];
                        Array.Copy(buffer, TrimmedBuffer, BytesRead);
                        buffer = TrimmedBuffer;
                        IO.buffer = buffer;
                }
                        IRIO Ret = await InfolinxRestProxy.PostFileInChuncks(IO);
                        IO.FileName = Ret.FileName;
                        IO.NewFile = false;
                        Offset += BytesRead;
                        }
                }
                catch (Exception ex)
                {
                        Notifications.NotificationService.WriteToEventLog("Error uploading
                        import file " + ex.ToString(), EventLogEntryType.Error,
                        this._session);
                }
        }
                uploadedFileName = IO.FileName;
        }
        catch (Exception EX)
        {
        this.OnExceptionOccurred(EX);
        return EX.ToString();
        }
        finally
        {
        if (fs != null)
        {
        fs.Dispose();
        }
        }
        return uploadedFileName;
  }
```