

MYOB Advanced

Contract-based REST API

Last Updated: 09 December 2016



Contents

Contract-based REST API	6
Representation of a Record in JSON Format.....	6
System Fields	6
General Fields.....	6
Linked Entities.....	6
Detail Entities.....	7
Custom Fields.....	7
Login to the Service.....	8
URL	8
Parameters.....	8
HTTP Method	9
Response	9
Example.....	9
Logout from the Service	10
URL	10
Parameters.....	10
HTTP Method	10
Response	10
Example.....	11
Creation of a Record	12
URL	12
Parameters.....	12
HTTP Method	12
Response	13
Example.....	13

Update of a Record	14
URL	14
Parameters.....	15
HTTP Method	15
Response	15
Example.....	16
Retrieval of a Record by Key Fields	17
URL	17
Parameters.....	18
HTTP Method	18
Response	18
Example.....	18
Retrieval of a Record by ID.....	20
URL	20
Parameters.....	20
HTTP Method	20
Response	20
Retrieval of Records by Conditions.....	21
URL	21
Parameters.....	21
HTTP Method	21
Response	21
Example.....	22
Retrieval of Data from an Inquiry Form.....	23
URL	23
Parameter.....	24
HTTP Method	24
Response	24
Example.....	24

Parameters for Retrieving Records	26
\$filter Parameter	26
\$top Parameter	27
\$skip Parameter	27
\$expand Parameter	27
\$custom Parameter	27
Removal of a Record.....	28
URL for Removing by Key Fields	28
URL for Removing by ID.....	28
Parameters.....	29
HTTP Method	29
Response	29
Example.....	29
Execution of an Action.....	31
URL	31
Parameters.....	31
HTTP Method	31
Response	31
Example.....	32
Attachment of a File to a Record.....	33
URL	33
Parameters.....	34
HTTP Method	34
Response	34
Example.....	34

Retrieval of a File Attached to a Record.....	36
URL	36
Parameters.....	36
HTTP Method	36
Response	37
Example.....	37
Retrieval of the Schema of Custom Fields.....	39
URL	39
Parameters.....	39
HTTP Method	39
Response	39

Contract-based REST API

The contract-based representational state transfer (REST) application programming interface (API) of MYOB Advanced provides the REST interface of the MYOB Advanced contract-based web services through which external systems can get data records from MYOB Advanced, process these records, and save new or updated records to MYOB Advanced.

Representation of a Record in JSON Format

By using the contract-based REST API, you obtain existing records from MYOB Advanced, create new records, update, and delete them. You work with the records in MYOB Advanced by using the entities that are defined in the contract of the endpoint that use to access the service. You pass records to and receive them from the contract-based REST API in JavaScript object notation (JSON) format. JSON is a text format for transmitting data objects that consist of key-value pairs.

To represent a record in JSON format, use the rules that are described in the following sections. You do not need to specify the values of all fields of an entity; you can specify the values of only the needed fields.

System Fields

Specify the value of a system field (such as ID, RowNumber, and Note) of an entity in the following format.

```
<Field name > : <Value>
```

For example, if you need to specify the note Imported for an entity, use the following string.

```
"Note" : "Imported"
```

General Fields

Specify the value of a general field (that is, a field that is not a system field) of an entity in the following format.

```
<Field name > : {value : <Value>}
```

For example, if you need to specify JOHNGOOD as the customer ID of a customer record, use the following string.

```
"CustomerID" : {value : "JOHNGOOD"}
```

Linked Entities

Specify the values of the fields of a linked entity in the following format.

```
<Field name > :  
  {  
    <List of fields of the linked entity with values  
  }  
}
```

For example, if you need to specify the values of an email address and the address of a customer main contact, use the following string.

```
"MainContact" :
{
  "Email" : {value : "demo@gmail.com" },
  "Address" :
  {
    "AddressLine1" : {value : "4030 Lake Washington Blvd NE" },
    "AddressLine2" : {value : "Suite 100" },
    "City" : {value : "Kirkland" },
    "State" : {value : "WA" },
    "PostalCode" : {value : "98033" }
  }
}
```

Detail Entities

Specify the values of the fields of a detail entity in the following format.

```
<Field name > : [
  {
    <List of fields of the detail entity with the values>
  },
  {
    <List of fields of the detail entity with the values>
  },
  ...
]
```

For example, if you need to specify the values of two detail lines of a sales order, use the following string.

```
"Details" : [
  {
    "InventoryID" : {value: "AALEGO500"},
    "Quantity" : {value: 10}, "UOM" : {value: "PIECE"},
  },
  {
    "InventoryID" : {value: "CONGRILL"},
    "Quantity" : {value: 1},
    "UOM" : {value: "PIECE"},
  }
]
```

Custom Fields

Specify the values of the custom fields (that is, the fields that are not included in the contract of the endpoint) in the following format.

```
"custom" :
{
  <View name >:
  {
    <Field name > :
    {
      "type" : <value>,
      "value" : <value>
    }
  }
}
```

Use this block in the JSON representation of the entity (top-level, detail, or linked) that contains

this custom field. For example, suppose that you added the **Personal ID** element to the Main Contact area of the Customers form (AR.30.30.00) in a customization project. The Contact entity, which is available through the `MainContact` property of the `Customer` entity, contains the **Personal ID** custom element. This element has the `String` type and the `UsrPersonalID` field name and belongs to the `DefContact` data view. Therefore, to specify the value AB123456 of the **Personal ID** custom element for the customer with ID JOHNGOOD through the REST API, use the following string.

```
{
  "CustomerID" : {value : "JOHNGOOD" } ,
  "MainContact" :
  {
    "custom" :
    {
      "DefContact" :
      {
        "UsrPersonalID" :
        {
          "type" : "String",
          "value" : "AB123456"
        }
      }
    }
  }
}
```

Login to the Service

Each time your application starts work with the MYOB Advanced contract-based REST service, you have to log in to MYOB Advanced. To log in to MYOB Advanced, access the needed URL address with the POST HTTP method and pass the credentials in the request body. See details on the URL, parameters, HTTP method, and response format in the following sections.

URL

When you need to log in to MYOB Advanced, use the following URL:

```
http://<MYOB Advanced URL>/entity/auth/login
```

Replace <MYOB Advanced URL> with the URL of your MYOB Advanced instance.

For example, suppose that you want to log in to a local MYOB Advanced instance with the name AdvancedDB. You should use the following URL:

```
http://localhost/AdvancedDB/entity/auth/login.
```

Parameters

Use no parameters when you log in to MYOB Advanced.

HTTP Method

Use the POST HTTP method and pass the credentials for accessing MYOB Advanced in JSON format, as shown in the following example.

```
{
  "name " : {value : "admin" } ,
  "password" : {value : "123" },
  "company" : {value : "MyCompany" },
  "branch" : {value : "MYSTORE" }
}
```

Response

The response of a successful method call is *204 No Content*.

Example

The following code shows an example of a class that implements a login to MYOB Advanced through the REST application programming interface (API).

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            New HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept = {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };

        //Log in to MYOB Advanced
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }
}
```

```
void IDisposable.Dispose()
{
    _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
        new ByteArrayContent(new byte[0])).Wait();
    _httpClient.Dispose();
}
}
```

The following code logs in to MYOB Advanced when an instance of the `RestService` class, which is defined in the code fragment above, is created.

```
RestService rs = new RestService(
    Properties.Settings.Default.AcumaticaBaseUrl,
    Properties.Settings.Default.UserName,
    Properties.Settings.Default.Password,
    Properties.Settings.Default.Company, Properties.Settings.Default.Branch
);
```

Logout from the Service

Each time your application finishes work with the MYOB Advanced contract-based REST service, you have to log out from MYOB Advanced. To log out from MYOB Advanced, you access the needed URL address with the POST HTTP method and pass the credentials in the request body. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

When you need to log out from MYOB Advanced, use the following URL.

```
http://<MYOB Advanced URL>/entity/auth/logout
```

Replace <MYOB Advanced URL> with the URL of your MYOB Advanced instance.

For example, suppose that you want to log out from a local MYOB Advanced instance with the name `AdvancedDB`. You should use the following URL:

```
http://localhost/AdvancedDB/entity/auth/logout
```

Parameters

No parameters are required when you log out from MYOB Advanced.

HTTP Method

Use the POST HTTP method to log out from MYOB Advanced.

Response

The response of a successful method call is *204 No Content*.

Example

The following code shows an example of a class that implements a logout from MYOB Advanced through the REST application programming interface (API). Logout is performed each time an instance of the RestService class is released.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };

        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    //Log out from MYOB Advanced
    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }
}
```

Creation of a Record

When you need to create a record by using the contract-based REST API, access the needed URL address with the PUT HTTP method and pass the record representation in JSON format in the request body. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to create a record in MYOB Advanced, use the following URL:

```
http://<Base endpoint URL>/<Top-level entity>
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name >/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to create a record.

For example, suppose that you want to create a stock item record in a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to create a record:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/StockItem.
```

Parameters

You can use the following parameters when you retrieve a record from MYOB Advanced:

- \$expand: To specify the linked and detail entities to be expanded
- \$custom: To specify the fields that are not defined in the contract to be returned

HTTP Method

Use the PUT HTTP method and pass a record in JSON format in the request body. You can find details on how to represent a record in JSON format in Representation of a Record in JSON Format. See below for an example of a customer record representation in JSON format.

```
{
  "CustomerID" : {value : "JOHNGOOD" } ,
  "CustomerName" : {value : "John Good" },
  "MainContact" :
  {
    "Email" : {value : "demo@gmail.com" },
    "Address" :
    {
      "AddressLine1" : {value : "4030 Lake Washington Blvd NE" },
      "AddressLine2" : {value : "Suite 100" },
      "City" : {value : "Kirkland" },
      "State" : {value : "WA" },
      "PostalCode" : {value : "98033" }
    }
  }
}
```

Response

The response of a successful method call contains the created record in JSON format in the response body. The response includes only the values of the fields of the created record that were specified during creation of the record or that were specified to be returned by using the parameters of the request.

Example

The following code shows an example of a class that implements the creation of a record in Acumatica ERP through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }
}
```

```

//Data submission
public string Put(string entityName, string parameters,
string entity)
{
    var res = _httpClient
        .PutAsync(_acumaticaBaseUrl + "/entity/Default/6.00.001/" +
            entityName + "?" + parameters,
            new StringContent(entity, Encoding.UTF8,
                "application/json"))
        .Result
        .EnsureSuccessStatusCode();

    return res.Content.ReadAsStringAsync().Result;
}
}

```

The following code uses the `RestService.Put()` method, which is defined in the previous code fragment, to create a customer record.

```

public static void CreateCustomer()
{
    //Path to a source text file that contains
    //the new customer record in JSON format
    string entitySource = @"..\..\Input\Customer.txt";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );
    using (StreamReader sr = new StreamReader(entitySource))
    {
        //Read the customer record in JSON format from the file
        string entityAsString = sr.ReadToEnd().ToString();

        //Create a customer record
        string customer = rs.Put("Customer", null, entityAsString);
    }
}

```

Update of a Record

When you need to update an existing record by using the contract-based REST API, you access the needed URL with the PUT HTTP method and pass the record representation in JSON format in the request body. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to update a record in MYOB Advanced, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name >/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to update a record.

For example, suppose that you want to update a stock item record in a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You would use the following URL to update a record:

```
http://localhost/AdvancedDB/entity/ Default/6.00.001/StockItem.
```

Parameters

You can use the following parameters when you are updating a record in MYOB Advanced:

- \$filter: To specify filtering conditions that identify the record to be updated
- \$expand: To specify the linked and detail entities to be expanded
- \$custom: To specify the fields that are not defined in the contract to be returned

HTTP Method

Use the PUT HTTP method and pass a record in JSON format in the request body. You can find details on how to represent a record in JSON format in Representation of a Record in JSON Format.

To make it possible for the record to be found by MYOB Advanced, you can use any of the following approaches:

- Specify the values of the key fields in the record representation in JSON format.
- Specify the value of the ID property in the record representation in JSON format.
- Specify the filtering conditions that identify the record in the \$filter parameter of the method.

If you want to delete a detail line during update, you should specify true as the value of the delete property of the corresponding detail entity: "delete" : true. To identify the detail line to be deleted, you can specify either the values of the key fields of the detail line or the value of the ID property.

Response

The response of a successful method call contains the updated record in JSON format in the response body. The response includes only the values of the fields of the updated record that were specified during the update or that were specified to be returned by using the parameters of the request.

Example

The following code shows an example of a class that implements the update of a record in MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;
    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    //Data submission
    public string Put(string entityName, string parameters, string
        entity)
    {
        var res = _httpClient
            .PutAsync(_acumaticaBaseUrl + "/entity/Default/6.00.001/" +
                entityName + "?" + parameters,
                new StringContent(entity, Encoding.UTF8,
                    "application/json"))
            .Result

            .EnsureSuccessStatusCode();
        return res.Content.ReadAsStringAsync().Result;
    }
}
```


The following code uses the `RestService.PutWithFilter()` method, which is defined in the previous code fragment, to update an existing customer record that has the `demo@gmail.com` email address.

```
public static void UpdateCustomer()
{
    //Path to a source text file that contains an updated customer record
    in JSON format
    string entitySource = @"..\..\Input\Customer_Upd.txt";

    //REST service initialization
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    using (StreamReader sr = new StreamReader(entitySource))
    {
        //Read customer record in JSON format from file
        string entityAsString = sr.ReadToEnd().ToString();

        //Specify filtering parameters that identify the customer
        string parameter = "$filter=MainContact/Email eq
            'demo@gmail.com'";

        //Update the customer record
        string customer = rs.Put("Customer", parameter, entityAsString);
    }
}
```

Retrieval of a Record by Key Fields

To retrieve a record by the values of its key fields from MYOB Advanced by using the contract-based REST API, you access the needed URL address with the GET HTTP method and specify the fields that should be returned in the parameters of the method. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to obtain a particular record with the known key fields, use the following URL:

```
http://<Base endpoint URL>/<Top-level entity>/<Key value 1>/<Key value 2>
```

The URL has the following components:

- `<Base endpoint URL>` is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- `<Top-level entity>` is the name of the entity for which you are going to retrieve a record.
- `<Key value 1>` and `<Key value 2>` are the values of the key fields of the record to be retrieved. Use the number and order of key fields as they are defined on the corresponding MYOB Advanced form.

For example, suppose that you want to retrieve the sales order with order type SO and order number 000123 from a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to retrieve the sales order:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/SalesOrder/SO/000123.
```

Parameters

You can use the following parameters when you retrieve a record from MYOB Advanced:

- \$expand: To specify the linked and detail entities to be expanded
- \$custom: To specify the fields that are not defined in the contract to be returned

HTTP Method

Use the GET HTTP method to retrieve records.

Response

The response of a successful method call contains the retrieved record in JSON format in the response body. For details on record representation in JSON format, see “Representation of a Record in JSON Format” on page 6.

Example

The following code shows an example of a class that implements the retrieval of a record in MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
    }
};
```

```

        _httpClient.PostAsJsonAsync(
            _acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    //Retrieval of a record by key fields
    public string GetByKey(string entityName, string keys, string
        parameters)
    {
        var res = _httpClient.GetAsync(
            _acumaticaBaseUrl + "/entity/Default/6.00.001/" +
            entityName + "/" + keys + "?" + parameters)
            .Result
            .EnsureSuccessStatusCode();

        return res.Content.ReadAsStringAsync().Result;
    }
}

```

The following code uses the `RestService.GetByKey()` method, which is defined in the previous code fragment, to retrieve a sales order with detail lines from MYOB Advanced.

```

public static void ExportSODetails()
{
    //Sales order data
    string orderType = "SO";
    string orderNbr = "000001";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    //Specify the parameter to obtain the details of a sales order
    string parameters = "$expand=Details";

    //Retrieve a sales order by keys
    string stockItems = rs.GetByKey("SalesOrder", orderType + "/" +
        orderNbr, parameters);
}

```

Retrieval of a Record by ID

To retrieve a record by the value of the session entity ID from MYOB Advanced by using the contract-based REST API, you access the needed URL address with the GET HTTP method and specify the fields that should be returned in the parameters of the method. See the following sections for details on the URL, parameters, HTTP method, and response format.

Note: The session entity ID is a GUID that is assigned to each entity you work with during an MYOB Advanced session. You can obtain the value of the session entity ID from the ID property of an entity returned from MYOB Advanced. The session entity ID is different for each new session with MYOB Advanced. That is, you cannot use the session entity ID that you received in the previous session to work with the entity after a new login to MYOB Advanced.

URL

If you need to obtain a particular record with the session entity ID, use the following URL:

```
http://<Base endpoint URL>/<Top-level entity>/<Session entity ID>
```

The URL has the following components:

- <Base endpoint URL> is the URL of a contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to retrieve a record.
- <Session entity ID> is the session ID of the record to be retrieved.

For example, suppose that you want to retrieve the sales order with session entity ID 03efa858-2351-4bd5-ae06-3d9fb3b3c1e6 from a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to retrieve the sales order:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/SalesOrder/03efa858-2351-4bd5-ae06-3d9fb3b3c1e6.
```

Parameters

You can use the following parameters when you retrieve a record from MYOB Advanced:

- \$expand: To specify the linked and detail entities to be expanded
- \$custom: To specify the fields that are not defined in the contract to be returned

HTTP Method

Use the GET HTTP method to retrieve records.

Response

The response of a successful method call contains the retrieved record in JSON format in the response body. For details on record representation in JSON format, see Representation of a Record in JSON Format.

Retrieval of Records by Conditions

To retrieve records that satisfy the specified conditions from MYOB Advanced by using the contract-based REST API, you access the needed URL address with the GET HTTP method and specify filtering conditions in the parameters of the method. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to retrieve the list of records that satisfies the specified conditions, use the following URL:

```
http://<Base endpoint URL>/<Top-level entity>
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to retrieve the list of records.

For example, suppose that you want to retrieve the list of stock item records from a local Acumatica ERP instance with the name `AdvancedDB` by using the system endpoint with the name `Default` and Version `6.00.001`. You should use the following URL to retrieve the list of records:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/StockItem.
```

Parameters

You can use the following parameters when you retrieve records from MYOB Advanced:

- `$filter`: To specify filtering conditions on the records to be returned
- `$skip`: To specify the number of records to be skipped from the list of returned records
- `$top`: To specify the number of records to be returned in the list
- `$expand`: To specify the linked and detail entities to be expanded
- `$custom`: To specify the fields that are not defined in the contract to be returned

HTTP Method

Use the GET HTTP method to retrieve records.

Response

The response of a successful method call contains the retrieved records in JSON format in the response body. For details on record representation in JSON format, see [Representation of a Record in JSON Format](#).

Example

The following code shows an example of a class that implements the retrieval of records from MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
        _httpClient.PostAsJsonAsync(
            _acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    public string Get(string entityName, string parameters)
    {
        var res = _httpClient.GetAsync(
            _acumaticaBaseUrl + "/entity/Default/6.00.001/"
            + entityName + "?" + parameters).Result
            .EnsureSuccessStatusCode();

        return res.Content.ReadAsStringAsync().Result;
    }
}
```

The following code uses the `RestService.Get()` method, which is defined in the previous code fragment, to retrieve the list of stock item records that have the Active status in MYOB Advanced and have been modified within the past month. The code uses the `$expand` parameter to retrieve the vendor details of each record.

```
public static void ExportStockItems()
{
    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    //Specify the parameter to filter records by status and last
    modified date
    string parameters1 =
        "$filter=ItemStatus eq 'Active' and LastModified gt
        datetimeoffset'" + WebUtility.UrlEncode(new
        DateTimeOffset(DateTime.Now.AddMonths(-1))
        .ToString("yyyy-MM-ddTHH:mm:ss.fffK")) + "'";

    //Specify the parameter to obtain vendor details
    string parameters2 = "$expand=VendorDetails";

    //Retrieve the list of stock items
    string stockItems = rs.Get("StockItem", parameters1 + "&" +
        parameters2);
}
```

Retrieval of Data from an Inquiry Form

To retrieve data from an inquiry form of MYOB Advanced by using the contract-based REST API, you access the needed URL with the PUT HTTP method and pass the parameters of the inquiry in JSON format in the request body. See the following sections for details on and examples of the URL, parameter, HTTP method, and response format.

URL

If you need to retrieve data from an inquiry form, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>
```

The URL includes the following components:

- `<Base endpoint URL>` is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- `<Top-level entity>` is the name of the entity that corresponds to the inquiry form from which you are going to retrieve data.

For example, suppose that you want to retrieve data from the Inventory Summary form (IN.40.10.00) in a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You would use the following URL to retrieve data:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/InventorySummaryInquiry.
```

Parameter

When you are retrieving data from an inquiry form, you should use the \$expand parameter to expand the detail entity, which contains the results of the inquiry. For a detailed description of the parameter, see Parameters for Retrieving Records.

HTTP Method

Use the PUT HTTP method and pass parameters of the inquiry in JSON format in the request body. See below for an example of the representation of parameters of the Inventory Summary inquiry form in JSON format.

```
{
  "InventoryID" : {value : "AALEGO500" } ,
  "WarehouseID" : {value : "MAIN" }
}
```

Response

The response of a successful method call contains the data returned from an inquiry form in JSON format in the response body.

Example

The following code shows an example of a class that implements the retrieval of data from an inquiry form of MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    MediaTypeWithQualityHeaderValue.Parse("text/json")}
        };
    }
};
```



```

        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    //Data submission
    public string Put(string entityName, string parameters,
        string entity)
    {
        var res = _httpClient
            .PutAsync(_acumaticaBaseUrl + "/entity/Default/6.00.001/" +
                entityName + "?" + parameters,
                new StringContent(entity, Encoding.UTF8,
                    "application/json"))
            .Result
            .EnsureSuccessStatusCode();

        return res.Content.ReadAsStringAsync().Result;
    }
}

```

The following code uses the `RestService.Put()` method, which is defined in the previous code fragment, to retrieve the quantities of a stock item from the Inventory Summary inquiry form.

```

public static void ExportItemQty()
{
    //Path to a source text file that contains the parameters of
    //the inquiry in JSON format
    string entitySource = @"..\..\Input\InventorySummaryInquiry.txt";

    //REST service initialization
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    using (StreamReader sr = new StreamReader(entitySource))
    {
        //Read inquiry parameters in JSON format from a file
        string entityAsString = sr.ReadToEnd().ToString();

        //Specify the parameter to expand results of the inquiry
        string parameters = "$expand=Results";
    }
}

```

```
//Retrieve data from the inquiry
string stockItems =
    rs.Put("InventorySummaryInquiry", parameters, entityAsString);
}
}
```

Parameters for Retrieving Records

When you retrieve records from MYOB Advanced by using the contract-based REST API, you can use the URL parameters, which are described in this topic, to filter records by the specified conditions, expand particular entities, and retrieve the values of the fields that are not defined in the contract of the endpoint.

\$filter Parameter

Use this parameter to specify the conditions that determine which records should be selected from MYOB Advanced. Use OData URI conventions to specify the value of the parameter. The following examples illustrate the use of this parameter:

- `$filter=ItemStatus eq 'Active'`: Obtains stock item records that have the Active status in MYOB Advanced.
- `$filter=MainContact/Email eq 'demo@gmail.com'`: Obtains a customer record that has the demo@gmail.com email address. (The Email field is defined in a linked entity, which is available through the MainContact property.)
- `$filter=ItemStatus eq 'Active' and LastModified gt datetimeoffset '2016-07-15T10%3A31%3A28.402%2B03%3A00'`: Obtains stock item records that have the Active status in MYOB Advanced and have been modified later than July 15, 2016.

Note: You should encode date and time values in URL format before passing them as the value of the parameter. For example, you can encode the current date and time by using the `System.Net.WebUtility.UrlEncode()` method as follows: `WebUtility.UrlEncode(new DateTimeOffset(DateTime.Now).ToString("yyyy-MM-ddTHH:mm:ss.fffK"))`.

When you specify the value of the parameter, you can use the following functions as they are defined in OData:

- `substringof`
- `startswith`
- `endswith`

You can also use the following custom function to filter records by the values of the elements that are not defined in the contract of the endpoint: `cf.<Type name >(f='<View name >.<Field name >')`, where `<Type name >` is the type of the custom element, `<View name >` is the name of the data view that contains the element, and `<Field name >` is the name of the element. For example, if you want to obtain all records on the Stock Items (IN.20.25.00) form that have the value of the Post Class ID element equal to STOCKITEM, you would use the following parameter string:

```
$filter=cf.String(f='ItemSettings.PostClassID') eq 'STOCKITEM'.
```

\$top Parameter

Use this parameter to specify the number of records to be returned from MYOB Advanced. That is, if you specify N as the value of this parameter, the first N records will be returned from MYOB Advanced. For example, if you want to obtain only first five records from the list, use the following parameter string: `$top=5`.

\$skip Parameter

Use this parameter to specify the number of records to be skipped from the list of returned records. That is, if you specify N as the value of this parameter, the first N records will be skipped from the list of returned records. For example, if you do not want to obtain the first five records from the list, use the following parameter string: `$skip=5`.

If use the `$skip` and `$top` parameters together, the `$skip` parameter is applied first.

\$expand Parameter

Use this parameter to specify the linked and detail entities that should be expanded. By default, no linked or detail entities are expanded. That is, only fields of the top-level entity are returned. Use OData URI conventions to specify the value of this parameter. For example, if you want to obtain the values of warehouse detail lines of stock item records, use the following parameter string: `$expand=WarehouseDetails`.

\$custom Parameter

Use this parameter to specify the fields that are not defined in the contract of the endpoint to be returned from MYOB Advanced. That is, you can use this parameter to obtain both the values of predefined elements on an MYOB Advanced form that are not included in the entity definition and the values of elements that were added to the MYOB Advanced form in a customization project. Specify the element whose value should be returned in the following format: `<View name>.<Field name>`, where you replace `<View name>` with the name of the data view that contains the element and `<Field name>` with the internal name of the element.

For example, if you want to obtain the value of the Post Class ID element of the Stock Items (IN.20.25.00) form, use the following parameter string:
`$custom=ItemSettings.PostClassID`.

If you want to obtain the values of multiple custom elements, specify the custom elements to be returned divided by commas.

Removal of a Record

In the contract-based REST API, you can delete the record by the value of its key fields or by its session identifier. To delete a record from MYOB Advanced, you access the needed URL address with the DELETE HTTP method. See the following sections for details on the possible URL, parameters, HTTP method, and response format.

Note: If you need to delete a detail line of a record, you should use the PUT HTTP method, as described in “Update of a Record” on page 14.

URL for Removing by Key Fields

If you need to delete a record with known key fields, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>/<Key value 1>/<Key value 2>
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to delete a record.
- <Key value 1> and <Key value 2> are the values of the key fields of the record to be deleted. Use the number and order of key fields as they are defined on the corresponding MYOB Advanced form.

For example, suppose that you want to delete the sales order with order type SO and order number 000123 from a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to delete the sales order:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/SalesOrder/SO/000123.
```

URL for Removing by ID

If you need to delete a record with a known session ID, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>/<Session entity ID>
```

You replace <Base endpoint URL> with the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`. You replace <Top-level entity> with the name of the entity for which you are going to retrieve the list of records. Replace <Session entity ID> with the session entity ID (which is the GUID).

For example, suppose that you want to delete the sales order with session entity ID 03efa858-2351-4bd5-ae06-3d9fb3b3c1e6 from a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to delete the sales order:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/SalesOrder/03efa858-2351-4bd5-ae06-3d9fb3b3c1e6.
```

Parameters

Use no parameters when deleting a record.

HTTP Method

Use the DELETE HTTP method to retrieve records. You pass no content in the request body.

Response

The response of a successful method call is 204 No Content.

Example

The following code shows an example of a class that implements the removal of a record from MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })

        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }
}
```

```

void IDisposable.Dispose()
{
    _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
        new ByteArrayContent(new byte[0])).Wait();
    _httpClient.Dispose();
}

//Removal of a record
public string Delete(string entityName, string keys)
{
    var res = _httpClient.DeleteAsync(_acumaticaBaseUrl
        + "/entity/Default/6.00.001/" + entityName + "/" + keys).Result
        .EnsureSuccessStatusCode();

    return res.Content.ReadAsStringAsync().Result;
}
}

```

The following code uses the `RestService.Delete()` method, which is defined in the previous code fragment, to delete a stock item record.

```

public static void DeleteStockItem()
{
    //Stock item data
    string inventoryID = "ASDF";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    //Remove the stock item
    string stockItem = rs.Delete("StockItem", inventoryID);
}

```

Execution of an Action

To perform an action by using the contract-based REST API, you access the needed URL address with the POST HTTP method and pass the record representation in JSON format and parameters of the action in the request body. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to perform an action on an MYOB Advanced form, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>/<Action name >
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name >/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to perform an action.
- <Action name > is the name of the action that you are going to perform.

For example, suppose that you want to confirm a shipment in a local MYOB Advanced instance with the name `AdvancedDB` by using the system endpoint with the name `Default` and Version `6.00.001`. You should use the following URL to confirm a shipment:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/Shipment/ConfirmShipment.
```

Parameters

Use no parameters when performing an action.

HTTP Method

Use the POST HTTP method and pass the record to which the action should be applied and the parameters of the action in the request body in JSON format as follows.

```
{
  "entity" : <record in JSON format>,
  "parameters" : <parameters in JSON format>
}
```

You can find details on how to represent a record in JSON format in “Representation of a Record in JSON Format” on page 6.

Response

If the long-running operation that was initiated by the action is completed or wasn't created, the response is *204 No Content*. If the long-running operation is in progress, the response is *202 Accepted*; it has the `Location` header, which contains a URL that can be used to check the status of the operation by using the GET HTTP method. When the GET HTTP method with this URL returns *204 No Content*, the operation is completed.

Example

The following code shows an example of a class that implements the execution of an action in MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    MediaTypeWithQualityHeaderValue.Parse("text/json")}
        };
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    //Invocation of an action
    public string Post(string entityName, string actionName,
        string entityAndParameters)
    {
        var result = _httpClient
            .PostAsync(_acumaticaBaseUrl + "/entity/Default/6.00.001/"
                + entityName + "/" + actionName,
            new StringContent(entityAndParameters,
                Encoding.UTF8, "application/json"));
        var res = result.Result;
        var cont = res.Content.ReadAsStringAsync().Result;
        res.EnsureSuccessStatusCode();

        var dt = DateTime.Now;
    }
}
```



```

while (true)
{
    switch (res.StatusCode)
    {
        case HttpStatusCode.NoContent:
            return "No content";
        case HttpStatusCode.Accepted:
            if ((DateTime.Now - dt).Seconds > 30)
                throw new TimeoutException();
            Thread.Sleep(500);
            res = _httpClient.GetAsync(res.Headers.Location)
                .Result.EnsureSuccessStatusCode();
            continue;
        default:
            throw new InvalidOperationException(
                "Invalid process result: " + res.StatusCode);
    }
}
}
}

```

The following code uses the `RestService.Post()` method, which is defined in the previous code fragment, to release a sales order invoice.

```

public static void ReleaseSOInvoice()
{
    //Invoice to be released
    string invoice = "{\"Type\" : {value : \"Invoice\"}, \"
        + \"ReferenceNbr\" : {value : \"INV000045\"} }";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    //Release the invoice
    invoice = rs.Post("SalesInvoice", "ReleaseSalesInvoice",
        "{\"entity\" : " + invoice + ", \"parameters\" : null}");
}

```

Attachment of a File to a Record

When you need to attach a file to a record by using the contract-based REST API, you access the needed URL address with the PUT HTTP method and pass the file in the request body. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to attach a file to a record in MYOB Advanced, use the following URL:

```

http://<Base endpoint URL>/<Top-level entity>/<Key value 1>/<Key value 2>/files/<File name >

```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name>/<Endpoint version>/`.
- <Top-level entity> is the name of the entity to which you are going to attach a file.
- <Key value 1> and <Key value 2> are the values of the key fields of the record to which you are going to attach a file. Use the number and order of key fields as they are defined on the corresponding MYOB Advanced form.
- <File name > is the name of the file that you are going to attach with the extension.

For example, suppose that you want to attach the Sample.jpg file to the stock item record with inventory ID AALEGO500 in a local MYOB Advanced instance with name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You should use the following URL to attach the file:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/StockItem/AALEGO500/files/Sample.jpg.
```

Parameters

Use no parameters when you attach a file to a record.

HTTP Method

Use the PUT HTTP method and pass the file to be attached in the request body.

Response

The response of a successful method call is *204 No Content*.

Example

The following code shows an example of a class that implements the attachment of a file to a record in MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
```

```

        {
            Accept =
                {MediaTypeWithQualityHeaderValue.Parse("text/json")}
        }
    };
    _httpClient.PostAsJsonAsync(
        acumaticaBaseUrl + "/entity/auth/login", new
        {
            name = userName,
            password = password,
            company = company,
            branch = branch
        }).Result
        .EnsureSuccessStatusCode();
}

void IDisposable.Dispose()
{
    _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
        new ByteArrayContent(new byte[0])).Wait();
    _httpClient.Dispose();
}

//Attachment of a file to a record
public string PutFile(string entityName, string keys,
    string fileName, System.IO.Stream file)
{
    var res = _httpClient.PutAsync(_
        acumaticaBaseUrl + "/entity/Default/6.00.001/"
        + entityName + "/" + keys + "/files/" + fileName,
        new StreamContent(file)).Result
        .EnsureSuccessStatusCode();
    return res.Content.ReadAsStringAsync().Result;
}
}

```

The following code uses the `RestService.PutFile()` method, which is defined in the previous code fragment, to attach a file to a stock item record.

```

public static void PutFile()
{
    //Input data
    string inventoryID = "AALEGO500";
    string fileName = "T2MCRO.jpg";
    string entitySource = @"..\..\Input\T2MCRO.jpg";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    using (StreamReader sr = new StreamReader(entitySource))
    {
        //Attach a file to a stock item record
        string stockItem = rs.PutFile("StockItem", inventoryID, fileName,
            sr.BaseStream);
    }
}
}

```

Retrieval of a File Attached to a Record

To retrieve a file that is attached to a record from MYOB Advanced by using the contract-based REST API, you access the URL address of the file with the GET HTTP method. See the following sections for details on the URL, parameters, HTTP method, and response format.

URL

If you need to obtain a file attached to a record, use the following URL.

```
http://<Base endpoint URL>/files/<File identifier>
```

The URL has the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name >/<Endpoint version>/`.
- <File identifier> is the internal identifier of the file in the system.

To get this URL for a particular file attached to a record, you should obtain the record from MYOB Advanced and, in the returned JSON representation of the record, find the value of the href property of the needed file in the files array. For information on how to retrieve a record from MYOB Advanced, see “Retrieval of a Record by Key Fields” on page 17 and “Retrieval of a Record by ID” on page 20.

For example, suppose that you retrieved the stock item record that contains the following files array from a local MYOB Advanced instance with the name AdvancedDB.

```
{
  ...,
  "files": [
    {
      "id": "9be45eb7-f97d-400b-96a5-1c4cf82faa96",
      "filename": "Stock Items (AAMACHINE1)\\T2MCRO.jpg",
      "href":
        "/AdvancedDB/entity/Default/6.00.001/files/9be45eb7-f97d-400b-96a5-1c4cf82faa96"
    }
  ]
}
```

You should use the following URL to retrieve the T2MCRO.jpg file attached to the stock item record:

```
http://localhost/AdvancedDB/entity/Default/6.00.001/files/9be45eb7-f97d-400b-96a5-1c4cf82faa96.
```

Parameters

Use no parameters when retrieving a file.

HTTP Method

Use the GET HTTP method to retrieve a file.

Response

The response of a successful method call contains the retrieved file in the response body.

Example

The following code shows an example of a class that implements the retrieval of a file attached to a record in MYOB Advanced through the REST API.

```
public class RestService: IDisposable
{
    private readonly HttpClient _httpClient;

    private readonly string _acumaticaBaseUrl;

    public RestService(
        string acumaticaBaseUrl, string userName, string password,
        string company, string branch)
    {
        _acumaticaBaseUrl = acumaticaBaseUrl;
        _httpClient = new HttpClient(
            new HttpClientHandler
            {
                UseCookies = true,
                CookieContainer = new CookieContainer()
            })
        {
            BaseAddress = new Uri(acumaticaBaseUrl +
                "/entity/Default/6.00.001/"),
            DefaultRequestHeaders =
            {
                Accept =
                    {MediaTypeWithQualityHeaderValue.Parse("text/json")}
            }
        };
        _httpClient.PostAsJsonAsync(
            acumaticaBaseUrl + "/entity/auth/login", new
            {
                name = userName,
                password = password,
                company = company,
                branch = branch
            }).Result
            .EnsureSuccessStatusCode();
    }

    void IDisposable.Dispose()
    {
        _httpClient.PostAsync(_acumaticaBaseUrl + "/entity/auth/logout",
            new ByteArrayContent(new byte[0])).Wait();
        _httpClient.Dispose();
    }

    //Retrieval of a record by key fields
    public string GetByKey(string entityName, string keys, string
        parameters)
    {
        var res = _httpClient.GetAsync(
            _acumaticaBaseUrl + "/entity/Default/6.00.001/" +
            entityName + "/" + keys + "?" + parameters)
            .Result
            .EnsureSuccessStatusCode();
    }
}
```

```

        return res.Content.ReadAsStringAsync().Result;
    }

    //Retrieving of a file
    public System.IO.Stream GetFile(string href)
    {
        var res = _httpClient.GetAsync(href).Result
            .EnsureSuccessStatusCode();
        return res.Content.ReadAsStreamAsync().Result;
    }
}

```

The following code uses the `RestService.GetByKey()` and `RestService.GetFile()` methods, which are defined in the previous code fragment, to retrieve a file attached to a stock item record from MYOB Advanced.

```

public static void GetFile()
{
    //Specify stock item data
    string inventoryID = "AAMACHINE1";

    //Initialize the REST service
    RestService rs = new RestService(
        Properties.Settings.Default.AcumaticaBaseUrl,
        Properties.Settings.Default.UserName,
        Properties.Settings.Default.Password,
        Properties.Settings.Default.Company,
        Properties.Settings.Default.Branch
    );

    //Retrieve the stock item
    string stockItem = rs.GetByKey("StockItem", inventoryID, null);

    //Find href and the file name of the needed file
    //(using Newtonsoft.Json.Linq and System.IO)
    JObject jItem = JObject.Parse(stockItem);
    JArray jFiles = jItem.Value<JArray>("files");
    string fileRef = jFiles[0].Value<string>("href");
    string fullFileName = jFiles[0].Value<string>("filename");
    string fileName = Path.GetFileName(fullFileName);

    //Obtain the file
    Stream file = rs.GetFile(fileRef);

    using (var outputFile = File.Create(@"..\..\Output\" + fileName))
    {
        file.Seek(0, SeekOrigin.Begin);
        file.CopyTo(outputFile);
    }
}

```

Retrieval of the Schema of Custom Fields

To retrieve the schema of custom fields of an entity—that is, the field name, view name, and type of the fields that are not defined in the contract of the endpoint for this entity—by using the contract-based REST API, you access the needed URL with the GET HTTP method. See the following sections for details on and examples of the URL, parameters, HTTP method, and response format.

URL

If you need to obtain the schema of custom fields of an entity, use the following URL.

```
http://<Base endpoint URL>/<Top-level entity>/$adHocSchema
```

The URL includes the following components:

- <Base endpoint URL> is the URL of the contract-based endpoint through which you are going to work with MYOB Advanced, which has the following format: `http://<MYOB Advanced instance URL>/entity/<Endpoint name >/<Endpoint version>/`.
- <Top-level entity> is the name of the entity for which you are going to retrieve the schema of custom fields.

For example, suppose that you want to obtain the schema of custom fields of a stock item entity from a local MYOB Advanced instance with the name AdvancedDB by using the system endpoint with the name Default and Version 6.00.001. You would use the following URL to retrieve the schema:

```
http:// localhost/AdvancedDB/entity/Default/6.00.001/StockItem/$adHocSchema.
```

Parameters

Use no parameters when retrieving the schema of custom fields of an entity.

HTTP Method

Use the GET HTTP method to retrieve the schema of custom fields.

Response

The response of a successful method call contains the schema of custom fields in JSON format in the response body.