

AcquireLocal Campaign & Activity API Documentation for *PartnerName*

The pages following are the initial documentation for the API which *PartnerName* will use to create marketing campaigns and activities to the AcquireLocal system.

As this API implementation is completed, there may be minor changes or additions. Changes will be communicated immediately to *PartnerName*.

The AcquireLocal Reporting API for *PartnerName* will be described in a separate document.

Notes on the *PartnerName* Implementation of this API:

1. The **AgencyID** referenced in the documentation is a key value intended to keep *PartnerName* data separate and secure relative to other AcquireLocal users. This is a fixed integer value. For the *PartnerName* implementation, the value of AgencyID is "2"
2. AcquireLocal tracks all marketing campaigns and activities within separate "Customer" accounts, identified with the CustomerID field. These accounts correspond to the store, dealer, or restaurant that is the end-customer of this program.

This ID is an alphanumeric value that can be provided by *PartnerName*, or can be generated automatically by AcquireLocal. The former option is preferred, allowing activity reports to link easily to *PartnerName's* own reporting.
3. The **ListSource** parameter in the **CreateCampaign** call indicates how the target list data will be sourced. In the *PartnerName* implementation, all data files will be provided by *PartnerName*, so the value of this parameter will always be "2".
4. The *PartnerName* PO Number should be passed to the **CreateCampaignActivity** procedure as the "**InternalActivityID**" parameter.

■ AcquireLocal Campaign and Activity API

DEFINITIONS

On the AcquireLocal platform some common terms are used in very specific ways.

- Agency:** The client of AcquireWeb that is licensing the AcquireLocal platform. An Agency provides local marketing services to its end customers.
- Agency ID:** A fixed-value integer assigned by AcquireWeb. This ID identifies the Agency.
- Customer Key:** A unique integer assigned to each end-customer by AcquireLocal
- Customer ID:** A unique identifier for each end-customer provided from the Agency.
- Campaign:** A “container” for marketing activities, a campaign is assigned to a specific end-customer. Before activities can be ordered, a campaign must be created generating a unique Campaign Key.
- Customer:** The clients of the Agency (see above, for whom the local marketing programs are being fielded.). Each customer location: the store, restaurant or dealership, is tied to a specific geography and a marketing list centered on that location.
- Activity:** An order for a specific marketing program, for one of four channels (Email, Facebook, Display, & Mobile). Each campaign activity ordered must be assigned to an active campaign. Multiple campaign activities, of one or multiple channels may be grouped within a campaign.
- Creative:** The marketing materials to be deployed. This includes the HTML text of the email, the subject line and any images associated with the email or other advertising
- Seed List:** The Seeds are email addresses for program administrators and customer personnel that should receive the marketing message, but who are not included in the target list. Often the seed list is used for approval testing. In AcquireLocal the seed list includes addresses that are specific to a customer and others that are common to all customers at an Agency.

NOTES:

- URL links for Creatives, Images and Datafile must be complete links including filename
- The SeedList parameter in the CreateEmail Activity call should be a comma-separated list of email addresses.
- The “ListSource” parameter in the AddCampaign call is an integer enumeration that indicates the source of the target list data.

API METHODS

These methods create a new campaign and attach marketing activities:

1. AddCampaign
2. AddCampaignEmailActivity
3. AddCampaignFacebookActivity
4. AddCampaignDisplayActivity

These methods return information about previously scheduled Campaigns and Activities

5. GetCampaign
6. GetCampaignListProgressReport
7. GetCampaignProgressReport

These methods return the list of ID and Name values for several enumerations used in the API

8. GetActivityStatusList
9. GetCampaignStatusList
10. GetActivityChannelList

This method obtains the Authentication token which must be included in all subsequent calls

11. Authenticate

Using the AcquireLocal API:

The first step in all communication with this API is to call the Authenticate method, passing a username and password, and receiving a string token that must be provided in the headers of subsequent calls. A Campaign should be created before ordering any Campaign Activities.

STEP1: Authenticate

Provide credentials; receive Token

STEP 2: Create Campaign

Provide Campaign details and Authentication token (from Step 1)

Receive Campaign ID

STEP 3: Attach Activities

Provide Activity details, Campaign ID and Authentication token

Receive Activity ID

NOTE: Save the Campaign and Activity IDs returned to use to query status reports from the API

... Or, use the "Get" calls listed above to retrieve a complete list of campaigns or activities for your Agency

DETAIL ON API METHODS

1. AddCampaign

Purpose: Create a new campaign

Input: AgencyID, CustomerID, CustomerKey, CampaignID, CampaignName, StartDate, EndDate, ListSource

Output: StatusID, ErrorMessage, CampaignKey

HTTP method: POST /al.api/campaign

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

AddCampaign Request JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AgencyID": {
      "type": "integer"
    },
    "CustomerID": {
      "type": "string"
    },
    "CustomerKey": {
      "type": "integer"
    },
    "CampaignID": {
      "type": "string"
    },
    "CampaignName": {
      "type": "string"
    },
    "StartDate": {
      "type": "string",
      "format": "date-time"
    },
    "EndDate": {
      "type": "string",
      "format": "date-time"
    },
    "ListSource": {
      "type": "integer"
    }
  },
  "required": [
    "AgencyID",
    "CustomerID",
    "CustomerKey",
    "CampaignID",
    "CampaignName",
    "StartDate",
    "EndDate",
    "ListSource"
  ]
}
```

AddCampaign Response JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "StatusId": {
      "type": "integer"
    },
    "ErrorMessages": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "CampaignKey": {
      "type": "integer"
    }
  },
  "required": [
    "StatusId",
    "ErrorMessages",
    "CampaignKey"
  ]
}
```

2. AddCampaignEmailActivity

Purpose: Add new Email Activity to a Campaign

Input: AgencyID, CampaignID, InternalActivityID, QuantityOrdered, StartDate, NumberOfDeployments, DaysBetweenDeployments, FromNameandEmail, ReplyToNameAndEmail, SubjectLine, DataFileURL, CreativeHTMLURL, CreativeImagesURL, SeedAddressList

Output: StatusID, ErrorMessage, CampaignActivityKey

HTTP method: POST /al.api/emailActivity

Expected HTTP response codes: 200,400,401

Required HTTP Request JSON Schema headers: Authentication (value prefixed with `Acquireweb.SSO`)

AddCampaignEmailActivity Request JSON Schema:

```
{ "$schema": "http://json-schema.org/draft-04/schema#", "type": "object",
  "properties": {
    "AgencyID": {
      "type": "integer"
    },
    "CampaignID": {
      "type": "string"
    },
    "InternalActivityID": {
      "type": "string"
    },
    "QuantityOrdered": {
      "type": "integer"
    },
    "StartDate": {"type": "string"},
    "NumberOfDeployments": {
      "type": "integer"
    },
    "DaysBetweenDeployments": {
      "type": "integer"
    },
    "FromNameandEmail": {
      "type": "string"
    },
    "ReplyToNameAndEmail": {
      "type": "string"
    },
    "SubjectLine": {
      "type": "string"
    },
    "DataFileURL": {
      "type": "string"
    },
    "CreativeHTMLURL": {
      "type": "string"
    },
    "CreativeImagesUrls": {
      "type": "array",
      "items": {"type": "string"}
    },
    "SeedAddressList": {
      "type": "array", "items": { "type": "string"}
    }
  },
  "required": [
    "AgencyID",
    "CampaignID",
    "QuantityOrdered",
    "StartDate",
    "NumberOfDeployments",
    "DaysBetweenDeployments",
    "FromNameandEmail",
    "ReplyToNameAndEmail",
    "SubjectLine",
    "DataFileURL",
    "CreativeHTMLURL",
    "CreativeImagesUrls",
  ]
}
```

AddCampaignEmailActivity Response JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "StatusId": {
      "type": "integer"
    },
    "ErrorMessages": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "CampaignActivityKey": {
      "type": "integer"
    }
  },
  "required": [
    "StatusId",
    "ErrorMessages",
    "CampaignActivityKey"
  ]
}
```

3. AddCampaignFacebookActivity

Purpose: to add new Facebook Activity to a campaign

Input: AgencyID, CampaignID, InternalActivityID, QuantityOrdered, StartDate, EndDate, DataFileURL, CreativesImagesURL

Output: StatusID, ErrorMessage, CampaignActivityKey, AdPlacementID

HTTP method: POST /al.api/facebookactivity

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

AddCampaignFacebookActivity Request JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AgencyID": {
      "type": "integer"
    },
    "CampaignID": {
      "type": "string"
    },
    "InternalActivityID": {
      "type": "string"
    }
  },
}
```

```

        "QuantityOrdered": {
            "type": "integer"
        },
        "StartDate": {
            "type": "string"
        },
        "EndDate": {
            "type": "string"
        },
        "DataFileURL": {
            "type": "string"
        },
        "CreativeImagesUrls": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "AdPlacementId": {
            "type": "integer"
        },
        "LandingPageURL": {
            "type": "string"
        }
    },
    "required": [
        "AgencyID",
        "CampaignID",
        "QuantityOrdered",
        "StartDate",
        "EndDate",
        "DataFileURL",
        "CreativeImagesUrls",
        "AdPlacementId",
        "LandingPageURL"
    ]
}

```

AddCampaignFacebookActivity Response JSON Schema:

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "StatusId": {
            "type": "integer"
        },
        "ErrorMessages": {
            "type": "array", "items": {"type": "string"}
        },
        "CampaignActivityKey": {
            "type": "integer"
        }
    },
    "required": [
        "StatusId",
        "ErrorMessages",
        "CampaignActivityKey"
    ]
}

```

4. AddCampaignDisplayActivity

Purpose: to add new Display Activity to a campaign

Input: AgencyID, CampaignID, InternalActivityID, QuantityOrdered, StartDate, EndDate, DataFileURL, LandingPageURL, AdPlacementID

Output: StatusID, ErrorMessage, CampaignActivityKey

HTTP method: POST /al.api/displayactivity

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

AddCampaignDisplayActivity Request JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AgencyID": {
      "type": "integer"
    },
    "CampaignID": {
      "type": "string"
    },
    "InternalActivityID": {
      "type": "string"
    },
    "QuantityOrdered": {
      "type": "integer"
    },
    "StartDate": {
      "type": "string"
    },
    "EndDate": {
      "type": "string"
    },
    "DataFileURL": {
      "type": "string"
    },
    "AdPlacementId": {
      "type": "integer"
    },
    "LandingPageURL": {
      "type": "string"
    }
  },
  "required": [
    "AgencyID",
    "CampaignID",
    "QuantityOrdered",
    "StartDate",
    "EndDate",
    "DataFileURL",
    "AdPlacementId",
    "LandingPageURL"
  ]
}
```

AddCampaignDisplayActivity Response JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "StatusId": {
      "type": "integer"
    },
    "ErrorMessages": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "CampaignActivityKey": {
      "type": "integer"
    }
  },
  "required": [
    "StatusId",
    "ErrorMessages",
    "CampaignActivityKey"
  ]
}
```

5. GetCampaign

Purpose: to retrieve existing campaign details (including all existing campaign activities)

Input: AgencyID (required), CampaignID or CampaignKey (required)

Output: *all campaign and campaign activity fields*

HTTP method: GET /al.api/campaign/{AgencyID}/Id{CampaignId}

GET /al.api/campaign/{AgencyID}/key{CampaignKey}

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

GetCampaign Request JSON Schema: none

GetCampaign Response JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "ErrorMessages": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "Campaign": {
```

```
"type": "object",
"properties": {
  "CampaignKey": {
    "type": "integer"
  },
  "AgencyID": {
    "type": "integer"
  },
  "CustomerID": {
    "type": "string"
  },
  "CustomerKey": {
    "type": "integer"
  },
  "CampaignID": {
    "type": "string"
  },
  "CampaignName": {
    "type": "string"
  },
  "StartDate": {
    "type": "string"
  },
  "EndDate": {
    "type": "string"
  },
  "ListSource": {
    "type": "integer"
  },
  "ListStatus": {
    "type": "integer"
  },
  "ListQuantity": {
    "type": "integer"
  },
  "ListName": {
    "type": "string"
  },
  "ExternalCampaignID": {
    "type": "string"
  },
  "StatusId": {
    "type": "integer"
  },
  "CampaignActivities": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "ActivityKey": {
          "type": "integer"
        },
        "AgencyID": {
          "type": "integer"
        },
        "CampaignID": {
          "type": "string"
        },
        "InternalActivityID": {
          "type": "string"
        }
      }
    }
  }
}
```

```
    "ExternalActivityID": {
      "type": "string"
    },
    "ChannelID": {
      "type": "integer"
    },
    "QuantityOrdered": {
      "type": "integer"
    },
    "StartDate": {
      "type": "string"
    },
    "NumberOfDeployments": {
      "type": "integer"
    },
    "DaysBetweenDeployments": {
      "type": "integer"
    },
    "FromNameandEmail": {
      "type": "string"
    },
    "ReplyToNameAndEmail": {
      "type": "string"
    },
    "SubjectLine": {
      "type": "string"
    },
    "DataFileURL": {
      "type": "string"
    },
    "CreativeHTMLURL": {
      "type": "string"
    },
    "CreativeImagesUrls": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "SeedAddressList": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "AdPlacement": {
      "type": "string"
    }
  },
  "required": [
    "ActivityKey",
    "AgencyID",
    "CampaignID",
    "InternalActivityID",
    "ExternalActivityID",
    "ChannelID",
    "QuantityOrdered",
    "StartDate",
    "NumberOfDeployments",
    "DaysBetweenDeployments",
    "FromNameandEmail",
```

```

        "ReplyToNameAndEmail",
        "SubjectLine",
        "DataFileURL",
        "CreativeHTMLURL",
        "CreativeImagesUrls",
        "SeedAddressList",
        "AdPlacement"
    ]
}
},
"required": [
    "CampaignKey",
    "AgencyID",
    "CustomerID",
    "CustomerKey",
    "CampaignID",
    "CampaignName",
    "StartDate",
    "EndDate",
    "ListSource",
    "ListStatus",
    "ListQuantity",
    "ListName",
    "ExternalCampaignID",
    "StatusId"
]
},
"required": [
    "StatusId",
    "ErrorMessages",
    "Campaign"
]
}
}

```

6. GetActivityStatusList

Purpose: to retrieve list of ID and Name values from CampaignActivityStatus table

Input: none

Output: ID, Name

HTTP method: GET /al.api/activityStatusList/[AgencyID]

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

GetActivityStatusList Request JSON Schema: none

GetActivityStatusList Response JSON Schema:

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "ErrorMessages": {
            "type": "array",

```

```

        "items": {
            "type": "string"
        }
    },
    "Statuses": {
        "type": "array",
        "items": {
            "type": "object",
            "properties": {
                "Id": {
                    "type": "integer"
                },
                "Name": {
                    "type": "string"
                }
            },
            "required": [
                "Id",
                "Name"
            ]
        }
    },
    "required": [
        "ErrorMessages",
        "Statuses"
    ]
}

```

7. GetCampaignStatusList

Purpose: to retrieve list of ID and Name values from CampaignStatus table

Input: none

Output: ID, Name

HTTP method: GET /al.api/campaignStatusList/(AgencyID)

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

GetCampaignStatusList Request JSON Schema: none

GetCampaignStatusList Response JSON Schema:

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "ErrorMessages": {
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    },
}

```

```

        "Statuses": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "Id": {
                        "type": "integer"
                    },
                    "Name": {
                        "type": "string"
                    }
                }
            },
            "required": [
                "Id",
                "Name"
            ]
        }
    },
    "required": [
        "ErrorMessages",
        "Statuses"
    ]
}

```

8. GetActivityChannelList

Purpose: to retrieve list of ID and Name values from CampaignActivityChannel table

Input: none

Output: ID, Name

HTTP method: GET /al.api/activityChannelList/(AgencyID)

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

Request JSON Schema:none

Response JSON Schema:

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "ErrorMessages": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "Statuses": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "Id": {

```

```

        "type": "integer"
    },
    "Name": {
        "type": "string"
    }
},
"required": [
    "Id",
    "Name"
]
}
},
"required": [
    "ErrorMessages",
    "Statuses"
]
}

```

9. GetCampaignListProgressReport

Purpose: to retrieve progress snapshot for a list of campaigns

Input: AgencyID(required), FromDate(required), ToDate(required), StatusID (optional)

Output: CampaignID, CampaignName, CampaignStatusId, ExternalCustomerID, for each campaign activity: [ActivityID, ChannelID, ActivityStatusID, ErrorMessage, QuantityOrdered, QuantityAchieved]

HTTP method: GET /al.api/campaignListReport/{AgencyId}/{StartDate}/{EndDate}/{StatusId}

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

GetCampaignListProgressReport Request JSON Schema: none

GetCampaignListProgressReport Response JSON Schema:

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "ErrorMessages": {
      "type": "array",
      "items": {"type": "string"}
    },
    "Reports": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "CampaignID": {
            "type": "string"
          },
          "CampaignName": {
            "type": "string"
          }
        }
      }
    }
  }
}

```



```

        "CampaignStatusId": {
            "type": "integer"
        },
        "ExternalCampaignID ": {
            "type": "string"
        },
        "ActivityReports": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "ACTivityKey": {
                        "type": "integer"
                    },
                    "StatusId": {
                        "type": "integer"
                    },
                    "ChannelId": {
                        "type": "integer"
                    },
                    "ErrorMessage": {
                        "type": "string"
                    },
                    "QuantityOrdered": {
                        "type": "integer"
                    },
                    "QuantityAchieved": {
                        "type": "integer"
                    }
                },
                "required": [
                    "ACTivityKey",
                    "StatusId",
                    "ChannelId",
                    "ErrorMessage",
                    "QuantityOrdered",
                    "QuantityAchieved"
                ]
            }
        },
        "required": [
            "CampaignID",
            "CampaignName",
            "CampaignStatusId",
            "ExternalCampaignID ",
            "ActivityReports"
        ]
    }
},
"required": [
    "ErrorMessages",
    "Reports"
]
}

```

10. GetCampaignProgressReport

Purpose: to retrieve campaign progress snapshot for a specific campaign

Input: AgencyID (required), CampaignID or CampaignKey (required)

Output: CampaignID, CampaignName, CampaignStatusId, ExternalCustomerID, for each campaign activity: [ActivityID, ChannelID, ActivityStatusID, ErrorMessage, QuantityOrdered, QuantityAchieved]

HTTP method: GET /al.api/campaignReport/{AgencyId}/id{CampaignId}

GET /al.api/campaignReport/{AgencyId}/key{CampaignKey}

Expected HTTP response codes: 200,400,401

Required HTTP headers: Authentication (value prefixed with **Acquireweb.SSO**)

GetCampaignProgressReport Request JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "AgencyID": {
      "type": "integer"
    },
    "CampaignID": {
      "type": "string"
    },
    "CampaignKey": {
      "type": "integer"
    }
  },
  "required": [
    "AgencyID",
    "CampaignID",
    "CampaignKey"
  ]
}
```

GetCampaignProgressReport Response JSON Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "ErrorMessages": {
      "type": "array",
      "items": {"type": "string"}
    },
    "CampaignProgressReport": {
      "type": "object",
      "properties": {
        "CampaignID": {
          "type": "string"
        }
      }
    }
  }
}
```

```

        "CampaignName": {
            "type": "string"
        },
        "CampaignStatusId": {
            "type": "integer"
        },
        "ExternalCampaignID ": {
            "type": "string"
        },
        "ActivityReports": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "ActivityKey": {
                        "type": "integer"
                    },
                    "StatusId": {
                        "type": "integer"
                    },
                    "ChannelId": {
                        "type": "integer"
                    },
                    "ErrorMessage": {
                        "type": "string"
                    },
                    "QuantityOrdered": {
                        "type": "integer"
                    },
                    "QuantityAchieved": {
                        "type": "integer"
                    }
                }
            },
            "required": [
                "ACTivityKey",
                "StatusId",
                "ChannelId",
                "ErrorMessage",
                "QuantityOrdered",
                "QuantityAchieved"
            ]
        }
    }, "required": [
        "CampaignID",
        "CampaignName",
        "CampaignStatusId",
        "ExternalCampaignID ",
        "ActivityReports"
    ]
}
},
"required": [
    "ErrorMessages",
    "CampaignProgressReport"
]

```

```
}
```

11. Authenticate

Purpose: to authenticate client requests and retrieve OAUTH token

Input: username (required), password (required)

Output: OAUTH token to be passed in Authentication HTTP header with every request

HTTP method: POST /aw.sso/authenticate

Expected HTTP response codes: 200, 400

Authenticate Request JSON Schema:

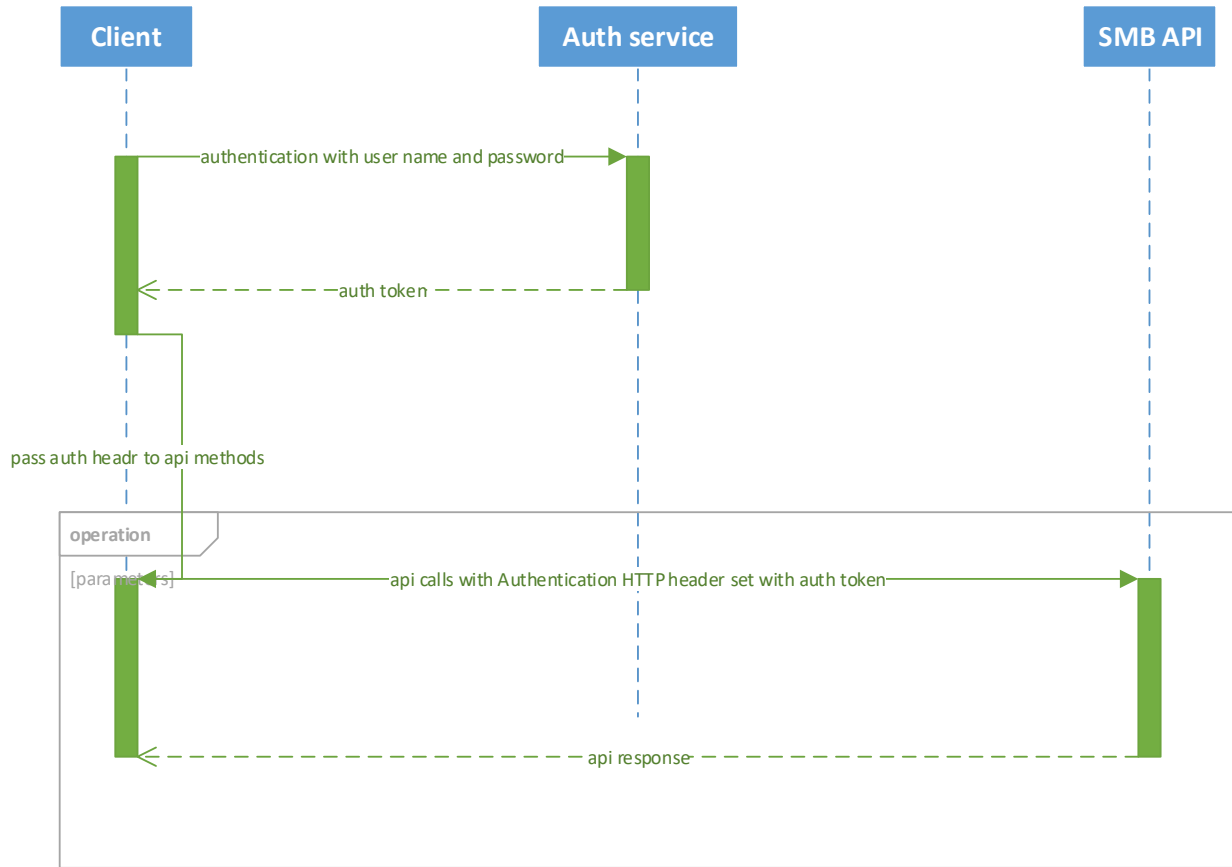
```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "UserName": {
      "type": "string"
    },
    "Password": {
      "type": "string"
    }
  },
  "required": [
    "UserName ",
    "Password"
  ]
}
```

Authenticate Response JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "Token": {
      "type": "string"
    },
    "Expires": {
      "type": "string",
      "format": "date-time"
    }
  },
  "required": ["Token", "Expires"]
}
```

AUTHENTICATION

1. OAUTH authentication diagram.



AcquireLocal Authentication — Sample web app integration

The AcquireLocal API Authentication sub-system includes a javascript helper library. This example of the system in use is intended to facilitate cross domain SSO functionality.

Initialize Authentication Service:

Below is a code snippet from a sample client-side html page.

```
<script type="text/javascript"
src="http://acquirewebapps.cloudapp.net:8082/Scripts/rsvp.js"></script>
<script type="text/javascript"
src="http://acquirewebapps.cloudapp.net:8082/Scripts/clientApp.js"></script>
<iframe src="http://acquirewebapps.cloudapp.net:8082/" id="authFrame" style="display:none"
></iframe>

<script type="text/javascript">
    var authManager = new AcquirewebSsoManager();

    authManager.initAuthApp().then(function(data) {
        if(authManager.loggedIn){
            _login();
            log('auth app inited');
        }else{
            _logout();
        }
    }, function(error) {
        _logout();
    });
```

This app loads a bridge iframe to contact the Auth service, and store the authentication token in local storage. Then the consumer app creates an instance of AcquirewebSsoManager and waits for it to initialize.

Client-side Login Example:

The code below shows a sample login function embedded into the client's web page.

```
function onLoginClick(){
    var user = document.getElementById('userName').value;
    var pwd = document.getElementById('password').value;
    authManager.login(user,pwd).then(function(data) {
        _login();
        log('logged in');
    }, function(error) {
        switch (error.returnCode){
            case 400:
                log('wrong username or password');
                break;
        }
    });
}
```

Call API using Authentication Token:

The code below demonstrates a sample API call using an authentication token

```
function getDataClick(){
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'api/protected', true);
    xhr.setRequestHeader("Authorization", "Acquireweb.SSO " + authManager.accessToken);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 ) {
            log(xhr.responseText);
            if (xhr.status == 401){
                _logout();
            }
        }
    }
    xhr.send();
}
```

C# Login Example:

The C# example below first calls the authentication service, receives an authentication token in response, and then calls the web API using that authentication token.

```
using (HttpClient authClient = new HttpClient())
{
    using (HttpClient client = new HttpClient())
    {
        //authenticate
        authClient.BaseAddress = new Uri("http://acquirewebapps.cloudapp.net:8082");

        var result = authClient.PostAsync(
            "/api/account/token",
            new StringContent(
                JsonConvert.SerializeObject(new LoginInfo() { UserName = USER_NAME,
                    Password = PASSWORD })),
            Encoding.UTF8,
            "application/json").Result;

        AuthenticationResponse response =
            JsonConvert.DeserializeObject<AuthenticationResponse>(result.Content.ReadAsStringAsync().Result);
        Console.WriteLine($"Token {response.Token}");

        //call smb api
        client.BaseAddress = new Uri("http://static2.testdomain.com");
        client.DefaultRequestHeaders.Add("Authorization",
            $"Acquireweb.SSO {response.Token}");
        result = client.GetAsync("/sso.app5/api/protected").Result;

        var getDataresponse =
            JsonConvert.DeserializeObject<GetDataResponse>(result.Content.ReadAsStringAsync().Result);
        Console.WriteLine($"User {getDataresponse.user}");
        Console.ReadKey();
    }
}
```