

IODClient Library  
for Windows Universal 8.1  
Version 1.0

## Contents

1. Overview .....	3
2. Integrate IODClient into Windows project .....	3
3. IODClient API reference .....	4
3.1. IODClient(string apiKey, String version="v1"); .....	4
3.2. GetRequest(Dictionary<String, Object> aParams, String iodApp, REQ_MODE mode) .....	5
3.3. PostRequest(Dictionary<String, Object> aParams, String iodApp, REQ_MODE mode) .....	6
3.4. GetJobResult(String jobId) .....	7

## 1. Overview

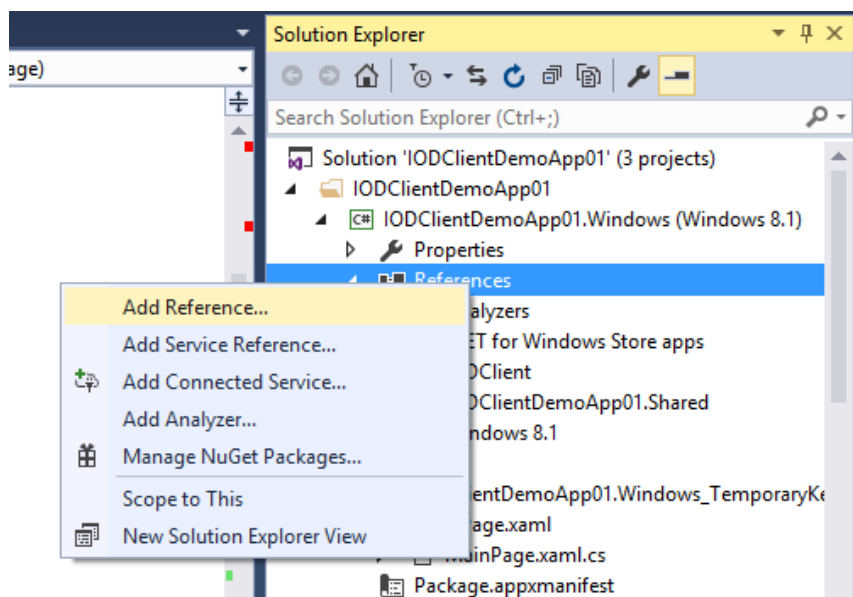
IODClient library for Windows Universal is a lightweight C# based API, which helps you easily integrate your Windows app with HP IDOL OnDemand Services.

IODClient library supports both Windows and Windows Phone 8.1.

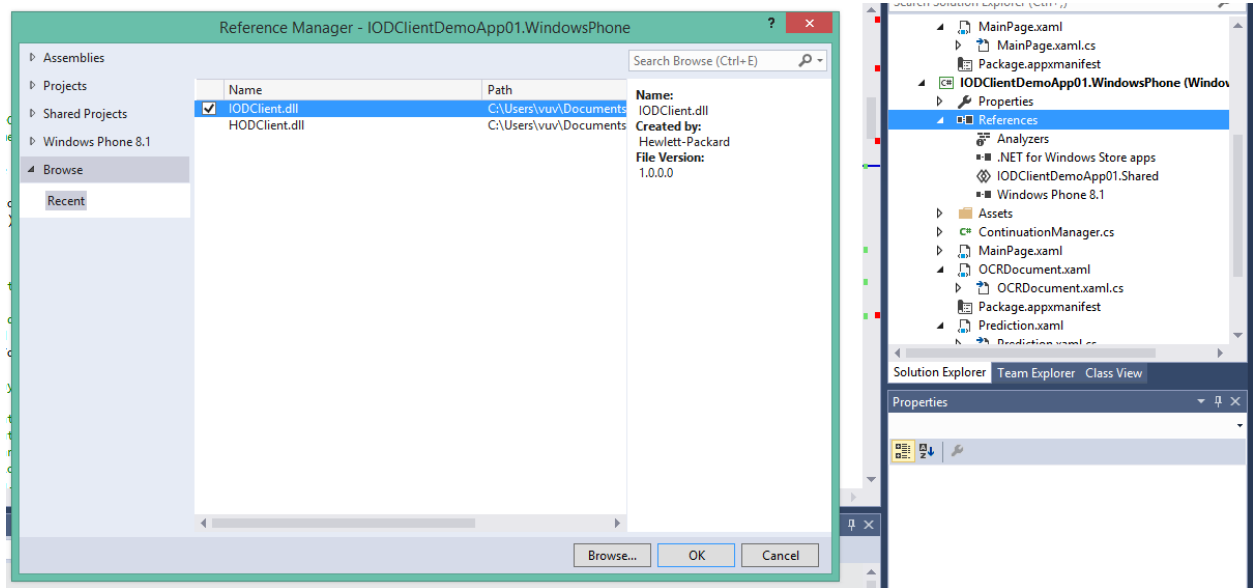
IODClient library requires the .NET 4.5.

## 2. Integrate IODClient into Windows project

- a) Download the IODClient.dll library for Windows Universal 8.1.
- b) Create a new or open an existing Windows project
- c) Add the IODClient.dll to the project.



- d) Browse to the folder where you saved the library and select the IODClient.dll.



### 3. IODClient API reference

#### 3.1. IODClient(string apiKey, String version="v1");

##### Description:

Constructor. Creates and initializes an IODClient object

##### Parameters:

**apiKey**: your developer apikey.

**version**: IDOL OnDemand API version. Currently it only supports version 1. Thus, the default value is "v1".

##### Example code:

```
using IOD.Client;

protected override void OnNavigatedTo(NavigationEventArgs e) {
    String apiKey = "my-api-key";
    IODClient iodClient = new IODClient(my-api-key);
    // implement delegated function
    iodClient.requestCompletedWithJobID +=
        iodClient_requestCompletedWithJobID;
    iodClient.requestCompletedWithContent +=
        iodClient_requestCompletedWithContent;
    iodClient.onErrorOccurred += iodClient_onErrorOccurred;
}
```

```

void iodClient_requestCompletedWithJobID(string response) { }
void iodClient_requestCompletedWithContent(string response) { }
void iodClient_onErrorOccurred(string errorMessage) { }

```

### 3.2. [GetRequest\(ref Dictionary<String, Object> Params, String iodApp, REQ\\_MODE mode\)](#)

#### Description:

Sends a GET request to an IDOL OnDemand API.

#### Parameters:

**Params:** a Dictionary object containing key/value pair parameters to be sent to an IDOL OnDemand API, where the keys are the parameters of an IDOL OnDemand API. E.g.

```

var Params = new Dictionary<String, Object>
{
    {"url", "http://www.cnn.com"}
};

```

#### Note:

In the case of a parameter type is an array<>, the key must be defined as “arrays” and the value must be a Dictionary<String,String> object with the key is the parameter name and the values separated by commas “,”. E.g.:

```

var entity_array = new Dictionary<String, String>
{
    {"entity_type", "people_eng,places_eng"}
}

```

**iodApp:** a string to identify an IDOL OnDemand API. E.g. “extractentities”. Current supported apps are listed in the IODApps class.

**mode** [REQ\_MODE.SYNC | REQ\_MODE.ASYNC]: specifies API call as Asynchronous or Synchronous.

Return: void.

#### Example code:

```

String iodApp = IODApps.ENTITY_EXTRACTION;
var arrays = new Dictionary<String, String>
{

```

```

        {"entity_type", "people_eng,places_eng"}
    }
    var Params = new Dictionary<String, Object>
    {
        {"url", "http://www.cnn.com"},
        {"arrays", arrays}
    };

    iodClient.GetRequest(ref Params, iodApp,
    IODClient.REQ_MODE.SYNC);

```

#### Response:

- If the mode is "ASYNC", response will be returned via the requestCompletedWithJobID(String response) callback function.
- If the mode is "SYNC", response will be returned via the requestCompletedWithContent(String response) callback function.
- If there is an error occurred, the error message will be sent via the onErrorOccurred(String errorMessage) callback function.

### 3.3. [PostRequest\(ref Dictionary<String, Object> Params, String iodApp, REQ\\_MODE mode\)](#)

#### Description:

Sends a POST request to an IDOL OnDemand API.

#### Parameters:

**Params:** a Dictionary object containing key/value pair parameters to be sent to an IDOL OnDemand API, where the keys are the parameters of an IDOL OnDemand API.

#### Notes:

1. In the case of the "file" parameter, the value must be a StorageFile object.
2. In the case of a parameter type is an array<>, the key must be defined as "arrays" and the value must be a Dictionary<String,String> object with the key is the parameter name and the values separated by commas ",". E.g.:

```

var entity_array = new Dictionary<String, String>
{
    {"entity_type", "people_eng,places_eng"}
};

```

**iodApp**: a string to identify an IDOL OnDemand API. E.g. "ocrdocument". Current supported apps are listed in the IODApps class.

**mode** [REQ\_MODE.SYNC | REQ\_MODE.ASYNC]: specifies API call as Asynchronous or Synchronous.

Return: void.

Example code:

```
String iodApp = IODApps.OCR_DOCUMENT;

StorageFile file = await
StorageFile.GetFileFromPathAsync("c:\\image.jpg");

var Params = new Dictionary<String, Object>
{
    {"file", file},
    {"mode", "document_photo"}
};

iodClient.PostRequest(ref Params, iodApp,
IODClient.REQ_MODE.ASYNC);
```

Response:

- If the mode is "ASYNC", response will be returned via the requestCompletedWithJobID(String response) callback function.
- If the mode is "SYNC", response will be returned via the requestCompletedWithContent(String response) callback function.
- If there is an error occurred, the error message will be sent via the onErrorOccurred(String errorMessage) callback function.

### 3.4. [GetJobResult\(String jobId\)](#)

Description:

Sends a request to IDOL OnDemand to retrieve the content identified by the jobId.

Parameter:

**jobID**: the job ID returned from an IDOL OnDemand API upon an asynchronous call.

Return: void.

Response:

Response will be returned via the requestCompletedWithContent(String response)

Example code:

```
void iodClient_requestCompletedWithJobID(string response)
{
    JsonValue root;
    JsonObject jsonObject;
    if (JsonValue.TryParse(response, out root))
    {
        jsonObject = root.GetObject();
        string jobId = jsonObject.GetNamedString("jobID");
        iodClient.GetJobResult(jobId);
    }
}
```