**Aspose HTML Cloud SDK**

This repository contains Aspose.HTML Cloud SDK source code. This SDK allows you to work with Aspose.HTML

Cloud REST APIs in your applications quickly and easily.

See

API Reference

for full API specification.

**How to use the SDK?**

The complete source code is available in this repository folder, you can either directly use it in your project.

**Requirements**

•

mac OS X 10.12.6

•

XCode 9.2

•

Swift 5.1 and later

•

Alamofire 4.9.1 and later

**Prerequisites**

To use Aspose HTML for Cloud SDK you need to register an account with

Aspose Cloud

and lookup/create App Key

and SID at

Cloud Dashboard

. There is free quota available. For more details, see

Aspose Cloud Pricing

.

**Installation & Usage**

**Getting Started**

Before create configuration, then use HtmlAPI or StorageAPI (see tests)

**Possible conversions:**

•

HTML -> PDF, XPS, DOCX, MD, MHTML, JPEG, BMP, PNG, GIF, TIFF

•

EPUB -> PDF, XPS, DOCX, JPEG, BMP, PNG, GIF, TIFF

•

MD -> PDF, XPS, DOCX, HTML, MHTML, JPEG, BMP, PNG, GIF, TIFF

•

MHTML -> PDF, XPS, DOCX, JPEG, BMP, PNG, GIF, TIFF

Example:

import Alamofire

import Foundation

import XCTest

import AsposeHtmlCloud

static let fm = FileManager.default

let resourceDir = fm.homeDirectoryForCurrentUser.appendingPathComponent("Documents/Aspose.HTML.Cloud.SDK.Swift/Tests/AsposeHtmlCloudTests/Resources")

let resultDir = fm.homeDirectoryForCurrentUser.appendingPathComponent("Documents/Aspose.HTML.Cloud.SDK.Swift/Tests/AsposeHtmlCloudTests/TestResult")

func url(forResource fileName: String) -> URL {

return resourceDir.appendingPathComponent(fileName)

}

func fileExist(name: String) -> Bool {

return FileManager.default.fileExists(atPath: name)

}

ClientAPI.setConfig(

basePath: "https://api.aspose.cloud/v4.0",

authPath: "https://api.aspose.cloud/connect/token",

apiKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",

appSID: "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",

debugging: true

)

let fileName = "test1.html"

let src = url(forResource: fileName).absoluteString

let expectation = self.expectation(description: "testConvert to \(format)")

let dst = resultDir.appendingPathComponent("LocToLocDoc.\(format)").absoluteString

HtmlAPI.convertLocalToLocal(src: src, dst: dst, options: nil) { (data, error) in

guard error == nil else {

XCTFail("Error get convert html to \(format)). Error=\(error!.localizedDescription)")

return

}

let resultPath = data!.file!

XCTAssertTrue(fileExist(name: resultPath))

expectation.fulfill()

}

self.waitForExpectations(timeout: testTimeout, handler: nil)

**Documentation for API Endpoints**

All URIs are relative to

*https://api.aspose.cloud/v4.0*

**Class**

**Method**

**Description**

*HtmlAPI*

**convertLocalToLocal**

Convert the HTML or EPUB document from local disk to a local disk.

*HtmlAPI*

**convertLocalToStorage**

Convert the HTML or EPUB document from local disk to user's storage.

*HtmlAPI*

**convertStorageToLocal**

Convert the HTML or EPUB document from user's storage to local disk.

*HtmlAPI*

**convertStorageToStorage**

Convert the HTML or EPUB document from user's storage to user's storage.

*HtmlAPI*

**convertUrlToLocal**

Convert the HTML document by URL to local disk.

*HtmlAPI*

**convertUrlToStorage**

Convert the HTML document by URL to user's storage.

*HtmlAPI*

**convert**

Convert the HTML, EPUB or URL to the specified format.

**Documentation For Authorization**

**oauth**

•

**Type**

: OAuth2

•

**Flow**

: application

•

**Authorization URL**

: "https://api.aspose.cloud/connect/token"

•

**Scopes**

: N/A

**Examples**

Test uses StorageApi for upload(download) file to(from) remote storage.

**Documentation for Storage API Endpoints**

All URIs are relative to

*https://api.aspose.cloud/v4.0*

**Class**

**Method**

**Description**

*StorageAPI*

**deleteFile**

Delete file

*StorageAPI*

**downloadFile**

Download file

*StorageAPI*

**uploadFile**

Upload file

*StorageAPI*

**createFolder**

Create the folder

*StorageAPI*

**deleteFolder**

Delete folder

*StorageAPI*

**getFilesList**

Get all files and folders within a folder

*StorageAPI*

**getDiscUsage**

Get disc usage

*StorageAPI*

**objectExists**

Check if file or folder exists

*StorageAPI*

**storageExists**

Check if storage exists

**Documentation for Models**

•

DiscUsage

•

ErrorDetails

•

FilesList

•

FilesUploadResult

•

FileVersion

•

FileVersions

•

ModelError

•

ObjectExist

•

StorageExist

•

StorageFile

Tests

contain various examples of using the Aspose.HTML and Aspose.Storage SDK.