



# 目次

---

- 1. 改訂情報
- 2. はじめに
  - 2.1. 本書の目的
  - 2.2. 対象読者
  - 2.3. サンプルコードについて
  - 2.4. 本書の構成
- 3. Office 365 連携 プログラミング概要
  - 3.1. Office 365 連携 の流れ
- 4. SharePoint Online のファイラを作成する (JavaEE開発モデル)
  - 4.1. SharePoint Online 上にディレクトリを新規作成する
  - 4.2. SharePoint Online 上にファイルをアップロードする
  - 4.3. SharePoint Online 上のファイルをダウンロードする
  - 4.4. SharePoint Online 上のファイルの詳細情報を表示する
  - 4.5. SharePoint Online 上のファイルを削除する
  - 4.6. SharePoint Online 上のドキュメントをiframeで表示する
  - 4.7. SharePoint Online のファイラを作成する
- 5. SharePoint Online のファイラを作成する (スクリプト開発モデル)
  - 5.1. SharePoint Online 上にディレクトリを新規作成する
  - 5.2. SharePoint Online 上にファイルをアップロードする
  - 5.3. SharePoint Online 上のファイルをダウンロードする
  - 5.4. SharePoint Online 上のファイルの詳細情報を表示する
  - 5.5. SharePoint Online 上のファイルを削除する
  - 5.6. SharePoint Online 上のドキュメントをiframeで表示する
  - 5.7. SharePoint Online のファイラを作成する
  - 5.8. ルーティングテーブルを作成する

変更年月日	変更内容
2015-08-01	初版
2015-12-01	第2版 下記を追加・変更しました <ul style="list-style-type: none"><li>▪ 「<a href="#">SharePoint Online のファイラを作成する (スクリプト開発モデル)</a>」を追加</li></ul>
2017-08-01	第3版 下記を変更しました。 <ul style="list-style-type: none"><li>▪ 「<a href="#">SharePoint Online のファイラを作成する (JavaEE開発モデル)</a>」にOffice 365 連携の低レベルAPIを新たに提供されたOneDrive APIに変更</li></ul>

---

## 本書の目的

---

本書では intra-mart Accel Platform における Office 365 連携機能 を用いたプログラミング方法について説明します。

## 対象読者

---

次の開発者を対象としています。

- intra-mart Accel Platform で Office 365 上のリソースを利用したアプリケーションを作成したい開発者

## サンプルコードについて

---

本書に掲載されているサンプルコードは可読性を重視しており、性能面や保守性といった観点において必ずしも適切な実装ではありません。

サンプルコードを参考に開発する場合は、上記について十分に注意してください。

なお、ドキュメントの流れに合わせてサンプルを実装し、実際に動作確認を行う場合の前提条件は以下の通りです。

- [Office 365 連携 セットアップガイド](#) を参考に Office 365 連携 の環境を作成してください。

## 本書の構成

---

本書は以下のような構成です。

- [Office 365 連携 プログラミング概要](#)
- [SharePoint Online のファイラを作成する \(JavaEE 開発モデル\)](#)
- [SharePoint Online のファイラを作成する \(スクリプト開発モデル\)](#)

## Office 365 連携 の流れ

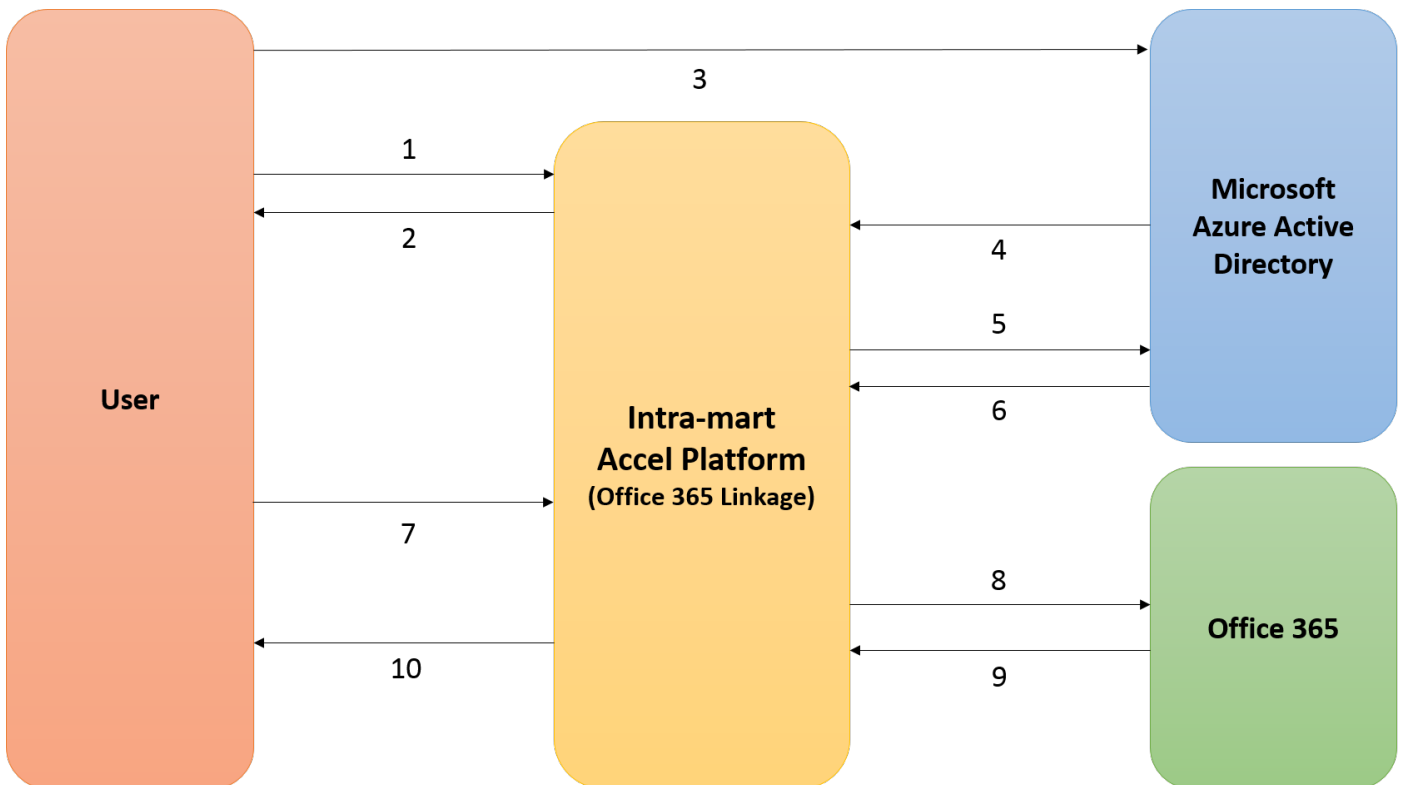
Office 365 連携機能のAPIは、Office 365 のWebAPIをラップする形で提供しています。

Office 365 のWebAPIの実行に必要なアクセストークンは、intra-mart Accel Platform（Office 365 連携）が OAuth2.0 の仕様によって、自動的に取得・更新します。

これにより開発者は OAuth2.0 関連の実装が不要となり、Office 365 連携機能の APIを利用したビジネスロジックの実装に集中できます。

開発者は以下の図の **7** と **10** に 必要な処理や画面を実装することで、Office 365 連携機能を利用したプログラムを作成できます。

### Office 365 連携のフロー図



1. ユーザが「個人設定」 - 「外部連携アプリケーション」画面で Office 365 連携 のプロバイダを「許可」します。
2. intra-mart Accel Platform（Office 365 連携）は ユーザを Microsoft Azure Active Directory（以降、Microsoft Azure AD とする）にリダイレクトさせます。
3. ユーザは Microsoft Azure AD にサインインを行い認証し、必要な場合は許諾を行います。
4. Microsoft Azure AD は intra-mart Accel Platform（Office 365 連携）のリダイレクトURI に認可コードを返します。
5. intra-mart Accel Platform（Office 365 連携API）は、Microsoft Azure AD に認可コードを提示し、アクセストークンを要求します。
6. Microsoft Azure AD は、アクセストークンとリフレッシュトークンを返却します。（アクセストークンの有効期限が切れた場合、リフレッシュトークンを利用し新しいアクセストークンに更新します。）
7. ユーザが SharePoint Online のファイル操作などの Office 365 連携機能のAPIを利用した画面をリクエストします。
8. intra-mart Accel Platform（Office 365 連携API）は、アクセストークンを利用し Office 365 の Web API を実行します。
9. Office 365 の Web API はリクエストされたデータを返します。
10. intra-mart Accel Platform（Office 365 連携）は、データを利用しユーザのリクエストに応答します。



#### コラム

Microsoft Azure AD のアクセストークンの取得フローについての詳細は以下のページを参照してください。

- <https://msdn.microsoft.com/en-US/library/azure/dn645542.aspx> (English)

Office 365 連携機能 では、Office 365 の SharePoint Online のドキュメントにアクセスするための以下の API を提供します。

- Office 365 のOneDrive REST APIをラップした低レベルAPI
  - jp.co.intra\_mart.foundation.office365.api.v2\_0.onedrive.operation.OneDriveOperation
- intra-mart Accel Platform の Storageインタフェースを継承した高レベルAPI
  - jp.co.intra\_mart.foundation.office365.service.storage.SharePointStorage

ここでは、Office 365 連携機能 の SharePointStorage APIを利用して実際にプログラムを作成する過程を説明します。

本項では、例として JavaEE開発モデル を利用しています。

また、サンプルコードは可読性を重視しているため、JSP内で全ての処理を行っています。

本項に記載のサンプルを組み合わせると 以下のような SharePoint Online のファイラが完成します。

## SharePoint Online のファイラの完成イメージ

The screenshot displays the SharePoint Online file explorer interface. At the top, there is a navigation bar with the 'intra-mart' logo and several menu items: 'Top', 'サンプル', and 'サイトマップ'. On the right side of the navigation bar, there is a search icon, a user profile dropdown (showing '円山益男'), a help icon, and a profile picture. The main content area is titled 'ファイル一覧' (File List) and features a search bar with a '移動' (Move) button. Below the search bar is a table with the following columns: '名前' (Name), '詳細' (Details), '削除' (Delete), and 'iframe表示' (iframe display). The table lists a directory '親ディレクトリへ移動' (Move to parent directory) and three files: 'sample', 'sample.docx', 'sample.xlsx', and 'sample.pptx'. Each file row includes an information icon, a delete icon, and an iframe display icon. Below the table, there is a section titled 'ファイルアップロード' (File Upload) with a '+ ファイル追加...' (Add file...) button, a '開始' (Start) button, and a '中断' (Cancel) button. At the bottom, there is a section titled 'ディレクトリ新規作成' (Create new directory) with a text input field and a '作成' (Create) button.

### 項目

- SharePoint Online 上にディレクトリを新規作成する
- SharePoint Online 上にファイルをアップロードする
- SharePoint Online 上のファイルをダウンロードする
- SharePoint Online 上のファイルの詳細情報を表示する
- SharePoint Online 上のファイルを削除する
- SharePoint Online 上のドキュメントをiframeで表示する
- SharePoint Online のファイラを作成する

**注意**

Office 365 連携 のAPIの詳細については以下を参照してください。

- 「[APIドキュメント - SharePointStorage](#)」
- 「[APIドキュメント - OneDriveOperation](#)」

Office 365 連携 のタグライブラリの詳細については以下を参照してください。

- 「[タグライブラリ - im-office365:excel](#)」
- 「[タグライブラリ - im-office365:powerPoint](#)」
- 「[タグライブラリ - im-office365:word](#)」

## SharePoint Online 上にディレクトリを新規作成する

SharePoint Online 上にディレクトリを新規作成する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのディレクトリパス、ディレクトリの名称 を元に新規ディレクトリを作成します。

### 実装例

SharePointStorage#makeDirectories() を利用します。

```
%CONTEXT_PATH%/sample/office365/filer/create_dir.jsp
```





```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
3
4 <%@ page import="java.io.IOException" %>
5
6 <%@ page import="jp.co.intra_mart.common.platform.log.Logger" %>
7 <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
8 <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
9 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage"%>
10
11 <%
12 String userCd = Contexts.get(AccountContext.class).getUserCd();
13 String providerId = request.getParameter("providerId");
14 String currentDirPath =
15     request.getParameter("targetDirPath") != null ? request.getParameter("targetDirPath") : "/";
16 String createDirName =
17     request.getParameter("createDirName") != null ? request.getParameter("createDirName") : null;
18 String dirPath = currentDirPath + createDirName;
19
20 String successMsg = null;
21 String errorMsg = null;
22
23 if (createDirName == null) {
24     errorMsg = "ディレクトリ名が指定されていません。";
25 } else {
26     SharePointStorage storage = new SharePointStorage(userCd, providerId, dirPath);
27     try {
28         //ディレクトリの作成
29         boolean result = storage.makeDirectories();
30         if (result) {
31             successMsg = "ディレクトリを作成しました。<br/><br/>" + dirPath;
32         } else {
33             errorMsg = "ディレクトリを作成できませんでした。<br/><br/>" + dirPath;
34         }
35     } catch (IOException e) {
36         errorMsg = buildErrorMessage("ディレクトリ作成時にエラーが発生しました。", e);
37     }
38 }
39 %>
40 <imui:head>
41 <title>Office365 - SharePointStorage API サンプル - ディレクトリ作成</title>
42
43 <script type="text/javascript">
44 $(document).ready(function() {
45     if(<%=successMsg != null%>){
46         imuiShowSuccessMessage('<%=successMsg%>');
47     }
48
49     if(<%=errorMsg != null%>){
50         imuiShowErrorMessage('<%=errorMsg%>');
51     }
52 });
53
54 </script>
55 </imui:head>
56
57 <div class="imui-title">
58     <h1><%=dirPath%></h1>
59 </div>
60
61 <div class="imui-toolbar-wrap">
62     <div class="imui-toolbar-inner">
63         <ul class="imui-list-toolbar">
64             <li>
65                 <a href="javascript:$('#backToIndex').submit();" class="imui-toolbar-icon" title="一覧に戻る">
66                 <span class="im-ui-icon-common-16-back"></span>一覧に戻る

```

```

67     </a>
68   </li>
69 </ul>
70 </div>
71 </div>
72
73 <form method="post" action="sample/office365/filer/index.jsp" id="backToIndex">
74   <input type="hidden" name="targetDirPath" value="<%=currentDirPath%>" />
75 </form>
76 <%!
77   private static Logger logger = Logger.getLogger("sample.office365.filer.create_dir");
78   private String buildErrorMessage(String errorMessage, Exception errorDetail) {
79     logger.error(errorDetail.getMessage(), errorDetail);
80     return errorMessage + "<br/><br/>" + errorDetail.getMessage();
81   }
82   %>

```

### 注意

Microsoft社が提供する Office 365 のWebAPIの動作仕様に起因し以下の制限があります。  
 SharePointStorage#makeDirectories() および OneDriveOperation#createFolderByPath() を利用しディレクトリを作成する場合、ルートディレクトリ直下に存在するディレクトリ名と同名のディレクトリを任意のディレクトリ配下に作成することはできません。

## SharePoint Online 上にファイルをアップロードする

SharePoint Online 上にファイルをアップロードする実装のサンプルです。

マルチパートフォームデータとして受け取った プロバイダID、ターゲットのディレクトリパス、ファイルを元にファイルをアップロードします。

### 実装例

SharePointStorage#create() を利用します。

```
%CONTEXT_PATH%/sample/office365/filer/upload.jsp
```

```

1  <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2
3  <%@ page import="java.io.InputStream" %>
4  <%@ page import="java.io.OutputStream" %>
5
6  <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
7  <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
8  <%@ page import="jp.co.intra_mart.common.aid.javaee.http.MultipartFormData" %>
9  <%@ page import="jp.co.intra_mart.common.aid.javaee.http.MultipartFormData.Entity" %>
10 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage"%>
11 <%@ page import="jp.co.intra_mart.common.aid.jdk.java.io.IOUtil" %>
12
13 <%
14     String userCd = Contexts.get(AccountContext.class).getUserCd();
15
16     //Web Application Server によってはレスポンスにテーマが含まれてしまうため notheme を指定
17     request.setAttribute("imui-theme-builder-module", "notheme");
18     response.setContentType("json");
19
20     MultipartFormData data = new MultipartFormData(request);
21     String providerId = data.getEntity("providerId").getContent();
22     String targetDirPath = new String(data.getEntity("targetDirPath").getBytes(), "utf-8");
23     Entity localFileEntity = data.getEntity("local_file");
24     String fileName = localFileEntity.getFileName();
25
26     SharePointStorage storage = new SharePointStorage(userCd, providerId, targetDirPath + "/" + fileName);
27
28     try (OutputStream outputStream = storage.create();
29         InputStream inputStream = localFileEntity.getInputStream()
30     ){
31         IOUtil.transfer(inputStream, outputStream);
32     }
33
34     response.setContentType("json");
35 %>
36 [{
37     "name": "<%= localFileEntity.getFileName() %>",
38     "size": <%= localFileEntity.getContentLength() %>,
39     "error": ""
40 }]

```

## SharePoint Online 上のファイルをダウンロードする

SharePoint Online 上のファイルをダウンロードする実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパス、ファイルの名称を元にファイルをダウンロードします。

### 実装例

SharePointStorage#open() を利用します。

```
%CONTEXT_PATH%/sample/office365/filer/download.jsp
```

```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ page import="java.io.InputStream" %>
3 <%@ page import="java.io.OutputStream" %>
4 <%@ page import="javax.activation.FileTypeMap" %>
5
6 <%@ page import="jp.co.intra_mart.common.aid.jdk.java.io.IOUtil" %>
7 <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
8 <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
9 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage" %>
10 <%@ page import="jp.co.intra_mart.foundation.http.ResponseUtil" %>
11
12 <%
13     String userCd = Contexts.get(AccountContext.class).getUserCd();
14     String providerId = request.getParameter("providerId");
15     String targetFilePath = request.getParameter("targetFilePath");
16     String targetFileName = request.getParameter("targetFileName");
17
18     response.resetBuffer();
19
20     response.setCharacterEncoding(null);
21     String mimeType = FileTypeMap.getDefaultFileTypeMap().getContentType(targetFileName);
22     response.setContentType(mimeType);
23
24     String encodedFileName = ResponseUtil.encodeFileName(request, "UTF-8", targetFileName);
25     response.setHeader("Content-Disposition", "attachment; " + encodedFileName);
26
27     SharePointStorage sharePointStorage = new SharePointStorage(userCd, providerId, targetFilePath);
28     try (InputStream inputStream = sharePointStorage.open();
29         OutputStream outputStream = response.getOutputStream();
30     ) {
31         IOUtil.transfer(inputStream, outputStream);
32     }
33 %>

```

## SharePoint Online 上のファイルの詳細情報を表示する

SharePoint Online 上のファイルの詳細情報を表示する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスをファイルの詳細情報を表示します。

### 実装例

```
%CONTEXT_PATH%/sample/office365/filer/show_detail.jsp
```



```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
3 <%@ taglib prefix="imtag" uri="http://www.intra-mart.co.jp/taglib/core/standard" %>
4 <%@ taglib prefix="i18n" uri="http://www.intra-mart.co.jp/taglib/im-i18n" %>
5
6 <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
7 <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
8 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage"%>
9
10 <%
11 String userCd = Contexts.get(AccountContext.class).getUserCd();
12 String providerId = request.getParameter("providerId");
13 String targetPathForShowDetail = request.getParameter("targetPathForShowDetail");
14
15 String errorMsg = "";
16 if(targetPathForShowDetail == null){
17     errorMsg = "ファイル、または、ディレクトリが指定されていません。";
18 }
19
20 SharePointStorage storage = new SharePointStorage(userCd, providerId, targetPathForShowDetail);
21 %>
22 <imui:head>
23 <title>Office365 - SharePointStorage API サンプル - 詳細表示</title>
24
25 <script type="text/javascript">
26 $(document).ready(function() {
27     if(<%= errorMsg != null %>){
28         imuiShowErrorMessage('<%= errorMsg %>');
29     }
30 });
31 </script>
32 </imui:head>
33
34 <div class="imui-title">
35     <imtag:condition validity="<%= Boolean.toString(errorMsg.isEmpty()) %>">
36         <h1><%=storage.getCanonicalPath()%></h1>
37     </imtag:condition>
38 </div>
39
40 <div class="imui-form-container-narrow">
41     <div class="imui-chapter-title">
42         <h2>詳細情報</h2>
43     </div>
44     <imtag:condition validity="<%= Boolean.toString(errorMsg.isEmpty()) %>">
45         <table class="imui-table">
46             <tr>
47                 <th class="wd-335px">正規化したパス</th>
48                 <td><%=storage.getCanonicalPath()%></td>
49             </tr>
50             <tr>
51                 <th>コンテンツをブラウザ上で確認可能なWeb URL</th>
52                 <td>
53                     <a href="<%=storage.getWebUrl()%>"><%=java.net.URLDecoder.decode(storage.getWebUrl(), "UTF-
54 8")%></a>
55                 </td>
56             </tr>
57             <tr>
58                 <th>作成時刻</th>
59                 <td><i18n:dateTime value="<%=new java.util.Date(storage.created())%>" pattern="yyyy/MM/dd
60 hh:mm:ss"/></td>
61             </tr>
62             <tr>
63                 <th>作成ユーザID</th>
64                 <td><%=storage.createdUserId()%></td>
65             </tr>
66             <tr>

```

```

67     <th>作成ユーザ名</th>
68     <td><%=storage.createdUserName()%></td>
69 </tr>
70 <tr>
71     <th>最終更新時刻</th>
72     <td><i18n:dateTime value="<%=new java.util.Date(storage.lastModified())%"
73 pattern="yyyy/MM/dd hh:mm:ss"/></td>
74 </tr>
75 <tr>
76     <th>最終更新ユーザID</th>
77     <td><%=storage.lastModifiedUserId()%></td>
78 </tr>
79 <tr>
80     <th>最終更新ユーザ名</th>
81     <td><%=storage.lastModifiedUserName()%></td>
82 </tr>
83 <tr>
84     <th>ファイルサイズ</th>
85     <td><%=storage.length()%>バイト</td>
86 </tr>
</table>
</imtag:condition>
</div>

```

## SharePoint Online 上のファイルを削除する

SharePoint Online 上のファイルを削除する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスを元にファイルを削除します。

### 実装例

SharePointStorage#remove() を利用します。

```
%CONTEXT_PATH%/sample/office365/filer/remove.jsp
```





```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2 <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
3
4 <%@ page import="java.io.IOException" %>
5
6 <%@ page import="jp.co.intra_mart.common.platform.log.Logger" %>
7 <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
8 <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
9 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage"%>
10
11 <%
12 String userCd = Contexts.get(AccountContext.class).getUserCd();
13 String providerId = request.getParameter("providerId");
14 String targetPath = request.getParameter("targetPathForRemove");
15 String currentDirPath =
16     request.getParameter("targetDirPath") != null ? request.getParameter("targetDirPath") : "/";
17
18 String successMsg = null;
19 String errorMsg = null;
20
21 if (targetPath == null) {
22     errorMsg = "ファイル、または、ディレクトリが指定されていません。";
23 } else {
24     SharePointStorage storage = new SharePointStorage(userCd, providerId, targetPath);
25     try {
26         //ファイルまたはディレクトリの削除
27         boolean result = storage.remove(true);
28         if (result) {
29             successMsg = "「" + targetPath + "」を削除しました。";
30         } else {
31             errorMsg = "「" + targetPath + "」を削除できませんでした。";
32         }
33     } catch (IOException e) {
34         errorMsg = buildErrorMessage("削除時にエラーが発生しました。", e);
35     }
36 }
37 %>
38 <imui:head>
39 <title>Office365 - SharePointStorage API サンプル - 削除</title>
40
41 <script type="text/javascript">
42 $(document).ready(function() {
43     if(<%=successMsg != null%>){
44         imuiShowSuccessMessage('<%=successMsg%>');
45     }
46     if(<%=errorMsg != null%>){
47         imuiShowErrorMessage('<%=errorMsg%>');
48     }
49 });
50 </script>
51 </imui:head>
52
53 <div class="imui-title">
54 <h1><%=targetPath%></h1>
55 </div>
56
57 <div class="imui-toolbar-wrap">
58 <div class="imui-toolbar-inner">
59 <ul class="imui-list-toolbar">
60 <li>
61 <a href="javascript:$('#backToIndex').submit();" class="imui-toolbar-icon" title="一覧に戻る">
62 <span class="im-ui-icon-common-16-back"></span>一覧に戻る
63 </a>
64 </li>
65 </ul>
66 </div>

```

```
67 </div>
68
69 <form method="post" action="sample/office365/filer/index.jsp" id="backToIndex">
70   <input type="hidden" name="targetDirPath" value="<%=currentDirPath%>"/>
71 </form>
72
73 <%!
74   private static Logger logger = Logger.getLogger("sample.office365.filer.remove");
75   private String buildErrorMessage(String errorMessage, Exception errorDetail) {
76     logger.error(errorDetail.getMessage(), errorDetail);
77     return errorMessage + "<br/><br/>" + errorDetail.getMessage();
78   }
79 %>
```

## SharePoint Online 上のドキュメントをiframeで表示する

---

SharePoint Online 上のドキュメントをiframeで表示する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスを元にドキュメントをiframeで表示します。

### 実装例

im-office365 タグを利用します。

```
%CONTEXT_PATH%/sample/office365/filer/show_iframe.jsp
```



```

1 <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
2 <%@ taglib prefix="imtag" uri="http://www.intra-mart.co.jp/taglib/core/standard" %>
3 <%@ taglib prefix="im-office365" uri="http://www.intra-mart.co.jp/taglib/im-office365"%>
4
5 <%
6 String width = "640";
7 String height = "480";
8
9 String providerId = request.getParameter("providerId");
10 String targetPath = request.getParameter("targetPathForShowIframe");
11
12 String errorMsg = null;
13 if (targetPath == null) {
14     errorMsg = "ファイルが指定されていません。";
15 }
16 %>
17 <imui:head>
18 <title>Office365 - iframe 表示</title>
19
20 <script type="text/javascript">
21 $(document).ready(function() {
22     if(<%=errorMsg != null%>){
23         imuiShowErrorMessage('<%=errorMsg%>');
24     }
25 });
26 </script>
27 </imui:head>
28
29 <div class="imui-title">
30     <h1><%=targetPath%></h1>
31 </div>
32
33 <div id="imui-container-inner">
34     <imtag:condition validity="<%=isWord(targetPath)%>">
35         <im-office365:word providerId="<%=providerId%>" path="<%=targetPath%>"
36             width="<%=width%>" height="<%=height%>" />
37     </imtag:condition>
38     <imtag:condition validity="<%=isExcel(targetPath)%>">
39         <im-office365:excel providerId="<%=providerId%>" path="<%=targetPath%>"
40             width="<%=width%>" height="<%=height%>" />
41     </imtag:condition>
42     <imtag:condition validity="<%=isPowerPoint(targetPath)%>">
43         <im-office365:powerPoint providerId="<%=providerId%>" path="<%=targetPath%>"
44             width="<%=width%>" height="<%=height%>" />
45     </imtag:condition>
46 </div>
47
48 <%!
49 private String isWord(final String fileName) {
50     return ".docx".equals(getExtension(fileName)) ? "true" : "false";
51 }
52
53 private String isExcel(final String fileName) {
54     return ".xlsx".equals(getExtension(fileName)) ? "true" : "false";
55 }
56
57 private String isPowerPoint(final String fileName) {
58     return ".pptx".equals(getExtension(fileName)) ? "true" : "false";
59 }
60
61 private String getExtension(String fileName) {
62     final int indexOfLastDot = fileName.lastIndexOf(".");
63     if (indexOfLastDot == -1) {
64         return null;
65     }
66     return fileName.substring(indexOfLastDot);

```

```
67 }  
68 %>
```

### 注意

ドキュメントを表示するにはユーザが Office 365 - SharePoint Online にログインする必要があります。

## SharePoint Online のファイラを作成する

前述の実装例を組み合わせ、SharePoint Online のファイラを作成します。

providerId は「[Office 365 連携 セットアップガイド](#)」で環境構築の際に設定した <im\_office365\_files\_api.xml> のプロバイダIDを指定してください。

### 実装例

SharePointStorage#directoriesStorages()、SharePointStorage#filesStorages() を利用します。

```
%CONTEXT_PATH%/sample/office365/filer/index.jsp
```



```

1  <%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
2  <%@ taglib prefix="imui" uri="http://www.intra-mart.co.jp/taglib/imui" %>
3  <%@ taglib prefix="imtag" uri="http://www.intra-mart.co.jp/taglib/core/standard" %>
4
5  <%@ page import="java.io.InputStream" %>
6  <%@ page import="java.io.IOException" %>
7  <%@ page import="java.io.OutputStream" %>
8  <%@ page import="java.util.ArrayList" %>
9  <%@ page import="java.util.Collection" %>
10
11 <%@ page import="jp.co.intra_mart.common.platform.log.Logger" %>
12 <%@ page import="jp.co.intra_mart.foundation.context.Contexts" %>
13 <%@ page import="jp.co.intra_mart.foundation.context.model.AccountContext" %>
14 <%@ page import="jp.co.intra_mart.foundation.oauth.client.service.OAuthPreconditionException" %>
15
16 <%@ page import="jp.co.intra_mart.foundation.office365.service.storage.SharePointStorage"%>
17
18 <%
19     String providerId = "Please_input_your_ID";
20     String userCd = Contexts.get(AccountContext.class).getUserCd();
21     String currentDirPath =
22         request.getParameter("targetDirPath") != null ?
23         addSlashToTheEnd(request.getParameter("targetDirPath")) : "/";
24
25     String errorMsg = null;
26     SharePointStorage storage = new SharePointStorage(userCd, providerId, currentDirPath);
27
28     // 一覧取得
29     Collection<SharePointStorage> directoriesStorages = new ArrayList<SharePointStorage>();
30     Collection<SharePointStorage> filesStorages = new ArrayList<SharePointStorage>();
31     try {
32         directoriesStorages = storage.directoriesStorages(false);
33         filesStorages = storage.filesStorages(false);
34     } catch (IOException e) {
35         errorMsg = buildErrorMessage("ディレクトリ、または、ファイル一覧の取得時にエラーが発生しました。", e);
36     }
37
38     // 親ディレクトリの存在チェック
39     String parentDir = storage.getParent();
40     String existsParentDir = parentDir != null ? "true" : "false";
41 %>
42 <imui:head>
43 <title>Office365 - SharePointStorage API サンプル</title>
44 <script src="csjs/im_window.js"></script>
45 <script type="text/javascript">
46     function moveDir(targetDirPath){
47         $('#targetDirPath').val('<%= currentDirPath %>' + targetDirPath);
48         $('#moveDirForm').submit();
49     }
50
51     function downloadFile(targetFilePath){
52         $('#targetFilePath').val('<%= currentDirPath %>' + targetFilePath);
53         $('#targetFileName').val(targetFilePath);
54         $('#downloadFileForm').submit();
55     }
56
57     function removeFileOrDir(targetPath){
58         $('#targetPathForRemove').val('<%= currentDirPath %>' + targetPath);
59         $('#removeFileOrDirForm').submit();
60     }
61
62     function showDetail(targetPath){
63         openNewWindow("office365_show_detail").focus();
64         $('#targetPathForShowDetail').val('<%= currentDirPath %>' + targetPath);
65         $('#showDetailForm').submit();
66     }

```



```

67
68 function showIframe(targetPath){
69     openNewWindow("office365_show_iframe").focus();
70     $('#targetPathForShowIframe').val('<%= currentDirPath %>' + targetPath);
71     $('#showIframeForm').submit();
72 }
73
74 $(document).ready(function() {
75     if(<%= errorMsg != null %>){
76         imuiShowErrorMessage('<%= errorMsg %>');
77     }
78
79     $('#toParent').click(function() {
80         $('#targetDirPath').val('/') + '<%= parentDir %>');
81         $('#moveDirForm').submit();
82     });
83
84 });
85 </script>
86 </imui:head>
87 <div class="imui-container">
88     <div class="imui-title">
89         <h1><%= currentDirPath %></h1>
90     </div>
91
92     <div class="imui-form-container-narrow">
93         <div class="imui-chapter-title">
94             <h2>ファイル一覧</h2>
95         </div>
96         <form method="post" action="sample/office365/filer/index.jsp">
97             <input type="text" name="targetDirPath" value="<%= currentDirPath %>" size="70" />
98             <input type="submit" value="移動" class="imui-small-button" />
99         </form>
100         <br />
101         <table class="imui-table">
102             <tr>
103                 <th>名前</th>
104                 <th>詳細</th>
105                 <th>削除</th>
106                 <th>iframe表示</th>
107             </tr>
108             <imtag:condition validity="<%= existsParentDir %>">
109                 <tr>
110                     <td><a href="javascript:void(0)" id="toParent">親ディレクトリへ移動</a></td>
111                     <td></td>
112                     <td></td>
113                     <td></td>
114                 </tr>
115             </imtag:condition>
116
117             <!-- ディレクトリ一覧 -->
118             <imtag:repeat list="<%= directoriesStorages %>" type="SharePointStorage" item="dirStorage"
119 index="idx">
120                 <tr>
121                     <td>
122                         <span class="im-ui-icon-common-16-folder"></span>
123                         <a href="javascript:void(0)"
124                             onclick="moveDir('<%= dirStorage.getName() %>');"><%= dirStorage.getName() %></a>
125                     </td>
126                     <td align="center">
127                         <a href="javascript:void(0)" onclick="showDetail('<%= dirStorage.getName() %>');">
128                             <span class="im-ui-icon-common-16-information"></span>
129                         </a>
130                     </td>
131                     <td align="center">
132                         <a href="javascript:void(0)" onclick="removeFileOrDir('<%= dirStorage.getName() %>');">

```

```

133     <span class="im-ui-icon-common-16-trashbox"></span>
134   </a>
135 </td>
136 <td></td>
137 </tr>
138 </imgtag:repeat>
139
140 <!-- ファイル一覧 -->
141 <imgtag:repeat list="<%= filesStorages %>" type="SharePointStorage" item="fileStorage"
142 index="idx">
143   <tr>
144     <td>
145       <span class="im-ui-icon-common-16-document"></span>
146       <a href="javascript:void(0)"
147         onclick="downloadFile('<%= fileStorage.getName() %>');"><%= fileStorage.getName() %></a>
148     </td>
149     <td align="center">
150       <a href="javascript:void(0)" onclick="showDetail('<%= fileStorage.getName() %>');">
151         <span class="im-ui-icon-common-16-information"></span>
152       </a>
153     <td align="center">
154       <a href="javascript:void(0)" onclick="removeFileOrDir('<%= fileStorage.getName() %>');">
155         <span class="im-ui-icon-common-16-trashbox"></span>
156       </a>
157     </td>
158     <td align="center">
159       <imgtag:condition validity="<%= isWordExcelPowerPointFile(fileStorage.getName()) %>">
160         <a href="javascript:void(0)" onclick="showIframe('<%= fileStorage.getName() %>');">
161           <span class="im-ui-icon-common-16-preview"></span>
162         </a>
163       </imgtag:condition>
164     </td>
165   </tr>
166 </imgtag:repeat>
167
168 </table>
169
170 <br/>
171
172 <div class="imui-chapter-title">
173   <h2>ファイルアップロード</h2>
174 </div>
175 <form id="uploadForm" action="sample/office365/filer/upload.jsp" enctype="multipart/form-data">
176   <imui:fileUpload outerFormId="uploadForm"/>
177   <input type="hidden" name="targetDirPath" value="<%= currentDirPath %>" />
178   <input type="hidden" name="providerId" value="<%= providerId %>" />
179 </form>
180 <div class="imui-chapter-title">
181   <h2>ディレクトリ新規作成</h2>
182 </div>
183
184 <form method="POST" name="createDir" id="createDir"
185   action="sample/office365/filer/create_dir.jsp">
186   <input type="text" name="createDirName" />
187   <input type="hidden" name="targetDirPath" value="<%= currentDirPath %>" />
188   <input type="hidden" name="providerId" value="<%= providerId %>" />
189   <input type="submit" value="作成" class="imui-small-button" />
190 </form>
191
192 <form method="POST" name="moveDirForm" id="moveDirForm"
193   action="sample/office365/filer/index.jsp">
194   <input type="hidden" name="targetDirPath" id="targetDirPath" />
195   <input type="hidden" name="providerId" value="<%= providerId %>" />
196 </form>
197
198 <form method="POST" name="downloadFileForm" id="downloadFileForm"

```

```

199     action="sample/office365/filer/download.jsp">
200     <input type="hidden" name="targetFilePath" id="targetFilePath" />
201     <input type="hidden" name="targetFileName" id="targetFileName" />
202     <input type="hidden" name="providerId" value="<%= providerId %>" />
203 </form>
204
205 <form method="POST" name="removeFileOrDirForm" id="removeFileOrDirForm"
206     action="sample/office365/filer/remove.jsp">
207     <input type="hidden" name="targetPathForRemove" id="targetPathForRemove" />
208     <input type="hidden" name="targetDirPath" id="targetDirPath" value="<%=currentDirPath%>" />
209     <input type="hidden" name="providerId" value="<%= providerId %>" />
210 </form>
211
212 <form method="POST" name="showDetailForm" id="showDetailForm"
213     action="sample/office365/filer/show_detail.jsp" target="office365_show_detail">
214     <input type="hidden" name="targetPathForShowDetail" id="targetPathForShowDetail" />
215     <input type="hidden" name="targetDirPath" id="targetDirPath" value="<%=currentDirPath%>" />
216     <input type="hidden" name="providerId" value="<%= providerId %>" />
217 </form>
218
219 <form method="POST" name="showIframeForm" id="showIframeForm"
220     action="sample/office365/filer/show_iframe.jsp" target="office365_show_iframe">
221     <input type="hidden" name="targetPathForShowIframe" id="targetPathForShowIframe" />
222     <input type="hidden" name="targetDirPath" id="targetDirPath" value="<%=currentDirPath%>" />
223     <input type="hidden" name="providerId" value="<%= providerId %>" />
224 </form>
225
226 </div>
227 </div>
228
229 <%!
230 private static Logger logger = Logger.getLogger("sample.office365.filer.index");
231
232 private String buildErrorMessage(final String errorMessage, final Exception errorDetail){
233     logger.error(errorDetail.getMessage(), errorDetail);
234     if(errorDetail instanceof OAuthPreconditionException){
235         return "「外部連携アプリケーション」画面でアプリケーションを許可してください。";
236     } else {
237         return errorMessage + "<br/><br/>" + errorDetail.getMessage();
238     }
239 }
240
241 private String isWordExcelPowerPointFile(final String fileName) {
242     final int indexOfLastDot = fileName.lastIndexOf(".");
243     if (indexOfLastDot == -1) {
244         return "false";
245     }
246
247     final String extension = fileName.substring(indexOfLastDot);
248     if (".docx".equals(extension) || ".xlsx".equals(extension) || ".pptx".equals(extension)) {
249         return "true";
250     } else {
251         return "false";
252     }
253 }
254 private String addSlashToTheEnd(final String s) {
255     final int indexOfLastSlash = s.lastIndexOf("/");
256     if((indexOfLastSlash == -1) || (indexOfLastSlash == s.length() - "/" .length())){
257         return s;
258     } else {
259         return s + "/";
260     }
261 }
262 %>

```

ここでは、Office 365 連携機能 の SharePointStorage APIを利用して実際にプログラムを作成する過程を説明します。

本項では、例として スクリプト開発モデル を利用しています。

本項に記載のサンプルを組み合わせると 以下のような SharePoint Online のファイラが完成します。

## SharePoint Online のファイラの完成イメージ



### 項目

- SharePoint Online 上にディレクトリを新規作成する
- SharePoint Online 上にファイルをアップロードする
- SharePoint Online 上のファイルをダウンロードする
- SharePoint Online 上のファイルの詳細情報を表示する
- SharePoint Online 上のファイルを削除する
- SharePoint Online 上のドキュメントをiframeで表示する
- SharePoint Online のファイラを作成する
- ルーティングテーブルを作成する

### 注意

Office 365 連携 のAPIの詳細については以下を参照してください。

- 「APIドキュメント - SharePointStorage」

Office 365 連携 のタグライブラリの詳細については以下を参照してください。

- 「タグライブラリ - imOffice365Excel」
- 「タグライブラリ - imOffice365PowerPoint」
- 「タグライブラリ - imOffice365Word」

SharePoint Online 上にディレクトリを新規作成する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのディレクトリパス、ディレクトリの名称 を元に新規ディレクトリを作成します。

#### 実装例

SharePointStorage#makeDirectories() を利用します。

`%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/create_dir.js`

```
1  var $currentDirPath;
2  var $successMsg;
3  var $errorMsg;
4  var $dirPath;
5
6  function init(request) {
7    var userCd = Contexts.getAccountContext().userCd;
8    var providerId = request.providerId;
9    $currentDirPath = request.targetDirPath != null ? request.targetDirPath : '/';
10   var createDirName = request.createDirName != null ? request.createDirName : null;
11   $dirPath = $currentDirPath + createDirName;
12   if (createDirName == null) {
13     $errorMsg = 'ディレクトリ名が指定されていません。';
14   } else {
15     try{
16       var storage = new SharePointStorage(userCd, providerId, $dirPath);
17       var result = storage.makeDirectories();
18       if(result){
19         $successMsg = 'ディレクトリを作成しました。<br><br>' + $dirPath;
20       }else{
21         $errorMsg = 'ディレクトリを作成できませんでした。<br><br>' + $dirPath;
22       }
23     }catch(e){
24       $errorMsg = 'ディレクトリ作成時にエラーが発生しました。' + '<br><br>' + e.errorMessage;
25     }
26   }
27 }
```

`%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/create_dir.html`

```

1  <script type="text/javascript">
2  $(document).ready(function() {
3      var successMsg = '<imart type="string" value=$successMsg />';
4      var errorMsg = '<imart type="string" value=$errorMsg />';
5      if (successMsg != '') {
6          imuiShowSuccessMessage(successMsg);
7      } else if (errorMsg != '') {
8          imuiShowErrorMessage(errorMsg);
9      }
10     });
11 </script>
12
13 <imart type="head">
14     <title>Office365 - SharePointStorage API サンプル - ディレクトリ作成</title>
15 </imart>
16
17 <div class="imui-title">
18     <h1>
19         <imart type="string" value=$dirPath />
20     </h1>
21 </div>
22
23 <div class="imui-toolbar-wrap">
24     <div class="imui-toolbar-inner">
25         <ul class="imui-list-toolbar">
26             <li>
27                 <a href="javascript:$('#backToIndex').submit();" class="imui-toolbar-icon" title="一覧に戻る">
28                     <span class="im-ui-icon-common-16-back"></span> 一覧に戻る
29                 </a>
30             </li>
31         </ul>
32     </div>
33 </div>
34
35 <form id="backToIndex" method="post" action="sample/office365/filer">
36     <imart type="hidden" targetDirPath=$currentDirPath />
37 </form>

```

### ! 注意

Microsoft社が提供する Office 365 のWebAPIの動作仕様に起因し以下の制限があります。  
 SharePointStorage#makeDirectories() を利用しディレクトリを作成する場合、ルートディレクトリ直下に存在するディレクトリ名と同名のディレクトリを任意のディレクトリ配下に作成することはできません。

## SharePoint Online 上にファイルをアップロードする

SharePoint Online 上にファイルをアップロードする実装のサンプルです。  
 マルチパートフォームデータとして受け取った プロバイダID、ターゲットのディレクトリパス、ファイルを元にファイルをアップロードします。

実装例

SharePointStorage#createAsBinary() を利用します。  
画面遷移を伴わないため、プレゼンテーションページは不要です。

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/upload.js
```

```

1  function init(request) {
2      var userCd = Contexts.getAccountContext().userCd;
3      var upfile = request.getParameter('local_file');
4      var providerId = request.providerId;
5      var currentDirPath = request.targetDirPath;
6      var response = Web.getHTTPResponse();
7      response.setContentType('json');
8      request.setAttribute('imui-theme-builder-module', 'notheme');
9
10     var targetFilePath = currentDirPath + upfile.getFileName();
11
12     upfile.openValueAsBinary(function(reader) {
13         var storage = new SharePointStorage(userCd, providerId, targetFilePath);
14         storage.createAsBinary(function(writer) {
15             reader.transferTo(writer);
16             response.sendMessageBodyString(Imjson.toJSONString([ {
17                 'name':upfile.getFileName(),
18                 'size':upfile.getLength()
19             } ]));
20         });
21     });
22 }
```

## SharePoint Online 上のファイルをダウンロードする

SharePoint Online 上のファイルをダウンロードする実装のサンプルです。  
リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパス、ファイルの名称を元にファイルをダウンロードします。

### 実装例

Module.download.send() を利用します。  
画面遷移を伴わないため、プレゼンテーションページは不要です。

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/download.js
```

```

1  function init(request) {
2      var userCd = Contexts.getAccountContext().userCd;
3      var providerId = request.providerId;
4      var targetFilePath = request.targetFilePath;
5      var storage = new SharePointStorage(userCd, providerId, targetFilePath);
6      Module.download.send(storage, storage.getName());
7  }
```

## SharePoint Online 上のファイルの詳細情報を表示する

SharePoint Online 上のファイルの詳細情報を表示する実装のサンプルです。  
リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスをファイルの詳細情報を表示します。

### 実装例

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/show_detail.js
```

```
1  var $errorMsg;
2  var $targetDirPathForShowDetail;
3  var $detail = {};
4
5  function init(request) {
6    var userCd = Contexts.getAccountContext().userCd;
7    var providerId = request.providerId;
8    $targetDirPathForShowDetail = request.targetDirPathForShowDetail;
9
10   if ($targetDirPathForShowDetail == null) {
11     $errorMsg = 'ファイル、または、ディレクトリが指定されていません。';
12   } else {
13     var storage = new SharePointStorage(userCd, providerId, $targetDirPathForShowDetail);
14
15     $detail.canonicalPath = storage.getCanonicalPath();
16     $detail.webUrl = storage.getWebUrl();
17     $detail.decodedWebUrl = decodeURIComponent(storage.getWebUrl());
18     $detail.created = new Date(storage.created());
19     $detail.createdUserId = storage.createdUserId();
20     $detail.createdUserName = storage.createdUserName();
21     $detail.lastModified = new Date(storage.lastModified());
22     $detail.lastModifiedUserId = storage.lastModifiedUserId();
23     $detail.lastModifiedUserName = storage.lastModifiedUserName();
24     $detail.length = storage.length();
25   }
26 }
```

`%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/show_detail.html`





```

1 <script type="text/javascript">
2 $(document).ready(function() {
3   var errorMsg = '<imart type="string" value=$errorMsg />';
4   if (errorMsg != "") {
5     imuiShowErrorMessage(errorMsg);
6   }
7 });
8 </script>
9
10 <imart type="head">
11 <title>Office365 - SharePointStorage API サンプル - 詳細表示</title>
12 </imart>
13
14 <div class="imui-title">
15   <h1>
16     <imart type="string" value=$targetDirPathForShowDetail />
17   </h1>
18 </div>
19
20 <div class="imui-form-container-narrow">
21   <div class="imui-chapter-title">
22     <h2>詳細情報</h2>
23   </div>
24
25   <table class="imui-table">
26     <tr>
27       <th>正規化したパス</th>
28       <td>
29         <imart type="string" value=$detail.canonicalPath />
30       </td>
31     </tr>
32     <tr>
33       <th>コンテンツをブラウザ上で確認可能なWeb URL</th>
34       <td>
35         <imart type="tag" tagname="a" href=$detail.webUrl>
36           <imart type="string" value=$detail.decodedWebUrl /></imart>
37         </td>
38     </tr>
39     <tr>
40       <th>作成時刻</th>
41       <td>
42         <imart type="dateTime" value=$detail.created pattern="yyyy/MM/dd hh:mm:ss" />
43       </td>
44     </tr>
45     <tr>
46       <th>作成ユーザID</th>
47       <td>
48         <imart type="string" value=$detail.createdUserId />
49       </td>
50     </tr>
51     <tr>
52       <th>作成ユーザ名</th>
53       <td>
54         <imart type="string" value=$detail.createdUserName />
55       </td>
56     </tr>
57     <tr>
58       <th>最終更新時刻</th>
59       <td>
60         <imart type="dateTime" value=$detail.lastModified pattern="yyyy/MM/dd hh:mm:ss" />
61       </td>
62     </tr>
63     <tr>
64       <th>最終更新ユーザID</th>
65       <td>
66         <imart type="string" value=$detail.lastModifiedUserId />

```

```

67     </td>
68 </tr>
69 <tr>
70   <th>最終更新ユーザ名</th>
71   <td>
72     <imart type="string" value=$detail.lastModifiedUserName />
73   </td>
74 </tr>
75 <tr>
76   <th>ファイルサイズ</th>
77   <td>
78     <imart type="string" value=$detail.length /> バイト
79   </td>
80 </tr>
81 </table>
82 </div>

```

## SharePoint Online 上のファイルを削除する

SharePoint Online 上のファイルを削除する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスを元にファイルを削除します。

### 実装例

SharePointStorage#remove() を利用します。

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/remove.js
```

```

1  var $successMsg;
2  var $errorMsg;
3  var $targetPath;
4  var $currentDirPath;
5
6  function init(request) {
7    var userCd = Contexts.getAccountContext().userCd;
8    var providerId = request.providerId;
9    $currentDirPath = request.targetDirPath != null ? request.targetDirPath : '/';
10   $targetPath = request.targetPathForRemove != null ? request.targetPathForRemove : null;
11   if ($targetPath == null) {
12     $errorMsg = 'ファイル、または、ディレクトリが指定されていません。';
13   } else {
14     try {
15       var storage = new SharePointStorage(userCd, providerId, $targetPath);
16       var result = storage.remove(true);
17       if (result) {
18         $successMsg = '「' + $targetPath + '」を削除しました。';
19       } else {
20         $errorMsg = '「' + $targetPath + '」を削除できませんでした。';
21       }
22     } catch (e) {
23       $errorMsg = '削除時にエラーが発生しました。' + '<br><br>' + e.errorMessage;
24     }
25   }
26 }

```

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/remove.html
```

```

1  <script type="text/javascript">
2  $(document).ready(function() {
3      var successMsg = '<imart type="string" value=$successMsg />';
4      var errorMsg = '<imart type="string" value=$errorMsg />';
5      if (successMsg != '') {
6          imuiShowSuccessMessage(successMsg);
7      } else if (errorMsg != '') {
8          imuiShowErrorMessage(errorMsg);
9      }
10     });
11 </script>
12
13 <imart type="head">
14 <title>Office365 - SharePointStorage API サンプル - 削除</title>
15 </imart>
16
17 <div class="imui-title">
18     <h1>
19         <imart type="string" value=$targetPath />
20     </h1>
21 </div>
22
23 <div class="imui-toolbar-wrap">
24     <div class="imui-toolbar-inner">
25         <ul class="imui-list-toolbar">
26             <li>
27                 <a href="javascript:$('#backToIndex').submit();" class="imui-toolbar-icon" title="一覧に戻る">
28                     <span class="im-ui-icon-common-16-back"></span> 一覧に戻る
29                 </a>
30             </li>
31         </ul>
32     </div>
33 </div>
34
35 <form method="post" action="sample/office365/filer" id="backToIndex">
36     <imart type="hidden" targetDirPath=$currentDirPath />
37 </form>

```

## SharePoint Online 上のドキュメントをiframeで表示する

SharePoint Online 上のドキュメントをiframeで表示する実装のサンプルです。

リクエストパラメータとして受け取った プロバイダID、ターゲットのファイルパスを元にドキュメントをiframeで表示します。

### 実装例

imOffice365 タグを利用します。

`%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/show_iframe.js`

```
1  var $errorMsg;
2  var $targetPathForShowIframe;
3  var $isWordFile;
4  var $isExcelFile;
5  var $isPowerPointFile;
6  var $providerId
7
8  function init(request) {
9      var userCd = Contexts.getAccountContext().userCd;
10     $providerId = request.providerId;
11     $targetPathForShowIframe = request.targetPathForShowIframe;
12     if ($targetPathForShowIframe == null) {
13         $errorMsg = 'ファイルが指定されていません。';
14     } else {
15         var storage = new SharePointStorage(userCd, $providerId, $targetPathForShowIframe);
16         $isWordFile = isWord(storage.getName());
17         $isExcelFile = isExcel(storage.getName());
18         $isPowerPointFile = isPowerPoint(storage.getName());
19     }
20 }
21
22 function isWord(fileName) {
23     return getExtension(fileName) == '.docx' ? true : false;
24 }
25
26 function isExcel(fileName) {
27     return getExtension(fileName) == '.xlsx' ? true : false;
28 }
29
30 function isPowerPoint(fileName) {
31     return getExtension(fileName) == '.pptx' ? true : false;
32 }
33
34 function getExtension(fileName) {
35     var indexOfLastDot = fileName.lastIndexOf('.');
36     if (indexOfLastDot == -1) {
37         return null;
38     }
39     return fileName.substring(indexOfLastDot);
40 }
```



```
1 <script type="text/javascript">
2 $(document).ready(function() {
3   var errorMsg = '<imart type="string" value=$errorMsg />';
4   if(errorMsg != ''){
5     imuiShowErrorMessage(errorMsg);
6   }
7 });
8 </script>
9
10 <imart type="head">
11 <title>Office365 - iframe 表示</title>
12 </imart>
13
14 <div class="imui-title">
15   <h1>
16     <imart type="string" value=$targetPathForShowIframe />
17   </h1>
18 </div>
19
20 <div id="imui-container-inner">
21   <imart type="condition" validity=$isWordFile>
22     <imart type="imOffice365Word" providerId=$providerId path=$targetPathForShowIframe
23       width="640" height="480" />
24   </imart>
25   <imart type="condition" validity=$isExcelFile>
26     <imart type="imOffice365Excel" providerId=$providerId path=$targetPathForShowIframe
27       width="640" height="480" />
28   </imart>
29   <imart type="condition" validity=$isPowerPointFile>
30     <imart type="imOffice365PowerPoint" providerId=$providerId path=$targetPathForShowIframe
31       width="640" height="480" />
32   </imart>
33 </div>
```

**注意**

ドキュメントを表示するにはユーザが Office 365 - SharePoint Online にログインする必要があります。



前述の実装例を組み合わせ、SharePoint Online のファイラを作成します。

providerId は「[Office 365 連携 セットアップガイド](#)」で環境構築の際に設定した <im\_office365\_files\_api.xml> のプロバイダIDを指定してください。

#### 実装例

SharePointStorage#directoriesStorages()、SharePointStorage#filesStorages() を利用します。

```
%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/index.js
```



```

1  var $providerId = 'dhimart_1st.onmicrosoft.com';
2  var $userCd = Contexts.getAccountContext().userCd;
3  var $directoriesStorages;
4  var $filesDetails=[];
5  var $currentDirPath;
6  var $errorMsg;
7  var $parentDir;
8  var $existsParentDir;
9
10 function init(request){
11   $currentDirPath = request.targetDirPath != null ? addSlashToTheEnd(request.targetDirPath) : '/';
12   try{
13     var storage = new SharePointStorage($userCd, $providerId, $currentDirPath);
14     $directoriesStorages = storage.directoriesStorages(false);
15     var filesStorages = storage.filesStorages(false);
16     for (var i = 0 ; i<filesStorages.length ; i++){
17       var fileDetail = {
18         fileName : filesStorages[i].getName() ,
19         filePath : filesStorages[i].getPath() ,
20         isWordExcelPowerPointFile:isWordExcelPowerPointFile(filesStorages[i].getName())
21       };
22       $filesDetails[i] = fileDetail;
23     }
24     $parentDir = storage.getParent();
25     if($parentDir == null){
26       $existsParentDir = false;
27     }else{
28       $existsParentDir = true;
29     }
30   }catch(e if e.errorCode == 'oauth.precondition'){
31     $errorMsg = '「外部連携アプリケーション」画面でアプリケーションを許可してください。';
32   }catch(e){
33     $errorMsg = 'ディレクトリ、または、ファイル一覧の取得時にエラーが発生しました。'
34     + '<br><br>' + e.errorMessage;
35   }
36 }
37
38 function isWordExcelPowerPointFile(fileName){
39   var indexOfLastDot = fileName.lastIndexOf('.');
40   if (indexOfLastDot == -1) {
41     return false;
42   }
43   var extension = fileName.substr(indexOfLastDot);
44   if (extension == '.docx' || extension == '.xlsx' || extension == '.pptx') {
45     return true;
46   } else {
47     return false;
48   }
49 }
50
51 function addSlashToTheEnd(s) {
52   var indexOfLastSlash = s.lastIndexOf('/');
53   if((indexOfLastSlash == -1) || s.substr((s.length - 1)) == '/'){
54     return s;
55   } else {
56     return s + '/';
57   }
58 }

```

`%CONTEXT_PATH%/WEB-INF/jssp/src/sample/office365/filer/index.html`

```

1 <script type="text/javascript">
2   function moveDir(targetDirPath) {
3     $('#targetDirPathForMove').val('/') + targetDirPath);
4     $('#moveDirForm').submit();
5   }
6
7   function downloadFile(targetFilePath) {
8     $('#targetFilePath').val(targetFilePath);
9     $('#downloadFileForm').submit();
10  }
11
12  function removeFileOrDir(targetPath) {
13    $('#targetPathForRemove').val(targetPath);
14    $('#removeFileOrDirForm').submit();
15  }
16
17  function showDetail(targetDirPath) {
18    $('#targetDirPathForShowDetail').val(targetDirPath);
19    $('#showDetailForm').submit();
20  }
21
22  function showIframe(targetFilePath) {
23    $('#targetPathForShowIframe').val(targetFilePath);
24    $('#showIframeForm').submit();
25  }
26
27  $(document).ready(function() {
28    var errorMsg = '<imart type="string" value=$errorMsg />';
29    if(errorMsg != ''){
30      imuiShowErrorMessage(errorMsg);
31    }
32  });
33
34  $(function($) {
35    $('#toParent').click(function() {
36      $('#targetDirPathForMove').val(
37        '/' + '<imart type="string" value=$parentDir />');
38      $('#moveDirForm').submit();
39    });
40  });
41
42 </script>
43
44 <imart type="head">
45 <title>Office365 - SharePointStorage API サンプル</title>
46 </imart>
47
48 <div class="imui-title">
49   <h1>
50     <imart type="string" value=$currentDirPath />
51   </h1>
52 </div>
53
54 <div class="imui-form-container-narrow">
55
56   <div class="imui-chapter-title">
57     <h2>ファイル一覧</h2>
58   </div>
59
60   <form method="post" action="sample/office365/filer">
61     <imart type="imuiTextbox" name="targetDirPath" value=$currentDirPath size="70" />
62     <input type="submit" value="移動" class="imui-small-button" />
63   </form>
64
65   <br>
66

```

```

67 <table class="imui-table">
68 <tr>
69 <th>名前</th>
70 <th>詳細</th>
71 <th>削除</th>
72 <th>iframe表示</th>
73 </tr>
74 <imart type="condition" validity=$existsParentDir>
75 <tr>
76 <td>
77 <a href="javascript:void(0)" id="toParent">親ディレクトリへ移動</a>
78 </td>
79 <td></td>
80 <td></td>
81 <td></td>
82 </tr>
83 </imart>
84 <!-- ディレクトリ一覧 -->
85 <imart type="repeat" list=$directoriesStorages item="dirStorage" index="idx">
86 <tr>
87 <td>
88 <span class="im-ui-icon-common-16-folder"></span>
89 <a href="javascript:void(0)"
90 onclick='moveDir("<imart type="string" value=dirStorage.getPath/>");'>
91 <imart type="string" value=dirStorage.getName />
92 </a>
93 </td>
94
95 <td align="center">
96 <a href="javascript:void(0)"
97 onclick='showDetail("<imart type="string" value=dirStorage.getPath/>");'>
98 <span class="im-ui-icon-common-16-information"></span>
99 </a>
100 </td>
101
102 <td align="center">
103 <a href="javascript:void(0)"
104 onclick='removeFileOrDir("<imart type="string" value=dirStorage.getPath/>");'>
105 <span class="im-ui-icon-common-16-trashbox"></span>
106 </a>
107 </td>
108 <td></td>
109 </tr>
110 </imart>
111
112 <!-- ファイル一覧 -->
113 <imart type="repeat" list=$filesDetails item="fileStorage" index="idx">
114 <tr>
115 <td>
116 <span class="im-ui-icon-common-16-document"></span>
117 <a href="javascript:void(0)"
118 onclick='downloadFile("<imart type="string" value=fileStorage.filePath/>");'>
119 <imart type="string" value=fileStorage.fileName />
120 </a>
121 </td>
122
123 <td align="center">
124 <a href="javascript:void(0)"
125 onclick='showDetail("<imart type="string" value=fileStorage.filePath/>");'>
126 <span class="im-ui-icon-common-16-information"></span>
127 </a>
128 </td>
129
130 <td align="center">
131 <a href="javascript:void(0)"
132 onclick='removeFileOrDir("<imart type="string" value=fileStorage.filePath/>");'>

```

```

133     <span class="im-ui-icon-common-16-trashbox"></span>
134   </a>
135 </td>
136
137 <td align="center">
138   <imart type="condition" validity=fileStorage.isWordExcelPowerPointFile>
139     <a href="javascript:void(0)"
140       onclick='showIframe("<imart type="string" value=fileStorage.filePath/>");'>
141       <span class="im-ui-icon-common-16-preview"></span>
142     </a>
143   </imart>
144 </td>
145 </tr>
146 </imart>
147
148 </table>
149
150 <br>
151
152 <div class="imui-chapter-title">
153   <h2>ファイルアップロード</h2>
154 </div>
155
156 <form id="uploadForm" action="sample/office365/filer/upload" enctype="multipart/form-data">
157   <imart type="imuiFileUpload" outerFormId="uploadForm" />
158   <imart type="input" style="hidden" value=$currentDirPath id="targetDirPath" name="targetDirPath" />
159   <imart type="hidden" providerId=$providerId />
160 </form>
161
162 <div class="imui-chapter-title">
163   <h2>ディレクトリ新規作成</h2>
164 </div>
165
166 <form method="POST" name="createDir" id="createDir"
167   action="sample/office365/filer/create_dir">
168   <imart type="imuiTextbox" name="createDirName" />
169   <imart type="hidden" targetDirPath=$currentDirPath providerId=$providerId />
170   <input type="submit" value="作成" class="imui-small-button" />
171 </form>
172
173 <form id="moveDirForm" name="moveDirForm" method="POST"
174   action="sample/office365/filer">
175   <imart type="input" style="hidden" name="targetDirPath" id="targetDirPathForMove" />
176   <imart type="hidden" providerId=$providerId />
177 </form>
178
179 <form id="showDetailForm" name="showDetailForm" method="POST"
180   action="sample/office365/filer/show_detail" target="office365_show_detail">
181   <imart type="input" style="hidden" name="targetDirPathForShowDetail"
182   id="targetDirPathForShowDetail" />
183   <imart type="hidden" providerId=$providerId />
184 </form>
185
186 <form name="removeFileOrDirForm" id="removeFileOrDirForm" method="POST"
187   action="sample/office365/filer/remove">
188   <imart type="input" style="hidden" id="targetPathForRemove" name="targetPathForRemove" />
189   <imart type="hidden" targetDirPath=$currentDirPath providerId=$providerId />
190 </form>
191
192 <form name="downloadFileForm" id="downloadFileForm" method="POST"
193   action="sample/office365/filer/download">
194   <imart type="input" style="hidden" name="targetFilePath" id="targetFilePath" />
195   <imart type="hidden" providerId=$providerId />
196 </form>
197
198 <form method="POST" name="showIframeForm" id="showIframeForm"

```

```

199     action="sample/office365/filer/show_iframe" target="office365_show_iframe">
200     <imart type="input" style="hidden" name="targetPathForShowIframe" id="targetPathForShowIframe" />
201     <imart type="hidden" providerId=$providerId />
202 </form>
</div>

```

## ルーティングテーブルを作成する

「スクリプト開発モデル用ルーティングテーブル」を利用してURLに対して スクリプト開発モデル のプログラムの割り当てを行います。

詳しくは「スクリプト開発モデル プログラミングガイド」-「ルーティング」を参照してください。

### 実装例

```
%CONTEXT_PATH%/WEB-INF/conf/routing-jssp-config/office365.xml
```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <routing-jssp-config
3   xmlns="http://www.intra-mart.jp/router/routing-jssp-config"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://www.intra-mart.jp/router/routing-jssp-config routing-jssp-config.xsd ">
6
7   <authz-default mapper="welcome-all" ></authz-default>
8
9   <file-mapping path="/sample/office365/filer" page="sample/office365/filer/index" />
10  <file-mapping path="/sample/office365/filer/remove" page="sample/office365/filer/remove"/>
11  <file-mapping path="/sample/office365/filer/download" page="sample/office365/filer/download" />
12  <file-mapping path="/sample/office365/filer/upload" page="sample/office365/filer/upload" />
13  <file-mapping path="/sample/office365/filer/create_dir" page="sample/office365/filer/create_dir" />
14  <file-mapping path="/sample/office365/filer/show_detail" page="sample/office365/filer/show_detail" />
15  <file-mapping path="/sample/office365/filer/show_iframe" page="sample/office365/filer/show_iframe" />
16 </routing-jssp-config>

```