

記事

[Megumi Kakechi](#) · 2020年9月16日 4m read

HTMLからRESTを使って画像ファイルをアップロードする方法

これはInterSystems FAQ サイトの記事です。

HTMLからRESTを使って画像ファイルをアップロードする方法をご紹介します。

1.はじめに、以下のようなhtmlとクラスを作成してください。

＊[UploadTest.html](#)

```
<html lang="ja">
<head>
  <title>Upload</title>
</head>
<body>
  <input id="up" type="file" />
  <button id="btn">Upload</button>
  <div></div>
  <script type="text/javascript">
    const sendfile = function(e) {
      let up = document.getElementById("up");
      let file = up.files[0];
      let fd = new FormData();
      fd.append("imgfile", file);
      let xmlhttp = new XMLHttpRequest();
      xmlhttp.onreadystatechange = function () {
        var result = document.querySelector('div');
        xmlhttp.onload = function () {
          result.innerHTML = xmlhttp.responseText;
        };
      };
      xmlhttp.open("POST", "http://127.0.0.1:52773/csp/user/isjtest/uploading", true);
      xmlhttp.send(fd);
    }
    let btn = document.getElementById("btn");
    btn.addEventListener("click", sendfile);
  </script>
</body>
</html>
```

＊[User.MyREST.cls](#) (IRIS/CachéサーバのUSERネームスペースに作成してください)

※こちらのサンプルでは、C:\tempフォルダにファイルをUploadしています。
適宜フォルダを作成して頂くか、任意のフォルダパスに変更して再コンパイルしてください。

```
Class User.MyREST Extends %CSP.REST
{
```

HTMLからRESTを使って画像ファイルをアップロードする方法

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
Parameter CONVERTINPUTSTREAM = 1;  
Parameter HandleCorsRequest = 1;
```

```
XData UrlMap  
{  
  <Routes>  
    <Route Url="/uploadimg" Method="POST" Call="readMimeData" />  
  </Routes>  
}  
ClassMethod readMimeData() As %Status  
{  
  set upload=$g(%request.MimeData("imgfile", 1))  
  set fname=%request.MimeData("imgfile",1).FileName  
  set file=##class(%File).%New("c:\temp\"_fname)  
  do file.Open("NWUK\BIN\  
  do file.CopyFrom(upload)  
  set st = file.%Save()  
  if st {  
    write fname_" ????????!!"  
  } else {  
    write fname_" ????????"  
  }  
  do file.Close()  
  quit $$$OK  
}
```

2. ウェブ・アプリケーション /csp/user/isjtest の定義を作成します。

管理ポータル：[システム管理]>[セキュリティ]>[アプリケーション]>

[ウェブ・アプリケーション]>[新しいウェブ・アプリケーションを作成]

RESTのディスパッチ・クラスに、1. で作成したUser.MyRESTクラスを指定します。(下記画像参照)

ウェブ・アプリケーション /csp/user/isjtest の定義を編集:

アプリケーションを保存しました。

新しいウェブ・アプリケーションを作成するにはここをクリックしてください。

一般 アプリケーション・ロール マatching・ロール

名前
必須です。(例: /csp/appname)

説明

ネームスペース USERのデフォルト・アプリケーション: /csp/user ☐ ネームスペースのデフォルト・アプリケーション

アプリケーション有効 ☒

有効 ☒ REST
ディスパッチ・クラス
必須です。

☐ CSP/ZEN
☐ アナリティクス ☒ 着信 Web サービス ☐ ログイン CSRF 攻撃を防ぐ

セキュリティの設定
必要なリソース IDでグループ化
許可された認証方法 ☒ 認証なし ☐ パスワード ☐ Kerberos ☐ LDAP ☐ ログイン Cookie

セッションの設定
セッションタイムアウト 秒 イベントクラス .cls
セッションにクッキーを使用する セッションクッキーパス

3. 必要に応じて UploadTest.html を編集し(※1)、Webサーバのドキュメントルート(※2)に配置します。

※1. xmlhttp.openには、環境にあったIPアドレス・ポートを指定してお試しください。

HTMLからRESTを使って画像ファイルをアップロードする方法

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
xmlhttp.open("POST", "http://<???IP>:<IRIS/Caché???>/csp/user/isjtest/uploading", true);
```

※2. 例：C:\netpub\wwwroot

4. クライアントブラウザより以下を実行し、任意のファイルをUploadします。

<http://localhost/UploadTest.html>

5. 指定したフォルダにファイルがUploadされたことをご確認ください。

[#CSP #REST API #Caché #Ensemble #InterSystems IRIS #InterSystems IRIS for Health](#)

ソースURL:

<https://jp.community.intersystems.com/post/html%E3%81%8B%E3%82%89rest%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6%E7%94%BB%E5%83%8F%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%82%92%E3%82%A2%E3%83%83%E3%83%97%E3%83%AD%E3%83%BC%E3%83%89%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95>