OFD服务系统接口规范

V1.0

山东英围特智能科技有限公司

www.sdinvent.com

2020.07

1 综述

OFD 服务系统包括：ofd 转 pdf 服务、ofd 转图服务、ofd 打印服务、ofd 数据提取服务、ofd 验证服务等。每个服务可独立部署，对外提供 web api 接口；由于采用了标准的 web 接口，所以对客户端没有任何限制，只要传输的数据符合规范即可。本文对接口定义做说明，并提供完整开发示例（c#语言版）。

2 接口定义

接口定义宗旨

1）简单：尽量隐藏处理细节；数据采用 json 格式，方便处理。

2）统一：不同的服务遵照共同的处理思路；达到触类旁通的目的，降低开发难度。

本文以 ofd 转 pdf 为例做讲解。

服务端函数定义：***string OfdFileToPdf (string input);***

函数的输入和输出皆为 string 类型，string 值为类型 json 的序列化结果。

Input 的定义为：

```
public class OfdToPdfInput
    {
        public string ClientData { get; set; }      //客户端数据，服务端会将此数据原样返回
        public List<OfdFileInput> ListOfdFileInput { get; set; } = new List<OfdFileInput>();
    }

    //ofd 文件信息
    public class OfdFileInput
    {
        public string OfdFileID { get; set; }   //文件唯一标识，可以为 guid。
        public string FileName { get; set; }    //文件名
        public string OfdFileDataBase64 { get; set; } //ofd 文件数据,base64 格式
    }
```

接口调用返回值

```
public class OfdToPdfOutput
    {
        public string ClientData { get; set; }//客户端数据，服务端会将此数据原样返回
        public List<OfdFileOutput> ListOfdFileOutput { get; set; } = new List<OfdFileOutput>();
//pdf 文件
        public string ServerData { get; set; }  //服务端返回的其它数据
    }

    public class OfdFileOutput
    {
```

```
        public bool OK { get; set; }            //是否转换成功
        public string Message { get; set; }     //提示信息

        public string OfdFileID { get; set; }   //文件 id ，客户端传来的值原样返回
        public string FileName { get; set; }    //文件名称 ，客户端传来的值原样返回
        public string PdfDataBase64 { get; set; }    //pdf 数据，base64 格式
    }
```

C#客户端调用代码示例

```
    /// <summary>
    /// web api 调用辅助类
    /// </summary>
    class OfdToPdfConvertor
    {
        public string HttpServerUrl { get; set; }

        public OfdToPdfOutput OfdFileToPdf(OfdToPdfInput inputParam)
        {
            //将发送到服务端的参数 json 序列化
            string jsonText = JsonConvert.SerializeObject(inputParam);

            //入参定义形式为(string input)
            NameValueCollection postParam = new NameValueCollection();
            postParam.Add("input", jsonText);

            //采用 post，将数据发送到服务端
            WebClient webClient = new WebClient();
            byte[] data = webClient.UploadValues(HttpServerUrl, "POST", postParam);
            string strData = Encoding.UTF8.GetString(data);

            //将从服务端收到的数据 json 反序列化，得到类型定义
            OfdToPdfOutput ofdOutputParam =
JsonConvert.DeserializeObject<OfdToPdfOutput>(strData);

            return ofdOutputParam;
        }
    }
```

各类接口定义和相关示例参见附件。