

## ServiceFile

Questa è una webApplication che viene utilizzata per la richiesta di dati, principalmente di file, in formato zip o standard. Può essere richiamata passandole vari parametri tra cui quello **operation** tramite il quale viene distinto il comportamento del servizio.

I possibili valori di operation e i relativi parametri da passare vengono di seguito descritti:

└ operation=download

Questa funzione permette di ricevere in input dei file o il contenuto dei file e li restituisce in formato file o zip. (Utilizzato per risolvere le problematiche legate a IE9). I parametri che accetta in input e le varie combinazioni sono:

- operation=download&forceZip=1&jsonObject=.....:

indica che deve essere restituito un file in formato zip composto da n file. Il contenuto del file zip viene inviato in un oggetto json dalla seguente struttura:

```
{
  "fileZipName": "nomeFileZip",
  "elencoFiles": [
    {
      "fileName": "nomeSingoloFile1",
      "content": "contenuto binario file1"
    },
    {
      "fileName": "nomeSingoloFile2",
      "content": "contenuto binario file2"
    }
  ]
}
```

La struttura json contiene quindi in nome del file e un elenco di file in esso contenuto dove per ognuno viene definito il content e il nome.

**Output: viene inviato nel Request il file (Response.BinaryWrite)**

- operation=download&forceZip=1&numeroTotale=numeroTotaleFile&fileName=nomeFileZip&fileNameSingle\_1=nomeFile1&fileContent\_1=contenutoFile1&fileNameSingle\_2=nomeFile2&fileContent\_2=contenutoFile2.....

Indica che deve essere restituito un file zip il cui nome è indicato nel parametro fileName e che è composto da n files (dove n è uguale al parametro numeroTotale) i quali nomi e contenuti sono definiti nei parametri fileNameSingle\_numero e fileContent\_numero

**Output: viene inviato nel Request il file (Response.BinaryWrite )**

- operation=download&forceZip=0&jsonObject=.....:

indica che deve essere restituito un singolo file NON zippato. Il contenuto del file zip viene inviato in un oggetto json dalla seguente struttura:

```
{
  "fileZipName": "",
  "elencoFiles": [
    {
      "fileName": "nomeFile",
      "content": "contenuto binario file"
    }
  ]
}
```

La struttura json è la stessa dell'esempio precedente ma in questo caso ovviamente il parametro fileZipName non è valorizzato (qualora lo fosse viene ignorato) e elencoFiles contiene solo un elemento (qualora ne contenesse di più verrà preso in considerazione solo il primo) con nome e

contenuto del file.

**Output: viene inviato nel Request il file (Response.Write)**

- `operation=download&forceZip=0&fileName=nomeFileZip&fileContent=contenutoFile1`

Indica che deve essere restituito un file così come viene inviato il cui nome è indicato nel parametro `fileName` il cui contenuto è definito nel parametro `fileContent`

**Output: viene inviato nel Request il file (Response.Write)**

- [ `operation=getFile&nomeFile=nomefile&rename=eventuale nome output del file`

Indica che deve essere restituito un file che è stato precedentemente salvato in una cartella preimpostata ( `c:\esriTempFile`). Il file da restituire è contenuto nel parametro **nomeFile**. Se viene passato anche il valore **rename** in output verrà restituito il file nominato come indicato in questo parametro. Il contenuto del campo **rename** NON deve contenere l'estensione che sarà poi fissata da codice come **.zip**.

**Output: viene inviato nel Request il file (Response.BinaryWrite )**

- [ `operation=upload&fd-callback=nomeFunzioneRitorno`

**fd-file:** parametro di tipo file letto tramite la `Request.File`

Indica che deve essere restituito un file preso in input. Se il parametro **fd-callback** è valorizzato verrà restituito in output un html con all'interno uno script che richiama la funzione callback presente sul chiamante passandole il contenuto del file e il nome (Codice in out= `. window.top. " + callback + "(" + output + "\", " + name.Replace("@\"", "@\"")`). Se il parametro callback è vuoto allora viene restituito il file direttamente in output.

**Output:**

- **callback impostata: viene restituita una pagina html con all'interno la chiamata della callback**
- **callback non impostata: viene restituito il file (Response.Write)**

- [ `operation=decrypt&valore=stringa da criptare`

Questa operazione restituisce il valore criptato della stringa “valore” che gli viene passata. Questa operazione è utile per configurare il file config con le credenziali criptate

- [ `operation=enc&valore=stringa da decifrare`

Questa operazione restituisce il valore normalizzato della stringa “valore” che gli viene passata. Questa operazione è utile per verificare il file config con le credenziali criptate

- [ `operation=excelfromdata&jData=jsonParametri`

Questa operazione si occupa di ricevere i dati direttamente dal client di organizzarli in tab e file excel da restituire in forma di file .zip. Il formato json è simile al seguente

```
{
  "nameZip": "",
  "directDownload": 1,
  "isCSV": 0,
  "convertData": "1",
  "callToken": 0,
  "queryParameter": {},
  "parameterAttribute": {},
  "files": [{
    "nameFile": "dati Prova",
    "objData": oggetto array dove al primo indice ci sono i nomi delle colonne, mentre
    dall'1 in poi ci sono i contenuti delle righe e colonne. Attenzione il nome degli attributi
    DEVE corrispondere con il nome colonna definito all'indice 0. Es: [
      ["OBJECTID", "NAME", "TYPE", "STATEFIPS", "COUNTYFIPS", "LOCALFIPS",
      "SQUAREMILE", "REVISIONDA", "SHAPE_Length", "SHAPE_Area"], {
        "OBJECTID": 1,
```

```

      "NAME": "Farmington",
      "TYPE": "City",
      "STATEFIPS": "26",
      "COUNTYFIPS": "125",
      "LOCALFIPS": "27380",
      "SQUAREMILE": 2.66448482,
      "REVISIONDA": 1146614400000,
      "SHAPE_Length": 59188.91597934275,
      "SHAPE_Area": 74281574.01680636
    }, {
      "OBJECTID": 2,
      "NAME": "Farmington Hills",
      "TYPE": "City",
      "STATEFIPS": "26",
      "COUNTYFIPS": "125",
      "LOCALFIPS": "27440",
      "SQUAREMILE": 33.29854028,
      "REVISIONDA": 1146614400000,
      "SHAPE_Length": 181859.09900748872,
      "SHAPE_Area": 928309272.2392159
    }, {
      "OBJECTID": 13,
      "NAME": "Southfield",
      "TYPE": "City",
      "STATEFIPS": "26",
      "COUNTYFIPS": "125",
      "LOCALFIPS": "74900",
      "SQUAREMILE": 26.27247276,
      "REVISIONDA": null,
      "SHAPE_Length": 148327.82225335555,
      "SHAPE_Area": 732461035.2827194
    }
  ],
  "tabs": []
}],
"forceZIP": false
}

```

[ operation=query&jData=jsonParametri

Questa operazione permette di effettuare query verso mapserver, di organizzare i dati di output secondo dei parametri configurabili “salvandoli” in file .zip e di ricevere i file in output. La parametrizzazione è demandata al json jData che è così strutturato:

```

{
  "nameZip": "nomeFile",
  "isCsv": "flag (0|1) che indica se il file deve essere in csv",
  "directDownload": "flag (0|1) che indica se deve essere inviato direttamente nella response il file oppure se deve essere salvato localmente nella cartella c:\esriTemp",
  "convertData": "flag (0|1) che indica se i campi, identificati come esriDateTime devono essere convertiti da numero a data nel formato dd/mm/yyyy",
  "queryParameter": "oggetto json contenente i parametri da passare alla query es: {
    \"geometryType\": \"esriGeometryEnvelope\",
    \"geometry\": \"{ 'type': 'extent', 'xmin': 607406.519857252, 'ymin': 5448928.053776373, 'xmax': 1214010.7763282498, 'ymax': 5835393.6687861215, 'spatialReference': { 'wkid': 102100 } }\",
    \"spatialRel\": \"esriSpatialRelIntersects\",
    \"inSR\": 102100,
    \"outSR\": 102100,
    \"returnGeometry\": false,

```

```

"callbackParamName": "callback",
"token": "-2oDCF3I9IxHr659O9bO-7j0R4Q-4MV8wRRxtvjOVcRuCFxOyo9X6dHrrP0my5KM",
"where": "",
"f": "json"
},
"parameterAttribute": oggetto json con due parametri per il reperimento dei dati {
  "shape": prefisso campo es. "PIR2015.",
  "allegati": prefisso campo es. "PIR2014."
},
"files": Array json contenente gli attributi per le query [
  {
    "nameFile": nome del singolo file di output es. "NomeFile1",
    "url": url per la chiamata query es.
    "https://nomeServer/arcgis/rest/services/Servizio/MapServer/0/query",
    "outFields": array json contenente una stringa con all'interno i campi da ricevere in output (da ottimizzare) es. [
      "campo1, campo2...." oppure "*"
    ],
    "tabs": Array json contenente la definizione dei tab che devono essere presenti nel file excel [
      {
        "nameTab": nome del tab es. "nomeTab1",
        "attributes": array json con l'elenco delle corrispondenze tra nome colonna e nome campo da inserire [
          {
            "nameAttribute": stringa contenente il nome della colonna es. "Nome Colonna 1",
            "nameOutFields": campo di riferimento (presente negli outfields) da inserire nella colonna es. "DESCRIZIONE"
            "typeAttribute": stringa contenente la tipologia di attributo (SHAPE o ALLEGATI) che viene utilizzata per costruire la stringa per prelevare il campo dal json di output della query es. "shape"
          },
          {
            "nameAttribute": "Nome Colonna 2",
            "nameOutFields": "campo di riferimento da inserire nella colonna",
            "typeAttribute": "tipo di attributo per richiamare il campo 2"
          },
          {
            "nameAttribute": "Nome Colonna n",
            "nameOutFields": "campo di riferimento da inserire nella colonna",
            "typeAttribute": "tipo di attributo per richiamare il campo n"
          }
        ]
      },
      {
        "nameTab": "nomeTabN",
        "attributes": [
          {
            "nameAttribute": "Nome Colonna 1",
            "nameOutFields": "campo di riferimento da inserire nella colonna",
            "typeAttribute": "tipo di attributo per richiamare il campo 1"
          },
          {
            "nameAttribute": "Nome Colonna 2",
            "nameOutFields": "campo di riferimento da inserire nella colonna",
            "typeAttribute": "tipo di attributo per richiamare il campo 2"
          }
        ]
      }
    ]
  }
]

```

```

    "nameAttribute": "Nome Colonna n",
    "nameOutFields": "campo di riferimento da inserire nella colonna",
    "typeAttribute": "tipo di attributo per richiamare il campo n"
  }
}
],
},
{
  "nameFile": "NomeFile2",
  "url": "https://nomeServer/arcgis/rest/services/Servizio/MapServer/0/query",
  "outFields": [
    "campo1, campo2...." oppure "*"
  ],
  "tabs": [
    {
      "nameTab": "nomeTab1",
      "attributes": [
        {
          "nameAttribute": "Nome Colonna 1",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo 1"
        },
        {
          "nameAttribute": "Nome Colonna 2",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo 2"
        },
        {
          "nameAttribute": "Nome Colonna n",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo n"
        }
      ]
    },
    {
      "nameTab": "nomeTabN",
      "attributes": [
        {
          "nameAttribute": "Nome Colonna 1",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo 1"
        },
        {
          "nameAttribute": "Nome Colonna 2",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo 2"
        },
        {
          "nameAttribute": "Nome Colonna n",
          "nameOutFields": "campo di riferimento da inserire nella colonna",
          "typeAttribute": "tipo di attributo per richiamare il campo n"
        }
      ]
    }
  ]
},
]
},

```

**Output: se c'è stato un errore durante l'esecuzione viene restituita una pagina html con la definizione di una funzione window.returnValue che ha un json che ha la seguente struttura:**

```
{  
    "isError": valore booleano che indica se è avvenuto un errore (in questo caso true),  
    "errMessage": descrizione dell'errore,  
    "errNumber": numero dell'errore,  
    "nomeFile": stringa vuota (in questo caso)  
}
```

**se non c'è stato un errore durante l'esecuzione e il directDownload è impostato a 1 viene restituita una pagina html con la definizione di una funzione window.returnValue che ha un json che ha la seguente struttura:**

```
{  
    "isError": valore booleano che indica se è avvenuto un errore (in questo caso false),  
    "errMessage": descrizione dell'errore,  
    "errNumber": numero dell'errore,  
    "nomeFile": stringa contenente il nome del file salvato (in questo caso)  
}
```

**se non c'è stato un errore durante l'esecuzione e il directDownload è impostato a 0 viene restituito il file generato direttamente nel Request (Response.BinaryWrite )**

Configurazione di web.config nel caso di query

Impostazioni di user name, password e dominio criptati (attraverso il metodo di decrypt)

```
<appSettings>  
  <add key="AGS_AUTHUSERN" value=""/>  
  <add key="AGS_AUTHUSERP" value=""/>  
  <add key="AGS_AUTHUSERDOM" value=""/>  
</appSettings>
```

Elenco di url e relativi portal a cui chiedere le credenziali per il token

```
<generateToken>  
  <server>  
    <add portal="https://gis20.a20g.lan/" url="https://gis20ags.a20g.lan/agsi"/>  
    <add portal="https://gis20.a20g.lan/" url="https://gis20ags.a20g.lan/agse"/>  
  </server>  
</generateToken>
```