

## FSA+ Web Service: Integration Guide

### Introduction

The purpose of this document is to provide technical guidance for development teams, enabling developers to quickly and easily integrate the SDM FSA+ Web Service into their applications.

This guide assumes prior knowledge of XML, and HTTP-based SOAP Web Services.

If you encounter any integration problems, please contact us (<http://www.sdm-fsa-register.co.uk/contact.aspx>), and we will be pleased to provide assistance.

### Details

The SDM FSA Register+ Web Service allows you to execute queries against the FSA Register database from within your applications. The service can be used to generate lists of organisations or individuals matching your specified search criteria, or to retrieve extended information on any organisation or individual listed on the register. We ensure the data is regularly updated (at the frequency specified in your subscription plan), and implement the search algorithms, so providing FSA Register search capabilities within your application could be achieved in fewer than twenty lines of code!

### Implementation

The SDM FSA Register+ Web Service is built using the .NET Framework, but is technically platform agnostic, as the communication takes place over HTTP rather than a platform specific protocol, and the results are returned as XML, which can be parsed using any programming language.

The service can be accessed at the following URL:

<https://www.sdm-fsa-register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx>

If your development platform includes support for WSDL, you can access the service's WSDL file at the following URL:

<https://www.sdm-fsa-register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx?wsdl>

Please note that whilst the return type of each of the methods is listed as 'DataSet' (which is a .NET specific in-memory data-cache) in the WSDL document, the schema for each of these is available for download (see 'Methods' section below for the relevant URLs) to enable you to write efficient parsing routines, if your chosen development platform does not include support for automatic generation of a service proxy.

Please also note that when calling any of the methods, your FSA Register+ user name and password must be passed in the request header, in the format shown in the 'request message format' sections below.

Finally, please also note that the service is secured via SSL. This ensures that the entire request message you send to the service is encrypted, so your user

credentials can be securely transmitted. However, if manually developing your own service proxy, you will need to ensure that the network communication takes place over an SSL port 443 connection; any unencrypted connections attempted via port 80 are automatically refused by the server. This guarantees the data is securely transmitted between your application and the FSA+ Web Service server at all times.

## Methods

The following sections describe the methods exposed by the SDM FSA Register+ Web Service which can be invoked from your application, and the format of the data returned by each.

The methods available have been categorised into three functional groups, as follows:

- i. The search methods are the primary methods you will use to query the FSA Register. These represent the core functionality of the SDM FSA Register+ Web Service.
- ii. The lookup methods are used to populate lookup lists within your application. These are useful for allowing users to build search queries when search parameter values can be selected from a finite list of available values. For example, you can query the appropriate web service lookup method to return all the valid FSA Register Authorisation Statuses, to allow your application users to filter their search queries by authorisation status.
- iii. The additional services methods provide supplementary functionality such as user authentication.

### **i. Search Methods**

---

#### **i.i SearchOrganisations()**

This method allows you to search all the current organisations within the FSA Register, and filter your search by a range of parameters.

#### **Request Message Format:**

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap:Header>
  <soap:Body>
    <SearchOrganisations xmlns="http://sdm-fsa-register.co.uk/">
      <criteria>
        <Postcode>string</Postcode>
        <StatusChangedFrom>dateTime</StatusChangedFrom>
        <StatusChangedTo>dateTime</StatusChangedTo>
        <PermissionsChangedFrom>dateTime</PermissionsChangedFrom>
        <PermissionsChangedTo>dateTime</PermissionsChangedTo>
        <AddedFrom>dateTime</AddedFrom>
        <AddedTo>dateTime</AddedTo>
        <AppointeeStatusChangedFrom>dateTime</AppointeeStatusChangedFrom>
        <AppointeeStatusChangedTo>dateTime</AppointeeStatusChangedTo>
      </criteria>
    </SearchOrganisations>
  </soap:Body>
</soap:Envelope>
```

```

    <AuthStatus>string</AuthStatus>
    <Activity>string</Activity>
    <InvestmentType>string</InvestmentType>
    <ModifiedFrom>dateTime</ModifiedFrom>
    <ModifiedTo>dateTime</ModifiedTo>
  </criteria>
</SearchOrganisations>
</soap:Body>
</soap:Envelope>

```

### Request Message Parameters:

*i. Postcode (string)*

The postcode, or postcode area, to filter the result set by. This can be either full or partial postcode (i.e. 'SW1 1TT' or 'SW1')

*ii. StatusChangedFrom (date: DD/MM/YYYY format)*

Searches for organisations which have changed authorisation status after the specified date.

*iii. StatusChangedTo (date: DD/MM/YYYY format)*

Searches for organisations which have changed authorisation status before the specified date.

*iv. PermissionsChangedFrom (date: DD/MM/YYYY format)*

Searches for organisations which have changed permissions after the specified date.

*v. PermissionsChangedTo (date: DD/MM/YYYY format)*

Searches for organisations which have changed permissions before the specified date. The date should be in the format DD/MM/YYYY.

*vi. AddedFrom (date: DD/MM/YYYY format)*

Searches for organisations which have been added to the register after the specified date.

*vii. AddedTo (date: DD/MM/YYYY format)*

Searches for organisations which have been added to register before the specified date.

*viii. AppointeeStatusChangedFrom (date: DD/MM/YYYY format)*

Searches for organisations whose appointee has changed status after the specified date.

*ix. AppointeeStatusChangedTo (date: DD/MM/YYYY format)*

Searches for organisations whose appointee has changed status before the specified date.

*x. AuthStatus (string)*

Searches for organisations with a specified authorisation status.

*xi. Activity (string)*

Searches for organisations performing a specified regulated activity.

*xii. InvestmentType (string)*

Searches for organisations with a specified investment type.

*xiii. ModifiedFrom (date: DD/MM/YYYY format)*

Searches for organisations whose record was modified on or after the specified date.

*xiv. ModifiedTo (date: DD/MM/YYYY format)*

Searches for organisations whose record was modified on or before the specified date.

### Response Message Format:

```

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SearchOrganisationsResponse xmlns="http://tempuri.org/">
      <SearchOrganisationsResult>dataset</SearchOrganisationsResult>
    </SearchOrganisationsResponse>
  </soap12:Body>
</soap12:Envelope>

```

### Response Dataset Schema:

<http://www.sdm-fsa-register.co.uk/documents/OrganisationSearch.xsd>

**Sample Invocation Code (c#):**

```
//create a new service proxy instance
FsaPlus.SdmFsaPlusWebService service = new
NewsServiceTestApp.FsaPlus.SdmFsaPlusWebService();

//set the service url (this is set automatically if you use visual
studio to generate your
//web service proxy
service.Url = "https://www.sdm-fsa-
register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx";

//pass your FSA Register+ credentials here
service.AuthHeaderValue = new AuthHeader();
service.AuthHeaderValue.UserName = "user_name";
service.AuthHeaderValue.Password = "password";

//create the search criteria object
OrganisationSearchCriteria criteria = new
OrganisationSearchCriteria();
criteria.StatusChangedFrom = DateTime.Now.Date.AddYears(-2);
criteria.StatusChangedTo = DateTime.Now.Date.AddYears(-1);

//invoke the SDM FSA Register+ web service
OrganisationSearch results = service.SearchOrganisations(criteria);

//consume the returned results...
foreach (OrganisationSearch.ResultsRow row in results.Results)
{
    int id = row.Reference;
    string name = row.OrganisationName;
}
```

**Sample XML Output:**

```
<?xml version="1.0" standalone="yes"?>
<OrganisationSearch xmlns="http://tempuri.org/OrganisationSearch.xsd">
  <Results>
    <Reference>000000</Reference>
    <OrganisationName>Test Company 1</OrganisationName>
    <Postcode>SS6 7DQ</Postcode>
    <Status>Authorised</Status>
    <IsOnWatchList>>false</IsOnWatchList>
  </Results>
  <Results>
    <Reference>111111</Reference>
    <OrganisationName>Test Company 2</OrganisationName>
    <Postcode>YO31 7XZ</Postcode>
    <Status>Cancelled</Status>
    <IsOnWatchList>>false</IsOnWatchList>
  </Results>
  <Results>
    <Reference>222222</Reference>
    <OrganisationName>Test Company 3</OrganisationName>
    <Postcode />
    <Status>EEA Authorised</Status>
    <IsOnWatchList>>false</IsOnWatchList>
  </Results>
</OrganisationSearch>
```

---

i.ii. SearchIndividuals()

This method allows you to search all individuals within the FSA Register, and filter your search by a range of parameters.

#### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap:Header>
  <soap:Body>
    <SearchIndividuals xmlns="http://sdm-fsa-register.co.uk/">
      <criteria>
        <AuthStatus>string</AuthStatus>
        <Position>string</Position>
        <AppointeeRelationship>string</AppointeeRelationship>
        <ModifiedFrom>dateTime</ModifiedFrom>
        <ModifiedTo>dateTime</ModifiedTo>
      </criteria>
    </SearchIndividuals>
  </soap:Body>
</soap:Envelope>
```

#### Request Message Parameters:

- i. *AuthStatus (string)*  
Searches for individuals with a specified authorisation status.
- ii. *Position (string)*  
Searches for individuals of a specified position.
- iii. *AppointeeRelationship (string)*  
Searches for individuals of a specified appointee relationship to their parent organisation.
- iv. *ModifiedFrom (date: DD/MM/YYYY format)*  
Searches for individuals whose record was modified on or after the specified date.
- v. *ModifiedTo (date: DD/MM/YYYY format)*  
Searches for individuals whose record was modified on or before the specified date.

#### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SearchIndividualsResponse xmlns="http://tempuri.org/">
      <SearchIndividualsResult>dataset</SearchIndividualsResult>
    </SearchIndividualsResponse>
  </soap12:Body>
</soap12:Envelope>
```

#### Response Dataset Schema:

<http://www.sdm-fsa-register.co.uk/documents/PersonSearch.xsd>

#### Sample Invocation Code (c#):

```
//create a new service proxy instance
FsaPlus.SdmFsaPlusWebService service = new
NewsServiceTestApp.FsaPlus.SdmFsaPlusWebService();
```

```
//set the service url (this is set automatically if you use visual
studio to generate your
//web service proxy
service.Url = "https://www.sdm-fsa-
register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx";

//pass your FSA Register+ credentials here
service.AuthHeaderValue = new AuthHeader();
service.AuthHeaderValue.UserName = "user_name";
service.AuthHeaderValue.Password = "password";

//create the search criteria object
PersonSearchCriteria criteria = new PersonSearchCriteria();
criteria.Name = "frederick";

//invoke the SDM FSA Register+ web service
PersonSearch results = service.SearchIndividuals(criteria);

//write the returned xml to disk
results.WriteXml(@"C:\Individuals.xml");
```

### Sample XML Output:

```
<?xml version="1.0" standalone="yes"?>
<PersonSearch xmlns="http://tempuri.org/PersonSearch.xsd">
  <Results>
    <Reference>ACH00000</Reference>
    <Name>Mr John Frederick Howard</Name>
    <Status>Active</Status>
    <LastUpdate>2006-06-16T00:00:00+01:00</LastUpdate>
    <IsOnWatchList>false</IsOnWatchList>
  </Results>
  <Results>
    <Reference>ACJ11111</Reference>
    <Name>Mr James Frederick Jackson</Name>
    <Status>Active</Status>
    <LastUpdate>2003-12-31T00:00:00+00:00</LastUpdate>
    <IsOnWatchList>false</IsOnWatchList>
  </Results>
  <Results>
    <Reference>WFS22222</Reference>
    <Name>Mr Simon Frederick Steele</Name>
    <Status>Active</Status>
    <LastUpdate>2005-03-09T00:00:00+00:00</LastUpdate>
    <IsOnWatchList>false</IsOnWatchList>
  </Results>
</PersonSearch>
```

### i.iii. GetOrganisation()

This method allows you to retrieve all information available relating to a specific organisation.

### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://tempuri.org/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
```

```
<soap12:Body>
  <GetOrganisation xmlns="http://tempuri.org/">
    <reference>int</reference>
  </GetOrganisation>
</soap12:Body>
</soap12:Envelope>
```

#### Request Message Parameters:

*i. reference (integer)*

The numeric organisation reference (i.e the organisation's unique FSA Register ID).

#### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetOrganisationResponse xmlns="http://tempuri.org/">
      <GetOrganisationResult>dataset</GetOrganisationResult>
    </GetOrganisationResponse>
  </soap12:Body>
</soap12:Envelope>
```

#### Response Dataset Schema:

<http://www.sdm-fsa-register.co.uk/documents/Organisation.xsd>

#### Sample Invocation Code (c#):

```
//create a new service proxy instance
FsaPlus.SdmFsaPlusWebService service = new
NewsServiceTestApp.FsaPlus.SdmFsaPlusWebService();

//set the service url (this is set automatically if you use visual
studio to generate your
//web service proxy
service.Url = "https://www.sdm-fsa-
register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx";

//pass your FSA Register+ credentials here
service.AuthHeaderValue = new AuthHeader();
service.AuthHeaderValue.UserName = "user_name";
service.AuthHeaderValue.Password = "password";

//set this to a valid organisation ID
int organisationID = 100000;

//invoke the SDM FSA Register+ web service
Organisation result = service.GetOrganisation(organisationID);

//write the returned xml to disk
result.WriteXml(@"C:\Organisation.xml");
```

#### Sample XML Output:

```
<?xml version="1.0" standalone="yes"?>
<Organisation xmlns="http://tempuri.org/Organisation.xsd">
  <OrganisationDetail>
    <Reference>100000</Reference>
    <CompanyName>Test Company 1</CompanyName>
    <Address1>4th Floor</Address1>
```

```

<Address2>100 A Street</Address2>
<Address3 />
<Address4 />
<Address5>London</Address5>
<Address6 />
<Postcode>E1 2AA</Postcode>
<Telephone>44 020 7000 0000</Telephone>
<Fax />
<ClientMoney>No</ClientMoney>
<Status>Authorised</Status>
<StatusModified>2007-05-01T00:00:00+01:00</StatusModified>
<Registered>2007-05-01T00:00:00+01:00</Registered>
<OrganisationType>Regulated</OrganisationType>
<LegalStatus>Company Limited by Share Capital</LegalStatus>
<IsOnWatchList>>false</IsOnWatchList>
</OrganisationDetail>
<Individuals>
  <IndividualReference>GCC01025</IndividualReference>
  <OrganisationReference>461914</OrganisationReference>
  <Name>Mr John Smith</Name>
  <Status>Active</Status>
</Individuals>
<Individuals>
  <IndividualReference>JSB01089</IndividualReference>
  <OrganisationReference>461914</OrganisationReference>
  <Name>Mr John Doe</Name>
  <Status>Active</Status>
</Individuals>
<InvestmentTypes>
  <Reference>461914</Reference>
  <Activity>Advising on investments (except on Pension Transfers and Pension Opt
  Outs)</Activity>
  <InvestmentType>Contractually Based Investments</InvestmentType>
  <InvestmentDescription>Commodity Future</InvestmentDescription>
  <Status>Authorised</Status>
  <Effective>2007-05-01T00:00:00+01:00</Effective>
  <LastUpdated>2007-05-01T00:00:00+01:00</LastUpdated>
</InvestmentTypes>
<InvestmentTypes>
  <Reference>461914</Reference>
  <Activity>Advising on investments (except on Pension Transfers and Pension Opt
  Outs)</Activity>
  <InvestmentType>Contractually Based Investments</InvestmentType>
  <InvestmentDescription>Commodity option and option on commodity
  future</InvestmentDescription>
  <Status>Authorised</Status>
  <Effective>2007-05-01T00:00:00+01:00</Effective>
  <LastUpdated>2007-05-01T00:00:00+01:00</LastUpdated>
</InvestmentTypes>
<InvestmentTypes>
  <Reference>461914</Reference>
  <Activity>Advising on investments (except on Pension Transfers and Pension Opt
  Outs)</Activity>
  <InvestmentType>Contractually Based Investments</InvestmentType>
  <InvestmentDescription>Contract for Differences (excluding a spread bet and a
  rolling spot forex contract)</InvestmentDescription>
  <Status>Authorised</Status>
  <Effective>2007-05-01T00:00:00+01:00</Effective>
  <LastUpdated>2007-05-01T00:00:00+01:00</LastUpdated>
</InvestmentTypes>
<RegulatedActivities>
  <Reference>461914</Reference>
  <Activity>Advising on investments (except on Pension Transfers and Pension Opt
  Outs)</Activity>
  <CustomerTypeDescription>Intermediate Customer</CustomerTypeDescription>
  <Effective>2007-10-31T00:00:00+00:00</Effective>
  <LastUpdated>2007-10-31T00:00:00+00:00</LastUpdated>
</RegulatedActivities>
<RegulatedActivities>
  <Reference>461914</Reference>
  <Activity>Advising on investments (except on Pension Transfers and Pension Opt
  Outs)</Activity>
  <CustomerTypeDescription>Market Counterparty</CustomerTypeDescription>
  <Effective>2007-10-31T00:00:00+00:00</Effective>
  <LastUpdated>2007-10-31T00:00:00+00:00</LastUpdated>

```

```

</RegulatedActivities>
<RegulatedActivities>
  <Reference>461914</Reference>
  <Activity>Arranging (bringing about) deals in investments</Activity>
  <CustomerTypeDescription>Intermediate Customer</CustomerTypeDescription>
  <Effective>2007-10-31T00:00:00+00:00</Effective>
  <LastUpdated>2007-10-31T00:00:00+00:00</LastUpdated>
</RegulatedActivities>
</Organisation>

```

---

#### i.iv. GetIndividual()

This method allows you to retrieve all information available relating to a specific individual.

#### Request Message Format:

```

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://tempuri.org/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <GetIndividual xmlns="http://tempuri.org/">
      <reference>string</reference>
    </GetIndividual>
  </soap12:Body>
</soap12:Envelope>

```

#### Response Message Format:

```

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetIndividualResponse xmlns="http://tempuri.org/">
      <GetIndividualResult>dataset</GetIndividualResult>
    </GetIndividualResponse>
  </soap12:Body>
</soap12:Envelope>

```

#### Request Message Parameters:

##### i. *reference (string)*

The individual's reference (i.e the individual's unique FSA Register ID).

#### Response Dataset Schema:

<http://www.sdm-fsa-register.co.uk/documents/Organisation.xsd>

#### Sample Invocation Code (c#):

```

//create a new service proxy instance
FsaPlus.SdmFsaPlusWebService service = new
NewsServiceTestApp.FsaPlus.SdmFsaPlusWebService();

//set the service url (this is set automatically if you use visual
studio to generate your

```

```
//web service proxy
service.Url = "https://www.sdm-fsa-
register.co.uk/FSAWebService/SdmFsaPlusWebService.asmx";

//pass your FSA Register+ credentials here
service.AuthHeaderValue = new AuthHeader();
service.AuthHeaderValue.UserName = "user_name";
service.AuthHeaderValue.Password = "password";

//set this to a valid individual reference
string individualID = "ASB00000";

//invoke the SDM FSA Register+ web service
Individual result = service.GetIndividual(individualID);

//write the returned xml to disk
result.WriteXml(@"C:\Individual.xml");
```

### Sample XML Output:

```
<?xml version="1.0" standalone="yes"?>
<Individual xmlns="http://tempuri.org/Individual.xsd">
  <IndividualDetail>
    <Reference>ASB00000</Reference>
    <Name>Mr John Smith</Name>
    <Modified>2006-05-04T00:00:00+01:00</Modified>
    <Status>Active</Status>
    <IsOnWatchList>>false</IsOnWatchList>
  </IndividualDetail>
  <Function>
    <Reference>ASB00000</Reference>
    <OrganisationReference>448160</OrganisationReference>
    <OrganisationName>Test Company 1</OrganisationName>
    <StartDate>2006-05-02T00:00:00+01:00</StartDate>
    <Description>CF21 Investment Adviser</Description>
  </Function>
  <Function>
    <Reference>ASB00000</Reference>
    <OrganisationReference>448160</OrganisationReference>
    <OrganisationName>Test Company 1</OrganisationName>
    <StartDate>2006-05-02T00:00:00+01:00</StartDate>
    <Description>CF8 Apportionment and Oversight</Description>
  </Function>
  <Function>
    <Reference>ASB00000</Reference>
    <OrganisationReference>448160</OrganisationReference>
    <OrganisationName>Test Company 1</OrganisationName>
    <StartDate>2006-05-02T00:00:00+01:00</StartDate>
    <Description>CF3 Chief Executive</Description>
  </Function>
  <Employment>
    <Reference>ASB00000</Reference>
    <OrganisationReference>210724</OrganisationReference>
    <OrganisationName>Pantheon Financial Management Ltd</OrganisationName>
    <StartDate>2004-05-05T00:00:00+01:00</StartDate>
    <Description>Employed by</Description>
  </Employment>
  <Employment>
    <Reference>ASB00000</Reference>
    <OrganisationReference>214350</OrganisationReference>
    <OrganisationName>The Davis Group Plc</OrganisationName>
    <StartDate>2002-07-04T00:00:00+01:00</StartDate>
    <EndDate>2004-04-30T00:00:00+01:00</EndDate>
    <Description>Employed by</Description>
  </Employment>
</Individual>
```

## ii. Lookup Methods

## ii.i. GetOrganisationAuthStatuses()

This method allows you to retrieve all the available authorisation statuses for organisations. The statuses are returned as strings, which can be used to populate the AuthStatus field when calling the SearchOrganisations method.

### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap:Header>
  <soap:Body>
    <GetOrganisationAuthStatuses xmlns="http://sdm-fsa-register.co.uk/" />
  </soap:Body>
</soap:Envelope>
```

### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetOrganisationAuthStatusesResponse xmlns="http://sdm-fsa-register.co.uk/">
      <GetOrganisationAuthStatusesResult>
        <string>string</string>
        <string>string</string>
      </GetOrganisationAuthStatusesResult>
    </GetOrganisationAuthStatusesResponse>
  </soap:Body>
</soap:Envelope>
```

---

## ii.ii. GetPersonAuthStatuses()

This method allows you to retrieve all the available authorisation statuses for individuals. The statuses are returned as strings, which can be used to populate the AuthStatus field when calling the SearchIndividuals method.

### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <GetPersonAuthStatuses xmlns="http://sdm-fsa-register.co.uk/" />
  </soap12:Body>
</soap12:Envelope>
```

### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetPersonAuthStatusesResponse xmlns="http://sdm-fsa-register.co.uk/">
      <GetPersonAuthStatusesResult>
        <string>string</string>
        <string>string</string>
      </GetPersonAuthStatusesResult>
    </GetPersonAuthStatusesResponse>
  </soap12:Body>
</soap12:Envelope>
```

### ii.iii. GetRegulatedActivites()

This method allows you to retrieve all the regulated activities for organisations. The results are returned as strings, which can be used to populate the Activity field when calling the SearchOrganisations method.

#### Request Message Format :

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <GetRegulatedActivities xmlns="http://sdm-fsa-register.co.uk/" />
  </soap12:Body>
</soap12:Envelope>
```

#### Response Message Format :

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetRegulatedActivitiesResponse xmlns="http://sdm-fsa-register.co.uk/">
      <GetRegulatedActivitiesResult>
        <string>string</string>
        <string>string</string>
      </GetRegulatedActivitiesResult>
    </GetRegulatedActivitiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

### ii.iv. GetIndividualFunctions()

This method allows you to retrieve all the controlled functions (i.e. positions) which apply to individuals within the register. The results are returned as key/value pairs. The value is for display purposes, and the key can be used to populate the Position field when calling the SearchIndividuals method.

#### Request Message Format :

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <GetIndividualFunctions xmlns="http://sdm-fsa-register.co.uk/" />
  </soap12:Body>
</soap12:Envelope>
```

### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetIndividualFunctionsResponse xmlns="http://sdm-fsa-register.co.uk/">
      <GetIndividualFunctionsResult>
        <PairOfStringString>
          <Key>string</Key>
          <Value>string</Value>
        </PairOfStringString>
        <PairOfStringString>
          <Key>string</Key>
          <Value>string</Value>
        </PairOfStringString>
      </GetIndividualFunctionsResult>
    </GetIndividualFunctionsResponse>
  </soap12:Body>
</soap12:Envelope>
```

### ii.v. GetAppointeeRelationships()

This method allows you to retrieve the valid relationships for organisation appointees which apply to individuals within the register. The results are returned as key/value pairs. The value is for display purposes, and the key can be used to populate the AppointeeRelationship field when calling the SearchIndividuals method.

### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <GetAppointeeRelationships xmlns="http://sdm-fsa-register.co.uk/" />
  </soap12:Body>
</soap12:Envelope>
```

### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
```

```
<GetAppointeeRelationshipsResponse xmlns="http://sdm-fsa-register.co.uk/">
  <GetAppointeeRelationshipsResult>
    <PairOfStringString>
      <Key>string</Key>
      <Value>string</Value>
    </PairOfStringString>
    <PairOfStringString>
      <Key>string</Key>
      <Value>string</Value>
    </PairOfStringString>
  </GetAppointeeRelationshipsResult>
</GetAppointeeRelationshipsResponse>
</soap12:Body>
</soap12:Envelope>
```

## i. Additional Services

---

### iii.i. AuthenticateUser()

This method allows you to provide login services within your application. The method accepts a user name and password as parameters, and returns a boolean value indicating if the credentials were successfully authenticated. Note: unlike the other methods within the service, this method does not require the credentials to be passed in the message header.

#### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AuthenticateUser xmlns="http://sdm-fsa-register.co.uk/">
      <userName>string</userName>
      <password>string</password>
    </AuthenticateUser>
  </soap12:Body>
</soap12:Envelope>
```

#### Response Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AuthenticateUserResponse xmlns="http://sdm-fsa-register.co.uk/">
      <AuthenticateUserResult>boolean</AuthenticateUserResult>
    </AuthenticateUserResponse>
  </soap12:Body>
</soap12:Envelope>
```

### iii.ii. DownloadFile()

This method allows you to download the most up-to-date version of the Organisations/Individuals data file, containing all the current entries within the FSA register. You can select what format you would like the output data file to be in, and the result is returned as an array of bytes, which should be written by your application to disk. Note: most files are sent compressed as zip files, which will need to be decompressed before use.

#### Request Message Format:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <AuthHeader xmlns="http://sdm-fsa-register.co.uk/">
      <UserName>string</UserName>
      <Password>string</Password>
    </AuthHeader>
  </soap12:Header>
  <soap12:Body>
    <DownloadFile xmlns="http://sdm-fsa-register.co.uk/">
      <request>
        <FileFormat>TAP or CSV or DBF or EXT</FileFormat>
        <SearchType>Organisation or Individual</SearchType>
      </request>
    </DownloadFile>
  </soap12:Body>
</soap12:Envelope>
```

#### **Request Message Parameters:**

*i. FileFormat (string)*

This determines what format the output file will be in. The valid values are ‘CSV’, ‘DBF’, ‘TAP’, or ‘EXT’.

*ii. SearchType (string)*

Specifying ‘Organisation’ or ‘Individual’ as the parameter value determines whether organisation or individual data is returned.

#### **Response Message Format:**

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DownloadFileResponse xmlns="http://sdm-fsa-register.co.uk/">
      <DownloadFileResult>
        <FileName>string</FileName>
        <Contents>base64Binary</Contents>
      </DownloadFileResult>
    </DownloadFileResponse>
  </soap12:Body>
</soap12:Envelope>
```

#### **Response Message Parameters:**

*i. FileName (string)*

The file name for the output file. Most files are sent compressed as zip files, so the file extension should be checked before writing the file to disk.

*ii. Content (byte array)*

The file content, as an array of base64 encoded bytes.