AE Exchange
RESTful API
Specification

version 1.0

Document Information

Document Title :	
Document File Name :	
Revision Number :	
Issued By :	
Issue Date :	
Status :	

Document Approvals

Prepared By:	Reviewed By:
<name> <title>	<name>
Approved By :	
<name> <project name>	
<name> <title> <customer>	

Revision History

Version	Primary Author(s)	Description of Version	Date Completed

Table of Contents

© **INTERSOFTWARE SOLUTIONS LTD**		Page 7 of 65

© **INTERSOFTWARE SOLUTIONS LTD**		Page 8 of 65

[RTI FPS v1](#)

Introduction

This document describes the RESTFul API and resources provided by AE Exchange. The REST APIs are for developers who want to integrate AE Exchange into their application or script interactions with the AE Exchange server.

AE Exchange's REST APIs provide access to resources (data entities) via URI paths. To use a REST API, your application will make an HTTPS request and parse the response. The response format is JSON. Your methods will be the standard HTTP methods like GET, PUT, POST and DELETE.

Because the REST API is based on open standards, you can use any web development language to access the API.

All API access is over HTTPS, and accessed via the API Endpoints. The relative path prefix /v1/ indicates that we are currently using version 1 of the API.

Overview

Prerequisites

To use AE Exchange API, we assume that :
- You are familiar with REST and OAuth 2.0
- You have obtained a client credential from us

API Endpoints

AE Exchange RESTFul API are supported in 2 environments, use Test environment for testing and live environment for production.

Test environment	https://test.aeexchange.com
Live environment	https://www1.aeexchange.com

Authentication

AE Exchange API use oAuth 2.0 protocol for authentication and authorisation. With each API call, you'll need to set request headers, including an OAuth 2.0 access token.

There are two types of authentication:
1. Basic Authentication which do not require user identity
   Our public API endpoints that don't pertain to specific user information, you can use oAuth2 client_credentials token grant type to make a request to these API.

2. Authentication with user identity
   Other than public API, the rest of the API are required user consent to grants the requesting application access to the user related data.

The type of authentication required will be decided based on the user/client requesting the client id from AE Exchange administrator.

The major roles within the oauth are:
a) Authorisation Server (AS) - the server that authorises and issues the tokens.
b) Resource Server (RS) - The server that serves the API
c) Resource Owner (Owner) - The user who owns the resource.
d) Client - The third party application.

Registration

Before you can begin the OAuth process, you must register the 'client' with the service. When registering a new application/client, you usually register the basic information such as application name, website, a logo etc. In addition you must register a redirect URI to be used for redirecting users to - for browser based, or mobile apps.

Redirect URIs

The service will only redirect the users to a registered URI, which helps prevent some attacks. Any HTTP redirect URIs must be protected with TLS security, so the service will only redirect to URIs beginning with "https". This prevents tokens being intercepted during the authorisation process.

© **INTERSOFTWARE SOLUTIONS LTD**		Page 12 of 65

Client ID and secret

After registering your client, you will receive a client ID and a client SECRET. The client ID is considered public information, and is used to build login URLs. The client secret must be kept confidential and if a deployed app cannot keep the secret confidential, then the secret is not used.

Grant Types supported

The first step of oAuth is to get authorisation from the Auth server (AS). AE Exchange oAuth server provides the following grant types for different use cases. The grant types defined are:

- Authorization code - for apps running on a web server
- Implicit - for browser based or mobile apps
- Client Credentials - for public API access

Authorization code

STEP 1) Create a login link sending the HTTP user to:

HTTP Request

```
POST /portal/oauth/authorize?response_type=code&client_id=<CLIENT_ID>&scope=<SCOPE>
```

<CLIENT_ID> - the client ID obtained from the provider
<SCOPE> - the scope that is appropriate for use with the API.

Response

```
{
code: pX345
}
```

STEP 2) Exchange the 'code' with a Token request using the AS.

HTTP Request

```
POST
/portal/oauth/token?grant_type=authorization_code&code=<CODE>&client_id=<CLIENT_ID>&
client_secret=<SECRET>scope=<SCOPE>
```

<CLIENT_ID> - the client ID obtained from the provider
<SECRET> - the client secret obtained from the provider
<CODE> - the code that is obtained in step 1
<SCOPE> - the scope that is appropriate for use with the API.

Response

```
{
 "access_token": "d3c333df-17fe-45aa-879a-e027003286c6",
 "token_type":"bearer",
 "refresh_token":"823d6237-d20d-49fb-9426-eec3c869d6eb",
 "expires_in":150,
 "scope":<SCOPE>,
 "resource-server-uri":"test.aeexchange.com"
 }
```

access_token - is the access token as assigned by the authorization server.
token_type - is a type of token assigned by the authorization server.
expires_in - is a number of seconds after which the access token expires, and is no longer valid. Expiration of access tokens is optional.
refresh_token - contains a refresh token in case the access token expires. The refresh token is used to obtain a new access token once the one returned in this response is no longer valid
resource_server_uri: URI to be prefixed to any API call offered by the RS.

Implicit

Browser-based apps run entirely in the browser after loading the source code from a web page. Since the entire source code is available to the browser, they cannot maintain the confidentiality of their client secret, so the secret is not used in this case

HTTP Request

```
POST /portal/oauth/authorize?response_type=token&client_id=<CLIENT_ID>&scope=<SCOPE>
```

You will get an access token response in the same format as the other grant types.

Client Credentials

In some cases, applications may wish to get an access token for their own account, outside the context of any specific user. OAuth provides the client_credentials grant type for this purpose.

```
POST
/portal/oauth/token?grant_type=client_credentidals&client_id=<CLIENT_ID>&client_secr
et=<SECRET>scope=<SCOPE>
```

You will get an access token response in the same format as the other grant types.

Error response

Most possible reasons for error response are due to incorrect credentials. The following http response will be provided in case of invalid credentials or Oauth token sent in the request. There will not be any XML body in the response.

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-store
Pragma: no-cache
WWW-Authenticate:     Bearer     realm="oauth",     error="invalid_token",
   error_description="9387e880-ab6f-4eff-9fff-d4f9b2180447"
Content-Type: application/json;charset=UTF-8
```

AE Quick Pension Calculator: Calculate

A simple calculator to assess employee ae status and calculate contributions.

Request

HTTP Request

```
POST /portal/public/api/v1/qpc/ae/calculator
```

Request Body

Provide an [Employee AE Basic Details](#) resource in the request body.
You must specify a value for these properties:
- birthDate
- gender
- erContributionsPercent
- eeContributionsPercent
- totalPay
- frequencyCode
- taxReliefArrangement
- earningBasis

Response

If successful, this method returns an [Employee AE Basic Details](#) resource in the response body.

Errors

The table below identifies error messages that the API could return in response to a call to this method. Please see the error message documentation for more detail.

Error Type	Error Description
Bad Request (400)	It could be invalid format or validation error

AE Quick Pension Calculator: Letter

Generate AE letter based on calculator result.

Request

HTTP Request

```
POST /portal/public/api/v1/qpc/ae/letters
```

Request Body

Provide an Employee AE Letter Details resource in the request body
You must specify a value for these properties:
- Staging Date
- Pension Provider Name
- Pension Provider Address
- Pension Provider Email
- Pension Provider Telephone
- Employer Name
- Employer Address
- Employer Email
- Employer Telephone
- Employer Contact Person
- Employee Title
- Employee Name
- Employee Address
- Employee Email
- Employee Mobile
- Assess Status
- Frequency Code
- Tax Year
- Earnings Basis
- Tax Relief Arrangement
- Employer Contributions Percent
- Employee Contributions Percent

Response

If successful, this method returns a PDF document.

Errors

The table below identifies error messages that the API could return in response to a call to this method. Please see the error message documentation for more detail.

Error Type	Error Description
Bad Request (400)	It could be invalid format or validation error

AE Assessment: Upload

To upload an employee earnings file for AE assessment.

Request

HTTP Request

```
POST /portal/api/v1/clients/clientId/ae/assessments
```

Content Type

application/vnd.genlife-1.0+xml	Use this to submit file in Genlife format
application/vnd.papdis-1.1+xml	Use this to submit file in PAPDIS 1.1 format
application/vnd.universal-ae-1.1.0+xml	This is similar to PAPDIS 1.1 format but added extra fields to capture additional data which require by certain pension provider. This is the prefer type if you want to use e-filing API in your application.

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Please supply either Genlife V1 or PAPDIS V1.1 or Universal AE 1.1.0 resource in request body.

Response

If successful, this method returns a Queuing Status resource in the response body.

© INTERSOFTWARE SOLUTIONS LTD		Page 19 of 65

AE Assessment: Upload Status

To get the assessment result

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/ae/assessments/status/correlationId
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a [Queuing Status](#) resource in the response body.

© **INTERSOFTWARE SOLUTIONS LTD**		Page 20 of 65

AE Assessment: Result

To get the assessment result

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/ae/assessments/results/correlationId
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Assessment Result resource in the response body.

AE Assessment: Commit

To accept the result and commit it. Once commit, same employee for the same period cannot assess again.

Request

HTTP Request

```
PUT /portal/api/v1/clients/clientId/ae/assessments/results/correlationId
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns HTTP OK Status. No body in the response.

(may need to create queuing)

AE Contributions: Upload

To upload a ae contributions file

Request

HTTP Request

```
POST /portal/api/v1/clients/clientId/ae/contributions
```

Content Type

application/vnd.papdis-1.0+xml	Use this to submit PAPDIS 1.0
application/vnd.papdis-1.1+xml	Use this to submit PAPDIS 1.1
application/vnd.universal-ae-1.1.0+xml	This is similar to PAPDIS 1.1 format but added extra fields to capture additional data which require by certain pension provider. This is the prefer type if you want to use e-filing API in your application.

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Please supply either PAPDIS V1.0 or PAPDIS V1.1 or Universal AE 1.1.0 resource in request body.

Response

If successful, this method returns a Queuing Status resource in the response body.

AE Contributions: Upload Status

To get the current status of the uploaded file.

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/ae/contributions/status/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id assigned to the submission

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Queuing Status resource in the response body.

AE Contributions: Upload Error

To get the error report of the uploaded file

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/ae/contributions/errors/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a AE Contribution Validation Error resource in the response body.

AE Contributions: Download for Payroll

To download the uploaded file

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/ae/contributions/payroll/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Accept

application/vnd.papdis-1.1+xml	To indicate the response body is PAPDIS 1.1 xml. Only this Accept type is acceptable in this API.

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a PAPDIS V1.1 resource in the response body.

© **INTERSOFTWARE SOLUTIONS LTD**		Page 26 of 65

AE Contributions: Download for Pension Provider

To download the files for pension provider

Request

HTTP Request

```
GET
/portal/api/v1/clients/clientId/ae/contributions/pension-provider/correlationId
```

Parameters

Parameter Name	Value	Description
clientId	integer	The id of the client
correlationId	string	The id of the uploaded file

Accept

text/csv	Depend on the file size, if less then XXX then return in csv format, otherwise return in compressed format.
application/zip	return in compressed format.

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

This method returns a contribution csv file or compressed csv file.

AE Contributions: ePost

To e-file the contributions to pension provider

Request

HTTP Request

```
PUT /portal/api/v1/clients/clientId/ae/contributions/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Queuing Status resource in the response body.

AE Contributions: ePost Status

To get the efiling status

Request

HTTP Request

```
POST
/portal/api/v1/clients/clientId/ae/contributions/efiling/status/correlationI
d
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id use for the correspondent efiling submission progress

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a <u>Queuing Status</u> resource in the response body.

AE Contributions: ePost Error

To get the error report of the efiling

Request

HTTP Request

```
GET
/portal/api/v1/clients/clientId/ae/contributions/efiling/errors/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the e-filing file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a AE Contributions Efile Validation Error resource in the response body.

AE Contributions: Delete

To delete the uploaded file

Request

HTTP Request

```
DELETE /portal/api/v1/clients/clientId/ae/contributions/correlationId
```

Parameters

Parameter Name	Value	Description
corrrelationId	string	The id of the e-filing file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/ae

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a HTTP 204 response code (No Content).

Errors

The table below identifies error messages that the API could return in response to a call to this method. Please see the error message documentation for more detail.

Error Type	Error Description
forbidden (403)	The file has been e-filed success, it is not allow to delete

FPS: Upload

To upload a fps file in HMRC XML format

Request

HTTP Request

```
POST /portal/api/v1/clients/clientId/rti/fps
```

Content Type

application/vnd.fps-1.0+xml	Use this to submit FPS for tax year 2015-16

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Please supply a RTI FPS v1 resource in request body

Response

If successful, this method returns a Queuing Status resource in the response body.

FPS: Upload Status

To get the current status of the uploaded file.

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps/status/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id assigned to the submission

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Queuing Status resource in the response body.

FPS: Upload List

To get all the uploaded files list

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Parameters

Parameter Name	Value	Description
clientId	integer	Required. The ID of the client.
offset	integer	Optional. If not provide, default 0 will be use. Use to indicate from which record to get.
limit	integer	Optional. If not provide, default 50 will be use. Use to limit no of records return. Minimum is 1 and maximum is 100.

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a FPS List resource in the response body.

© **INTERSOFTWARE SOLUTIONS LTD**		Page 36 of 65

FPS: Download

To download the uploaded file

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Accept

text/csv	To indicate the response body is csv
application/json	To indicate the response body is json
application/xml	To indicate the response body is xml

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a <u>FPS</u> resource in the response body.

FPS: Upload Error

To get the error report of the uploaded file

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps/errors/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients//rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a FPS Validation Error resource in the response body.

FPS: Delete

To delete the uploaded file

Request

HTTP Request

```
DELETE /portal/api/v1/clients/clientId/rti/fps/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients//rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a HTTP 204 response code (No Content).

FPS: Efile

To efile the uploaded file

Request

HTTP Request

```
PUT /portal/api/v1/clients/clientId/rti/fps/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the uploaded file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Queuing Status resource in the response body.

FPS: Efile List

To get the records in the efile list

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps/efiling
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Parameters

Parameter Name	Value	Description
offset	integer	Optional. If not provide, default 0 will be use. Use to indicate from which record to get.
limit	integer	Optional. If not provide, default 50 will be use. Use to limit no of records return. Minimum is 1 and maximum is 100.

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a FPS Efile List resource in the response body.

© INTERSOFTWARE SOLUTIONS LTD		Page 43 of 65

FPS: Efiling Status

To get the efiling status

Request

HTTP Request

```
POST /portal/api/v1/clients/clientId/rti/fps/efiling/status/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id use for the correspondent efiling submission progress

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a Queuing Status resource in the response body.

FPS: Efiling Error

To get the error report of the efiling

Request

HTTP Request

```
GET /portal/api/v1/clients/clientId/rti/fps/efiling/errors/correlationId
```

Parameters

Parameter Name	Value	Description
correlationId	string	The id of the e-filing file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a FPS Efile Validation Error resource in the response body.

FPS: Delete Efiling

To delete the uploaded file

Request

HTTP Request

```
DELETE /portal/api/v1/clients/clientId/rti/fps/efiling/correlationId
```

Parameters

Parameter Name	Value	Description
corrrelationId	string	The id of the e-filing file

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/rti

Request Body

Do not supply a request body in this method

Response

If successful, this method returns a HTTP 204 response code (No Content).

© INTERSOFTWARE SOLUTIONS LTD		Page 47 of 65

Errors

The table below identifies error messages that the API could return in response to a call to this method. Please see the error message documentation for more detail.

Error Type	Error Description
forbidden (403)	The file has been e-filed success, it is not allow to delete

Sign Up

Sign Up to AE Exchange

Request

HTTP Request

```
POST /portal/public/api/v1/signup
```

Authorisation

This request does not require authorisation, but you must set your api key in Authentication Header.

Request Body

Provide an Sign Up resource in the request body.
You must specify a value for these properties:
- Name
- Address
- Telephone
- AdministratorTitle
- AdministratorSurname
- AdministratorForename
- AdministratorSecondForename
- AdministratorEmail
-
- ServiceSignUp

Response

If successful, this method returns an Sign Up resource in the response body.
- Name

Errors

The table below identifies error messages that the API could return in response to a call to this method.
Please see the error message documentation for more detail.

© **INTERSOFTWARE SOLUTIONS LTD**		Page 49 of 65

Error Type	Error Description
Bad Request (400)	It could be invalid format or validation error

Sign Up: Get Status

Get the Sign Up status. Use this method to find out the account is activate or not. Only activated account is allowed to sign in.

Request

HTTP Request

```
GET /portal/public/api/v1/signup/signUpRef
```

Authorisation

This request does not require authorisation, but you must set your api key in Authentication Header.

Request Body

Do not supply a request body in this method.

Response

This method returns an Sign Up resource in the response body.
AccStatus and userActivateDate will indicate the account is activated or not.

Sign Up: Get Security Questions

Get available security questions for Sign Up

Request

HTTP Request

```
GET /portal/public/api/v1/signup/security-questions
```

Authorisation

This request does not require authorisation, but you must set your api key in Authentication Header.

Request Body

Do not supply a request body in this method.

Response

This method returns an Security Questions resource in the response body.

Clients: list

Get the list of clients

Request

HTTP Request

```
GET /portal/api/v1/clients
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients

Parameters

Parameter Name	Value	Description
offset	integer	Optional. If not provide, default 0 will be use. Use to indicate from which record to get.
limit	integer	Optional. If not provide, default 50 will be use. Use to limit no of records return. Minimum is 1 and maximum is 100.

Request Body

Do not supply a request body in this method.

Response

This method returns an Client List resource in the response body.

Header

Pagination info will be provided in the Link Header.

Clients: Add

Create a client.

Request

HTTP Request

```
POST /portal/api/v1/clients
```

Authorisation

This request requires authorisation with at least one of the following scopes.

Scope
/api/v1/clients

Request Body

Provide an <u>Client</u> resource in the request body.
You must specify a value for these properties:

Response

This method returns a <u>Client</u> resource in the response body. A client ID will be include in the resource for the future reference on this client.

Pension Providers: list

Get the list of supported pension providers

Request

HTTP Request

```
GET /portal/public/api/v1/ae/pension-providers
```

Request Body

Do not supply a request body in this method.

Response

This method returns an Pension Providers List resource in the response body.

Setup Pension Scheme
POST
/portal/api/v1/clients/clientId/pension-providers/pensionProviderId/pensions
Response: Pension Scheme Resource with pension scheme id

Get All Pension Schemes
GET
/portal/api/v1/clients/clientId/pension-providers/pensionProviderId/pensions
Response: Pension Scheme Resource

Update Pension Scheme
PUT
/portal/api/v1/clients/clientId/pension-providers/pensionProviderId/pensions/schemeId

Create Client User
POST /portal/api/v1/clients/clientId/users
Response: User Resource with user sign in id

Update Pension Provider
PUT /portal/api/v1/clients/clientId/pension-providers/pensionProviderId
Response: Pension Provider Resource

Get list of employees
GET /portal/api/v1/clients/clientId/employees
default return all active employees, use query string "status" to retrieve all
employees or terminated employees.
status=all - return all employees
status=leaver - return only leaver employees.
Response : Employee List Resource

Resources Specification

Queuing Status

Queuing Status contains the information about your submission status.

Resource Representation

```
{
      "CorrelationId":"string",
      "StatusCode":"string",
      "Status":"string",
      "SuccessTimestamp":"datetime"
}
```

Property Name	Data Type	Description	Note
CorrelationId	String		
StatusCode	String		
Status	String		
SuccessTimestamp	DateTime		

Employee AE Basic Details

Employee AE Basic Details contains minimum data use for AE assessment and pension contribution calculation.

Resource Representation

```
{
        "totalPay":"number",
        "pensionableEarnings":"number",
        "frequencyCode":"string",
        "taxReliefArrangement":"string",
        "earningBasis":"string",
        "birthDate":"date",
        "gender":"string",
        "erContributionsPercent":"number",
        "eeContributionsPercent":"number",
        "erContributionsAmount":"number",
        "eeContributionsAmount":"number",
        "assessStatus":"string",
        "taxYear":"string"
}
```

Property Name	Data Type	Description	Note
totalPay	Number	**[Required]** Qualifying Earnings for this period, usually their total pay subject to National Insurance	No comma, 2 decimal only
pensionableEarnings	Number	The earnings used for calculate pension contributions. It will be same as TotalPay if Net Pay Arrangement is use.	Response value
frequencyCode	String	**[Required]** Employee pay frequency. Accepted values are 'M' -- Monthly 'W' -- Weekly 'N' -- 2 Weekly 'F' -- 4 Weekly 'Q' -- Quarterly 'H' -- Bi Annually	

		'Y' -- Annually	
taxReliefArrangement	String	**[Required]** Tax Relief Arrangement, it can be either of this value 'N' -- Net Pay Arrrangement 'Y'' -- Relief at Source	
earningBasis	String	**[Required]** Earning Basis - To indicate whether to use all TotalPay to calculate the pension contribution or base on Qualifying Earnings to calculate. The accepted value are: 'Q' -- Qualifying Earnings 'A' -- Full Earnings	
birthDate	Date	**[Required]** Employee date of birth. Accepted format is dd/mm/yyyy	
gender	String	**[Required]** Employee Gender. Accepted value are: 'M' -- Male 'F' -- Female	
erContributionsPercent	Number	**[Required]** The percentage of employer contribution.	
eeContributionsPercent	Number	**[Required]** The percentage of employee contribution.	
erContributionsAmt	Number	The employer contribution amount.	Response value
eeContributionsAmt	Number	The employee contribution amount.	Response value
assessStatus	String	The employee AE status. The possible return values are: "EJH" -- Eligible Job Holder "NEJH" -- Non Eligible Job Holder "EW" -- Entitled Worker "W" -- Worker	Response value
taxYear	String	Current Tax Year	Calculation Result.

Employee AE Letter

Employee AE Letter contains details based on calculator result.

Resource Representation

```
{
     "stagingDate":"date",
     "ppName":"string",
     "ppAddress":{
          "address1": "string",
          "address2": "string",
          "address3": "string",
          "address4": "string",
          "postcode": "string",
          "country": "string"
     },
     "ppEmail":"string",
     "ppTelephone":"string",
     "erName":"string",
     "erAddress":{
          "address1": "string",
          "address2": "string",
          "address3": "string",
          "address4": "string",
          "postcode": "string",
          "country": "string"
     },
     "erEmail":"string",
     "erTelephone":"string",
     "erContactPerson":"string",
     "eeTitle":"string",
     "eeName":"string",
     "eeAddress":{
          "address1": "string",
          "address2": "string",
          "address3": "string",
          "address4": "string",
          "postcode": "string",
          "country": "string"
     },
     "eeEmail":"string",
```

```
    "eeMobile":"string",
    "assessStatus":"string",
    "frequencyCode":"string",
    "taxReliefArrangement":"string",
    "earningBasis":"string",
    "erContributionsPercent":"number",
    "eeContributionsPercent":"number",
    "taxYear":"string"
}
```

Property Name	Value	Description	Note
stagingDate	Date	Staging Date. Accepted format is dd/mm/yyyy	
ppName	String	Name of Pension Provider. 35 characters max. **[Required]**	
ppAddress	Object	Pension Provider Address	
ppAddress.address1	String	Line 1 of Pension Provider Address. 35 characters max. **[Required]**	
ppAddress.address2	String	Line 2 of Pension Provider Address. 35 characters max.	
ppAddress.address3	String	Line 3 of Pension Provider Address. 35 characters max.	
ppAddress.address4	String	Line 4 of Pension Provider Address. 35 characters max.	
ppAddress.postcode	String	Postcode of Pension Provider Address. 10 characters max. **[Required]**	
ppAddress.country	String	Country of Pension Provider Address. 35 characters max.	

ppEmail	String	Email of Pension Provider. 75 characters max. **[Required]**	
ppTelephone	String	Telephone number of Pension Provider . 50 characters max.	
erName	String	Name of Employer. 35 characters max. **[Required]**	
erAddress	Object	Employer Address	
erAddress.address1	String	Line 1 of Employer Address. 35 characters max. **[Required]**	
erAddress.address2	String	Line 2 of Employer Address. 35 characters max.	
erAddress.address3	String	Line 3 of Employer Address. 35 characters max.	
erAddress.address4	String	Line 4 of Employer Address. 35 characters max.	
erAddress.postcode	String	Postcode of Employer Address. 10 characters max. **[Required]**	
erAddress.country	String	Country of Employer Address. 35 characters max.	
erEmail	String	Email of Employer. 75 characters max. **[Required]**	
erTelephone	String	Telephone number of Employer . 50 characters max.	
erContactPerson	String	Contact Person's Name of Employer	
eeTitle	String	Title of Employee. 35 characters max. **[Required]**	
eeName	String	Name of Employee. 35 characters max. **[Required]**	
eeAddress	Object	Employee Address	
eeAddress.address1	String	Line 1 of Employee Address. 35 characters max. **[Required]**	

eeAddress.address2	String	Line 2 of Employee Address. 35 characters max.	
eeAddress.address3	String	Line 3 of Employee Address. 35 characters max.	
eeAddress.address4	String	Line 4 of Employee Address. 35 characters max.	
eeAddress.postcode	String	Postcode of Employee Address. 10 characters max. **[Required]**	
eeAddress.country	String	Country of Employee Address. 35 characters max.	
eeEmail	String	Email of Employee. 75 characters max. **[Required]**	
eeMobile	String	Mobile number of Employee. 50 characters max.	
assessStatus	String	**[Required]** The employee AE status. The possible return values are: "EJH" -- Eligible Job Holder "NEJH" -- Non Eligible Job Holder "EW" -- Entitled Worker "W" -- Worker	
frequencyCode	String	**[Required]** Employee pay frequency. Accepted values are 'M' -- Monthly 'W' -- Weekly 'N' -- 2 Weekly 'F' -- 4 Weekly 'Q' -- Quarterly 'H' -- Bi Annually 'Y' -- Annually	
taxReliefArrangement	String	**[Required]** Tax Relief Arrangement, it can be either of this value 'N' -- Net Pay Arrrangement 'Y'' -- Relief at Source	
earningBasis	String	**[Required]** Earning Basis - To indicate whether to use all TotalPay to calculate the pension contribution or base on Qualifying Earnings to calculate.	

		The accepted value are: 'Q' -- Qualifying Earnings 'A' -- Full Earnings	
erContributionsPercent	Number	**[Required]** The percentage of employer contribution.	
eeContributionsPercent	Number	**[Required]** The percentage of employee contribution.	
taxYear	String	Current Tax Year	Calculation Result.

RTI FPS v1

Please refer to HMRC schema here….